

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS

DOCTORAL THESIS

Development of an Alternative Method for Multispectral Image Segmentation based on Cartesian Complexes and its associated Oriented Matroids

Author:
José VALERO

Supervisor:
Dr. Iván LIZARAZO

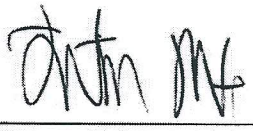
February 2, 2019

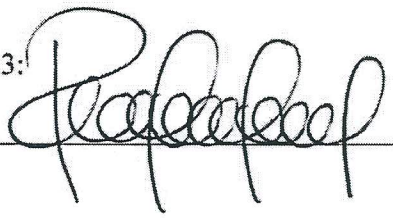
Doctorate Committee

This thesis, entitled "Development of an Alternative Method for Multispectral Image Segmentation based on Cartesian Complexes and its associated Oriented Matroids", written by José Antonio Valero Medina, has been approved in terms of style and intellectual content.

We have read this thesis and approve it,

Jury 1: 

Juror 2: 

Juror 3: 

PhD. Iván Lizarazo Salcedo.
Tutor

Date of Defense: February 5, 2019.

Declaration of Authorship

I, José VALERO, declare that this thesis titled, "Development of an Alternative Method for Multispectral Image Segmentation based on Cartesian Complexes and its associated Oriented Matroids" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

"*The thoughts of the diligent tend only to plenteousness; But every one that is hasty hasteth only to want.*"

Proverbs 21:5, American Standard Version (ASV)

"*Be diligent in these things; give thyself wholly to them; that thy progress may be manifest unto all.*"

1 Timothy 4:15, American Standard Version (ASV)

Abstract

In digital image analysis, a strategy used to address spatial and topological properties is to define image objects, as they are known in the remote sensing community, grouping pixels as coarser geometric space elements or super pixels. This process is known as image segmentation. In this process it is common to group near pixels based on (a, b)-connected graphs as neighborhood definitions. Such an approach, however, cannot meet some topological axioms needed to ensure a correct representation of connectedness relationships. Super pixel boundaries may present ambiguities because one-dimensional contours are represented by pixels, which are 2-dimensional. The inherent complexity of segmentation algorithms along with the high volume of data of the high resolution images, demand considerable computing resources. Because of the above, geometric algorithms traditionally used for image segmentation work on 2-dimensional entities (i.e., there are neither 0-dimensional nor 1-dimensional entities to build boundaries on) and, therefore, take decisions based on topological relationships ambiguously represented.

This research managed to conceptually design and computationally implement an alternative method for multispectral image segmentation based on axiomatic locally finite spaces (ALFS) provided by Cartesian complexes, which take into account topological and geometric properties. This alternative representation model provides a geometric space that complies with the T_0 digital topology free of topological ambiguities, on which a novel way for segmenting imagery data is built. Proposed model is developed and implemented in such a way that the required subset of geometrical characteristics are transformed into combinatorial structures encoding topological and geometric features present in combinatorial half spaces using its oriented matroid. The proposed approach uses a layered architecture going from a physical level, going next through the logical geospatial abstraction level and then through the Cartesian complex logical level. Additionally, there is a layer of oriented matroids composed of conceptual elements in terms of combinatorics for encoding relevant features to multispectral image segmentation. First, it is conducted an edge detection task using a multi-scale texture analysis and oriented gradient calculation, next a spectral affinity analysis, including oriented derivative filter appliance to finally obtain a probability contour map using a Cartesian complex space rather than the pixel conventional image representation. Therefore, a computational framework by which it is possible the representation of a multispectral digital image in a way that explicitly takes into account topological properties in order to better conduct image segmentation was produced.

Accuracy assessment of boundaries produced by proposing approach was carried out through two validation strategies: (1) Segment scale generalization to the scale of the available segmentation ground truth and (2) proposed approach versus conventional pixel boundary detection benchmarking. The results show that, by departing from the conventional pixel representation, it is possible to segment an image based on a topologically correct digital space, while simultaneously taking advantage of combinatorial features of their associated oriented matroids. Even though the precision produced from Cartesian complexes still does not exceed that obtained from the conventional approach based on pixels, the approach here proposed does achieve a better recall and average precision. This permits to affirm that the model proposed and implemented as part of the research here presented constitutes a reliable alternative for the segmentation of multispectral images. It was possible to confirm that the usage of axiomatic locally finite spaces and their associated matroids enables image topological-geometric segmentation.

Keywords: Cartesian Complexes, Image Segmentation, Locally Finite Spaces, Oriented Matroids.

Resumen

En el análisis de imágenes digitales, una estrategia utilizada para abordar las propiedades espaciales y topológicas es definir objetos de imagen, como se les conoce en la comunidad de percepción remota, que agrupen píxeles como elementos más gruesos del espacio geométrico o superpíxeles. Este proceso se conoce como segmentación de imágenes. En este proceso es común agrupar píxeles cercanos usando grafos (a, b)-conectados como definiciones de vecindad. Tal enfoque, sin embargo, no puede cumplir algunos axiomas topológicos necesarios para asegurar una representación correcta de las relaciones de conexión. Los límites de superpíxeles pueden presentar ambigüedades debido a que los contornos unidimensionales están representados por píxeles, los cuales son bidimensionales. La complejidad inherente de los algoritmos de segmentación, además del gran volumen de datos típico de las imágenes de alta resolución, hacen que se requieran considerables recursos informáticos. Como resultado de lo anterior, los algoritmos geométricos tradicionalmente utilizados para la segmentación de imágenes funcionan con base en entidades bidimensionales (es decir, no hay entidades de 0 o 1 dimensiones para construir los límites) y, por lo tanto, toman decisiones con base en relaciones topológicas representadas ambigüamente.

Esta investigación logró diseñar conceptualmente e implementar computacionalmente un método alternativo para la segmentación de imágenes multiespectrales basado en los espacios axiomáticos localmente finitos (ALFS, por su sigla en inglés) proporcionados por los complejos cartesianos, y el cual toma en cuenta las propiedades topológicas y geométricas. Este modelo de representación alternativa proporciona un espacio geométrico que cumple con la topología digital T_0 libre de ambigüedades topológicas, sobre el cual se construye una nueva forma de segmentar datos de imágenes. El modelo propuesto se desarrolla e implementa de tal manera que el subconjunto requerido de características geométricas se transforman en estructuras combinatorias las cuales codifican las características topológicas y geométricas presentes en los semiespacios combinatorios usando su matroide orientado asociado. El enfoque propuesto utiliza una arquitectura por capas la cual va desde un nivel físico, pasando luego por un nivel lógico de abstracción geoespacial y luego a través del nivel lógico complejo cartesiano. Además, existe una capa de matroides orientados compuesta por elementos conceptuales en términos de combinatoria que codifican características relevantes para la segmentación de imágenes multiespectrales. Primero, se lleva a cabo una tarea de detección de bordes usando un análisis de textura y cálculo de gradiente orientado multi-escala, luego un análisis de afinidad espectral que incluye la aplicación de filtros derivativos orientados para obtener finalmente un mapa de probabilidad de límite usando un espacio complejo cartesiano en lugar de la representación de imagen convencional basada en píxeles. De esta manera, se construye un marco computacional mediante el cual es posible la representación de una imagen digital multiespectral de una manera que explícitamente toma en cuenta las propiedades topológicas para una mejor segmentación de imágenes.

La evaluación de exactitud de los límites producidos por el enfoque propuesto se llevó a cabo a través de dos estratégias de validación: (1) generalización de escala de los segmentos para llevarlos a la escala de la verdad de terreno de segmentación disponible y (2) comparación entre los resultados obtenidos a partir del enfoque propuesto y a partir de superpixeles convencionales. Los resultados muestran que, aparte de la representación de píxeles convencional, es posible segmentar una imagen con base en un espacio digital topológicamente correcto, al mismo tiempo que se aprovechan las características combinatorias de sus matroides orientados asociados. Si bien es cierto que la precisión producida a partir de complejos cartesianos aún no supera la obtenida a partir del enfoque convencional basado en píxeles, el enfoque propuesto por esta investigación logra una mejor cobertura y precisión promedio. Esto permite

afirmar que el modelo propuesto e implementado como parte de la investigación aquí presentada constituye una alternativa confiable para la segmentación de imágenes multiespectrales. Se pudo confirmar que el uso de los espacios axiomáticos localmente finitos y sus matroides asociados permite una segmentación topológico-geométrica de la imagen.

Palabras clave: complejo cartesiano, espacio localmente finito, matroide orientado, segmentación de imagen.

Acknowledgements

First, I thank my God because I have felt his support from the beginning, without which I would never have started this journey. I thank my supervisor for his guidance without which I could not have found the beginning of the road nor its end. Thanks to Professors of engineering faculty who believed in this idea.

Thanks to the nice group of colleagues of the second cohort Adriana Vega and Nelson Vera and to the colleague Edwin Millán of the first cohort for their patience, tolerance and example of which I learned a lot. Thanks to all the administrative staff and especially to Mr. Juan Pablo Ramírez who always facilitated and streamlined all the processes. Thanks to the Centro de Cómputo de Alto Desempeño (CECAD) of the Universidad Distrital and to all its staff for their support and for providing me with the computational resources without which it would have been impossible to develop the computational framework we produced.

Finally, thanks to Cristian Dallos, student of Maestría en Ciencias de la Información y las Comunicaciones, for being part of this project and for its implementation of the first version of software components for gray level image, directional derivative filters, grouping k-means and watershed transformation.

Thank you again to my God, because in all that was said before, your supernatural hand was always present.

To my wife and daughter, the only people who gave me the motivation to continue and not give up . . . and the only people by whom I would do it again.

Contents

Declaration of Authorship **v**

Abstract **ix**

Acknowledgements **xiii**

List of Abbreviations **xxix**

1 Introduction **1**
1.1 Problem statement . . . 2
1.1.1 Justification . . . 3
1.1.2 Background . . . 4
Digital image segmentation . . . 4
Geometric characteristics encoding using oriented matroids . . . 6
Computational calculation of triangulations . . . 7
Alternative models of space representation . . . 8
Terrain visibility graphs . . . 9
Comments of the state of the art . . . 10
1.2 Research question . . . 10
1.3 Hypothesis . . . 11
1.4 Research objectives . . . 11
1.4.1 General objective . . . 11
1.4.2 Specific objectives . . . 11
1.5 Research group motivation . . . 11
1.6 Thesis summary . . . 12

2 Theoretical framework **15**
2.1 Axiomatic locally finite spaces . . . 16
2.1.1 ALFS properties . . . 18
2.1.2 Abstract cell complexes . . . 18
2.1.3 Cartesian complexes and combinatorial coordinates . . . 19
2.1.4 Combinatorial balls and spheres . . . 21
2.1.5 Combinatorial homomorphism . . . 22
2.1.6 Generalized combinatorial boundary and combinatorial boundary of a space . . . 24
2.1.7 Orientation of AC complexes . . . 24
2.1.8 Block complexes . . . 25

2.1.9 Completely connected spaces . . . 26
2.1.10 Connectedness preserving mapping . . . 27
2.1.11 Digital geometry based on topology of complexes . . . 30
2.1.12 Digital analytical geometry . . . 31
2.1.13 Linear inequalities in Locally Finite Spaces . . . 34
2.2 Digital image segmentation . . . 38
2.2.1 Image segmentation process . . . 39
2.2.2 Oriented edge-detection based image segmentation . . . 40
2.2.3 Oriented gradient based boundary strength . . . 41
2.2.4 Normalized cut based global cues . . . 41
2.2.5 Watershed transform based image partitioning . . . 42
2.3 Preserved topological and geometric properties by contour detection and hierarchical image segmentation . . . 43
2.4 Oriented matroids as geometric feature encoding structures . . . 47
2.4.1 Matroids . . . 47
2.4.2 Matrices and matroids . . . 48
2.4.3 Graphic matroids . . . 49
2.4.4 Affine matroids . . . 50
2.4.5 Oriented matroids . . . 51
Central hyperplane arrangement and sphere arrangement . . . 52
What is an oriented matroid? . . . 54

3 Image segmentation conceptual model based on ALFSs and oriented matroids 57
3.1 Cartesian complex topology (CCT) . . . 58
3.2 Cartesian complex geometry (CCG) . . . 61
3.3 Oriented matroid layer (OML) . . . 65
3.3.1 Central SSF arrangement . . . 65
3.3.2 Oriented matroid . . . 69
3.4 Layer of segmentation tasks (STL) . . . 75
3.4.1 Directional filtering (DF) . . . 76
Filter kernel . . . 76
Bank of Directional filters . . . 79
3.4.2 Oriented gradient (OG) Cartesian complex . . . 81
3.4.3 Boundary cell affinity (BCA) . . . 85
3.4.4 Boundary probability . . . 88
Multiscale Boundary geometrical probability (MBGP) . . . 89
Boundary spectral probability (BSP) . . . 90
Boundary global probability (BGP) . . . 91
3.4.5 Watershed transform . . . 91

4 Computational framework based on ALFS and oriented matroid 93
4.1 Cartesian complex . . . 94
4.1.1 Topology . . . 95
4.1.2 Geometry . . . 97
4.2 Matroid package . . . 97

4.3 Segmentation task layer (STL) . . . 98
4.3.1 Parallelism scheme . . . 98
4.3.2 Monochromatic package . . . 99
4.3.3 Directional filtering package . . . 100
4.3.4 Oriented gradient Cartesian complex package . . . 101
4.3.5 Boundary cell affinity package . . . 102
4.3.6 Boundary probability package . . . 103
4.3.7 Clustering package . . . 104
4.3.8 Watershed package . . . 104
4.3.9 Utility package . . . 105
4.4 Discussion of the computational framework implementation . . . 106
4.4.1 Representation of a multispectral image as a Cartesian complex . . . 106
4.4.2 Link between an central SSF arrangement and its associated oriented matroid . . . 107
4.4.3 Multispectral image segmentation tasks based on 1-cells . . . 107

5 Results and Validation of proposed model 109
5.1 Dataset . . . 109
5.1.1 Pavia dataset . . . 110
5.1.2 Multispectral image dataset . . . 110
Vaihingen dataset . . . 111
Potsdam dataset . . . 111
5.2 Computational outcomes of segmentation tasks . . . 113
5.2.1 Cartesian complex conversion (1) . . . 113
5.2.2 Gray scale conversion (2) . . . 114
5.2.3 Multiscale texture analysis (3) . . . 115
5.2.4 Texture grouping (4) . . . 116
5.2.5 Multiscale oriented gradient (5) . . . 116
5.2.6 Boundary Oriented (6) and maximum (7) geometrical strength . . . 117
5.2.7 Oriented affinity analysis (8) . . . 118
5.2.8 Boundary oriented global strength (9) . . . 118
5.2.9 Hierarchical segmentation (10) . . . 119
5.3 Validation of proposed segmentation model . . . 119
5.3.1 Segment scale generalization . . . 120
5.3.2 Boundary detection benchmarking . . . 122
Multispectral gradient validation . . . 123
Texture gradient validation . . . 124
Geometric and spectral boundary strength validation . . . 125
Global boundary probability . . . 127

6 Discussion 129
6.1 Objective achievement . . . 129
6.1.1 Filtering stage . . . 130
6.1.2 Gradient half-subspace definition . . . 131
6.1.3 1-cell affinity . . . 131
6.1.4 Oriented matroid-based gradient calculation . . . 132

6.1.5 Watershed calculation . . . 133
6.1.6 Parallel computing scheme . . . 133
6.1.7 Specific objectives achievement . . . 134
6.2 Testing the hypothesis . . . 134
6.2.1 Segment scale generalization . . . 135
6.2.2 Boundary detection benchmarking . . . 136
6.3 Discussion of computational prototype outcomes . . . 137

7 Conclusions **139**

8 Main contributions **141**
8.1 Conceptual model . . . 141
8.2 Image analysis . . . 142
8.2.1 Support for Cartesian complex space representation . . . 142
8.2.2 1-cell filtering . . . 142
8.2.3 Oriented gradient calculation . . . 143
8.2.4 Affinity analysis . . . 143
8.2.5 Boundary strength calculation . . . 143
8.2.6 Segmentation calculation . . . 143
8.2.7 Octave software support . . . 143
8.3 Socialization of research results . . . 144
8.4 Academic contribution . . . 145

9 Future work and recommendations **147**

Bibliography **149**

A Implementation source code **159**

List of Tables

2.1 Mapping between two examples of 1-dimensional complexes X and Y 27
2.2 Noncontinuous mapping between two 1-dimensional complexes X and Y 28
2.3 Noncontinuous and simple mapping between two 1-dimensional complexes X and Y . . 29
2.4 Mapping between 1-dimensional subcomplexes displayed in Figures 2.14(a) and 2.14(c) 33

3.1 SSF family for the central SSF arrangement in Figure 3.10(b) 68
3.2 SSF family for vertical central SSF arrangement in Figure 3.11 69
3.3 Subspace values for central SSF arrangement on Figure 3.13 73
3.4 SSFs defined in terms of subspaces of Table 3.3 74
3.5 SSFs defined in terms of subspaces of Figure 3.14 74

5.1 Vaihingen windows extracted dataset 111
5.2 Potsdam windows extracted dataset 112
5.3 Segment intersecting boundary ground truth geometry assessment includes several segmentations (*one by row*) 121
5.4 Boundary assessment reports precision, recall, and F-measure; region quality reports covering of ground-truth (*GT covering*) 122
5.5 Execution time measured for three radii (*in pixels for Matlab and cracks for proposed*) and several orientations (*always the maximum*) 122
5.6 Multispectral multi-scale oriented gradient assessment summary. 123
5.7 Texture multi-scale oriented gradient assessment summary. 125
5.8 Geometric and spectral boundary detection assessment summary. 126
5.9 Boundary detection assessment summary for Potsdam and Vaihingen dataset. 127

List of Figures

1.1 Connectedness definition . . . 3
1.2 Boundary definition . . . 3
1.3 Counterclockwise order of CCS . . . 7
1.4 Conventional pixels versus Cartesian complex space representations . . . 12
1.5 Combinatorial approach includes Cartesian complex geometry to oriented matroid position vector transformation . . . 13

2.1 Graphical representations of a 1-dimensional connected ACC . . . 20
2.2 Graphical representations of a 2-dimensional Cartesian ACC and its dual . . . 21
2.3 Elementary cell subdivision in a 2-dimensional Cartesian ACC . . . 23
2.4 Example of block complex on a 2-dimensional Cartesian complex . . . 26
2.5 A principal 2-cell is isomorphic to a single polytope . . . 26
2.6 Complexes X (*a*) and Y (*b*) and graph (*c*) for mapping presented in Table 2.1 . . . 28
2.7 Complexes X (*a*) and Y (*b*) and graph (*c*) for mapping presented in Table 2.2 . . . 28
2.8 Complexes X and Y (a,b) and graph (c) for mapping presented in Table 2.3 . . . 29
2.9 B-isomorphism example between 2-dimensional complex and 1-dimensional complex . . . 29
2.10 Incidence structure for 0-cell (dark gray encircled dot), 1-cell (red encircled dot), and 2-cell (green encircled dot) in 2-dimensional Cartesian complex . . . 30
2.11 Solid and nonsolid subcomplexes . . . 30
2.12 VCU widths of “0”, “1”, “2”, and “3” . . . 31
2.13 Subcomplex translation . . . 32
2.14 1-dimension subcomplex magnification . . . 32
2.15 2-dimension subcomplex magnification . . . 33
2.16 Subcomplex rotation . . . 34
2.17 Collinearity . . . 35
2.18 Semi-collinear cells with vector **(n, m) = (8, 10)** . . . 35
2.19 Half-plane for $5x - 4y \geq 0$ . . . 36
2.20 Adjacent pixels (2-cells) of boundary curve (red dots) - Positive (dots with blue round background) and negative (dots with green round background) pixels (2-cells) . . . 36
2.21 Bases of a DSS Positive (*dots with blue round background*) and negative (*dots with green round background*); $H(x,y)$ (*blue headed line*) and $H'(x,y)$ (*green headed line*) . . . 37
2.22 Strength boundary-based edge detection pipeline . . . 44
2.23 UCM production pipeline . . . 44
2.24 Non-isomorphic graphs with isomorphic cycle matroid . . . 50
2.25 Graph with its vertex-edge incidence matrix . . . 50
2.26 Dual graph of graph displayed in Figure 2.25 . . . 51

2.27 Point set in plane isomorphic to uniform matroid $U_{3,8}$, except $\{\{b,d,h\},\{a,d,e\}\}$ 51
2.28 Central hyperplane arrangement 52
2.29 Sphere arrangement in 2-dimensional space (in $\mathbb{R}^2$) 53
2.30 (a) System of spheres and (b) their respective realization as a polytope $\boldsymbol{P_A}$ 54
2.31 Face correspondence between sphere arrangement and their zonotopes (In $\mathbb{R}^2$) 54

3.1 Multispectral image segmentation conceptual model 57
3.2 2D image represented in 2D Cartesian complex 59
3.3 Conceptual elements in CCT layer 59
3.4 Coordinate orientation 61
3.5 Conceptual elements in CCG layer 62
3.6 Subsets produced by conventional SSF definition 62
3.7 Subsets produced by SSF definition - nearest non-SSF 1-cell case 63
3.8 Subsets produced by SSF definition - farthest on-SSF 1-cell case 64
3.9 Conceptual elements in the OML layer 65
3.10 Number of SSFs in central SSF arrangement 66
3.11 Vertical central SSF arrangement 69
3.12 Central SSF arrangement formation 70
3.13 SSF arrangement subspace indexing 71
3.14 Vertical 1-cell centered SSF arrangement subspaces indexing 75
3.15 Subcomponents of the layer of segmentation tasks 75
3.16 Directional filter conceptual elements 76
3.17 Directional Prewitt kernel 77
3.18 $\frac{\pi}{10}$ rotated second derivative Gaussian filter kernel 78
3.19 Prewitt directional filter bank for $n = 1$ and $o = 2$ 80
3.20 Gaussian directional filter bank 80
3.21 Oriented ball division definition for calculating gradient Cartesian complex for central SSF arrangement with three SSFs 81
3.22 Oriented gradient conceptual elements 82
3.23 Oriented ball division definition for calculating gradient Cartesian complex for central SSF arrangement with three SSFs 83
3.24 Conceptual elements in BCA component 85
3.25 Affinity search area defined using Manhattan distance-based ball 86
3.26 Conceptual elements for boundary probability 88
3.27 Cracks (*dots in darker area*) in the union of SON (*clearer area*) of elements in central 1-cell closure (*all circled dots*), except the central 1-cell, used in exploring basin expansion from the central 1-cell (*central circled dot*). 92

4.1 Package hierarchy of conceptual elements 93
4.2 Cartesian complex package elements 94
4.3 Cell structure 95
4.4 Cartesian complex structure 96
4.5 Geometry elements structure 97
4.6 Matroids package elements 98
4.7 Space element grouping by CPU 99

4.8 Utilitarian package structure . . . 100
4.9 Filtering package structure . . . 100
4.10 Oriented gradient package structure . . . 101
4.11 Affinity analysis package structure . . . 102
4.12 Boundary probability package structure . . . 103
4.13 Boundary probability process . . . 103
4.14 Clustering package structure . . . 104
4.15 Watershed package structure . . . 105
4.16 Utilitarian package structure . . . 105

5.1 Dataset used covers Pavia Centre (Italy) (*first principal component of hyperspectral image on left and ground truth image on right*) . . . 110
5.2 Input image window to segmentation tasks . . . 113
5.3 Cartesian complex conversion . . . 114
5.4 Gray scale conversion . . . 114
5.5 Gaussian filtering-based texton analysis . . . 115
5.6 Texture grouping using k-means . . . 116
5.7 Maximum oriented multispectral and texture gradient outcomes . . . 117
5.8 Boundary oriented geometrical strength . . . 117
5.9 Outcome affinity analysis eigenvectors . . . 118
5.10 Boundary oriented spectral probability . . . 118
5.11 Boundary oriented global probability . . . 119
5.12 Cartesian complex segmentation outcome . . . 119
5.13 Boundary strength image (*left*) and multilevel segmentation image (*on right: over, "9" height ranges and "5" height range segmentations*) . . . 121
5.14 Boundary detection assessment pipeline . . . 123
5.15 Multispectral multi-scale oriented gradient assessment (*from left to right radii of "3", "5" and "10"* 1-*cells/pixels*) for Potsdam (top) and Vaihingen (bottom). . . . 124
5.16 Oriented Texture multi-scale oriented gradient assessment(*from left to right radii of "5", "10" and* 20 1-*cells/pixels*). . . . 125
5.17 Geometric (left) and spectral (right) boundary detection assessment. . . . 126
5.18 Boundary detection assessment for Potsdam (left), Vaihingen (middle) and both (right) dataset. . . . 127

6.1 Conventional versus Cartesian complex Sobel filter kernel definition . . . 130
6.2 Conventional versus Cartesian complex gradient half-space definition . . . 131
6.3 Conventional versus Cartesian complex connecting line in affinity analysis . . . 132
6.4 Affinity between pair of cells (*red cells are centers*) . . . 132
6.5 Conventional versus Cartesian complex basin expansion . . . 133
6.6 Parallel execution scenarios ($m < n < p$) . . . 134

List of Equations

1.1 Chirotope to transfer the geometry to the combinatoric . . . 8
2.1 1D bounding relationship . . . 20
2.2 2D bounding Relationship . . . 20
2.3 Closed *m*-dimensional AC sphere . . . 22
2.4 Segmentation predicate . . . 39
2.5 Convolution operator . . . 40
2.6 Convolution operator compact notation . . . 40
2.7 Surface gradient . . . 40
2.8 Smoothed image gradient . . . 40
2.9 Gaussian first derivative . . . 41
2.10 Oriented Gaussian first derivative . . . 41
2.11 Local histogram distance . . . 41
2.12 Cut between two pixel groups . . . 41
2.13 Normalized cut . . . 42
2.14 Generalized eigenvalue system . . . 42
2.15 Geodesic distance to a set . . . 42
2.16 Geodesic influence zone . . . 43
2.17 Skeleton by influence zone . . . 43
2.18 Multiscale oriented image . . . 45
2.19 Maximum *mPb* image . . . 45
2.20 Pixel affinity measurement . . . 45
2.21 Boundary detector spectral component . . . 45
2.22 Global probability of boundary . . . 46
2.23 Maximum global probability of boundary . . . 46
3.1 Cell value vector . . . 59
3.2 Cell type . . . 60
3.3 SON definition . . . 60
3.4 XSON scope . . . 61
3.5 CONNECTED scope . . . 61
3.6 SCN definition . . . 61
3.7 Conventional subspaces . . . 62
3.8 SSF shifting . . . 63
3.9 H value at a 1-cell on an SSF . . . 64
3.10 H value at a vertical 1-cell on a SSF . . . 64
3.11 SSF subspaces . . . 64
3.12 SSF sign subspaces . . . 65

3.13 Central 1-cell combinatorial coordinate . . . 66
3.14 Number of SSFs based on 0-cells . . . 66
3.15 Number of SSFs based on outer 1-cells . . . 67
3.16 Outer 1-cell set definition . . . 67
3.17 SSF family definition . . . 67
3.18 SSF vertical difference . . . 67
3.19 SSF horizontal difference . . . 67
3.20 a parameter definition . . . 67
3.21 b parameter definition . . . 67
3.22 a parameter symmetric definition . . . 68
3.23 b parameter symmetric definition . . . 68
3.24 l parameter definition . . . 68
3.25 Vertical SSF vertical difference . . . 68
3.26 SSF family sampling rate . . . 69
3.27 SSF_i sign subspaces . . . 70
3.28 Sign value . . . 70
3.29 Position value vector . . . 70
3.30 On positive half SSF_j^0 position value definition . . . 71
3.31 On positive half SSF_j^+ position value definition . . . 71
3.32 On negative half SSF_j^0 position value definition . . . 72
3.33 On negative half SSF_j^+ position value definition . . . 72
3.34 Position value vector on negative side . . . 72
3.35 central SSF arrangement all reference line subspaces indexing . . . 72
3.36 Sign vector value as a 3 base number . . . 72
3.37 Sign vector value for half SSF_j^0 . . . 73
3.38 Sign vector value for half SSF_j^+ . . . 73
3.39 central SSF arrangement subspace definition . . . 74
3.40 SSF subset definition . . . 74
3.41 Filter kernel definition . . . 77
3.42 Prewitt neighbourhood definition . . . 77
3.43 Prewitt kernel value . . . 77
3.44 Vertical one dimensional kernel . . . 78
3.45 Horizontal one dimensional kernel . . . 78
3.46 Gaussian neighbourhood definition . . . 78
3.47 Gaussian kernel value . . . 78
3.48 Center surround kernel . . . 78
3.49 Two dimensional kernel application . . . 79
3.50 Two dimensional kernel value vector . . . 79
3.51 Prewitt kernel value for both sides . . . 79
3.52 Filter bank Cell value vector . . . 81
3.53 Filtered Cartesian complex . . . 81
3.54 Gradient neighbourhood definition . . . 82
3.55 Histogram partitions . . . 82
3.56 Subspace histogram . . . 82
3.57 s_j to SSF_i sides assignment . . . 83

3.58 SSF + side histogram . . . 83
3.59 SSF - side histogram . . . 83
3.60 Oriented gradient at c 1-cell . . . 84
3.61 Gradient cell value vector . . . 84
3.62 Oriented gradient Cartesian complex . . . 84
3.63 SSF + side histogram by channel . . . 84
3.64 SSF - side histogram by channel . . . 84
3.65 Oriented gradient at c 1-cell by channel . . . 84
3.66 Gradient cell value vector by channel . . . 84
3.67 Maximum oriented gradient at a cell . . . 84
3.68 Maximum oriented gradient Cartesian complex . . . 85
3.69 Affinity neighbourhood definition . . . 86
3.70 Affinity between pairs of 1-cells . . . 86
3.71 Affinity a_j parameter . . . 86
3.72 Affinity b_j parameter . . . 86
3.73 Displacement direction to central 1-cell . . . 87
3.74 Previous 1-cell location on turn . . . 87
3.75 Previous 1-cell location along even axis . . . 87
3.76 Affinity with respect to previous 1-cell . . . 87
3.77 Eigen vector cell value vector by channel . . . 87
3.78 Generalized eigenvector Cartesian complex . . . 88
3.79 Multiscale filter bank Cell value vector . . . 89
3.80 Multi-scale Filtered Cartesian complex . . . 89
3.81 Weighted sum of multispectral and texture channels gradient . . . 89
3.82 Oriented geometric probability Cartesian complex . . . 90
3.83 Maximum geometrical probability at a cell . . . 90
3.84 Maximum geometrical probability Cartesian complex . . . 90
3.85 Weighted sum of oriented filtered eigenvector Cartesian complex . . . 90
3.86 Oriented spectral probability Cartesian complex . . . 90
3.87 Boundary oriented global probability at a cell . . . 91
3.88 Oriented global probability Cartesian complex . . . 91
3.89 Global boundary probability at a cell . . . 91
3.90 Maximum oriented global probability Cartesian complex . . . 91
3.91 Watershed expansion neighborhood definition . . . 91
4.1 Minimum 2-cell group size . . . 99
4.2 Maximum 2-cell group size . . . 99
5.1 Superpixel geometry assessment . . . 121

List of Abbreviations

0D	Zero-dimensional
1D	One-dimensional
2D	Two-dimensional
AC	Abstract Cell
ACC	Abstract Cell Complex
ALFS	Axiomatic Locally Finite Space
AOS	Aerial Oblique System
API	Application Program Interface
BCA	Boundary Cell Affinity
BGP	Boundary Global Probability
BP	Boundary Probability
BSP	Boundary Spectral Probability
CCG	Cartesian Complex Geometry
CCS	Counter Clockwise System
CCT	Cartesian Complex Topology
CPM	Connectedness Preserving Mapping
CPU	Central Processing Units
DAIS	Digital Airborne Imaging System
DF	Directional Filtering
DSS	Digital Straight-Line Segment
FNEA	Fractal Network Evolution
GDAL	Geoespatial Data Abstraction Library
GIS	Geographic Information Systems
GPS	Global Positioning System
GPU	Graphical Processing Units
HCC	Homogeneous Completely Connected
HR	High Resolution
IJIMAI	International Journal of Interactive Multimedia and Artificial Intelligence
ISPRS	International Society for Photogrammetry and Remote Sensing
JRE	Java Execution Environment
LCP	Linear Complementarity
LFS	Locally Finite Space
LP	Linear Programming
MBGP	Multiscale Boundary Geometrical Probability
MSNI	Mobile Semantic Navigation and Information
NIDE	Núcleo de Investigaciones en Datos Espaciales

OBIA	Object-based Image Analysis
OF	Opening Frontier
ODS	Optimal Dataset Scale
OIS	Optimal Image Scale
OM	Oriented Matroid Layer
OMP	Oriented Matroid Programming
OWT	Oriented Watershed Transform
POSET	Partially Ordered Set
QP	Quadratic Programming
REST	Representational State Transfer
RAG	Region Adjacency Graph
SN	Smallest Neighborhood
SON	Smallest Open Neighborhood
SSF	Standard Separating Form
STL	Segmentation Tasks Layer
TOP	True Ortho-Photo (TOP)
TOPCOM	Triangulations of Point Configurations and Oriented Matroids
UCM	Utrametric Contour Map
UML	Unified Modelling Language
VCU	Visual Curve

Chapter 1

Introduction

The digital analysis of high-resolution (HR) images relies largely on a nearby pixel grouping scheme to form elements of coarse spatial resolution known as superpixels or image-objects, this process is known as image segmentation (Lézoray and Grady, 2012; Lizarazo and Elsner, 2009; Brun, Domenger, and Mokhtari, 2003). The remote sensing community has adapted the computer vision segmentation approach for dealing with multispectral images (Lizarazo and Elsner, 2011). A common strategy is to define coarser geometric space elements by grouping near pixels based on (a, b)-connected graphs as neighborhood relationships. However, ambiguities could appear when using such a space representation because (a, b)-connected graphs may not address all the topological axioms required (Kovalevsky, 2008). Specifically, digital image segmentation is heavily dependent on proper boundary definitions, which are difficult to obtain in a two-dimensional (2D) space from structures such as (a, b)-connected 2D-elements graphs (Kovalevsky, 1989). Hence, the geometric algorithms traditionally used for image segmentation function on 2D entities ,i.e., there are neither zero-dimensional (0D) nor one-dimensional (1D) entities to build boundaries on. Therefore, the geometric algorithms make decisions based on topological relationships ambiguously represented.

Having an adequate space representation is not the only obstacle to overcome. Multi and hyperspectral images have a large number of spectral bands and today it is common to have increasing spatial and radiometric discernibility (Wang and Wang, 2015; Ren et al., 2014; Plaza et al., 2005). So, multispectral image segmentation process multiband image data sets wohose channels usually beyond the visible range common in computer vision. Although this means that the ambiguity is decreasing, it also means that for a given geographic area, the volume of data that must be processed during multi and hyperspectral image segmentation is considerably large, even after data reduction stages such as principal component analysis. This ambiguity reduction suggests that the analysis performed by segmentation algorithms, e.g., texture analysis, should achieve improved results; however, high resolution could require intensive use of computing resources (Kaewpijit, Moigne, and El-Ghazawi, 2003). Even with the computing power available today, the algorithmic complexity of segmentation analysis and the large volume of data to be processed can cause problems of lack of memory or delay in obtaining results.

This document presents a doctoral research aimed at contributing to the solution of the two limitations influencing image analysis as summarized in the two preceding paragraphs. In particular, this document describes the research undertaken into two interconnected topics: (1) the development of an alternative conceptual model for image representation and segmentation and (2) the design and implementation of a computational framework for topologically aware image segmentation.

This introductory chapter is divided into six sections. Sections 1.2 to 1.5 correspond to what was proposed and approved at the research institutionalization stage, presented herein with minor changes. Section 1.2 presents the research question. Section 1.3 states the main hypothesis. Section 1.4 formulates the research objectives. Section 1.5 presents the research group motivation and Section 1.6 summarizes this thesis document.

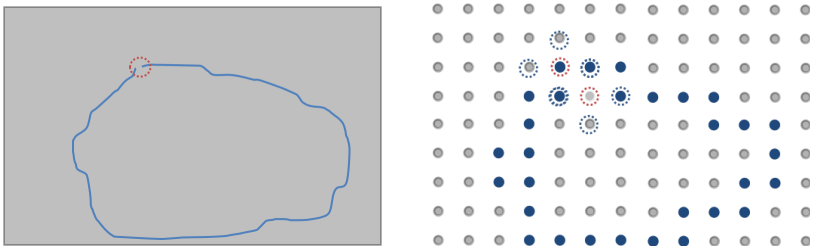
1.1 Problem statement

The land cover classification of remote sensing images has been traditionally conducted using primarily the spectral information associated with discrete spatial units (i.e., pixels). The spatial context surrounding the pixels was ignored for many years. However, in the last fifteen years, there have been a number of attempts to include both the geometric and topological characteristics of the pixel neighborhood in the image analysis process (Jong and Meer, 2004). This can be explained by the increasing availability of remotely sensed high spatial resolution images. In recent years, a number of concepts, methods, and techniques for object-based image analysis (OBIA) have been developed and evaluated (Blaschke, 2009). The OBIA approach is particularly useful for HR image analysis, i.e., when the pixel size is less than the objects of interest. The OBIA analysis process begins by grouping spectrally similar and spatially close pixels into segments. A meaningful segmentation is essential for the subsequent analysis stages (Lizarazo and Elsner, 2009).

Usually, image segmentation algorithms assume that the images are a continuous space, similar to the real world they depict. Unfortunately, the digital representation that is ultimately available in any computer is made by sampling the space and discretizing reality. Further, it is assumed that multi and hyperspectral digital images are composed of square picture elements of similar size, which might not be the case (Cracknell, 1998). Inaccuracies may also be present in representations of images such as 4-connected, 8-connected, and (a,b)-connected graphs (Rosenfeld, 1970; Kong and Rosenfeld, 1991). Figure 1.1(a) displays, in a continuous space, how by withdrawing points from the boundary between the interior and exterior sets, previously disconnected by a Jordan Curve, these sets become connected. Figure 1.1(b) displays, in a conventional digital space and using the 4-connected criterion, how by removing the element encircled in red from the boundary (depicted using blue points),the corresponding interior and exterior sets do not become connected; this is illustrated using two neighborhoods (the central element of each is depicted as a point encircled in red). In fact, in a conventional digital space, it is also not possible to establish unambiguously such a connectivity using the 8-connected criterion. In this case, for the example displayed, the two sets remain connected, even without removing the element under consideration from the boundary.

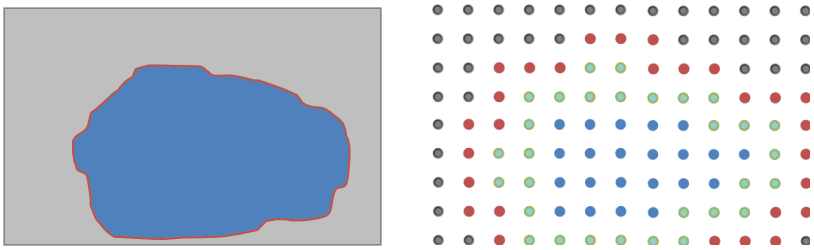
Figure 1.2(a) indicates how, in a continuous space, the boundaries of the interior and exterior sets are 1D and coincide perfectly (red line). Figure 1.2(b) illustrates how, in the conventional digital space, each of the two sets have a different boundary (exterior boundary is red and interior boundary is green), and that both boundaries are 2D (all the elements comprising this space are 2D). These ambiguities are particularly significant for image segmentation when considering topological relationships such as neighborhood, contiguity, connectivity, and boundary.

Digital image processing based on representations of space that do not adhere to the topological axiomatic postulates causes geometric algorithms to be influenced by the ambiguities that lead to erroneous



(a) Continuous space (b) Conventional digital space

FIGURE 1.1: Connectedness definition



(a) Continuous space (b) Conventional digital space

FIGURE 1.2: Boundary definition

decisions (Munkres, 1999). Although it is commonly accepted that a raster image matches the human perception to a degree, the truth is that several important concepts for image analysis, such as the connectivity of regions, their boundaries, and the adjacency between them, are not explicitly presented. This lack of topological awareness of pixel-based image representation is clearly an important limitation for image analysis in computer environments (Kovalevsky, 1989). Furthermore, it is well known that algorithms based on geometric characteristics have high complexity because the number of possible situations to be considered increases significantly, as it increases with the object dimension under evaluation (Worboys and Duckham, 2004). Another problem that arises during the implementation phase of geometric algorithms is the impossibility of having an accurate arithmetic in terms of real numbers. This results in the use of an arithmetic in terms of integers or rational numbers, which decreases the performance (Berg et al., 2008). In summary, conventional methods for multispectral image segmentation are based on digital structures that do not adhere to well-established topological axioms and geometric algorithms that assume a continuous space-computing model. This research addresses this problem.

1.1.1 Justification

Kovalevsky (2006) suggested an alternative space representation using axiomatic locally finite spaces (ALFS) based on Cartesian complexes, a type of abstract cell complexes (ACC). It has been stated that this alternative space avoids the topological ambiguities commonly found in conventional solutions (Pavlidis, 1977; Kovalevsky, 1984). Furthermore, it has been suggested that this alternative space could be a strong foundation for image segmentation in computer vision (Kovalevsky, 2006). However, to the author's best knowledge, the usage of such ALFS has not yet been explored for segmentation of remotely sensed

multispectral images. Therefore, it was relevant to investigate the potential of ALFS to conduct a multispectral image segmentation including topological and geometric relationships in addition to spectral attributes. A number of authors have attempted to reduce the algorithmic complexity of geometric spaces by selecting a subset of geometrical characteristics and transforming these into combinatorial structures using oriented matroids (Santos, 1997; Smyth and Webster, 2007; Stell and Webster, 2007; De Loera, Rambau, and Santos, 2010). However, this potential has not been explored in its application to the solution of performance problems occurring in segmentation of large image datasets. Although previous work has demonstrated the possibility of addressing a portion of the geometric properties using purely combinatorial structures provided by the oriented matroids, it is necessary to establish if they are suitable for representation of the topological properties present in a multispectral image. A preliminary review of finite sets based on mathematical foundations that support matroids, conducted during this research, suggested that their potential for improving computational implementation models for multispectral image segmentation was high. The successful completion of this research provided concepts and computational techniques to address problems that are overly complex when the conventional geometric model is considered as the basis. The community of geo-spatial specialists can benefit from having an alternative spatial framework potentially useful for solving problems where the conventional representations and algorithms have limitations or are even unusable. Although there has been a significant amount of research on the usage of oriented matroids in geospatial applications, such as the case of Knuth (1992) and Pfeifle and Rambau (2002), having identified an alternative space representation associating the oriented matroids with the Cartesian complexes proposed in (Kovalevsky, 2008) has resulted in a valuable contribution for alternative multispectral image processing techniques.

1.1.2 Background

Digital image segmentation

Kovalevsky (1989) demonstrated how abstract cell complexes allow for implementing the topological relations required to unambiguously perform the segmentation of digital images in computer vision. In fact, more recent work has included the development of several algorithms for color image segmentation (Kovalevsky, 2001; Kovalevsky, 2005). However, the author of this research is not aware of recent studies that rigorously evaluated the appropriateness of Kovalevsky´s approach to multispectral image segmentation. Felzenszwalb and Huttenlocher (2004) addressed the image segmentation problem by defining a predicate for measuring the evidence for a boundary between two regions, developing a greedy algorithm, and using a graph-based representation of the image (Urquhart, 1982; Zahn, 1971). An important characteristic of this method was its ability to preserve detail in low-variability image regions, while ignoring the detail in high-variability regions. The evidence for a boundary between two regions was measured by comparing the intensity differences across the boundary, and intensity differences between neighboring pixels within each region.

Yu et al. (2006) assessed the capacity of Digital Airborne Imaging System (DAIS) HR images and topographic data for vegetation classification. Image objects were generated using the fractal network evolution (FNEA) segmentation approach and considered spectral, textural, topographic, and geometric attributes. Initially, for FNEA, each pixel was an image object; then, pairwise objects were merged step-by-step to form larger objects using a merging criterion in such a manner that the average image objects heterogeneity weighted by their size in pixels was minimized (Baatz and Schäpe, 1999; Baatz

and Schape, 2000; Benz et al., 2004). Through a hierarchical classification scheme and set of features selected for each vegetation category, the authors performed a detailed classification obtaining considerably more accurate results than those provided by the conventional nearest neighbor and maximum likelihood methods. Furthermore, they claimed full correction of the salt-and-pepper effect present in the latter classification technique.

Kong, Xu, and Wu (2006) extracted land use information from a high spatial resolution image using a multiscale image segmentation approach. Urban land use was divided into different levels forming a hierarchical network structure, where objects on the upper level were composed of objects in the lower level. Classification of image objects was performed based on the attributes of color, shape, hierarchy, and correlation characteristics between neighboring objects. Results demonstrated that by including a variety of spectral and spatial features, it was possible to differentiate urban land use categories that could not be separated through conventional per-pixel classification methods based on spectral data only.

Letscher and Fritts (2007) introduced a hybrid *split-merge* method for image segmentation based on computational geometry and topology using persistent homology. The algorithm initially used edge-directed topology to split the image into three types of regions based on the Delaunay triangulations of points in the edge map. One type of region corresponded to objects of interests; the remaining two types corresponded to smaller regions, which could be attached either to the first regions or to themselves to form new objects. Preliminary results demonstrated a high quality image segmentation.

Johansen et al. (2007) assessed the capacity of high spatial resolution satellite images to discriminate structural stages of vegetation in forest ecosystems. Based on semivariogram experiments (Tso and Mather, 2009), they established that the most appropriate window sizes for textural analysis were 3×3 and 11×11 pixels. They subsequently applied an object-oriented spectral and textural classification algorithm to produce a map of the structural vegetation classes. The joint use of spectral and texture features improved thematic accuracy between 2% and 19% compared to the accuracy based only on spectral features.

Li and Sun (2010) proposed an "active image" segmentation method that distorted the image to match the "initial" outlines and could segment multiple objects simultaneously. The deformation field was modeled using B-Spline free-form deformations. By penalizing the bending energy, they claimed to preserve both the shape and local topology of the objects of interest. Preliminary results, obtained using both synthetic and real images, indicated that the proposed method could address low-contrast and occlusion issues that could not be overcome using image segmentation based on simpler criteria.

Arbelaez et al. (2011) presented a unified approach to contour detection and image segmentation. To produce high-quality image segmentations, the contour detector was linked with a generic grouping algorithm consisting of two steps. In the first, a new image transformation called the Oriented Watershed Transform (OWT) was introduced for building a hierarchical segmentation by exploiting the information in the contour signal (Arbelaez et al., 2009). In the second step, using an agglomerative clustering procedure, an initial graph was built where the nodes were the initial regions, the links were the initial arcs separating adjacent regions, and the weights were a measure of the dissimilarity between regions. The algorithm proceeded by sorting the links by similarity and iteratively merging the most similar regions. The process produced a tree of regions where the leaves were the initial regions, the root was the entire image, and the inclusion relation ordered the regions in a multiscale fashion.

Chen, Freedman, and Lampert (2011) proposed a new method for integrating topological image properties within a segmentation model of random field image, which did not pose topological restrictions in the energy minimization stage. Using this approach, they claimed the achievement of an image segmentation that guaranteed topological properties. It should be noted, however, that they used a graph-based image representation based on the conventional (and inaccurate) 4-connected and 8-connected neighborhood relationships. Nunez-Iglesias et al. (2013) proposed a bottom-up strategy improving the agglomeration using more information than the boundary only. Using supervised machine learning techniques, they predicted whether two super pixels should be merged. In the case of obtaining feature-combination non-matching training data, additional training examples were generated by applying an active learning paradigm at every agglomeration hierarchy level. In active learning, the algorithm determined what example it wanted to learn from, at each iteration, based on the previous training data. The learning process was validated for accuracy using a coarse-scale ground truth image. They reported the usage of a region adjacency graph (RAG); however, they did not report any special topological representation for the RAG.

In summary, a significant amount of work has been performed addressing the segmentation problem. However, as far as the author has been able to know, except Kovalevsky's proposal, all other proposed approaches that were reviewed have used a space representation based on conventional pixels or employed the graph-based image representation using 4- and 8-connected connectivity, which suffer from the connection and boundary ambiguities described before (Figures 1.1(b) and 1.2(b)).

Geometric characteristics encoding using oriented matroids

Oriented matroids have been used to extend the theory of linear programming (LP). Bachem and Kern (1992) fostered this advancement considerably. It has given rise to oriented matroid programming (OMP), and in the process of its formulation, many finite algorithms for LP have been discovered. OMP has been extended to generalize both the quadratic programming (QP) and linear complementarity (LCP) theories (Fukuda, 2004). In addition to this field of application, several works have indicated interest in the usage of oriented matroids as finite mathematical constructions to formulate alternative models of space representation and devise computational tools to solve efficiently spatial analysis problems.

Counterclockwise systems Knuth (1992) conducted research into the relationship between geometric computing and matroids in his work on counterclockwise systems (CCS). A CCS was formed by a finite set with a ternary relationship (it was written xyz to indicate that (x,y,z) was part of the relationship or $\neg xyz$ to indicate that it was not) such that:

$(CC1) xyz \Rightarrow yzx$,
$(CC2) xyz \Rightarrow \neg xzy$,
$(CC3) xyz \vee yxz$,
$(CC4) zxy \wedge wzy \wedge wxz \Rightarrow wxy$,
$(CC5) zyv \wedge zyw \wedge zyx \wedge zvw \wedge zwx \Rightarrow zvx$

In agreement with this proposal, each pair of distinct $\{a,b\}$ points formed a straight line and their corresponding open half-spaces were $\{x|abx\}$ and $\{x|axb\}$ sets. The convex hull of a set was the set of those points that cannot be separated from it: x is in the convex hull of X if there is no half-space containing

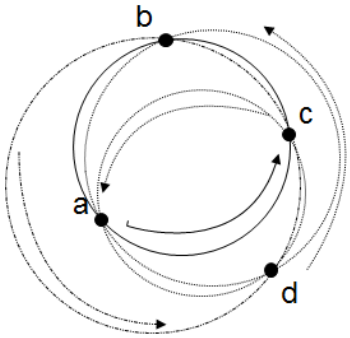


FIGURE 1.3: Counterclockwise order of CCS

x and not intersecting X. Stell and Webster (2007) noted that, based on the system thus defined, Knuth developed algorithms that were expressed only in the structure of the model to determine computationally the geometry of CCS with a significant efficiency improvement.

Knuth´s definition was as follows: in any finite subset of the plane that does not contain triples of collinear points, for any triplet of points (x,y,z), it is possible to associate a unique circle that contains these points in its circumference. Thus, the relation xyz occurs if when moving counterclockwise on the circumference after identifying x, y is before z. This relationship is called the counterclockwise order and fulfills Knuth´s five axioms.

For the four points indicated in Figure 1.3, counterclockwise order is given by acb, cba, bac, adc, dca, cad, adb, dba, bad, bdc, dcb, and cbd. In the resulting CCS, the open half-spaces corresponding to the straight line $\{a,c\}$ are $\{b\}$ and $\{d\}$. A hyperplane is any set that contains exactly two points and the complement of a $\{x,y\}$ hyperplane is called a co-circuit, which can be partitioned into the two $\{z|xyz\}$ and $\{z|xzy\}$ sets. For a triplet of points $x=(x_1,x_2), y=(y_1,y_2)$, and $x=(x_1,x_2)$, and

$$det(x,y,z) = \begin{vmatrix} x_1 & x_2 & 1 \\ y_1 & y_2 & 1 \\ z_1 & z_2 & 1 \end{vmatrix} = x_1(y_2-z_2) - y_1(x_2-z_2) + z_1(x_2-y_2)$$

$B(x,y,z)$ is true if $det(x,y,z) > 0$. Knuth focuses on the plane-owned property of ordering counterclockwise and how this order can be calculated using the determinants of the matrices. Geometric algorithms can be designed in terms of only this property, which leads to significantly improved algorithms. According to Smyth and Webster (2007), the resultant set of co-circuits satisfies the axioms of flat oriented matroids. In fact, CCSs are equivalent to flat oriented matroids of rank 3, where all sets of cardinality 3 are bases of the underlying matroid.

Computational calculation of triangulations

Pfeifle and Rambau (2002) presented a procedure by which it was possible to perform the triangulation calculation from a point set in a computational manner based only on the associated oriented matroid combinatorial structures. They were interested in triangulations as those were a basic tool for decomposing objects into simpler objects. For a configuration $\mathcal{A}$ of n d-dimensional points, a subset T of subsets of $\mathcal{A}$, each of $d+1$ elements, was a triangulation if and only if

$\cup_{\sigma \in T} conv\sigma = conv\mathcal{A}$, and

$conv\sigma \cap conv\sigma' = conv(\sigma \cap \sigma')\forall \sigma, \sigma' \in T$

The important part of the Pfeifle proposal was that instead of using expensive linear programming with exact arithmetic to ensure the two aforementioned conditions, purely combinatorial checks based on the oriented matroid of $\mathcal{A}$ were used. To validate if the second condition above was not met, the set of all circuits of $\mathcal{A}$ was used and checked if for any two subsets $\sigma, \sigma' \in T$ existed in a circuit (Z^+, Z^-) of $\mathcal{A}$, $Z^+ \subseteq \sigma$ and $Z^- \subseteq \sigma'$. To verify the fulfillment of the first condition in a purely combinatorial fashion, the set of all co-circuits of $\mathcal{A}$ was used. There was a co-circuit for each $(d-1)$-dimensional affine hyperplane extending over subsets of $\mathcal{A}$, which included all points of $\mathcal{A}$ that were in the positive side (C^+), all points of $\mathcal{A}$ that were in the negative side (C^-) of the respective hyperplane, and all points of $\mathcal{A}$ that were on the hyperplane (C^0). For the final testing of the first condition, it was verified that for each $\sigma \in T$, every C^0 of every co-circuit of $\mathcal{A}$, where sets C^- and C^+ were both nonempty, must be a $\sigma' \in T$ containing C^0.

According to Pfeifle and Rambau (2002), as circuits of $\mathcal{A}$ determined in a purely combinatorial fashion their co-circuits and vice versa, and because any one of them defined the matroid of $\mathcal{A}$, possible triangulations of $\mathcal{A}$ depend only on its oriented matroid. The transfer of the geometry to the combinatoric associated with $\mathcal{A}$ was established through the concept of its chirotope (De Loera, Rambau, and Santos, 2010) (Equation 1.1). This assigns its orientation to each ordered basis of $\mathcal{A}$.

$$\chi : \begin{cases} \begin{pmatrix} \mathcal{A} \\ d+1 \end{pmatrix} \rightarrow \{+,-,0\} \\ (i_1, i_2 \ldots i_{d+1}) \mapsto sign(det(a_1, a_2 \ldots a_{d+1})) \end{cases} \tag{1.1}$$

Circuits of $\mathcal{A}$ can be calculated based on all subsets of $d+2$ elements of $\mathcal{A}$ and its co-circuits, based on all its subsets of d elements and the calculation of the associated chirotopes. Pfeifle and Rambau (2002) also proposed algorithmic solutions that ranged from obtaining any possible triangulation to obtaining all the possible triangulations, using only combinatorial structures associated with the oriented matroid of the point configuration $\mathcal{A}$. These algorithms were implemented by their authors in the package TOPCOM (Triangulations of Point Configurations and Oriented Matroids) attempting to achieve a minimum complexity and maximum efficiency.

Alternative models of space representation

Stell and Webster (2007) suggested possible usages of oriented matroids in the context of space representation in five specific application areas. The most direct application was using them as geometric structures for discrete points set in a similar manner to the grid of points in the *Realm* (Schneider, 1997). Geographical objects represented as points could be referenced to elements of the oriented matroid; 1D and 2D features could be represented using subsets of the oriented matroid. For example, the pair of corresponding elements of the matroid could represent the open interval between a pair of adjacent points, and similarly, the interior of a square bounded by four adjacent points could be represented by the four corresponding elements of the matroid.

A second area of application was to identify the elements of the oriented matroid with particular geographic entities (houses, trees, tents). In this approach, certain geometric relationships could be represented regardless of the objects' position in space. The third area of application was the representation of incomplete data, which is frequent in spatial databases. The potential of the Stell and Webster approach was based on considering uncertainty by defining regions of possibility (cells) represented by the corresponding subset of the oriented matroid, indicating that the "true" position was somewhere in the corresponding continuous part. This could provide a model where it was preferred to provide incomplete information than to provide incorrect information.

A fourth potential work area was the triangulation of regions based on oriented matroids that had previously been studied, for example in (Santos, 1997). The remaining area of further research would be the usage of oriented matroids based on complex cells to provide a unified data model that includes the two classical forms of spatial structure in geographic information systems (GIS): the raster and vector, as proposed in (Winter, 1998).

According to Smyth and Webster (2007), in extending topology to geometry, oriented matroids are appropriate finite models of the linear and convex structure of Euclidean space. However, according to them, the theory of oriented matroids includes important limitations for an efficient spatial data storage, and algebraic spatial representation could facilitate improved storage and computation of spatial objects. Hence, the spherical version of oriented matroids should be used. Similarly, the Borovik, Gelfand, and White (2003) proposal of purely algebraic and full symmetric matroids appears to be promising.

Terrain visibility graphs

Saeedi (2012) studied a terrain profile viewed as a monotone polygonal line in x in the xy plane, where two terrain vertices were mutually accessible if and only if either they were adjacent or the line segment that connected them was located above the ground. Based on a graph of ground visibility, where the vertexes represented terrain vertexes and the edges represented pairs of mutually accessible terrain vertices, Saeedi explained its theoretical properties as a combinatoric structure of geometric objects to address problems associated with geometric visibility.

Visibility between vertexes, which was analyzed in the core induced sub-graphs from visibility graphs (Kumar, 1995), involved requirements of order that must be met by the slopes of the lines connecting the pairs of vertexes. Through the creation of an order based on the slope of the lines connecting all the pairs of vertexes of the polygon, such that this order was consistent with the desired visibility graph, it was possible to determine if certain properties of graphs were sufficient to indicate if this graph was a graph of visibility.

The main Saeedi's contribution was the formulation of a simple test that these properties should meet for guaranteeing such ordering of slope and thus obtaining a faster algorithm for its construction. Saeedi's approach clarified the implications of the theoretical properties of graphs on slope ordering by interpreting them as properties that define an underlying oriented matroid.

Comments of the state of the art

In summary, it was discussed that conventional methods for remote sensing image segmentation are based on digital structures that do not adhere to well-established topological axioms and geometric algorithms that assume a pixel-based spatial computing model. Thus, previous work attempting to produce topology preserving image segmentation in remote sensing used digital image representations that suffered from topological ambiguities. A more rigorous proposal for addressing such problems in image segmentation of computer vision was suggested by Kovalevsky. It was demonstrated that ALFS meet the classical axiomatic topological postulates and hence are able to unambiguously represent adjacency, connectedness, and boundary relationships that are critical for appropriate multispectral image analysis, in particular, image segmentation. Therefore, it was necessary to conduct a research to understand how the combinatorial properties of ALFS could allow the involvement of topological properties such that the accuracy and efficiency of the multispectral image segmentation algorithms could be improved. This implied devising a conceptual model for multispectral image representation based on Cartesian complexes of abstract cells that considered topological and geometric properties. It was then possible to build and evaluate a computational framework that enabled the implementation of Cartesian complexes to represent adequately the topological and geometric image-object properties used in the segmentation of multispectral images.

It was therefore relevant and promising to conduct further in-depth research on the usage of Cartesian complexes for obtaining a topologically correct image representation to produce more accurate and effective image segmentation algorithms. The production of a computational framework that made possible the representation of a digital image in a manner that explicitly considered the topological data and used this for digital image segmentation, would constitute a significant technological development. It would strengthen the ability of the Engineering Faculty at District University of Bogotá to produce useful solutions to technical problems and to improve geo-spatial knowledge.

i

1.2 Research question

To what extent can multispectral image segmentation be improved by linking combinatorial properties of axiomatic locally finite spaces and their associated oriented matroids which allows reinforcement of topological properties?

Answering this research question involves the necessary research work to also answer the following questions.

- How can digital image segmentation take advantage of topological properties available in the Cartesian complexes?
- How to link the combinatorial properties of ALFS and their associated oriented matroids?
- How can the Cartesian complex based image segmentation approach be computationally implemented?

1.3 Hypothesis[1]

A spatial computational framework, based on the geometric half-spaces provided by the ALFS and their associated oriented matroids, improves the performance of space algorithms for multispectral image segmentation because using only the combinatorial coordinates established by Cartesian complexes and oriented matroids combinatorial properties preserves the topological and geometric properties of the image objects and reduces algorithmic complexity.

1.4 Research objectives

1.4.1 General objective

To develop and evaluate an alternative multispectral image representation by associating Cartesian complexes with oriented matroids to determine an efficient solution to a segmentation process that considers topological and geometric properties.

1.4.2 Specific objectives

- To devise a conceptual model for multispectral image representation based on Cartesian complexes of abstract cells that considers topological and geometric properties.
- To design a mechanism for encoding topological and geometric features present in combinatorial half-spaces using oriented matroids to reduce algorithmic complexity in multispectral image segmentation.
- To build and evaluate a computational framework that enables the implementation of Cartesian complexes to adequately represent the topological and geometric image-object properties used in the segmentation of multispectral images.

1.5 Research group motivation

The techniques of digital image analysis have had a very important development in the last decade. Its success is due, among other things, to the increasing availability of optical and micro-wave images of high spatial and spectral resolution, which has given rise to the "big data" environmental remote sensing and to mostly interdisciplinary and data driven applications (Chi et al., 2016; Liu et al., 2018).

The research line in *Image analysis for characterization and detection of changes in geographical phenomena and objects* of the research group *Núcleo de Investigaciones en Datos Espaciales* (NIDE) aims to deepen and develop new concepts, methods and techniques of image analysis that consider not only the spectral information that the individual pixels possess, but also the spatial context information. In this way, beside the spectral information stored in each pixel, other features such as the shape, size and

[1] The research question has been defined here following the formal requirements stated by Bunge (2004).

texture of the objects and geographical phenomena of interest are incorporated to the analysis. Therefore, having an ambiguities-free image segmentation stage is extremely important for this research line , in particular, what is related to the later object detection stage with a view to performing the monitoring of the dynamics of changes.

1.6 Thesis summary

This thesis presents a space alternative representation of digital images by associating Cartesian complexes and oriented matroids to determine a solution for the segmentation process by considering topological and geometric properties. This thesis describes a referential framework that provides theoretical support to the selected approach to solve the research question. Further, this thesis document describes a conceptual model that has been defined using locally finite spaces provided by Cartesian complexes and how oriented matroids are linked in the multispectral image segmentation process, as well as each component of the model. Furthermore, the designed and implemented computational framework, which allows representing digital images in a correct manner based on the ALFS definition provided by Cartesian complexes (Kovalevsky, 1989; Kovalevsky, 1984; Kovalevsky, 2001; Kovalevsky, 2005; Kovalevsky, 2006) is described.

This digital space definition supports a digital topology that meets the T_0 separation property, which allow to define properly the boundary of space elements from their nearest lower dimension pairs. Figure 1.4 indicates how, in a Cartesian complex, unlike a conventional pixel space, two 1D and one 0D additional inter-pixel space elements are included in the digital representation. The figure illustrates how conventional pixels (*bottom*) are digitally allocated (*middle*) by row and column to a grid (*the small connecting pyramid*). Comparatively, in a Cartesian complex, in addition to the pixels (*the tall connecting pyramid*), inter-pixel space elements are also digitally represented (*two connecting triangles for two inter-pixel 1D elements (*1*-cells) and a connecting line for one inter-pixel 0D element (0-cell)*).

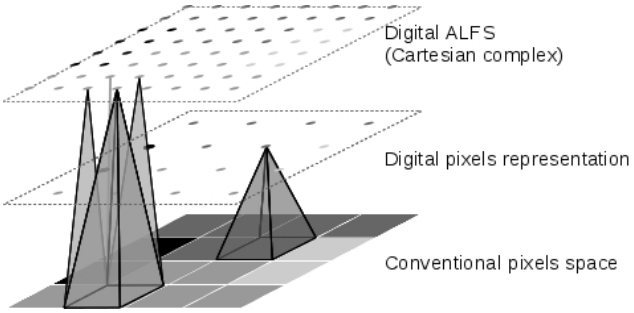


FIGURE 1.4: Conventional pixels versus Cartesian complex space representations

To improve the digital image segmentation process performance, two strategies are proposed. The first is to codify topological and geometrical characteristics using oriented matroids (Whitney, 1935; Oxley, 2006), which are linked with hyperplanes defined by the Standard Separating Form (SSF) available in the Cartesian complex geometric space (Kovalevsky, 2008). Figure 1.5 displays a central hyperplane arrangement (Fukuda, 2004) formed by two SSFs and two subspace examples produced using a Chirotope transformation (De Loera, Rambau, and Santos, 2010). Each SSF is defined based on the inter-pixel space

elements available in the Cartesian complex. Specifically, a central hyperplane arrangement is used to establish the set of valid orientations for both the oriented gradient calculation and filter banks production (Leung and Malik, 1996).

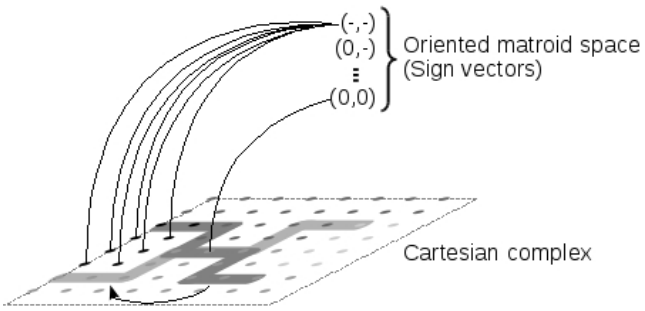


FIGURE 1.5: Combinatorial approach includes Cartesian complex geometry to oriented matroid position vector transformation

The second strategy is to design and implement all algorithms using a parallel computational framework. In the segmentation procedural framework, each task outcome calculated in a space element is independent of that calculated in another, except for the watershed transform. Therefore, each pipeline task can be conducted simultaneously for all space elements, being limited only by the number of available computing processors. This research focuses on reformulating the segmentation framework proposed in (Arbelaez et al., 2011) in terms of Cartesian complexes and encoding geometric features present in a Cartesian complex using oriented matroids to improve the segmentation processes.

The accuracy assessment of the boundaries produced by the proposed approach was performed using two validation strategies: (1) segment scale generalization to the scale of the available segmentation ground truth and (2) the proposed approach versus conventional pixel boundary detection benchmarking using the Berkeley Segmentation Benchmarks 500 (BSDS500) (Martin et al., 2001) software. The results demonstrate that, by departing from the conventional pixel representation, it is possible to segment an image based on a topologically correct digital space, while simultaneously exploiting the combinatorial features of their associated oriented matroids. The results indicate that although it is true that the precision produced from Cartesian complexes does not exceed that obtained from the conventional approach based on pixels, the proposed approach does achieve a superior recall and average precision. This affirms that the model proposed and implemented as part of the research presented herein constitutes a reliable alternative for the segmentation of multispectral images. It is also possible to confirm that the use of ALFS and their associated matroids enables image topological-geometric segmentation.

This thesis document is divided in nine chapters. Chapter 1 presents the research question, objectives, and hypothesis as they were proposed and approved at the research institutionalization stage. Chapter 2 first develops the topological concepts that are essential to complete the general framework over which the digital geometry defines its concepts. Then, a formal formulation of the image segmentation process is presented. The adopted segmentation procedural framework is introduced and finally the oriented matroid concept is developed. Chapter 3 discusses the procedural framework conceptual model for topological and geometrical feature-preserving segmentation. This chapter introduces the layered architecture used to individually address different conceptual components and resolve them more effectively. All components of the image segmentation procedural framework are defined. A description of how the link between the Cartesian complexes and oriented matroids was established is also provided. Chapter 4 presents the

main elements used to provide computational support to the conceptual model established in Chapter 3, in addition to their execution outcomes. Chapter 5 describes the benchmark-based assessment procedure used to validate the proposed model and presents the validation outcomes using hyper and multispectral datasets, which are also described. The results and a discussion are presented in Chapter 6. Conclusions are drawn in Chapter 7. The main contributions are discussed in Chapter 8 and finally, future work and recommendations are proposed in Chapter 9

Chapter 2

Theoretical framework

The first challenge in this research was to define and build an alternative space for image representation that reinforced topology and geometric relationships among the space elements[1] to suitably define super-pixels (i.e., image-objects[2]). To achieve this, the approaches proposed in (Kovalevsky, 2008); (Khalimsky, Kopperman, and Meyer, 1990), and (Bertrand and Couprie, 1999) were found to be the only ones addressing the sparation axiom T_0. The approach proposed in (Kovalevsky, 2008) was adopted because it indicated how for any locally finite topological space satisfying axioms in Section 2.1.2 there exist an isomorphic abstract cell complex proposed there, in addition to the fact that the algorithmic development for image analysis presented there served as a starting point. It is an axiomatic approach to the topology and geometry of locally finite spaces, placing special emphasis on computational solutions for topological and geometric problems and outlining several mechanisms for explicit computational representation of the topological spaces. The properties of the latter correspond to the axioms of classical topology including the separation axiom T_0. Digital topology, so defined, is free of the topological ambiguities shown in Section 1.1 (Figures 1.1 and 1.2) and introduces a concept of digital geometry, based on the topology of locally finite spaces, which is independent from Euclidean geometry.

The second challenge was to define a strategy for image segmentation that could exploit the underlying topological model. After reviewing the previous works on digital image segmentation presented in Section 1.1.2, the procedural framework proposed in (Arbelaez et al., 2011) was adopted because it was in the "top 10" of the state-of-the-art approaches to segmentation in computer vision. Further, it proposed a line of processes that included several tasks where there was the opportunity to propose reformulations in terms of 1D inter-pixel elements available in a Cartesian complex as discussed in Chapter 3. This approach uses a combination of both top-down and bottom-up approaches presented in Section 2.2.1.

The following sections introduce the main concepts supporting the alternative method proposed in this thesis document, which extends the proposal of Horowitz and Pavlidis (1976), where a formal conceptualization of the segmentation problem was presented and is described in Section 2.2. The concepts that support oriented matroids are also presented because they are used as a combinatorial alternative representation of the topological and geometric features to provide a mechanism to improve the tasks of the segmentation process. These topics are developed as the theoretical framework from which the research was developed to determine a solution to the problem raised in Section 1.1 and to answer the

[1] It is used the generic term *element* because the locally finite space proposed by Kovalevsky (2008) must posses elements with different topological properties.

[2] Image-objects are discrete groups of pixels with clearly delimited boundaries but no thematic description (Lizarazo and Elsner, 2011).

question and contrast the hypothesis, respectively, raised in Sections 1.2 and 1.3. Section 2.1 develops the axiomatic locally finite space concepts as support of the abstract cell complexes that are used to finally define the Cartesian complexes. Section 2.2 addresses the concepts that define the segmentation problem. Section 2.3 presents the proposed procedural framework in (Arbelaez et al., 2011) and finally, Section 2.4 develops the oriented matroid concept. All the concepts, figures, and examples are presented according to the interpretation given during the development of the research.

2.1 Axiomatic locally finite spaces

The classical conception of space used in GIS and multispectral imaging analysis is based on several concepts including continuity (Stell and Webster, 2007). Continuity refers to the fact that a region can always be subdivided into smaller subregions; this concept is also known as dense sets (Worboys and Duckham, 2004). However, the digital representation is performed through space sampling that discretizes the reality; therefore, it only represents certain elements of the real world. The use of a representation of space that "occurs" in terms of continuously varying coordinates and is expressed in real numbers, involves their obligatory transformation to the digital world of coordinates expressed in integer numbers. The traditional models of digital representation of geographical phenomena (Schneider, 1997), being discrete objects or continuous fields, suffer from imprecision and inaccuracy and do not include, in the particular case of digital images, a measure of their topology.

Such inaccuracy is due to the impossibility of determining a method to transform the conceptual reality, which is countless (based on $\mathbb{R}^n$), to a computational representation that is countable (based on $\mathbb{Z}^n$). These imprecisions and inaccuracies are particularly significant when geographical relationships are modeled based on concepts such as neighborhood, contiguity, connectivity, and boundary. If the digital representation of these relationships is not performed properly, GIS technology may not provide the appropriate responses to typical problems in remote sensing image analysis, such as segmentation. It would then be necessary to evaluate alternative models of representation.

Kovalevsky (2008) proposed to build a discrete geometry using a Cartesian complex from the definition of the topological spaces based on the axiomatic locally finite spaces provided by abstract cell complexes. His proposal attempts to provide an alternative representation of space based on a digital topology that is not affected by the ambiguities found in conventional representations of images, such as 4-connected, 8-connected, and (a, b)-connected graphs (Rosenfeld, 1970; Kong and Rosenfeld, 1991). In his work, Kovalevsky demonstrates how Cartesian complexes allow for implementing the topological relations required to perform digital image segmentation without ambiguity. In this approach, a digital space should be a locally finite space where each element has a neighborhood composed finitely of several elements with different topological properties. Below are the definitions given in the Kovalevsky's proposal on which this research work is based. The definitions, lemmas and theorems are presented as they can be found in (Kovalevsky, 2008).

Definition LFS (locally finite space): A nonempty set S is called a *locally finite space* (LFS) if to each element e of S certain subsets of S are assigned as neighborhoods of e and some of them are finite.

An LFS is called an axiomatic LFS (ALFS) when it satisfies the following four axioms.

Axiom 1: For each space element $e \in S$ there are certain subsets containing e which are neighborhoods of e. The intersection of two neighborhoods of e is also a neighborhood of e.

Because the space is locally finite, there exists the *smallest neighborhood* (SN) of e (denoted by $SN(e)$) that is the intersection of all the neighborhoods of e; i.e., $SN(e)$ is contained in each of the neighborhoods of that element.

Axiom 2: There are space elements whose SN consists of more than one element.

Definition IN (incidence): If $b \in SN(a)$ or $a \in SN(b)$, then the elements a and b are called *incident* to each other. The incidence relationship is symmetric and reflexive.

Definition IP (incidence path): A *incidence path* between a pair of elements of a subset of an LFS can be defined if it is possible to identify a sequence of elements between one and the other where each pair of subsequent elements is incident.

Definition CN (connected): If for each pair of elements of a given subset of an LFS it is possible to identify a path of incidence entirely contained within that subset, it is said that the subset is *connected*.

Definition FR (frontier): The topological boundary or *frontier* $Fr(T,S)$ of a nonempty subset T of an LFS S is the set of all elements of S, such that each neighborhood contains elements of both T and its complement $S-T$.

Definition NR (neighborhood relation): An LFS *neighborhood relation N* between two elements a and b is given if and only if $a \in SN\ (b)$. A ordered pair of this relation is denoted by aNb.

Definition TF (thick frontier): When a pair of elements a and b of $Fr\ (T,\ S)$ meets the condition that aNb, bNa, an element belongs to T and the other to the complement $S-T$, a and b are *opponents*. A frontier is *thick* when $Fr\ (T,\ S)$ contains at least one pair of opponent elements, otherwise it is *thin*.

Axiom 3: The frontier $Fr(T\ ,\ S)$ of any subset $T\ \subset\ S$ is thin.

In agreement with Axiom 3, $Fr\ (T,\ S)\ =\ Fr\ (S-T,\ S)$, as it was expected in a topological space. This condition can only be accomplished when a neighborhood relationship (Definition NR) is anti-symmetric.

Axiom 4: The frontier of the frontier of a subset of an LFS is equal to the frontier of the subset. $Fr(Fr\,(T,S)\,,S) = Fr(T,S)$.

The condition imposed by Axiom 4 can only be achieved when the neighborhood relationship is not transitive.

However, the classical topology (Munkres, 1999, p.75) of a space S is defined if a set of subsets of S called the open subsets satisfy the following axioms:

Axiom C_1: The entire set S and empty subset $\emptyset$ are open.

Axiom C_2: The union of any number of open subsets is open.

Axiom C_3: The intersection of a finite number of open subsets is open.

Axiom C_4: The space has a separation property.

For Axiom C_4, in the case of LFSs, it is only needed the separation axiom T_0, i.e., for any pair of elements of the space there is an open subset that contains exactly one of the elements.

Definition OP (open): A subset $O \subset S$ is called *open* in S if along with each element e, O also contains $SN(e)$. When it contains all the elements of its frontier $Fr(O,S)$, it is *closed in S*.

Open subsets, so defined, satisfy the conditions of Axioms C_1 to C_3 and therefore are open in the classical sense.

2.1.1 ALFS properties

Definition BD (bounding): Given two elements a and b of an ALFS S, b *bounds* a if $a \neq b$ and $a \in SN(b)$ i.e., $b \in Fr(\{a\},S)$.

The bounding relationship is irreflexive, anti-symmetric, and transitive; hence, being a partial order relation ensures that S will contain the elements that are not bounded by any other element (*minimum*) and elements that do not bound any other (*maximum*) (Kovalevsky, 2008, lemma MM, p.14).

The frontier of a subset of an ALFS does not contain maximum elements and the SN of each element of the space contains at least one maximum. The SN of any element of an ALFS is open in the classic sense and is called the *smallest open neighborhood* (SON) of a given element of the space. All elements bounded by a given element are part of its SON. The SON of an element of the space satisfies the axiom of separation T_0.

2.1.2 Abstract cell complexes

Abstract cell complexes are an important particular case of an LFS (Listing, 1862). In this case, the space is characterized by a relationship of partial order between its elements and a function of dimension $dim\ (a)$, which maps to an element a of the space a non-negative integer such that if another element $b \in SON\ (a)$, then $dim\ (a) \leq dim\ (b)$. If $dim\ (a) = k$, then the element a is of dimension k (k-dimensional). An element a of the space is a face of another element b if $b \in SON\ (a)$. The face relationship is a reflexive partial order (Kovalevsky, 2008, Definition SON/FC, p.23).

An *abstract cell complex* (ACC) $C = (E, B, dim)$ is a set E of elements called abstract cells (AC) with a binary bounding relationship $B \subset E \times E$ that is asymmetric, irreflexive, and transitive among its elements and a function $dim : E \longrightarrow \mathbb{Z}^+$ provided $dim(e') < dim(e'')$ for $(e', e'') \in B$. Two elements a and k of an ACC establish a bounding path if it is possible to identify a sequence of elements between a and k such that each element bounds the following. The number of elements in the sequence minus one is the bounding path length (Kovalevsky, 2008, Definition AC, p.23).

The dimension $dim(c,C)$ of a cell c in an ACC C corresponds to the longest bounding path length from any element of C to c (Kovalevsky, 2008, Definition DC, p.24). The dimension of a complex is given by the greatest dimension of all its cells. A *K-complex* $C = (E, B)$ is an ACC that applies this definition of dimension (Kovalevsky, 2008, Definition KC, p.27). Given $E' \subseteq E$ and $B' = B \cap (E' \times E')$, $S = (E', B')$ is a subcomplex of C (Kovalevsky, 2008, Definition SC, p.29). The subcomplex consisting of a cell c^n n-dimensional of complex C^n n-dimensional is open, whereas the subcomplex of any complex consisting of a 0D cell c^0 is closed. A subcomplex will be open or closed depending on the complex from which is a subcomplex.

For example, given the K-complex $C = (E,B)$ where $E = \{e_1,e_2,e_3,e_4,e_5,l_1,l_2,l_3,l_4\}$, the dimension function is such that $dim\,(e_i) < dim\,(l_j)$ with $B = \{(e_1,l_1),(e_2,l_1),(e_2,l_2),\ (e_3,l_2),(e_3,l_3),(e_4,l_3),(e_4,l_4),$ $(e_5,l_4)\}$, the subcomplex $T = (E',B')$ with $E' = \{e_3,l_3\}$ and $B' = \{(e_3,l_3)\}$ is neither open nor closed in C. However, for the subcomplex $S = (E'',B'')$ with $E'' = \{e_3,e_4,l_3,l_4\}$, $B'' = \{(e_3,l_3),(e_4,l_3),(e_4,l_4)\}$ and the same dimension function definition as before, T is open in S.

That is, in agreement with Definition OP, in a complex a subcomplex is open if and only if it contains all the cells bounded by a cell of the subcomplex . Conversely, the subcomplex is closed if it contains all the cells of the complex that bound each subcomplex cell (Kovalevsky, 2008, theorems BO and BC, p.30).

Given two subsets t and T of a space S such that $t \subseteq T \subseteq S$, the set containing each cell $a \in t$ and all cells of T bounding a is called the closure of t in T denoted by $Cl\ (t,\ T)$. The set $t - Fr(t,\ T\ )$ is called the interior of t in T and is denoted by $Int(t,\ T\ )$. The set $T - t - Fr(t,\ T\ )$ is called the exterior of t in T and is denoted by $Ext(t,\ T\ )$ (Kovalevsky, 2008, Definition CL, p.31).

The size and shape of the SON of a cell c depend on the complex C base on which this is defined and is denoted by $SON(c,C)$. Two sub complexes of a complex are mutually incident, if any of the two contains at least one cell that is incident with any cell of the other.

Definition DU (dual): A complex D is called *n-dual* to C when it is obtained from a given n-dimensional complex C changing the order of the two cells in each pair of bounding relation B and by replacing the dimension d of each cell by $n-d$.

For example, given the K-complex $C = (E,B)$, where $E = \{e_1,e_2,e_3,e_4,e_5,l_1,l_2,l_3,\ l_4\}$, the dimension function is such that $dim(e_i) < dim(l_j)$ with $B = \{(e_1,l_1),(e_2,l_1),(e_2,l_2),(e_3,l_2)\ ,(e_3,l_3),(e_4,l_3),(e_4,l_4),$ $(e_5,l_4)\}$, which is 1-dimensional, has a dual $C^* = (E,B^*)$ with $B^* = \{(l_1,e_1)\ ,(l_1,e_2),\ \ (l_2,e_2),(l_2,e_3),$ $(l_3,e_3),(l_3,e_4),(l_4,e_4),(l_4,e_5)\}$ with the dimension function $dim\,(l_i) < dim\,(e_j)$; C^* is also 1-dimensional.

Definition HD: An n-dimensional complex C^n is *homogeneously n-dimensional* when each of its cells of dimension less than n bounds at least one n-dimensional cell (n-cell) of C^n.

Definition nPA: An incidence path in a complex C^n is called an *n-dimensional path* in C^n when it starts and ends in n-cells and contains only n- and $(n-1)$-cells.

Definition SCO (strongly connected): A homogeneously n-dimensional complex C is *strongly connected* if for any two n-cells of C there is in C an n-dimensional path containing these two cells. A n-dimensional complex C is *locally strongly connected* if the $SON(c,C)$ of each cell c is strongly connected.

The subcomplex frontier (Definition FR) can then be expressed in terms of the cell dimension. The combinatorial boundary ∂S^k of a k-dimensional subcomplex $S^k \subset C^n$ is the closure in C^n of the set of all $(k-1)$-dimensional cells ($(k-1)$-cells) of S^k, each of which exactly bounds a k-cell in S^k.

The open frontier $Of(S,C)$ of the subcomplex S in the complex C is the subcomplex of C containing each cell c of C whose closure $Cl\ (c,\ C)$ contains cells of both S and complement $C-S$.

2.1.3 Cartesian complexes and combinatorial coordinates

A K-complex $C = (E,B)$ is a 1-*dimensional connected ACC* when $E = \{e_0,e_1,e_2,\cdots,e_{2m}\}$ with $m \geq 1$, and B is a bounding relationship such that each element e_i, with even index i, bounds the elements e_{i-1}

and e_{i+1}, with odd indexes (Equation 2.1). Here each 0-dimensional cell (0-cell) is closed, whereas each 1-dimensional cell (1-cell) is open. The cell indexes are called *combinatorial coordinates* in the 1-dimensional space.

$$B_1(e_i, e_j) = \begin{cases} true \text{ if } j = i \pm 1 \text{ and } i \text{ is even} \\ false \text{ otherwise} \end{cases} \tag{2.1}$$

Figure 2.1 displays two possible graphical representations of a 1-dimensional ACC from its combinatorial coordinates. In Figure 2.1(a), each complex's cell is represented as a point, for convenience, the 0-cells are in black and the 1-cells are in red. Figure 2.1(b) displays each 0-cell as a point, whereas each 1-cell is displayed as a line segment, retaining the same colors to facilitate graphical correspondence. Each line segment representing a 1-cell has been joined with dots representing the bounding 0-cells. Although it is accepted that the representation of Figure 2.1(b) is more intuitive, the truth is that it suggests a nonexistent "continuity" in a locally finite space; for this reason, the representation provided in Figure 2.1(a) is preferred.

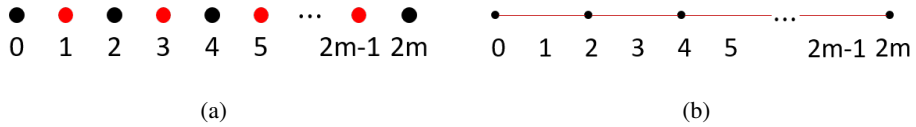


FIGURE 2.1: Graphical representations of a 1-dimensional connected ACC

A Cartesian ACC is obtained by performing the Cartesian product of two or more of these 1-dimensional complexes (Kovalevsky, 2008, Definition CC, p.39). The cell set of a n-dimensional Cartesian ACC C^n is the Cartesian product of n 1-dimensional connected ACC sets. The 1-dimensional complexes are the coordinate axes of C^n, which become a n-dimensional space. A cell of C^n is an n-tuple $c = (a_1, a_2, a_3, \cdots, a_n)$ of a_i cells, each belonging to a particular 1-dimensional complex and becoming a component cell of c. In this case, a cell $c_1 = (a_1, a_2, a_3, \cdots, a_n)$ bounds another cell $c_2 = (b_1, b_2, b_3, \cdots, b_n)$ if for each pair of component cells a_i and b_i $a_i = b_i$ or a_i bounds b_i (Equation 2.2). The dimension of a cell c is the sum of the dimensions of its component cells. Cells of the greatest dimension n in C^n are called *principal cells* of C^n.

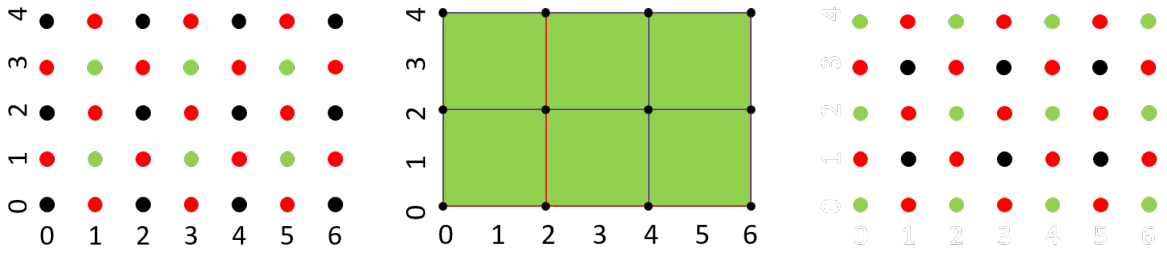
$$B_n(c_1, c_2) = \begin{cases} true \text{ if } \forall (a_i, b_i), a_i = b_i \vee B_1(a_i, b_i) \\ false \text{ otherwise} \end{cases} \tag{2.2}$$

For the 1-dimensional ACCs $A_1 = \{e_0, e_1, e_2, e_3, e_4, e_5, e_6\}$ and $A_2 = \{e_0, e_1, e_2, e_3, e_4\}$, the 2-dimensional Cartesian ACC is

$$C^2 = A_1 \times A_2 = \left\{ \begin{array}{c} (e_0,e_0),(e_0,e_1),(e_0,e_2),(e_0,e_3),(e_0,e_4), \\ (e_1,e_0),(e_1,e_1),(e_1,e_2),(e_1,e_3),(e_1,e_4), \\ (e_2,e_0),(e_2,e_1),(e_2,e_2),(e_2,e_3),(e_2,e_4), \\ (e_3,e_0),(e_3,e_1),(e_3,e_2),(e_3,e_3),(e_3,e_4), \\ (e_4,e_0),(e_4,e_1),(e_4,e_2),(e_4,e_3),(e_4,e_4), \\ (e_5,e_0),(e_5,e_1),(e_5,e_2),(e_5,e_3),(e_5,e_4), \\ (e_6,e_0),(e_6,e_1),(e_6,e_2),(e_6,e_3),(e_6,e_4) \end{array} \right\}$$

.

Figure 2.2 displays two possible 2-dimensional Cartesian ACC graphical representations. In Figure 2.2(a), each complex's cell is represented as a dot; the cells have been differentiated by color; the 0-cells are black, the 1-cells are red, and the 2-cells are green. In Figure 2.2(b), each 0-cell is represented as a dot, each 1-cell as a line segment, and each 2-cell as a quadrilateral, also retaining the same colors to facilitate the graphical correspondence. Each line segment representing a 1-cell is linked with points representing the 0-cells that bound it. In the same manner, each quadrilateral representing a 2-cell has been surrounded by a representation of the 0-cells and 1-cells (dots and line segments) bounding it. Again, the representation of Figure 2.2(b) can be more intuitive; however, as in the 1-dimensional case, it suggests a nonexistent "continuity" as the Cartesian ACC remains a locally finite space. For this reason, the preferred representation is the one provided in Figure 2.2(a).



(a) 0-cells are represented by black dots,1-cells with red dots, and 2-cells with green dots.

(b) 0-cells are represented by black dots, 1-cells with red line segments, and 2-cells by green areas.

(c) Dual.

FIGURE 2.2: Graphical representations of a 2-dimensional Cartesian ACC and its dual

In an n-dimensional Cartesian complex C^n, a k-cell c^k exactly bounds $\sum_{m=k+1}^{n} \binom{n-k}{m-k} 2^{m-k}$ cells and is bounded by exactly $\sum_{j=0}^{k-1} \binom{k}{k-j} 2^{k-j}$. Every k-cell c^k bounds exactly two $(k+1)$-cells c_1^{k+1} and c_2^{k+1}.

2.1.4 Combinatorial balls and spheres

The concept of ACC enables the definition of strange ACCs such as the case of a 1-dimensional complex that is limited by more than two 0-dimensional complexes or 2-dimensional complexes with "holes". Topological concepts such as ball and sphere restrict the set of valid complexes. For this purpose, in addition to the concepts of smaller open neighborhood $SON(c,M)$ and closure $Cl\ (c,\ M)$ of a cell c,

it is also necessary to have modified versions that exclude cell c: $SON^*(c,M) = SON(c,M) - \{c\}$ and $Cl^*(c,M) = Cl(c,M) - \{c\}$.

Definition B0: A 0-*dimensional closed AC ball* (0-ball) (denoted by B^0) is a K-complex composed of a single cell. The K-complex that is the union of two different 0-balls is called a 0-dimensional sphere S^0.

Definition S1: A 1-*dimensional AC sphere* (denoted by S^1) is a one-dimensional K-complex where each cell is incident to exactly two other cells.

Definition PR: An m-dimensional cell (m-cell) c^m of an n-dimensional K-complex C^n ($m \leq n$) is a *proper cell* of C^n if $Cl^*(c^m,C^n)$ is either empty (if $m = 0$) or it is an $(m-1)$-dimensional sphere S^{m-1} and $SON^*(c^m,C^n)$ is either empty (if $m = n$), or it is an $(n-m-1)$-dimensional sphere S^{n-m-1}. A complex C is *proper* when all its cells are proper.

Definition BM: A *closed m-dimensional AC ball* (denoted by B^m), $m \leq n$, is either the closure of an m-dimensional proper cell of an n-dimensional complex, or the union of two m-dimensional closed balls B_1^m and B_2^m, provided $\partial B_1^m \cap \partial B_2^m$ is a $(m-1)$-dimensional closed ball.

Definition SM: A *m-dimensional AC sphere* (denoted by S^m), $m \geq 2$ (Equation 2.3), is the union of two m-dimensional closed balls B_1^m and B_2^m intersecting only in their complete boundaries.

$$S^m = B_1^m \cup B_2^m, \partial B_1^m = \partial B_2^m, (B_1^m - \partial B_1^m) \cap (B_2^m - \partial B_2^m) = \emptyset. \tag{2.3}$$

Definition OB: An *open m-dimensional ball* (denoted by O^m) is the subcomplex of an n-dimensional complex that is n-dual to an m-dimensional closed ball.

An n-dimensional sphere cannot be embedded in an n-dimensional space and requires a complex at least $(n+1)$-dimensional. For the 2-dimensional Cartesian complex represented in Figure 2.2, the subcomplex $\{(e_i,e_j)\}$, $0 \leq i \leq 6$, $0 \leq j \leq 4$ is a 0-dimensional ball, the subcomplex $\{(e_i,e_j),(e_k,e_l)\}, (e_i,e_j) \neq (e_k,e_l)$, $0 \leq i \leq 6, 0 \leq j \leq 4$, $0 \leq k \leq 6, 0 \leq l \leq 4$ is a 0-dimensional sphere, the subcomplex $\{(e_4,e_2)$, (e_5,e_2), $(e_6,e_2)\}$ is a 1-dimensional closed ball, the subcomplex $\{(e_5,e_2)$, (e_6,e_2), $(e_6,e_1)\}$ is a 1-dimensional open ball, the subcomplex $\{(e_0,e_0)$, (e_0,e_1), (e_0,e_2), (e_0,e_3), (e_0,e_4), (e_1,e_4), (e_2,e_4), (e_3,e_4), (e_4,e_4), (e_4,e_3), (e_4,e_2), (e_3,e_2), $(e_2,e_2)\}$ is a 1-dimensional sphere, the subcomplex $\{(e_2,e_2)$, (e_2,e_3), (e_2,e_4), (e_3,e_4), (e_4,e_4), (e_4,e_3), (e_4,e_2), (e_3,e_2), $(e_3,e_3)\}$ is a 2-dimensional closed ball, and the subcomplex $\{(e_{2i+1},e_{2j+1}) |\ 0 \leq i \leq 2, 0 \leq j \leq 1\}$ is a 2-dimensional open ball (its n-dual is a 2-ball as it can be seen in Figure 2.2(c)).

It can be stated that each cell inside a complex Cartesian C^n is a proper cell (Kovalevsky, 2008, Theorem PR, p.50).

2.1.5 Combinatorial homomorphism

A combinatorial homomorphism is based on the concept of elementary subdivision of the cells in an ACC (Stillwell, 1995, p.24). The elementary subdivision of a 1-cell c^1, replaces it with two 1-cells c_1^1, c_2^1 and a 0-cell c^0, such that the latter is a common face of the other two. In the case of the 2-dimensional Cartesian complex displayed in Figure 2.2, elementary cell subdivision must be the Cartesian product of the subdivision of each of its two component cells, considering that, in general, 0-cells are not subdivided.

For example, the 1-cell (e_3, e_2) of the 2D Cartesian complex displayed in Figure 2.3 (left) would use the $\{e_{\frac{5}{2}}, e_3, e_{\frac{7}{2}}\}$ subdivision set for its first component cell and $\{e_2\}$ for its second component cell, which is not subdivided because it is 0-dimensional. Finally, the elementary subdivision would be represented by the two 1-cells $(e_{\frac{5}{2}}, e_2)$, $(e_{\frac{7}{2}}, e_2)$ and the 0-cell (e_3, e_2) produced by Cartesian product of these two sets, provided $((e_3, e_2), (e_{\frac{5}{2}}, e_2))$ and $((e_3, e_2), (e_{\frac{7}{2}}, e_2))$ belong to the bounding relationship B defined on the respective Cartesian complex elementary subdivision (see Figure 2.3 (right)).

The *elementary subdivision* of an m-cell c^m of an n-dimensional complex (n-complex), $1 < m \leq n$, replaces the cell c^m by two m-cells c_1^m, c_2^m and an $(m-1)$-cell c^{m-1}, which bounds the first two; all of this occurs when the three new cells meet the following conditions (Kovalevsky, 2008, Definition ES, p.57):

(1) $\partial\left(\{c_1^m\} \cup \{c^{m-1}\} \cup \{c_2^m\}\right) = \partial c^m$ or is a subdivision of ∂c^m.

(2) $c^{m-1} \notin \partial c^m$.

(3) $\partial c^{m-1} \subset \partial c^m$ or ∂c^{m-1} is a subdivision of ∂c^m.

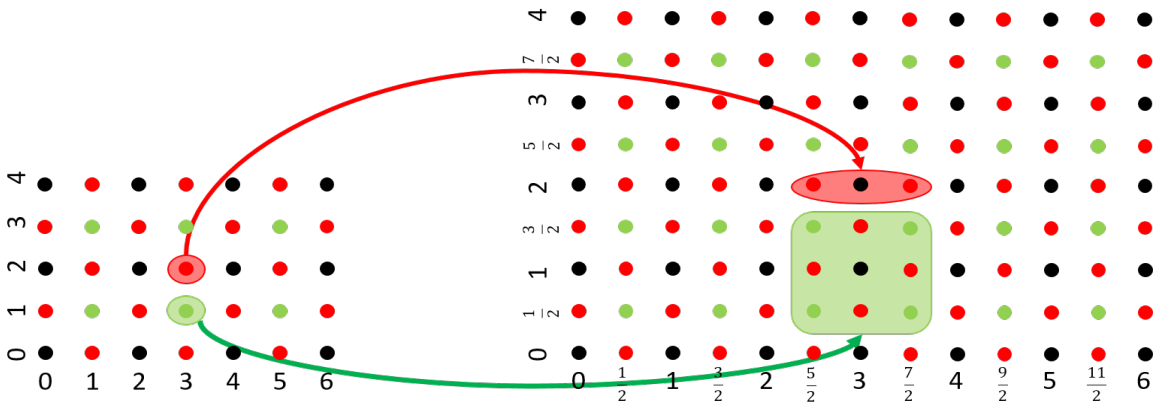


FIGURE 2.3: Elementary cell subdivision in a 2-dimensional Cartesian ACC

In a 2-dimensional Cartesian complex, the elementary subdivision of a 2-cell (e_i, e_j) starts defining a set of three 1-dimensional coordinates for each component cell: $\{e_{i-\frac{1}{2}}, e_i, e_{i+\frac{1}{2}}\}$ for e_i, and $\{e_{j-\frac{1}{2}}, e_j, e_{j+\frac{1}{2}}\}$ for e_j; in this case, the average indexes are open and the integers are closed. The Cartesian product of this two sets produces nine 2-dimensional coordinates, four of them correspond to 2-cells, four to 1-cells, and one is a 0-cell. For example, for the Cartesian complex displayed in Figure 2.2, the elementary subdivision of the 2-cell (e_3, e_1) produces, for each cell component, three new component cells, which produces nine cells as indicated in Figure 2.3 (on the left, the initial Cartesian complex, on the right, the corresponding elementary subdivision. The red background represents the subdivision of a 1-cell and green represents the subdivision of a 2-cell).

Two AC complexes C_1 and C_2 are combinatorically homomorphic if there exist elementary subdivisions of C_1 and C_2 that are isomorphic.

2.1.6 Generalized combinatorial boundary and combinatorial boundary of a space

A cell c^k of a complex C^n is proper if its sets $Cl^*(c^k,C^n)$ and $SON^*(c^k,C^n)$ are spheres. However, in certain cases, it may be appropriate to relax this concept for situations where $Cl^*(c^k,C^n)$ or $SON^*(c^k,C^n)$ is a "hemisphere" and then discuss these as semi-proper cells. A cell c^k of a subcomplex $T^m \subset C^n$ is semi-proper if it meets any of the following cases (Kovalevsky, 2008, Definition SP, p.59):

(1) $k = 0$ and $SON^*(c^k,T^m)$ is an $(m-1)$-dimensional ball rather than a sphere.

(2) $0 < k < m$, $Cl^*(c^k,T^m)$ is a $(k-1)$-dimensional sphere and $SON^*(c^k,T^m)$ is an $(m-k-1)$-dimensional ball rather than an $(m-k-1)$-dimensional sphere.

(3) $0 < k < m$, $SON^*(c^k,T^m)$ is an $(m-k-1)$-dimensional sphere and $Cl^*(c^k,T^m)$ is a $(k-1)$-dimensional ball rather than a $(k-1)$-dimensional sphere.

(4) $k = m$ and $Cl^*(c^k,T^m)$ is a $(k-1)$-dimensional ball rather than a sphere.

A subcomplex $T^m \subseteq C^n,\ m \leq n$ is a *semi-proper subcomplex* if all its cells are either proper or semi-proper. The *closing boundary* of a subcomplex $T^m \subseteq C^n,\ m \leq n$ is the complex of all the k-cells $c^k \in T^m, k < m$, which are semi-proper cells of T^m. Conversely, its opening boundary is the complex of all the k-cells $c^k \in C^n, k > 0$, such that the complex $Cl^*(c^k,T^m)$ is not a $(k-1)$-dimensional sphere. The generalized boundary of a subcomplex $T^m \subseteq C^n,\ m \leq n$ is the union of the closing and opening boundaries of T^m (Kovalevsky, 2008, Definitions SPC,CB,BO and BG, p.60-63).

2.1.7 Orientation of AC complexes

The notion of orientation (Rinow, 1975, p.425) for Euclidean complexes, applied to simplicial complexes, cannot be applied to ACCs. Following is Kovalevsky's concept for the orientation of proper Cartesian complexes. In the case of closed 2-dimensional complexes, it is common to define the orientation of a 1-dimensional AC (1-cell), identifying which of its two bounding 0-cells is the initial and which is the final; this is its intrinsic orientation. Analyzing the orientation of a 0-cell, it is clear that it can correspond simultaneously to the end cell of a 1-cell and to the start cell of other; this means that the 0-cells have no intrinsic orientation; it must be induced relative to each 1-cell it bounds.

Similarly, the 2-cells have an induced orientation relative to the orientation of the 1-cells that bound it. However, it would be impossible in a closed 2D Cartesian complex for two adjacent 2-cells to determine the induced orientation of each one relative to the orientation of the common bounding 1-cell; the latter has the same induced orientation (its intrinsic orientation) for both 2-cells. The solution is to introduce, together with the 1-cell intrinsic orientation, their relative orientations for each specific 2-cell. In this manner, the induced orientation of a 1-cell over a 2-cell either will match its intrinsic or will be opposed to this. The operation that transforms the intrinsic orientation of a cell a in respect to a cell b sets the induced orientation either equal to the intrinsic or its opposite. The choice between the two possibilities is commonly performed based on a parameter whose value is conventionally $+1$ when it becomes equal to the intrinsic or -1 when it should be changed to its opposite. This parameter is known as the bounding value of the a cell in cell b and is represented as $\mathscr{E}(a,b)$.

If b is a 1-cell and a an 0-cell incident to b, then $\mathscr{E}(a,b) = 1$ if a is the initial cell of b, otherwise $\mathscr{E}(a,b) = -1$. To assign the orientation of a 2-cell c^2, it is necessary to determine the bounding values of

all the 1-cells in the boundary of c^2 such that the induced orientation of any two 1-cells c_1^1 and c_2^1 having a common incident 0-cell c^0 are coherent in the sense that c^0 must be the end of a 1-cell and the beginning of another. The general rule can be expressed as $\mathscr{E}\left(c^0,c_1^1\right)\cdot\mathscr{E}\left(c_1^1,c^2\right)+\mathscr{E}\left(c^0,c_2^1\right)\cdot\mathscr{E}\left(c_2^1,c^2\right)=0$.

In a proper or semi proper n-dimensional complex C^n, if an $(m-2)$-cell c^{m-2} bounds an m-cell c^m, then there is in C^n exactly two $(m-1)$-cells c_1^{m-1} and c_2^{m-1} incident to c^m and c^{m-2} (Kovalevsky, 2008, Theorem T1, p.66). Based on this, it is possible to generalize for an m-cell c^m, a *bounding function* definition $\mathscr{E}(c^{m-1},c^m)$ that assigns to each $(m-1)$-cell a value from the $\{-1,+1\}$ set. The bounding function is *pairwise coherent* if $\mathscr{E}(c^{m-2},c_1^{m-1})\cdot\mathscr{E}(c_1^{m-1},c^m)+\mathscr{E}(c^{m-2},c_2^{m-1})\cdot\mathscr{E}(c_2^{m-1},c^m)=0$ for any $(m-2)$-cell c^{m-2} bounding c^m (Kovalevsky, 2008, Definitions BF and CBF, p.66).

If the bounding functions of all $(m-1)$-cells bounding an m-cell c^m are defined, c^m can have a maximum of two possible orientations. Two m-cells c_1^m and c_2^m, which have an $(m-1)$-cell c^{m-1} incident in common, are *coherently oriented* along c^{m-1} if the induced orientations of c^{m-1} with respect to c_1^m and c_2^m are opposite (Kovalevsky, 2008, Definition CO, p.67).

Definition ORC: An n-dimensional complex K^n is called *orientable*, if such orientations of all its principal cells exist that each pair of strongly connected principal cells of K^n is coherently oriented along its common $(n-1)$-cell.

Any connected n-dimensional complex C^n, having each of its cells proper or semi-proper, is called a *manifold with boundary* (Kovalevsky, 2008, Definition MB, p.70). This definition is more general than the classic because it allows manifolds with opening or generalized boundaries because a finite set is always compact. The manifolds can be orientable or non-orientable.

2.1.8 Block complexes

As was described in Section 2.2, the image segmentation bottom-up method divides the image into a large number of small regions, which are then merged to form larger regions. When the image is represented as a cell complex, it is necessary to partition it in subcomplexes that are considered uniform with respect to a particular criterion. The result of partitioning a cell complex into blocks, each consisting of a homogeneous group of cells based on a specific criterion, is called a *block complex* (Rinow, 1975; Kovalevsky, 1989).

Definition BC: The triplet $B(A)=(EB,BR,Dim)$ is called the *block complex* of A if there is a partition of A in subcomplexes S_i^k, every one homomorphic to an open k-dimensional ball. EB is the set of block cells, each cell $b_i\in EB$ corresponds to a subcomplex S_i^k called the k-dimensional block or k-block of A. The ordered pair (b_i,b_j) of block cell is in the bounding relationship BR if there is in A two cells $c_1\in S_i^k$ and $c_2\in S_j^m$ such that c_1 bounds c_2. The dimension function Dim assigns to each block cell b_i of $B(A)$ the dimension of the corresponding block S_i^k.

Figure 2.4 displays a partition divided into 13 blocks. In the figure, highlighted conveniently with a background that surrounds the original cells, are the subcomplexes corresponding to the block cells of the block complex that could be obtained. The subcomplexes correspond to 0-cells, and 1- and 2-dimensional open balls. From 35 0-cells, 58 1-cells, and 24 2-cells of the original cell complex, there are four 0-block cells, six 1-block cells, and three 2-block cells in the retrieved block complex.

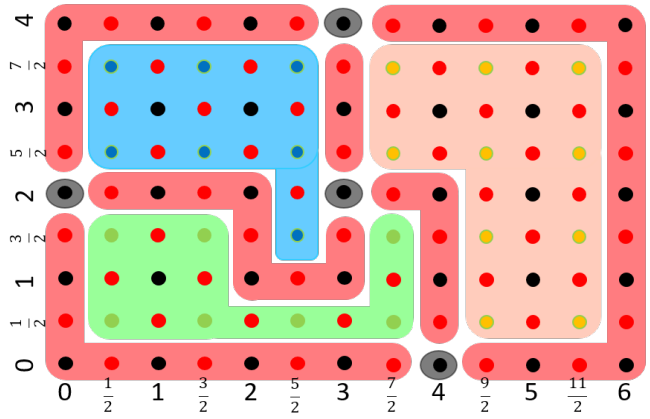


FIGURE 2.4: Example of block complex on a 2-dimensional Cartesian complex

2.1.9 Completely connected spaces

An $(n-1)$-dimensional LFS is completely connected when any two principal cells that have a common incident cell of any dimension also have a common incident $(n-1)$-cell. In a Cartesian complex, any connected homogeneously n-dimensional subcomplex is strongly connected. A *homogeneous completely connected (HCC) space* is an n-dimensional LFS S where each principal cell of S is isomorphic to a single polytope and any two principal cells that have a common incident cell of any dimension have exactly one common $(n-1)$-cell (Kovalevsky, 2008, Definition HCC, p.82). From the discussion context presented in (Kovalevsky, 2008, section 3.12, pp 81-83), the notion of isomorphism being used is the correspondence between each principal cell closure and a single polytope face lattice. A cell c_i bounds a cell c_j in a c principal cell closure if and only if there exist two faces F_k and F_l in a single polytope P such that $F_k \subset F_l$. This is shown in Figure 2.5 for a $2D$ Cartesian complex principal cell (a 2-cell). The binary bounding relationship for the c^2 principal 2-cell closure (on the bottom and left) and the face lattice for P polytope (on the bottom and right) are shown as well as some correspondences (yellow, blue and red double headed lines) between them.

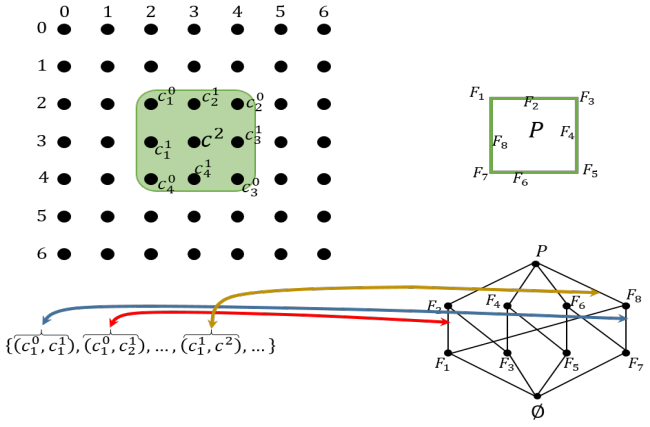


FIGURE 2.5: A principal 2-cell is isomorphic to a single polytope

The only 2D HCC is the hexagonal grid (the polytope is an hexagon) and when each principal cell in 3D LFS is isomorphic to a truncated octahedron, it is a 3D HCC space (Kovalevsky, 2008, Fig. 3.31 and 3.32, p.82-83).

2.1.10 Connectedness preserving mapping

Continuous mappings between topological spaces in the case of LFSs are limitedly applicable; isomorphism is the only continuous mapping appropriate to use between two LFSs. In mapping among multidimensional LFSs, it is impossible to map continuously one such space into a greater space. The only possible solution is to consider correspondence that is more general by assigning to each cell of a space a cell subset of another space.

A multivalued mapping between the complexes X and Y is a subset F of ordered pairs (x,y) containing all cells $x \in X$ and certain cells $y \in Y$. Given a correspondence F, the set of all $y \in Y$ encountered in pairs of F containing a fixed x is called the image of x and denoted by $F(x)$. The set of all $x \in X$ encountered in pairs of F containing a fixed y is called the preimage of y and denoted by $F^{-1}(y)$. $\bigcup_{x \in SX \subset X} F(x)$ is called the image of SX and denoted by $F(SX)$. $\bigcup_{y \in SY \subset Y} F(y)$ is called the preimage of SY and denoted by $F^{-1}(SY)$. It is possible to represent correspondences using their graphs. The graph of an F mapping between an n-dimensional complex X and an m-dimensional complex Y is an $(n+m)$-dimensional complex Z; hence, Z is a subset of the Cartesian product between complexes X an Y (Kovalevsky, 2008, p.85).

Table 2.1 presents a mapping example for two 1-dimensional complexes $X = \{x_0, x_1, x_2, x_3, x_4, x_5, x_6\}$ with the bounding relationship $\{(x_0,x_1),(x_2,x_1),(x_2,x_3),\ (x_4,x_3),(x_4,x_5),(x_6,x_5)\}$ and $Y = \{y_0,\ y_1,\ y_2,\ y_3,\ y_4,\ y_5\}$ with the bounding relationship $\{(y_0,y_1),(y_2,y_1),(y_2,y_3),(y_4,y_3),(y_4,y_5),(y_0,y_5)\}$; Figure 2.6 is the respective correspondence graph. In the figure, each member of F is highlighted with a circular blue background that surrounds it.

TABLE 2.1: Mapping between two examples of 1-dimensional complexes X and Y

Cells of X	x_0	x_1	x_2	x_3						x_4	x_5	x_6
Cells of Y	y_3			y_0	y_1	y_2	y_3	y_4	y_5	y_3		

When the preimage of any open subset of Y is open in X, the correspondence is *continuous*. The correspondence indicated in Figure 2.6 is continuous because any open subset of Y, which contains y_3, has X as a preimage and X (the entire set) is open. Any subset of Y that does not contain y_3 has $\{x_3\}$ as a preimage and $\{x_3\}$ is open.

The correspondence indicated in Figure 2.7 is noncontinuous because there exists open subsets of Y that have as a preimage non-open subsets of X, e.g., the open subset $\{y_3\}$ has $\{x_2\}$, which is closed, as a preimage. As in Figure 2.6, each member of F is highlighted with a circular blue background that surrounds it.

When the image of any connected subset of X is connected in Y, the correspondence is a *connectedness preserving mapping* (CPM) (Kovalevsky, 2008, Definition CMP, p.87). The correspondence displayed in Figure 2.6 is a CPM because any pair of incident elements of X that does not contain x_3 has $\{y_3\}$ as an image, and $\{y_3\}$ is connected (because the incidence relationship is reflexive). Further, any pair of incident elements of X that contains x_3 has Y as an image, and Y is connected (because between any two

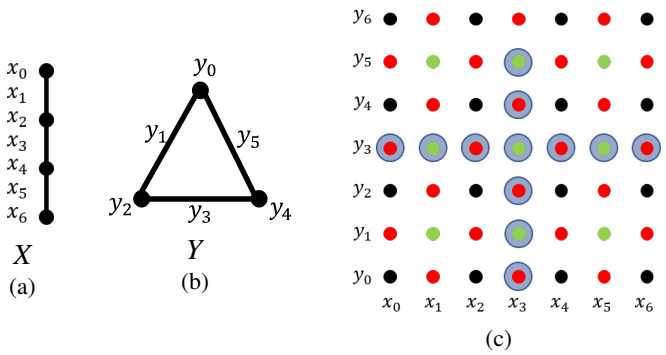


FIGURE 2.6: Complexes X (a) and Y (b) and graph (c) for mapping presented in Table 2.1

TABLE 2.2: Noncontinuous mapping between two 1-dimensional complexes X and Y

Cells of X	x_0	x_1	x_2						x_3	x_4
Cells of Y	y_4		y_0	y_1	y_2	y_3	y_4	y_5	y_4	

elements of Y there exists an incidence path containing them). The correspondence in Figure 2.6 is also a CPM, however, noncontinuous.

If $V(x,y)$ is the connected component of $F(x)$ containing y and $H(x,y)$ is the connected component of $F^{-1}(y)$ containing x, the correspondence F is *simple* if for each pair (x,y), at most one of the sets $V(x,y)$ and $H(x,y)$ contains more than one element (Kovalevsky, 2008, Definition SIC, p.88). Table 2.2 presents a noncontinuous mapping example for two 1-dimensional complexes. CPMs displayed in Figure 2.6 and Figure 2.7 are not simple. The correspondence presented in Table 2.3 with its graph displayed in Figure 2.8 is a continuous and simple CPM.

A *combinatorial homeomorphism* $X \sim Y$ uniquely specifies a continuous CPM $F : X \to Y$, whose inverse correspondence is also a continuous CPM. If X^n and Y^n are two n-dimensional LFSs and $F : X^n \to Y^n$ is a homeomorphism, then F is a *isomorphism* that preserves bounding relationship, i.e., for any two cells $a,b \in X^n$, the relationship $a < b$ implies $F(a) < F(b)$ (Kovalevsky, 2008, Theorems HCP and HI,

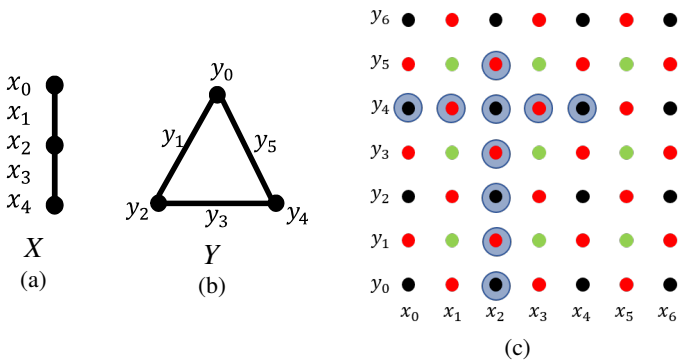


FIGURE 2.7: Complexes X (a) and Y (b) and graph (c) for mapping presented in Table 2.2

TABLE 2.3: Noncontinuous and simple mapping between two 1-dimensional complexes X and Y

Cells of X	x_0	x_1	x_2	x_3	x_4	x_5		
Cells of Y	y_0	y_1			y_2	y_3	y_4	y_5

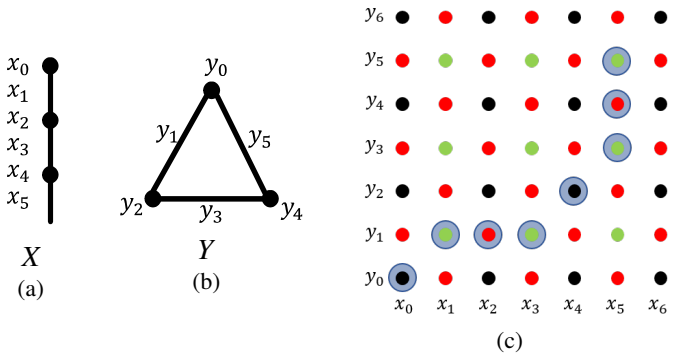


FIGURE 2.8: Complexes X and Y (a,b) and graph (c) for mapping presented in Table 2.3

p.90-91). When X and Y have unequal dimensions, this is called a *B-isomorphism*, which allows the comparison of subcomplexes having different space dimensions (Kovalevsky, 2008, Definition BI, p.92).

Figure 2.9 displays a B-isomorphism, a subcomplex, whose elements are highlighted with a circular blue background, is 2-dimensional (green dots are 2-cells), and the other, whose elements are highlighted with a circular red background, is a 1-dimensional (red dots are 2-cells and black dots are 0-cells). It should be clear that if these complexes are seen as k-complexes, both of them are 1-dimensional.

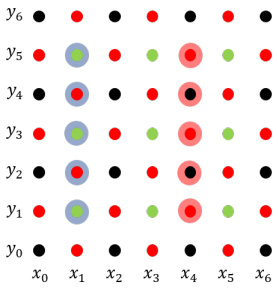
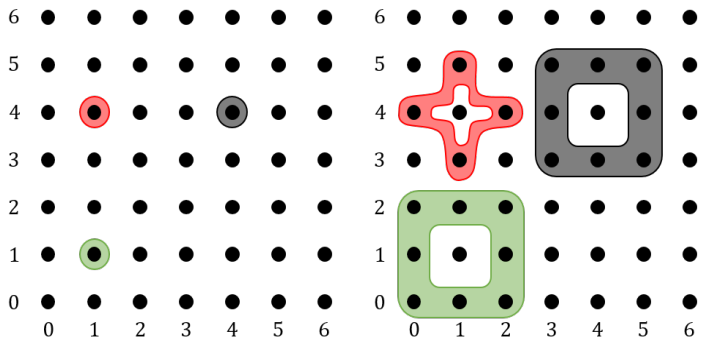


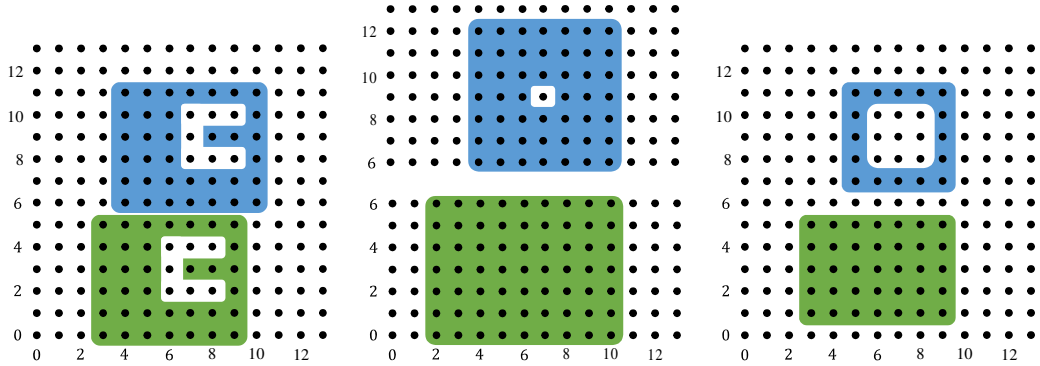
FIGURE 2.9: B-isomorphism example between 2-dimensional complex and 1-dimensional complex

Each cell c in an n-dimensional complex C^n has its *incidence structure* defined as $SON^*(c,C^n) \cup Cl^*(c,C^n)$ (Kovalevsky, 2008, Definition IST, p.93). Figure 2.10(a) displays a 0-cell (with a black circular background), a 1-cell (with a red circular background), and a 2-cell (with a green circular background) in a 2-dimensional Cartesian complex whose cells are all represented as black dots and use combinatorial coordinates. The incidence structures for each cell are displayed in Figure 2.10(b), representing each incidence structure subcomplex as a background with the respective color. In all of these cases, it can be seen that the incidence structure is isomorphic to a 1-dimensional sphere.



(a) Original cells (b) Incidence structures

FIGURE 2.10: Incidence structure for 0-cell (dark gray encircled dot), 1-cell (red encircled dot), and 2-cell (green encircled dot) in 2-dimensional Cartesian complex



(a) Two subcomplexes (b) their closures (c) interior of their closures

FIGURE 2.11: Solid and nonsolid subcomplexes

A homogeneous n-dimensional subcomplex that has no cell of its complement in the interior of its closure is a *solid*, i.e., a solid contains the interior of its closure (Kovalevsky, 2008, Definition SL, p.95). Figure 2.11(a) displays two subcomplexes, the cell subset represented with a blue background is a solid and that with a green background is not. This can be verified in Figure 2.11(b) and Figure 2.11(c); the former indicates the closure of each and the latter the interiors. The blue subset contains the interior of its closure, whereas the green does not.

2.1.11 Digital geometry based on topology of complexes

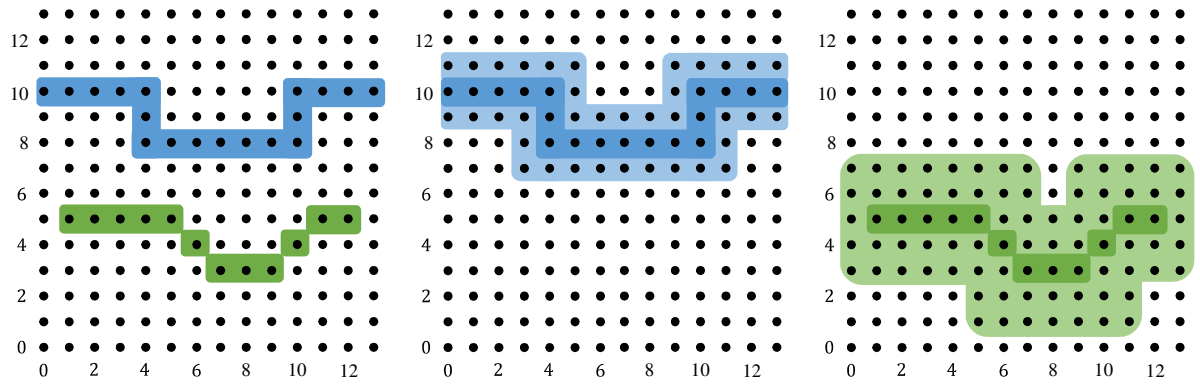
The concept of geometry presented here is based on AC topology without any relation to a Euclidean space and without the use of infinitesimals. A fundamental concept is that of a digital curve, which serves as a precise limit of 2-dimensional objects; therefore, they should be 1-dimensional objects.

A *boundary curve* is a connected subcomplex of a 1-dimensional sphere. The 0-cells are called *points*, the 1-cells are called *cracks* and 2-cell are called *pixels*. Boundary curves are the suitable means to specify

exactly the boundary of the subsets to analyze and to encode the contents of 2-dimensional digital images. A boundary curve, which is a 1-dimensional object, cannot be visualized on a computer screen because only 2-dimensional objects can appear on a screen; therefore, it is necessary to use the opening frontier (Of) concept. The *open hull* $Op(S,C)$ of a subcomplex S of a complex C is the smallest open subset of C that contains S, $Op(S,C) = S \cup Of(S,C)$ (Kovalevsky, 2008, Definitions BCU and OPH, p.109).

A visual curve (VCU) is defined in terms of its width: (1) a VCU width of "0" is a boundary curve, (2) a VCU width of "1" is a connected set of straight (horizontal or vertical) 2-dimensional paths with pairs of them connected by only one point, which is connected to only one element of each 2-dimensional path, and (3) a VCU width of W, $W \geq 2$, is the open hull of the closure of a VCU width of $W-2$ (Kovalevsky, 2008, Definitions VCU, p.110).

Figure 2.12 displays VCU widths of "0", "1", "2", and "3". Figure 2.12(a) illustrates two subcomplexes, the first, which is highlighted with a blue background, corresponds to a VCU width of "0" composed of only a 0-cell and 1-cell. Actually, it is a boundary curve. The second, which is highlighted with a green background, corresponds to a VCU width of "1" as it is a set of three 2-dimensional paths connected by two points. Figure 2.12(b) displays a VCU width of "2" as the open hull of the closure of the VCU ($Op(Cl(S,C))$) width of "0" displayed in Figure 2.12(a). Figure 2.12(c) displays a VCU width of "3" as the open hull of the closure of the VCU width of "1" displayed in Figure 2.12(a).



(a) VCU width of "0" (upper), VCU width of "1" (bottom)

(b) Op of blue subcomplex

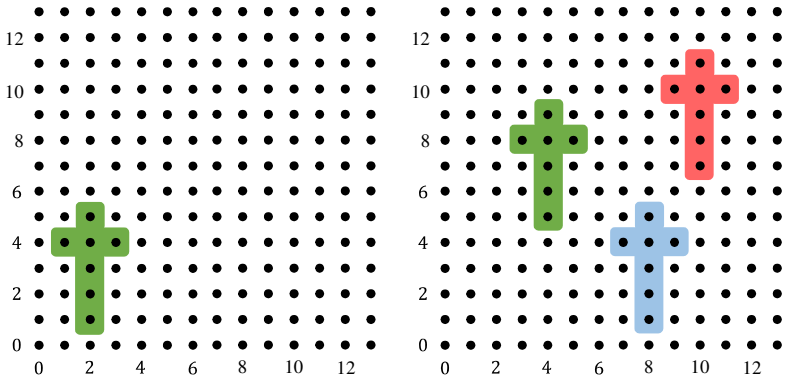
(c) Op of green subcomplex

FIGURE 2.12: VCU widths of "0", "1", "2", and "3"

2.1.12 Digital analytical geometry

The digital plane is a 2-dimensional AC that is composed of 0-cells called *points*, 1-cells called *cracks*, and 2-cells called *pixels*. Actually, the cell set of a 2-dimensional Cartesian ACC C^2 is the Cartesian product of two 1-dimensional connected ACC sets. The 1-dimensional complexes are the coordinate axes of C^2, which become a 2-dimensional Cartesian complex. A cell of C^2 is a 2-tuple $c = (a_1, a_2)$ of a_i cells each belonging to a particular 1-dimensional complex and becoming a component cell of c. In this case, a cell $c_1 = (a_1, a_2)$ bounds another cell $c_2 = (b_1, b_2)$ if for each pair of component cells a_i and b_i $a_i = b_i$ or a_i bounds b_i. The dimension of a cell c is the sum of the dimensions of its component cells.

To employ formulae from classical geometry without leaving the frame of cell complexes, rational numbers change to real numbers. The translation $f(p^n) : C^n \to C^n$ maps an n-dimensional complex onto itself, giving to a cell with the coordinate vector $v^n \in C^n$ the coordinate vector $w^n \in C^n$ such that $w^n = v^n + p^n$. Figure 2.13 displays the movement of a subcomplex, which is displayed in Figure 2.13(a), using three different parameter vectors, $(2,4)$, $(6,0)$, and $(8,6)$, whose movement are indicated in Figure 2.13(b) in green, blue, and red, respectively. It should be noticed that the movement must be performed by multiples of two to not change the dimension of the moved subcomplex in relation to the space (considered as a k-complex).

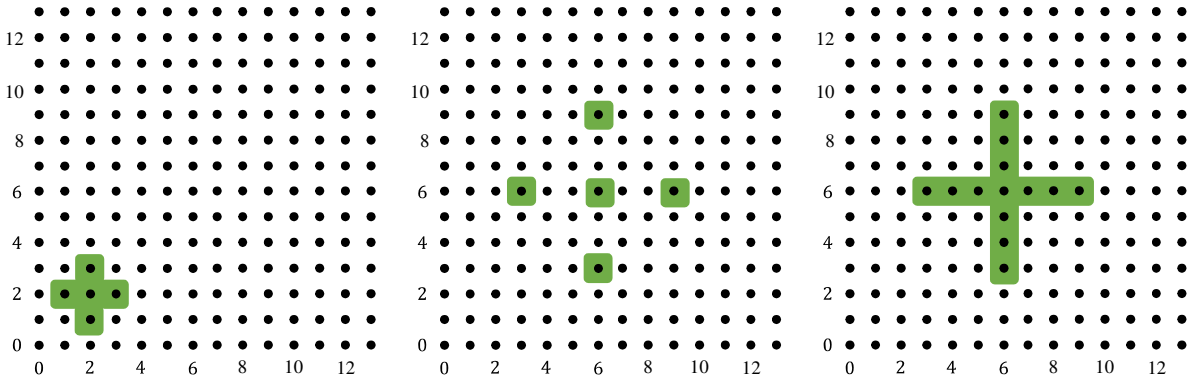


(a) Subcomplex

(b) Movements by vectors $(2,4)$ (green), $(6,0)$ (blue), and $(8,6)$ (red)

FIGURE 2.13: Subcomplex translation

The multiplication of coordinates of a connected subcomplex by an integer number $w^n = av^n$ could result in a disconnected subcomplex. Figure 2.14(a) illustrates a 1-dimension connected subcomplex example S composed of five cells, one 0-cell and four 1-cells; Figure 2.14(b) is its multiplication $T = 3S$. Hence, the magnification of a subcomplex must be redefined based on a "one-to-many" CPM to obtain a connected magnified subcomplex. Figure 2.14(c) displays the magnification outcome of the original subcomplex by a factor 3 using the one-to-many CPM indicated in Table 2.4.



(a) Original subcomplex S

(b) Multiplication $T = 3S$

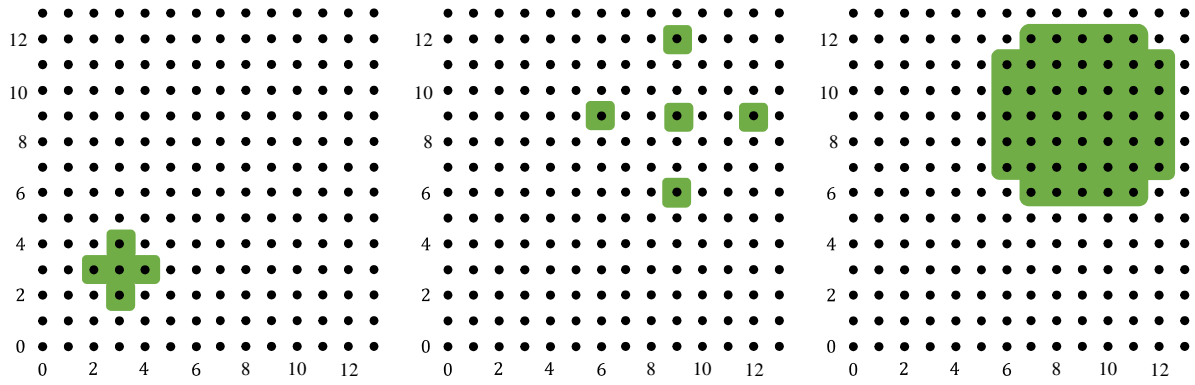
(c) S magnification by 3

FIGURE 2.14: 1-dimension subcomplex magnification

TABLE 2.4: Mapping between 1-dimensional subcomplexes displayed in Figures 2.14(a) and 2.14(c)

S's cells	$(2,1)$			$(1,2)$			$(2,2)$	$(3,2)$			$(2,3)$		
T's cells	$(6,3)$	$(6,4)$	$(6,5)$	$(3,6)$	$(4,6)$	$(5,6)$	$(6,6)$	$(7,6)$	$(8,6)$	$(9,6)$	$(6,7)$	$(6,8)$	$(6,9)$

When the original subcomplex is 2-dimensional, its magnification must be performed in both dimensions as indicated in Figure 2.15 using a suitable one-to-many CPM. The CPM can be conducted by an elementary subdivision, which uses for each cell of the subcomplex, its two component cells. If a component cell is a 0-cell, it must not be subdivided; however, if it is a 1-cell, it must be subdivided into k 1-cells and $k-1$ 0-cells where k is the given magnification factor. Hence, a 1-cell c_i is replaced by $c_{i-1+\frac{1}{k}}, c_{i-1+\frac{2}{k}}, \cdots, c_{i-1+\frac{2k-1}{k}}$, which is a sequence of 1-cell, 0-cell, 1-cell, 0-cell, continuing until finally ending in a 1-cell. To complete the magnification process, each subdivision of a component cell must be combined with each subdivision of the other component to form elements on the 2-dimensional Cartesian complex after multiplying each one by the k factor. Figure 2.15(c) displays the magnification of the subcomplex in Figure 2.15(a) with a factor of "3". Figure 2.15(b) displays the multiplication of the subcomplex in Figure 2.15(a) by "3".



(a) Original subcomplex S (b) Multiplication $T = 3S$ (c) S magnification by 3

FIGURE 2.15: 2-dimension subcomplex magnification

If an orthogonal matrix M is defined as a matrix whose transposed matrix is equivalent to its inverse, its determinant is equal to 1 and the matrix has the form: $\begin{bmatrix} a & s\cdot\sqrt{1-a^2} \\ -s\cdot\sqrt{1-a^2} & s\cdot a \end{bmatrix}$; where $s = \pm 1$ and $\mid a \mid \leq 1$. It is the matrix of a rotation eventually with reflection in the X-axis if $s < 0$ (Kovalevsky, 2008, p.113).

Figure 2.16(b) displays the rotation (and subsequent translation) of the subcomplex in Figure 2.16(a) with $\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} e_1 \\ e_2 \end{bmatrix} + \begin{bmatrix} 6 \\ 0 \end{bmatrix}$ i.e., $a = 0$ and $s = -1$. In this research, however, it was necessary to use only the rotation on 2-cell sub-complexes.

As the rotation must retain the length of a vector, it must select a metric for this. The metric is the Euclidean distance between two cells c_1 and c_2 defined as $Dist\,(c_1,c_2) = \sqrt{(x_2-x_1)^2+(y_2-y_1)^2}$, where

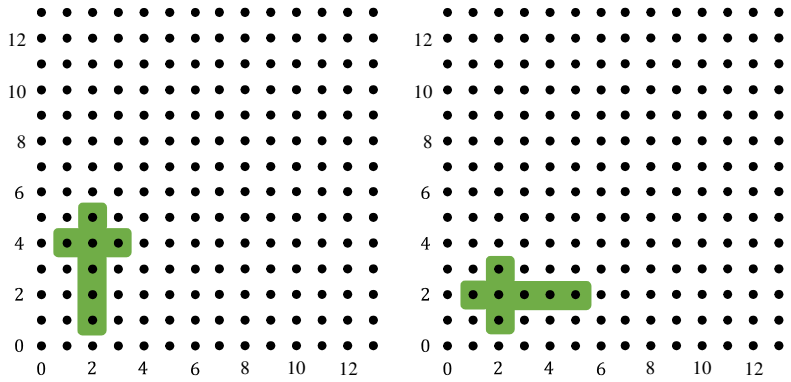


FIGURE 2.16: Subcomplex rotation

(x_1, y_1) and (x_2, y_2) are the combinatorial coordinates of c_1 and c_2, respectively. In a 2-dimensional Cartesian complex the distance d between two cells is *digitally equal* to a number b when $|d - b| \leq 2\sqrt{2}$.

2.1.13 Linear inequalities in Locally Finite Spaces

For a Cartesian complex, it is possible to define its subcomplex using equations and inequalities based on collinearity, half-space, and convex set concepts. Given three cells c_1, c_2, and c_3 of a 2-dimensional space with combinatorial coordinates x_1, y_1, x_2, y_2, and x_3, y_3, respectively, the line slope containing c_1 and c_2 must be the same as the line slope containing c_1 and c_3 for establishing a collinearity among the three cells; hence, $\frac{y_2 - y_1}{x_2 - x_1} = \frac{y_3 - y_1}{x_3 - x_1}$, i.e.,

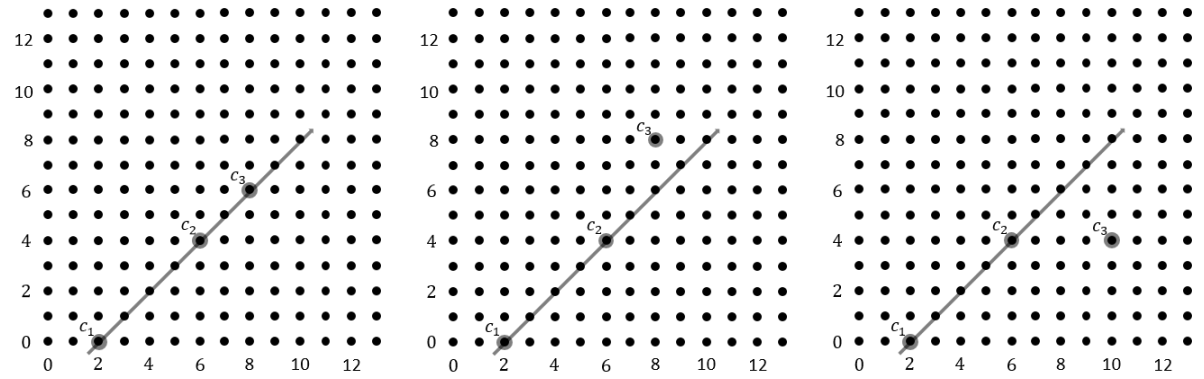
$$(x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1) = 0$$

.

Figure 2.17(a) displays the collinearity among the three highlighted cells. When $\frac{y_2 - y_1}{x_2 - x_1} < \frac{y_3 - y_1}{x_3 - x_1}$, the c_3 cell is on the positive side of the line containing c_1 and c_2 (Figure 2.17(b)). When $\frac{y_2 - y_1}{x_2 - x_1} > \frac{y_3 - y_1}{x_3 - x_1}$, the c_3 cell is on the negative side (Figure 2.17(c)) (Kovalevsky, 2008, Definitions COL, p.115).

In a Cartesian complex, a 2-dimensional vector (x, y) is *right semi-collinear* (or *left semi-collinear*) with another vector (n, m) such that $0 \leq (x \cdot M - y \cdot N) \leq |M| + |N| - 1$ (or $0 \leq (y \cdot N - x \cdot M) \leq |M| + |N| - 1$), where M and N are the numerator and denominator of the irreducible fraction $M/N = m/n$. Figure 2.18 displays the collinear (gray rounded background), right semi-collinear (green rounded background), and left semi-collinear (yellow rounded background) cells with the vector $(n, m) = (8, 10)$, which has $M/N = 5/4$ as its irreducible fraction.

A *closed digital half-space* of an n-dimensional space is the closure of the set of all n-cells whose coordinates satisfy a linear inequality. When the Cartesian complex is 2-dimensional, the half-space is called a *digital half-plane* (Kovalevsky, 2008, Definitions HS, p.116). Figure 2.19(a) displays the digital half-plane for the inequality $5x - 4y \geq 0$, the 2-cells that satisfy the inequality are highlighted with a green



(a) Collinearity of three cells (b) Cell on positive side of line (c) Cell on negative side of line

FIGURE 2.17: Collinearity

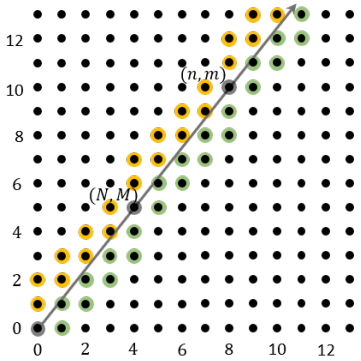


FIGURE 2.18: Semi-collinear cells with vector **(n, m) = (8, 10)**

rounded background and the cells belonging to the respective closure, which is the actual half-plane, are highlighted with a blue background. In the figure, the vector line is indicated in gray, as a guide only.

Any connected subset of the frontier of a digital half-plane is called a *digital straight-line segment* (DSS) (Kovalevsky, 2008, Definitions DSS, p.116). Figure 2.19 displays the digital straight line as the frontier of the digital half-plane in a finite space corresponding to the $5x - 4y \geq 0$ inequality. As can be seen, the DSS, which is represented as red dots, is composed of only 0-cells and 1-cells, as expected; all are semi-collinear (right or left) with the vector line, which is represented with a grey line, as a guide only.

Since a DSS is a boundary curve, it can be assigned an orientation and thus, its cracks (1-cells). Doing this, each 1-cell of a DSS is incident to exactly two pixels (2-cells), one lies to the negative side and the other to the positive side of the ordered pair of the points (the end 0-cells of the 1-cell). The set of all positive pixels of a boundary curve K is called the *positive pixel set* of K, and the set of all negative pixels is called the *positive pixel set* of K (Kovalevsky, 2008, Definitions PP, p.118). Figure 2.20 displays a boundary curve (corresponding to the DSS in Figure 2.19(b)), which is represented as red dots, with an assigned orientation (indicated as a red dashed headed line). The respective positive pixel set (the cells displayed highlighted with a blue round background) and negative pixel set (the cells displayed highlighted with a green round background) are also indicted. When the boundary curve K is a DSS, as

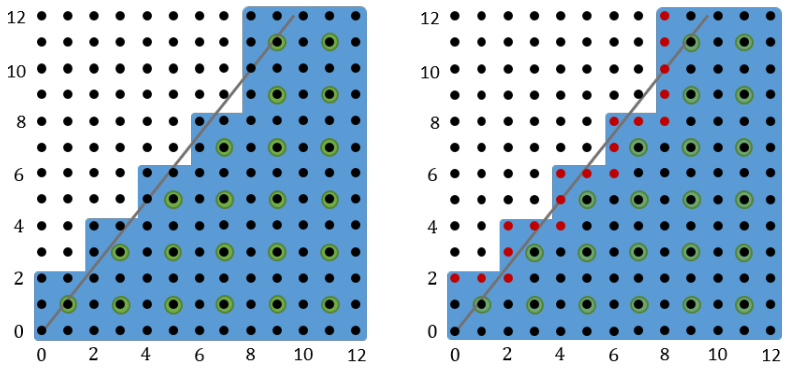


FIGURE 2.19: Half-plane for $5x-4y \geq 0$

is the case in Figure 2.20, there is a linear form $H(x,y)$, which is called the *separating* form, such that $H(x,y) \geq 0$ for all positive pixels of K and $H(x,y) < 0$ for all of its negative pixels.

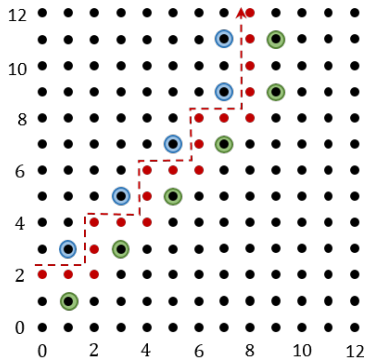


FIGURE 2.20: Adjacent pixels (2-cells) of boundary curve (red dots) - Positive (dots with blue round background) and negative (dots with green round background) pixels (2-cells)

In general, $H(x,y) = a \cdot x + b \cdot y + c$ is the *standard separating form* (SSF) of a DSS D if: (1) $H(x,y) \geq 0$ for all positive pixels of D and $H(x,y) < 0$ for all of negative pixels, (2) there are at least two positive pixels of D, where $H(x,y)$ takes its minimum value or two negative pixels of D where $H(x,y)$ takes its maximum value, and (3) $H(x,y)$'s coefficients are integers with a and b mutually prime. The *positive base* of D is the set of all positive pixels of D where $H(x,y)$ takes its minimum value and the *negative base* of D is the set of all negative pixels of D where $H(x,y)$ takes its maximum value (Kovalevsky, 2008, Definitions SSF and BS, p.119).

Figure 2.21 displays the oriented DSS in Figure 2.20, represented as red dots. The DSS negative base expressed in combinatorial coordinates is composed of $(1,1)$ and $(9,11)$ 2-cells (pixels), represented in the figure by 2-cells highlighted with a green round background. The positive base is composed of the $(7,9)$ 2-cell (pixel) represented in the figure by the 2-cells highlighted with a blue round background. The pixel of the bases that is the nearest to the oriented DSS starting point is called the *starting pixel*

(cell $(1,1)$), and the pixel that is the nearest to the oriented DSS ending point is called the *ending pixel* (cell $(9,11)$). The vector parallel to starting / ending pixels is called the *base vector* of the DSS and is represented as a headed red line. The corresponding SSF is $H(x,y) = 4y-5x-1$, which is calculated with the positive base pixels' combinatorial coordinates, and as expected, $H(x,y) \geq 0$ for all positive pixels and $H(x,y) < 0$ for all negative pixels and is represented as a headed blue line. $H'(x,y) = 4y-5x+1$ is another SSF; however, in this case $H'(x,y) > 0$ for all positive pixels and $H'(x,y) \leq 0$ for all negative pixels and is represented as a headed green line.

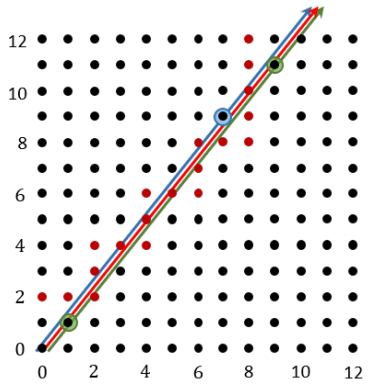


FIGURE 2.21: Bases of a DSS
Positive (*dots with blue round background*) and negative (*dots with green round background*); $H(x,y)$ (*blue headed line*) and $H'(x,y)$ (*green headed line*)

In the following, several properties for any DSS D in a finite 2-dimensional Cartesian complex are described:

(1) There exists an SSF for which, the absolute values of the coefficients a and b are no greater than the greatest dimension of the space. If D contains at least two 1-cells, then the SSF is unique (Kovalevsky, 2008, Lemma SSF, p.120).

(2) It contains oriented 1-cells of at most two non-opposite directions (Kovalevsky, 2008, Lemma TWD, p.121).

(3) At least one of the directions of the cracks occurs singularly, i.e., each 1-cell of that direction is followed immediately by a 1-cell of the other direction. The singular direction is vertical if $|a| \leq |b|$ and is horizontal if $|a| \geq |b|$. If $|a| = |b|$, both directions are singular (Kovalevsky, 2008, Lemma SING, p.121).

(4) Each positive pixel of a DSS D is incident to a 1-cell of the main direction (the orthogonal direction to the singular direction). This 1-cell is incident to a negative pixel of D (Kovalevsky, 2008, Lemma INC, p.122).

(5) For D's SSF $H(x,y) = ax+by+r$, positive pixels $H(x,y)$ take all values in the closed interval $[0, 2(\max(|a|,|b|)-1)]$, which are multiples of two; negative pixels $H(x,y)$ take all values in the closed interval $[-2(\max(|a|,|b|)-1), -2]$, which are multiples of two. x and y are given in combinatorial coordinates (Kovalevsky, 2008, Lemma VAL, p.123).

(6) There are no pixels in the space between the bases of a DSS (Kovalevsky, 2008, Lemma BB, p.125).

All the properties described above are important to define algorithms for the recognition of DSSs on digital images, in addition to the following three:

(7) If two vectors V_1 and V_2 correspond to two pixels (2-cells), then the vector $V = 2V_1 - V_2$ is also a pixel (2-cell) (Kovalevsky, 2008, Lemma SYM, p.126).

(8) For $(x,y) = ax + by + r$, whose integer coefficients a and b are mutually prime, and two vectors $P_1 = (x_1,y_1)$ and $P_1 = (x_2,y_2)$, if $H(P_2) - H(P_1) = M \neq 0$, then the differences $dx = x_2 - x_1$ and $dy = y_2 - y_1$ have no common divider greater than $|M|$ (Kovalevsky, 2008, Lemma CD, p.126).

(9) If two vectors $P_1 = (x_1,y_1)$ and $P_1 = (x_2,y_2)$ belong to different bases of a DSS, then the values $b = (x_2 - x_1)/2$ and $a = (y_2 - y_1)/2$ are mutually prime (Kovalevsky, 2008, Corollary JB, p.126).

2.2 Digital image segmentation

Image segmentation is the image partitioning into regions that later becoming different objects or parts of objects. At the end, each image pixel is in a category. A segmentation is the outcome of clustering neighbor pixels in the same category because they have similar greyscale of multivariate values and form a connected region, and neighboring pixels which are in different categories have dissimilar values (Glasbey and Horgan, 1995). Image segmentation has been widely studied as a fundamental procedure in computer vision. By region splitting or merging, or optimizing some global criterion, an attempt is made to measure intra-region consistency or inter-region dissimilarity. Szeliski (2010) presents some techniques that have been developed for image segmentation including algorithms based on active contours (Kass, Witkin, and Terzopoulos, 1988; Mortensen and Barrett, 1995) and level sets (Osher and Paragios, 2003), region splitting and merging (Horowitz and Pavlidis, 1976), mean shift (Comaniciu and Meer, 2002), normalized cuts (Shi and Malik, 2000), and binary Markov random fields solved using graph cuts (Boykov and Funka-Lea, 2006). Work has also been carried out based on a unified approach that includes both detection of contours (edge-detection) and definition of regions (segments) (Arbelaez et al., 2011) including multi-scale texture analysis and gradient calculation and a hierarchical partitioning of the image. Besides being in the top ten of the best algorithms for image segmentation (according to what is reported by Arbelaez et al. (2011)), this approach is of special interest for the present research because it includes a large number of tasks that can be reformulated in terms of cell geometrical and topological features available in the Cartesian complex image representation.

Below are the concepts that give support to the unified approach to image segmentation. In the Section 2.2.1 is the formal definition of the segmentation process introduced by Horowitz and Pavlidis (1976), Section 2.2.2 presents the concepts related to the edge detection based on texture analysis from first and second derivative Gaussian filters. Section 2.2.3 presents the gradient calculation based on the local histogram difference. Section refsec:NormalizedCut describes a segmentation global cue based on normalized cuts. Finally, the Section 2.2.5 introduces the image partitioning based on the watershed transform to obtain the final segments.

2.2.1 Image segmentation process

Image segmentation can be done subdividing an image into smaller regions (top-down) based on the homogeneity or cohesiveness among groups of near pixels (Lézoray and Grady, 2012, p.111), or grouping image pixels into connected homogeneous components (bottom-up) called regions. These regions are determined by two kinds of methods (Brun, Domenger, and Mokhtari, 2003): (1) edge-detection methods and (2) region-based image segmentation methods. The former determine edges between regions and then closes them to define a partition. The latter group pixels according to a homogeneity criterion aiming to obtain a partition of the image into homogeneous regions. The segmentation of an image X (the space domain) subdivides it, based on a function f (the feature domain) defined on X, using a logical predicate P on subsets S of X as in Equation 2.4 (Horowitz and Pavlidis, 1976), where e is a prescribed error tolerance.

$$P(S) = \begin{cases} true \text{ if there exists a constant } a \text{ such that } |f(x,y) - a| \leq e, \forall (x,y) \in S \\ false \text{ otherwise} \end{cases} \tag{2.4}$$

It should be noted that in general, for multispectral images, the space domain $X \subset \mathbb{R}^n$ and the feature domain $F \subset \mathbb{R}^m$ such that $f : X \to F$ and $a, e \in F \subset \mathbb{R}^m$. A segmentation of X is a partition of X into subsets $S_i, i = 1, \ldots, m$ for some m such that

(1) $X = \bigcup_{i=1}^{m} S_i$,

(2) $S_i \cap S_j = \emptyset, \forall i \neq j$,

(3) $P(S_i) = true, \forall i$,

(4) $P(S_i \cup S_j) = false, \forall i \neq j$, S_i and S_j are adjacent in X.

Conditions (1) and (2) ensure that the image is partitioned into a set of regions. Condition (3) ensures that each region is homogeneous according to the homogeneity criterion P. Condition (4) ensures that all regions are maximal, thus that any merge of two adjacent regions produces a non-homogeneous region (Brun, Domenger, and Mokhtari, 2003).

Region detection methods can be addressed in two fashions (Horowitz and Pavlidis, 1976). The first, merging or bottom-up, divides the image into a large number of small regions, which are then merged to form larger regions. The other, splitting or top-down, successively divides the image into smaller and smaller regions until certain criteria are satisfied. It is possible to devise a mixed split-and-merge. A merging scheme starts with a partition satisfying (c) and proceeds to fulfil (d); a splitting scheme starts with a partition satisfying (d) and proceeds to fulfil (c). A split-and-merge procedure begins with an arbitrary partition satisfying neither condition and produces a partition satisfying both.

The information provided by a partition can be primarily geometrical or topological. The geometrical information relates to each region considered separately from the partition. The set of pixels composing a region, the region including a pixel or the boundary of a given region, can be classified as geometrical information. The topological information describes the relationships between regions. The set of regions adjacent to a given region or the set of regions included in a region belong to the topological information field (Brun, Domenger, and Mokhtari, 2003).

Image segmentation can be viewed as a labelling problem (Ishikawa, 2012), obtaining from the image only the neighborhood topological relationships and placing them on an undirected region adjacency graph (RAG) (Brice and Fennema, 1970; Cheevasuvit, Maitre, and Vidal-Madjar, 1986) along with a set of labels. The problem is then to determine the best labelling according to the criteria in the problem's requirements. "Energy" is a translation of the criteria into a function that evaluates how well the given labelling is, for example a smaller energy for a labelling means a better corresponding solution to the problem; thus, the problem becomes an energy minimization problem that can be solved using general algorithms. As the minimization of such energies, in general, is known to be NP-hard, graph cut methods utilizing the $s-t$ min-cut algorithms known in operations research are used (Lézoray and Grady, 2012,p.26).

2.2.2 Oriented edge-detection based image segmentation

In edge-based segmentation, an edge filter is applied to each pixel image in order to classify it as edge or non-edge depending on the filter output. Pixels which are not separated by an edge are allocated to the same category (Glasbey and Horgan, 1995). Local (Neighborhood) operators are used to filter images in order to accentuate edges. The most commonly used type of neighborhood operator is a linear filter, in which an output pixel value is determined as a weighted sum of input pixel values by using Equation 2.5 called the convolution operator (Szeliski, 2010).

$$g(i,j) = \sum_{k,l} f(i-k, j-l)h(k,l) = \sum_{k,l} f(k,l)h(i-k, j-l). \tag{2.5}$$

The Equation 2.5 is more compactly notated using Equation 2.6.

$$g = f * h. \tag{2.6}$$

Segmentation is based on the detection of discontinuities (edges) that break invariant intensity value regions, i.e., edges occur at boundaries between regions of different color, intensity, or texture. To detect edges, it is commonly used only local information defining an edge as a location of rapid intensity variation. A mathematical way to define the slope and direction of a surface is through its gradient (Equation 2.7).

$$\bigtriangledown I(x,y) = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right)(x,y). \tag{2.7}$$

It is common to smooth the image with low-pass filter prior to computing the gradient because image derivatives amplify noise. The Gaussian filter is used in most edge detection algorithms because this is circularly symmetric. The gradient of the smoothed image can be written as Equation 2.8 using the convolution operator (Equation 2.6).

$$\bigtriangledown [G_\sigma(x,y) * I(x,y)] = \bigtriangledown [G_\sigma](x,y) * I(x,y). \tag{2.8}$$

For $G_\sigma(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$, $\bigtriangledown[G_\sigma](x,y)$ is given by Equation 2.9.

$$\bigtriangledown[G_\sigma](x,y) = \begin{bmatrix} -x & -y \end{bmatrix} \frac{1}{\sigma^2} G_\sigma(x,y). \tag{2.9}$$

where σ indicates the scale of the Gaussian.

Higher-order steerable filters can be used when it is necessary to restrict the edge orientation. These filters can model curve intersections because they allow to represent multiple orientations at the same pixel. One approach to find the response of a filter at many orientations is to apply many versions of the same filter, each of which is different from the others by some small rotation in angle (Freeman and Adelson, 1991). $\bigtriangledown[G_\sigma](x,y)$ at a given orientation θ can be established by Equation 2.10.

$$\bigtriangledown^\theta[G_\sigma](x,y) = \begin{bmatrix} cos(\theta) & sin(\theta) \end{bmatrix} \bigtriangledown[G_\sigma]^T(x,y). \tag{2.10}$$

It is possible to steer any order of Gaussian derivative, however, Hilbert transform need to be used for second-order and higher filters (Freeman and Adelson, 1991).

2.2.3 Oriented gradient based boundary strength

Martin, Fowlkes, and Malik (2004) introduce a gradient-based paradigm for detecting local changes in image intensity (color, texture , etc.). They draw a circle of radius r at each location (x,y) in the image, and divide it along the diameter at orientation θ. The gradient function $G(x,y,\theta,r)$ compares the contents of the two resulting disc halves and when the difference between them is large, this indicates a discontinuity in the image along the disc's diameter. In each disc half, their kernel density estimates are binned and histograms in each disc half are computed. so, the half-disc g and h regions are described by histograms, which are compared with the χ^2 histogram difference operator by Equation 2.11 (Rubner et al., 2001) as a measurement of boundary strength.

$$\chi^2(g,h) = \frac{1}{2}\sum_i \frac{(g(i)-h(i))^2}{g(i)+h(i)}. \tag{2.11}$$

2.2.4 Normalized cut based global cues

Top-down splitting techniques divide regions into coherent subregions. So does the normalized cuts technique (Shi and Malik, 2000). In this, nearby pixels are assessed for affinity and when groups that are connected by weak affinities are found, they are separated. The cut between two groups A and B is defined as the sum of all the weights being cut as established by Equation 2.12 (Szeliski, 2010).

$$cut(A,B) = \sum_{i\in A, j\in B} w_{i,j}. \tag{2.12}$$

The weights $w_{i,j}$ between two pixels (or regions) i and j measure their similarity. When a minimum cut is used as a segmentation criterion, the smallest cuts usually involve isolating a single pixel, which can result in unreasonable clusters. A better measure of segmentation is the normalized cut, which is defined by Equation 2.13.

$$Ncut\,(A,B) = \frac{cut\,(A,B)}{assoc\,(A,V)} + \frac{cut\,(A,B)}{assoc\,(B,V)}. \tag{2.13}$$

$assoc(A,V) = assoc(A,A) + cut(A,B)$ is the sum of all the weights associated with nodes in A. $assoc(A,A) = \sum_{i\in A, j\in A} w_{i,j}$ is the association (sum of all the weights) within a cluster. Since to compute the optimal normalized cut is NP-complete, Shi and Malik (2000) suggest computing a real-valued assignment of nodes to groups. For the indicator vector $\mathbf{x}$ ($x_i = +1$ iff $i \in A$ and $x_i = -1$ iff $i \in B$), the row sums of the symmetric matrix $\mathbf{W}$ given by $\mathbf{d} = \mathbf{W}_1$ and the corresponding diagonal matrix $\mathbf{D} = diag(\mathbf{d})$, minimizing the normalized cut over all possible indicator vectors $\mathbf{x}$ is equivalent to minimizing

$$\min_{\mathbf{y}} \left\{ \frac{\mathbf{y}^T (\mathbf{D} - \mathbf{W})\,\mathbf{y}}{\mathbf{y}^T \mathbf{D}\mathbf{y}} \right\} .$$

where $\mathbf{y} = ((\mathbf{1} + \mathbf{x}) - b(\mathbf{1} - \mathbf{x}))/2$ is a vector consisting of all 1s and $-b$s such that $\mathbf{y} \cdot \mathbf{d} = 0$. Minimizing this quotient is equivalent to solving the generalized eigenvalue system given by Equation 2.14.

$$(\mathbf{D} - \mathbf{W})\,\mathbf{y} = \lambda \mathbf{D}\mathbf{y}. \tag{2.14}$$

2.2.5 Watershed transform based image partitioning

In mathematical morphology, it is usual to consider the gray level (the image intensity) image as a topographical surface, i.e., the places where the intensity sharply changes are where it makes sense to search for contour lines. For segmentation purposes, gradient images have become a suitable source from which to perform this calculation (Najman and Schmitt, 1996). This technique consists of segmenting an image into several regions (interpreted as a height field or landscape) where rain would flow into the same lake (catchment basin) (Szeliski, 2010).

The watershed approach allocates an influence zone to each of the regional minima of an image. The boundaries of these influence zones define the watersheds. This approach assumes that each regional minimum has a small hole, which allows the water progressively to flood it when the image is slowly immerse into a "lake". To avoid merging adjacent "catchment basins", the crest line is progressively raised up to process end. The set of raised crest lines is the watershed of the image. When the crest line is a plateau, it is kept its (geodesic) middle line.

For two pixels p_i and p_j on an image A, the geodesic distance $d_A(p_i, p_j)$ is the lower bound of the length of the paths γ linking p_i and p_j. For $B \subset A$ the geodesic distance from a pixel p_j to B is given by Equation 2.15.

$$d_A(p_j, B) = \min_{p_k \in B} \left\{ d_A(p_j, p_k) \right\}. \tag{2.15}$$

The geodesic influence zone of a connected component B_i of B is the set of pixels of A for which the geodesic distance to B_i is smaller than the geodesic distance to other connected components of B and is given by Equation 2.16.

$$iz_A(B_i) = \{a \in A, \forall j \in [1,k] \setminus \{i\}, d_A(a,B_i) < d_A(a,B_j)\}. \tag{2.16}$$

The set of pixels that do not belong to any influence zone make up the skeleton by influence of B in A, denoted by $SKIZ_A(B)$ (Equation 2.17).

$$SKIZ_A(B) = A \setminus IZ_A(B). \tag{2.17}$$

where $IZ_A(B) = \bigcup_{i\in[1,k]} iz_A(B_i)$ indicates the scale of the Gaussian.

In general, the algorithm on a digital image A, which is seen as a surface image f (at each pixel p provides a height h, i.e., $f(p) = h$), can be defined in a iterative way in agreement with procedure steps given in the following (Digabel and Lantuéjoul, 1978) ():

1. $X_{h_{min}} = f^{h_{min}}$,
2. $X_{h+1} = Reg_Min_{h+1}(f) \cup IZ_{f^{h+1}}(X_h), \forall \in [h_{min}, h_{max} - 1]$.

where $h_{min} = \min f$ and $h_{max} = \max f$,
$f^h = \{p \in A \mid f(p) \leq h\}$ is the upper threshold of f at level (height) h,
$Reg_Min_h(f)$ is the set of the regional mimima of f at height h.

The watershed of f is the complementary of $X_{h_{max}}$. A more formal specification can be found in (Vincent and Soille, 1991).

2.3 Preserved topological and geometric properties by contour detection and hierarchical image segmentation

Arbelaez et al. (2011) introduced a unified approach to contour detection and image segmentation. Because such an approach preserves topological and geometric properties, it was selected to be used in this research. The main characteristics of this approach are reviewed in this section from Equations introduced in Sections2.2.1 to 2.2.5.

To produce high-quality image segmentations, the contour detector is linked with a generic grouping algorithm consisting of two steps. In the first step, a new image transformation called the Oriented Watershed Transform (OWT) is introduced for building a hierarchical segmentation by exploiting the information in the contour image (Arbelaez et al., 2009). In the second, using an agglomerative clustering procedure, an initial graph is built where the nodes are the initial regions, the links are the initial arcs separating the adjacent regions, and the weights are a measure of dissimilarity between the regions. The algorithm proceeds by sorting the links by similarity and iteratively merging the most similar regions. The process produces a tree of regions where leaves are the initial regions, the root is the entire image,

and the inclusion relation orders the regions in a multiscale fashion. Figure 2.22 displays the complete process scheme used for edge detection, and Figure 2.23 presents the complete process scheme used for ultrametric contour map (UCM) production.

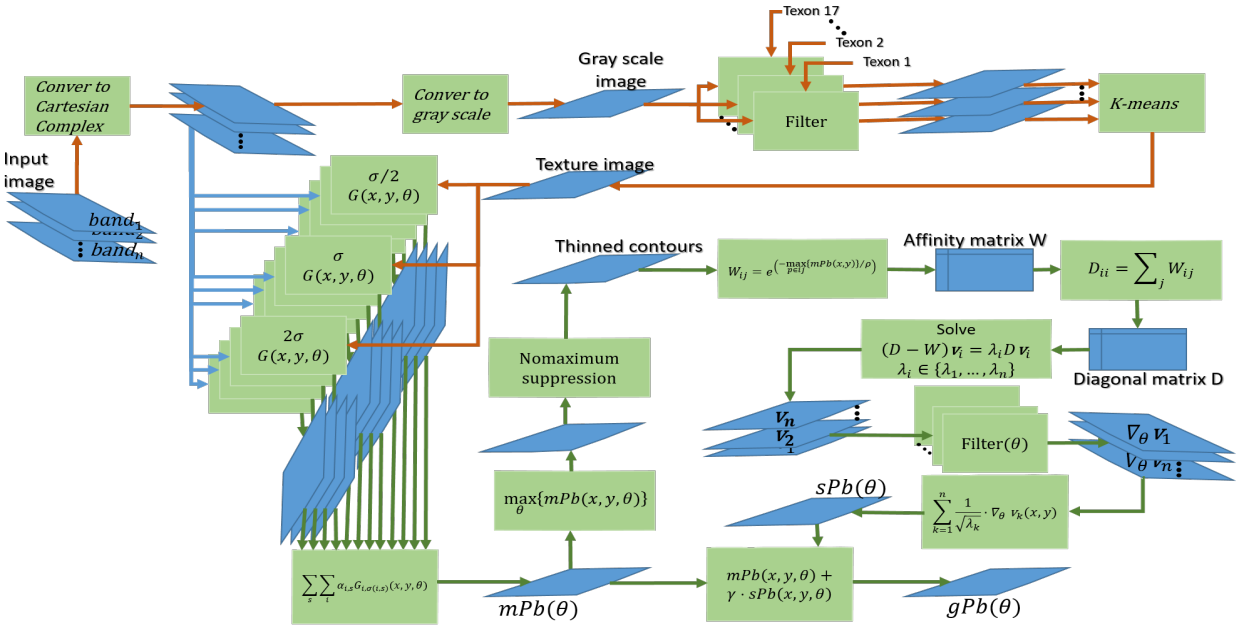


FIGURE 2.22: Strength boundary-based edge detection pipeline

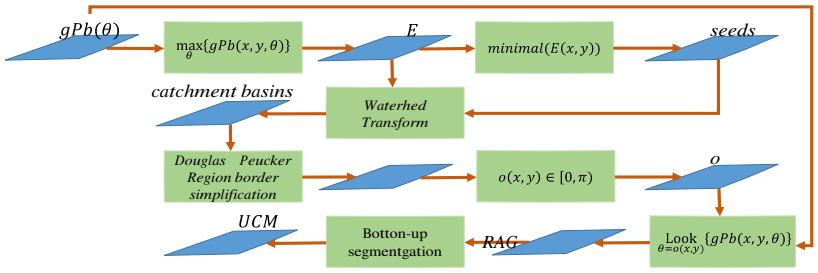


FIGURE 2.23: UCM production pipeline

The function $Pb(x,y,\theta)$ predicts the posterior probability of a boundary with orientation θ at each image pixel (x,y) by measuring the difference in the local image brightness, color, and texture channels. It introduces a multiscale version of the *Pb* detector and a new globalization method on top of this multiscale local detector. The oriented gradient image $G(x,y,\theta)$ is computed from an intensity image I for each pixel placing a disk centered in x,y and split by an θ oriented diameter in two half-discs. For each half-disc, a histogram of the intensity values of the I's pixels covered by it is built. The gradient magnitude G at location (x,y) is defined by the χ^2 distance (Equation 2.11) between the two half-disc histograms g and h.

As the original image is transformed into four separate feature channels and the processing of each channel proceeds independently, the distance is calculated for each channel. Then, it fits a cylindrical parabola whose axis is orientated along direction θ, to a local 2D window surrounding each pixel and replacing the response at the pixel with that estimated by the fit using a second-order filtering (Savitzky, 1964). A strong oriented gradient response means that a pixel is likely to lie on the boundary between two distinct regions. The *Pb* detector combines the oriented gradient images obtained from each channel, the first three correspond to the channels of brightness, color a, and color b channels.

The fourth is a texture channel, which assigns each pixel a *texton* id. These assignments are computed by another filtering stage that converts the input image to grayscale and convolves it with the set of 17 Gaussian derivatives and a center surround filter. Each pixel is associated with a (17-dimensional) vector of responses, containing one entry for each filter. These vectors are then clustered using K-means. The cluster centers define a set of image-specific *textons* and each pixel is assigned the integer id in $[1,64]$ of the closest cluster center. For each of the channels, three gradients scales $\left[\frac{\sigma}{2},\sigma,2\sigma\right]$ are considered to detect the fine and coarse structures, where σ is a given number of pixels. The three local gradients are lineally combined into a single multiscale oriented image (Equation 2.18).

$$mPb\left(x,y,\theta\right)=\sum_{s}\sum_{i}\alpha_{i,s}G_{i,\sigma(i,s)}(x,y,\theta)\ , \tag{2.18}$$

where s indexes the scales, i indexes the feature channels, and $G_{i,\sigma(i,s)}(x,y,\theta)$ measures the histogram difference in channel i between two halves of a disc of radius $\sigma\left(i,s\right)$ centered at (x,y) and divided by a diameter at angle θ. The $\alpha_{i,s}$ parameter is used for weighting the relative contribution of each gradient image. The maximal *mPb* image is produced by Equation 2.19. An optional non-maximum suppression step (Canny, 1986) produces thinned, real-valued contours.

$$mPb\left(x,y\right)=\max_{\theta}\left\{mPb\left(x,y,\theta\right)\right\}\ . \tag{2.19}$$

Using the maximal value of *mPb* along a line connecting two pixels (the *intervening contour* cue), a sparse symmetric affinity matrix W is constructed as input to the spectral clustering stage. The affinity (Equation 2.20) for each pair of pixels i,j within a fixed radius r is

$$W_{ij}=e^{\left(-\max_{p\in\overline{ij}}\{mPb(p)\}\ /\rho\right)}\ , \tag{2.20}$$

where $\overline{ij}$ is the line connecting i and j and ρ is a constant. Global information is introduced defining $D_{ii}=\sum_{j}W_{ij}$ and solved for the generalized eigenvectors $\{\mathrm{v}_0,\mathrm{v}_1,\ldots,\mathrm{v}_n\}$ of the system $(D-W)\,\mathrm{v}=\lambda D\mathrm{v}$ (Equation 2.14), which corresponds to the $n+1$ smallest eigenvalues $0=\lambda_0\leq\lambda_1\leq\cdots\leq\lambda_n$. Each vector v_k is convolved with Gaussian directional derivative filters at multiple orientations θ obtaining $\{\nabla_\theta \mathrm{v}_k(x,y)\}$ to ignore the smooth variation of large regions on the image. Equation 2.21 provides the boundary detector spectral component.

$$sPb\left(x,y,\theta\right)=\sum_{k=1}^{n}\frac{1}{\sqrt{\lambda_k}}\cdot\nabla_\theta \mathrm{v}_k(x,y)\ . \tag{2.21}$$

The weighting by $1/\sqrt{\lambda_k}$ is motivated by the physical interpretation of the generalized eigenvalue problem as a mass-spring system (Belongie and Malik, 1998). The images *mPb* and *sPb*, that represent the edges and the most salient curves in the image, respectively, are lineally combined to obtain the final globalized probability of boundary (Equation 2.22).

$$gPb(x,y,\theta) = mPb(x,y,\theta) + \gamma \cdot sPb(x,y,\theta) \ . \tag{2.22}$$

As the outcome can offer non-closed contours, OWT is applied to produce a set of initial regions whose boundaries are the base for building a UCM (Arbelaez, 2006). Using $gPb(x,y,\theta)$, the finest partition for the hierarchy, an oversegmentation whose regions determine the highest level of detail considered, is created from Equation 2.23.

$$gPb(x,y) = \max_{\theta} \{gPb(x,y,\theta)\} \ , \tag{2.23}$$

whose minimal values are used as seeds for the OWT stage, which first applies the watershed transform (Beucher and Meyer, 1992; Najman and Schmitt, 1996) on the topographic surface defined by $gPb(x,y)$. An enforcing consistency between the watershed arcs K_0 (the possible boundary locations) and the underlying $gPb(x,y,\theta)$ image is applied. This reweighting process estimates an orientation at each pixel on an arc from the local geometry of the arc itself. Approximating the watershed arcs with line segments connecting the ends of watershed arcs applying a technique based on the Douglas-Peucker algorithm (Visvalingam and Whyatt, 1990) and using as approximation criterion the maximum distance of a point on the arc to the line segment as a fraction of the line segment length. Each pixel (x,y) on a subdivided arc takes its orientation $o(x,y) \in [0,\pi)$ from the corresponding line segment, producing $E(x,y,o(x,y))$. Each arc pixel (x,y) obtains a boundary strength simply by looking for $E(x,y,o(x,y))$ in $gPb(x,y,\theta)$.

A UCM defines a duality between closed, non-self-intersecting weighted contours and a hierarchy of regions. The contours in the oversegmented base level define the weakest contour strength, whereas the upper levels impose progressively stronger contour strength. Each level i has a RAG $G_i = (\mathrm{P}_i, \mathrm{K}_i, W(\mathrm{K}_i))$, where the nodes are regions P_i, the links are arcs K_i that separate adjacent regions, and weights $W(\mathrm{K}_i)$ are a measure of dissimilarity between the regions. The algorithm proceeds by sorting the links by similarity and iteratively merging the most similar regions:

1. $i \leftarrow 1$.
2. $\mathrm{P}_i \leftarrow \mathrm{P}_{i-1}$, $\mathrm{K}_i \leftarrow \mathrm{K}_{i-1}$, $W(\mathrm{K}_i) \leftarrow W(\mathrm{K}_{i-1})$.
3. $C^* = \mathrm{argmin}_{C \in \mathrm{K}_i} W(C)$.
4. $R = R_1 merge R_2$, where R_1 and R_2 are the regions separated by C^*.
5. $\mathrm{P}_i \leftarrow \mathrm{P}_i \setminus \{R_1, R_2\} \cup \{R\}$, $\mathrm{K}_i \leftarrow \mathrm{K}_i \setminus \{C^*\}$.
6. For each $C_1, C_2 \in \mathrm{K}_i$ and $R_s \in \mathrm{P}_i$, if C_1 separates R_1 from R_s and C_2 separates R_2 from R_s:
   (a) $K = C_1 merge C_2$.
   (b) $W(K) = (W(C_1) + W(C_2))/2$.
   (c) $\mathrm{K}_i \leftarrow \mathrm{K}_i \cup \{K\}$, where K separates R from R_s.

(d) $W(\mathrm{K}_i) \leftarrow W(\mathrm{K}_{i-1}) \setminus \{W(C_1), W(C_2)\} \cup \{W(K)\}$.

7. If $\mathrm{K}_i \neq \emptyset$, $i \leftarrow i+1$ and repeat from 2.

The mean between R_1's and R_2's radiometric information is that of the R's.

Because the weight of the contour currently being removed cannot decrease during the merging process, the constructed region tree can be described by a dendrogram, where the height $H(R) = W(C)$ is the value of the dissimilarity where R first appears, where C is the contour whose removal formed R. The distance between the two regions given by the height of the smallest containing segment $D(R_1, R_2) = \min\{H(R) : R_1, R_2 \subseteq R\}$, satisfies the ultrametric property $D(R_1, R_2) = \max(D(R_1, R), D(R, R_2))$. Therefore, the entire hierarchy can be represented as a UCM (Arbelaez, 2006), the real-valued image obtained by weighing each boundary by its scale of disappearance.

2.4 Oriented matroids as geometric feature encoding structures

The requirement to reduce the complexity of the geometric spaces algorithms has raised an alternative representation based on mathematical artifacts called matroids (Whitney, 1935), which seek to generalize, in more abstract conceptual elements, the independent set concept present in matrix arrays and graph theory. Matroids include a structure of convexity, through the definition of an orientation, giving rise to oriented matroids.

The geometric structure can be encoded in associated oriented matroids through the creation of hyperplane or hypersphere arrangements (Fukuda, 2004). The former conforms to a structure of homogeneous faces that are associated with a position vector relative to the half-spaces that each hyperplane separates the original space. In the latter, a sphere arrangement can be viewed as the cross-section of a central hyperplane arrangement with a unit hypersphere. In this case, the hypersphere surface is divided into sections by the hypersphere arrangement of lower dimension formed by its intersection with the central hyperplane arrangement, such that the sections can be encoded based on their relative position with each separation hypersphere. By Folkman & Lawrence representation theorem, for a hypersphere system there is an oriented matroid and vice versa (Folkman and Lawrence, 1978,p.215).

2.4.1 Matroids

Before providing the definition of an oriented matroid, it is necessary to define the structures associated with any matroid; this can be based on several axiomatizations, all equivalent. Whitney (1935) introduced the concept of matroid in his research on the abstract properties of linear dependence. According to Oxley (2003), matroids were introduced by Whitney to attempt to capture the dependency essence in an abstract manner. Whitney presented several alternatives for defining a matroid in terms of range, independent sets, bases, and circuits. The postulate for independent sets is:

"*Let M be a set of elements. Let any subset N of M be either 'independent' or 'dependent'. Let the two following postulates be satisfied:*

(I_1) *Any subset of an independent set is independent.*

(I_2) *If* $N = e_1 + \ldots + e_p$ and $N' = e'_1 + \ldots + e'_{p+1}$ *are independent, then for some i such that* e'_i *is not in* N, $N + e'_i$ *is independent.*

The resulting system is equivalent to a matroid."

Whereas it is true that this definition is equal to the one given in the introduction to his article, the sense given here to "independent" is not linear independence. To attach the sense of "independent" given here, in a clearer context, it is appropriate to adopt that given in (Pitsoulis, 2014, p.25):

A matroid M is a pair (E, I) where E is a finite set and I a set of subsets of E that satisfies:

(I_1) I is nonempty.

(I_2) Every subset of every member of I is also in I.

(I_3) If X and Y are in I and $|X| > |Y|$, then there is an element x in $X - Y$ such that $Y \cup \{x\}$ is in I.

The set E and the set members of I are the ground set and independent sets, respectively, of the matroid. According to Oxley (2003), Whitney used conditions (I_1) - (I_3) to capture the essence of dependency in an abstract manner. If $E = \emptyset$ (empty), there is exactly one matroid on E with $I = \{\emptyset\}$. If $E = \{\clubsuit\}$, there are exactly two matroids on E, one with $I = \{\emptyset\}$ and another with $I = \{\emptyset, \{\clubsuit\}\}$.If $E = \{\clubsuit, \spadesuit\}$, there are exactly five matroids on E having their sets of independent sets $\{\emptyset\}$, $\{\emptyset, \{\clubsuit\}\}$, $\{\emptyset, \{\spadesuit\}\}$, $\{\emptyset, \{\clubsuit\}, \{\spadesuit\}\}$, and $\{\emptyset, \{\clubsuit\}, \{\spadesuit\}, \{\clubsuit, \spadesuit\}\}$. However, the second $M_2 = (E, \{\emptyset, \{\clubsuit\}\})$ and third $M_3 = (E, \{\emptyset, \{\spadesuit\}\})$ matroids have exactly the same structure and there is a bijection between the two of them such that a set is independent in the first matroid if and only if its image is independent in the second. These matroids are called isomorphic ($M_2 \cong M_3$)**.**

Given a set E of n elements and a set I of the subsets of E with maximum r elements ($0 \leq r \leq n$), the pair (E, I) forms a matroid called the uniform matroid $U_{r,n}$. In this context, a maximum independent set is an independent set that is not a proper subset of any other independent set and a set is minimum dependent when all its proper subsets are independent.

Given a finite set E, a set $\mathscr{C}$ of subsets of E that satisfies (C_1)-(C_3) given below, and a set I of those subsets of E containing no member of $\mathscr{C}$; then, (E, I) is a matroid that has $\mathscr{C}$ as its set of circuits (Oxley, 2003, p.9).

(C_1) The empty set is not in $\mathscr{C}$.

(C_2) No member of $\mathscr{C}$ is a subset of another member of $\mathscr{C}$.

(C_3) If C_1 and C_2 are distinct members of $\mathscr{C}$ and $e \in C_1 \cap C_2$, then $(C_1 \cup C_2) - \{e\}$ contains one member of $\mathscr{C}$.

2.4.2 Matrices and matroids

(Whitney, 1935) proposed the following approach with respect to a given matrix A with its columns $C_1, \ldots, C_n$. Any subset N of these columns makes up a matrix with rank $r(N)$. If the columns are the abstract elements, it conforms to a matroid M with range given by the number of linearly independent columns in N. Whitney gives two different geometric interpretations to this matroid:

(1) Let E_m be the m-dimensional Euclidean space with a X_i point corresponding to each column C_i of A with coordinates $a_{1i}, \ldots, a_{mi}$. The subset $C_{i1}, \ldots, C_{ip}$ of M is linearly independent if and only if the points $0(0,...,0)$, $X_{i1}, \ldots, X_{ip}$ are linearly independent at E_m; that is, if and only if these $p+1$ points determine a hyperplane in E_m dimension p.

(2) Let E_n be the n-dimensional Euclidean space. Let $R_1, \ldots, R_m$ be the rows of M. If $Y_1, \ldots, Y_m$ are the corresponding points of $E_n: Y_i = (a_{i1}, \ldots, a_{in})$, then the points $0, Y_1, \ldots, Y_m$ determine a hyperplane $H = H(M)$, called the hyperplane associated with M.

From a matrix A over any field F, it is possible to define a matroid. In particular, the interest from a computational point of view is in finite fields for which Galois $GF(p^k)$ fields provide exactly p^k elements when p is a prime number. With $k = 1$, the respective field $GF(p)$ can be seen as the set $\{0,1,2,...,p-1\}$ with the operations of addition and multiplication module p. The following theorem appears in (Oxley, 2003, p.4):

"*Let A be a matrix over a field F. Let E be the set of column labels of A and $\Im$ be the collection of subsets I of E for which the multiset of columns labelled by I is linearly independent over F. Then $(E, \Im)$ is a matroid.*"

The matroid thus obtained is denoted by $M[A]$ and is called the vector matroid of A. A matroid M being isomorphic to $M[A]$ for some matrix A over a field F is called F-representable and A is an F-representation of M. The matroids that are representable over $GF(2)$ and $GF(3)$ are called binary and ternary, respectively. An example of this is the matroid $U_{2,4}$, which is ternary (Oxley, 2003, Proposition 2.6, p.4).

2.4.3 Graphic matroids

If the set of arcs of a graph G is the finite set E and I, the set of subsets of E, does not contain all the edges of the simple closed path or cycle of G; then, the pair (E, I) is a matroid called the *cycle matroid* of the graph G and is denoted by $M(G)$. A matroid that is isomorphic to the cycle matroid of some graph is called *graphic* (Pitsoulis, 2014, Definition 3.2, p.26). Non-isomorphic graphs can have isomorphic cycle matroids (see Figure 2.24). The matroid cycle of the graph displayed in Figure 2.25 has $E = \{1,2,3,4,5,6\}$ and its set of independent sets equal to the I set corresponding to $U_{2,4}$, except $\{\{3,4\}\}$.

In (Oxley, 2003) it is also demonstrated that every graphic matroid is binary using the vertex-edge matrix of a graph (see Figure 2.25). For a graph G and its matrix A_G over $GF(2)$, its vector matroid $M[A_G]$ has as its independent sets all subsets of $E(G)$ that do not contain all the edges of a cycle, and that therefore $M[A_G] = M(G)$ and all graphic matroids are binary.

For a matroid M on a set E and the set $\mathscr{B}$ of the bases of M, there is a set $\mathscr{B}^* = \{E - B | B \in \mathscr{B}\}$ that corresponds to the set of bases of a matroid M^* on E. The matroid M^* is called the *dual matroid* of M. The bases and circuits of M^* are called the cobases and cocircuits of M (Delucchi, 2003, Definition 1.1.16, p.10). The circuits $\{5\}$, $\{6\}$, $\{3,4\}$, $\{1,2,3\}$, and $\{1,2,4\}$ can be seen in the graph of Figure 2.25; in Figure 2.26, its dual with the circuits $\{1,2\}$, $\{1.3, 4\}$,*and* $\{2,3, 4\}$, and clearly $(M^*)^* = M$. Unlike graphs, where only plane graphs have a dual graph, every matroid has a dual.

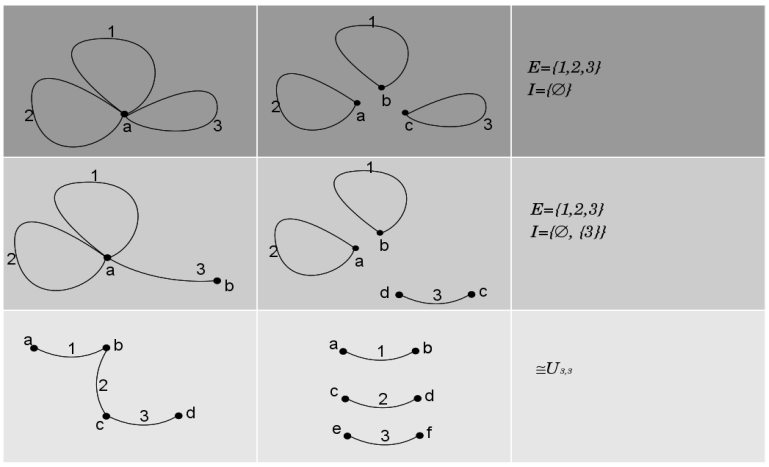


FIGURE 2.24: Non-isomorphic graphs with isomorphic cycle matroid

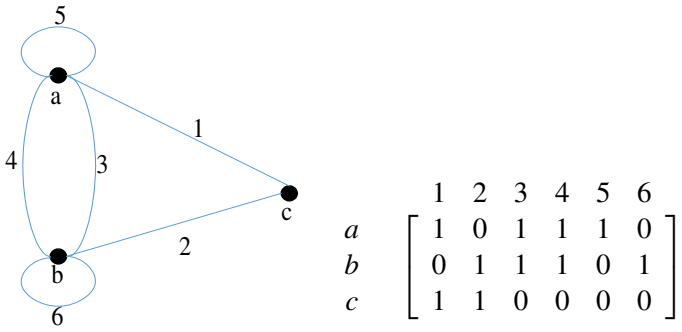


FIGURE 2.25: Graph with its vertex-edge incidence matrix

For the graph displayed in Figure 2.25, the cycle matroid base set is $\mathscr{B} = \{\{1,2\},\{1,3\},\{1,4\},\{2,3\},\{2,4\}\}$; hence, $\mathscr{B}^* = \{\{3,4,5,6\},\ \{2,4,5,6\},\ \{2,3,5,6\},\ \{1,4,5,6\},\ \{1,3,5,6\}\}$, which corresponds to the base set of the dual graph cycle matroid (Figure 2.26).

In addition to the dual of a matroid, in (Oxley, 2003, p.15) appear two other operations (deletion and contraction), which generalize their similar in graph theory. If M is a matroid $(E,\mathfrak{I})$ and e an element of E, then the pair $(E-\{e\},\mathfrak{I}')$ is a matroid with $\mathfrak{I}' = \{I \subseteq E-\{e\} | I \in \mathfrak{I}\}$ denoted by $M\backslash e$ and called the matroid of *deletion* of e from E. The pair $(E-\{e\},\mathfrak{I}'')$ is a matroid with $\mathfrak{I}'' = \{I \subseteq E-\{e\} | I \cup \{e\} \in \mathfrak{I}\}$ denoted by M/e and called the matroid of *contraction* of e in E. A *minor* of M is a matroid obtained from a sequence of deletions and contractions $M\backslash X/Y$ where X and Y are disjunct subsets of E. If $X \cup Y \neq \emptyset$, then $M\backslash X/Y$ is a *proper minor* of M. The three operations (duality, deletion, and contraction) are linked by expressions $M^*/T = (M\backslash T)^*$ and $M^*\backslash T = (M/T)^*$.

2.4.4 Affine matroids

In (Stell and Webster, 2007, p.15) it is introduced the concept of *affine matroid* based on the affine independence of the affine set concept (in $\mathbb{R}^2$, the empty set, the points themselves, straight lines, and the

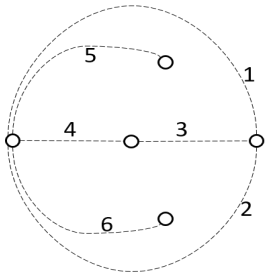


FIGURE 2.26: Dual graph of graph displayed in Figure 2.25

plane itself are affine independent). The *affine hull* of a set is the smallest affine set that contains it. A set is *affinely independent* if and only if its affine hull is not the affine hull of any of its proper subsets.

In fact, a set $x_1,...,x_k$ is affine dependent in $V(=F^n)$ if and only if $(x_1,1),...,(x_k,1)$ is linearly dependent on the space of vectors F^{n+1} (Smyth and Webster, 2007, p.764). Then, given a set of points E in the plane and a set $\mathfrak{I}$ of independent affine subsets of E, the pair $(E,\mathfrak{I})$ so defined, forms a matroid M called the *affine matroid*. For example, for a set P of n $(n \geq 3)$ points of the plane that has no subset of three collinear points, its affine matroid is isomorphic to the uniform matroid $U_{3,n}$. For the set of points displayed in Figure 2.27, the set of independent sets are all subsets of three or fewer elements, except $\{\{b,d,h\}\ ,\ \{a,d,e\}\}$.

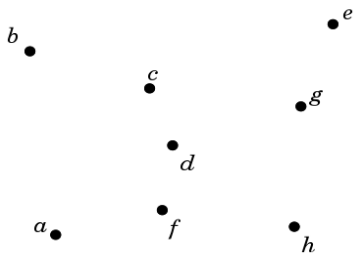


FIGURE 2.27: Point set in plane isomorphic to uniform matroid $U_{3,8}$, except $\{\{b,d,h\},\{a,d,e\}\}$

2.4.5 Oriented matroids

The geometry of matroids is based on what affine or linear sets provide; however, it does not have any structure of convexity. There is neither a notion of betweenness or duality between the points of a straight line, nor the hyperplanes separating the space into two half-spaces. Vector spaces do not have sufficient structure to support a theory of convexity; therefore, it is difficult to expect matroids to do so. Thus, it is necessary to provide an additional framework to matroids (orientation) from which the convexity can arise; this is the theory of oriented matroids (Smyth and Webster, 2007, p.768).

The oriented matroid of a finite set of points $\mathscr{P}$ extracts information for the relative position and orientation from the configuration, which can be provided by a list of signs encoding the orientation of all its bases. Although in the passage of a specific point's configuration to its oriented matroid metric, information is lost, many of the structural properties of $\mathscr{P}$ have their counterpart in the purely combinatorial

level of the oriented matroid (Richter-Gebert and Ziegler, 2004, p.129). That is, the oriented matroids not only describe the structure of incidence between the elements of the matroid and their respective hyperplanes but also encode the position of the elements relative to the hyperplanes; i.e., what elements fall into the positive side, what elements fall into the negative side, and what elements are in the hyperplane (Richter-Gebert and Ziegler, 2004, p.132).

There are two conceptions of a matroid, flat and spherical (Smyth and Webster, 2007, p.768). Each oriented matroid can be represented by an array of topological spheres, which is a generalization of the arrangements of the spheres. Therefore, sphere arrangements are a natural structure to address the study of oriented matroids (Fukuda, 2004, p.0-9). Following are excerpts regarding the concepts of central hyperplane arrangements and sphere arrangements taken from (Fukuda, 2004), where a complete overview of the underlying concepts can be found.

Central hyperplane arrangement and sphere arrangement

A central hyperplane arrangement is an array of hyperplanes where all the hyperplanes contain the origin ($\bigcap_{i=1}^{m} h_i = \{0\}$). Central arrangements have a homogeneous face structure, which makes them considerably easier to attempt mathematically. In effect, the position vector of a central hyperplane arrangement $\mathscr{F}(A) = \{\delta(x)|x \in \mathbb{R}^d\}$ is closely related to a subspace of vectors. If $(A,\ 0)$ is the representation of a central arrangement A, and V is the space of columns of A, $V = \{y \in \mathbb{R}^m | y = Ax,\ x \in\ \mathbb{R}^d\ \}$. If $\sigma(V)$ is the sign vectors set of vectors in V, $\sigma(y)_j = sign(y_j)$ indicates that $\delta(x) = \sigma(Ax)$ for all $x\ \in\ \mathbb{R}^d$, such that $\mathscr{F}(A) = \sigma(V)$.

That is, the combinatorial structure of a central arrangement corresponds to the signature patterns of the vectors in a vector subspace and vice versa. In Figure 2.28, the respective central hyperplane arrangement with its open half-spaces and sign vectors for three particular points of matrix A are indicated.

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ -1 & 1 \end{bmatrix}$$

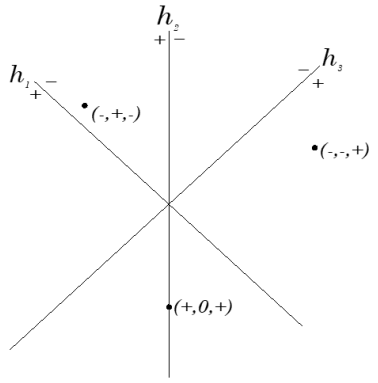


FIGURE 2.28: Central hyperplane arrangement

The sphere arrangement on the unitary sphere $\mathcal{S}^d$ of dimension d (d-sphere) in $\mathbb{R}^{d+1}$ is a set $\mathcal{A} = \{s_i | i = 1,2,...,m\}$ of spheres of dimension $d-1$ on $\mathcal{S}^d$. In fact, a sphere arrangement can be viewed as the cross-section of a central hyperplane arrangement in $\mathbb{R}^{d+1}$ with the unit sphere $\mathcal{S}^d$. For a central arrangement with representation $(A,\ 0)$, each sphere is the set $s_i = \{x|A_i x = 0\} \cap \mathcal{S}^d$.

A subset F of $\mathcal{S}^d$ is a face if it is the intersection of $\mathcal{S}^d$ and some corresponding central hyperplane arrangement face. Therefore, as a combinatorial object, a sphere arrangement is equivalent to some hyperplane's central arrangement. There are two reasons to prefer sphere arrangements rather than central hyperplane arrangements (Fukuda, 2004). The first is purely a visual aspect, as in $\mathbb{R}^3$, it is easier to visualize a sphere arrangement than a central hyperplane arrangement. The second reason is that all oriented matroids can be represented by a topological sphere arrangement, which is a generalization of sphere arrangements. Figure 2.29 displays the sphere arrangement (the spheres are the pairs of points enclosed by the dashed ellipses included in the figure to provide clarity) corresponding to the central hyperplane arrangement displayed in Figure 2.28.

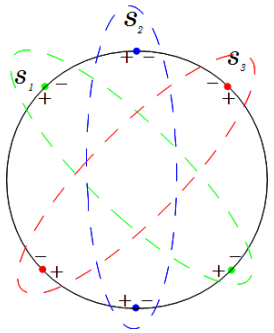


FIGURE 2.29: Sphere arrangement in 2-dimensional space (in $\mathbb{R}^2$)

A sphere arrangement is combinatorically equivalent to a convex polytope. However, because the partially ordered set (POSET) $\mathcal{F}(\mathcal{A})$ of the arrangement does not contain the largest element, it is necessary to add it artificially including element "1"; the new set $\widehat{\mathcal{F}}(\mathcal{A})$ corresponds to the face lattice of arrangement $\mathcal{A}$. Given the polytope $P_A = \left\{x \mid y^T Ax \leq 1 \; \forall y \in \{-1,+1\}^m\right\}$, the face lattice $\widehat{\mathcal{F}}(\mathcal{A})$ is isomorphic to the face lattice of the polytope P_A. Figure 2.30(a) displays the sphere system for the sphere arrangement of Figure 2.29 and the corresponding polytope P_A (Figure 2.30(b)) retrieved from the complete solution for the respective inequality system also displayed in the figure (only four of the eight inequalities are displayed; the other four are obtained by multiplying the left part of each displayed inequality by -1). The polar of polytope P_A is given by

$$\begin{aligned}(P_A)^* &= conv\left\{y^T A \in \mathbb{R}^d \mid y \in \{-1,+\}^m\right\} \\ &= \left\{y^T A \in \mathbb{R}^d \mid y \in \{-1,+\}^m\right\} \\ &= L_1 + L_2 + \cdots + L_m \text{ , where } L_i \text{ is the line segment } [-A_i, A_i]\end{aligned}$$

This set corresponds to the Minkowski sum of a finite line segment set known as a zonotope. The sign vector $\sigma(y)$ of a vector y in $\{-1,+1\}^{\mathrm{m}}$ is in $\mathcal{F}(\mathcal{A})$ if and only if the vector yA is a vertex of the zonotope $(PA)^*$. Figure 2.31 displays the face correspondence between the arrangement of Figure 2.30(a) and their corresponding zonotope.

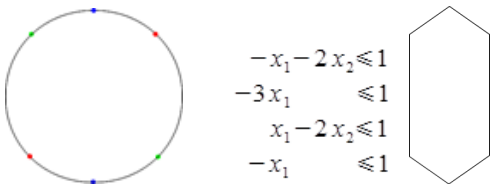


FIGURE 2.30: (a) System of spheres and (b) their respective realization as a polytope $\boldsymbol{P_A}$

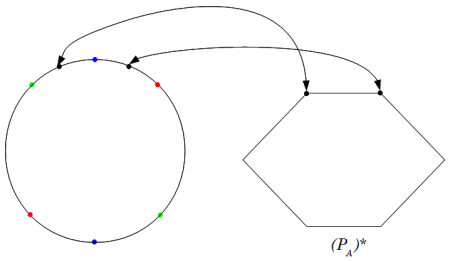


FIGURE 2.31: Face correspondence between sphere arrangement and their zonotopes (In $\mathbb{R}^2$)

What is an oriented matroid?

The oriented matroid concept is introduced based on the notions developed in the previous section (Fukuda, 2004, p.1-1). Given a finite set E, a *sign vector* on E is a vector in $\{+,0,-\}^E$. The negative $-Y$ of a sign vector is the sign vector Z on E with $Z_j = -Y_j$ for all j. An index $j \in E$ separates two sign vectors Y and Y' on E if $Y_j = -Y'_j \neq 0$. The set of all elements that separate Y from Y' is denoted by $D(Y,Y')$. The composition $Y \circ Y'$ of two sign vectors Y and Y' on E is the sign vector Z on E with

$$Z_j = \begin{cases} Y_j, & Y_j \neq 0 \\ Y'_j, & \text{other wise} \end{cases}$$

An oriented matroid on E is a pair $M = (E, \mathscr{F})$, where E is a finite set and $\mathscr{F}$ is a set of sign vectors on E that satisfies the following axioms:

$(F1)$	$0 \in \mathscr{F}$;	
$(F2)$	$Y \in \mathscr{F} \longrightarrow -Y \in \mathscr{F}$	(simetry);
$(F3)$	$Y,Y' \in \mathscr{F} \Longrightarrow Y \circ Y' \in \mathscr{F}$	(composition); *and*
$(F4)$	$Y,Y' \in \mathscr{F} \wedge f \in D(Y,Y' \Longrightarrow \exists Z \in \mathscr{F})$ such that $Z_f = 0$ y $Z_j = (Y \circ Y')$	(strong elimination)

Each member of $\mathscr{F}$ is an oriented matroid face (sometimes called a co-vector). The Folkman & Lawrence topological representation theorem states that oriented matroids are in bijection to oriented sphere arrangements by what can be considered both as topological such as combinatorial objects.

If $\mathcal{S}^d$ is the unit d-esfera in $\mathbb{R}^{d+1}$ and $s_{(d-1)}$ the $(d-1)$-sphere embedded in $\mathcal{S}^d$, the $(d-1)$-sphere $\left\{x \in \mathscr{S}^d \mid x_1 = 0\right\}$ partitions $\mathcal{S}^d$ into three sets $\left\{s^+_{(d-1)}, s^0_{(d-1)}, s^-_{(d-1)}\right\}$, where $s^0_{(d-1)} = s_{(d-1)}$, $s^+_{(d-1)} = \left\{x \in \mathcal{S}^d \mid x_1 > 0\right\}$, and $s^-_{(d-1)} = \left\{x \in \mathcal{S}^d \mid x_1 < 0\right\}$. A d-esfera is a topological space homomorphic to $\mathcal{S}^d$, and a sphere is a d-esfera for some d. A subset s of a sphere S is an equator of S if there is a homomorphism such that $f\left(s_{(d-1)}\right) = s$. An equator partitions s into three sets s^+, s^0, and s^-, which are the images of $s^+_{(d-1)}, s^0_{(d-1)}$ and $s^-_{(d-1)}$ under f. A sphere system of dimension d is a triplet $(E,\ S,\ A)$, where E is a finite set, S a d-esfera, and A a family $\{\{s^+_i, s^0_i, s^-_i\} | i \in E\}$ that satisfies the following axioms:

(S_1) Each s^0_i is an equator of $\mathcal{S}$, whose sides are s^+_i, s^0_i, s^-_i for each $i \in E$;

(S_2) For each subset F of E, $\bigcap_{i \in F} s^0_i$ is a sphere, called *flat*;

(S_3) For each flat t and any $i \in E$, either $t \subset s^0_i$ or $t \cap s^0_i$ is an equator of t, whose sides are $t \cap s^+_i$ and $t \cap s^-_i$.

For a sphere system $(E,\ S,\ A)$, there is a natural method to define the position vector of each point in S as:

$$\delta(x)_i = \left\{ \begin{array}{l} +\,, x \in s^+_i \\ 0\,, x \in s^0_i \\ -\,, x \in s^-_i \end{array} \right\}, \ i \in E$$

Finally, the theorem states that for an oriented matroid $M = (E, \mathscr{F})$, there is a sphere system $(E,\ S,\ A)$ such that $\delta(S) = \mathscr{F}$, and vice versa, i.e., that if $(E,\ S,\ A)$ is a sphere system, then $(E,\ \delta\ (S))$ is an oriented matroid. This theorem is one of the most important in the theory of oriented matroids. Using this, it is possible to pass sphere arrangements to oriented matroids.

Chapter 3

Image segmentation conceptual model based on axiomatic locally finite spaces and oriented matroids

The research problem addressed in this thesis, which was presented in Section 1.1, required effort to overcome additional challenges to those mentioned in Chapter 2. For this reason, a conceptual model using layers was proposed, which facilitated conceptually isolating subsets of these challenges and progressively achieving their solution addressing one at a time. This model was organized in layers from the more concrete at the bottom to the more abstract at the top, in agreement with the layered arrangement displayed in Figure 3.1.

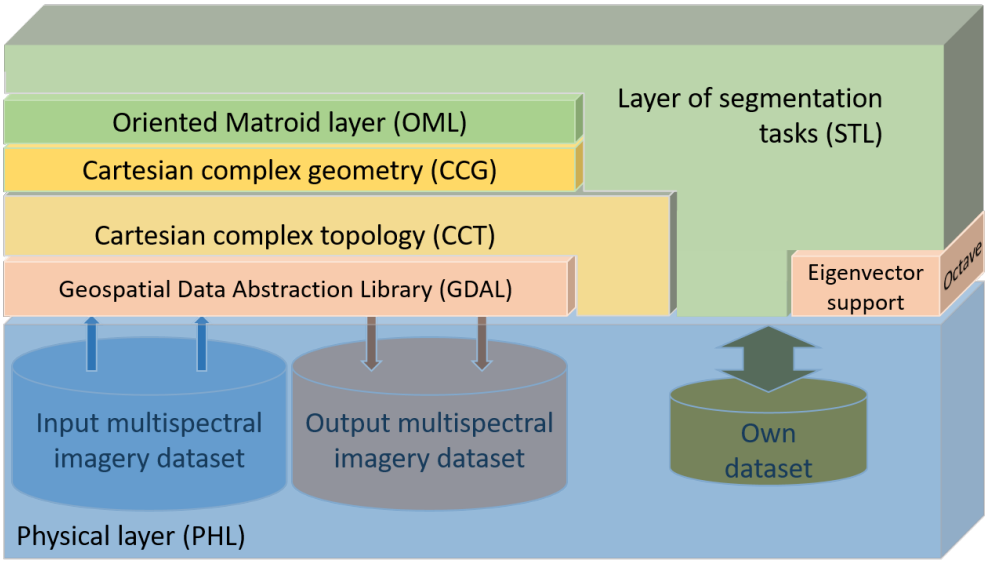


FIGURE 3.1: Multispectral image segmentation conceptual model

The conceptual model developed in this thesis starts at the physical level represented by a multispectral imagery dataset and the dataset created and managed to support the segmentation tasks (the Own dataset in Figure 3.1), and then passes through the open source Geospatial Data Abstraction Library (GDAL) (GDAL Development Team, 2016), a logical geospatial abstraction level that allows manipulation of the

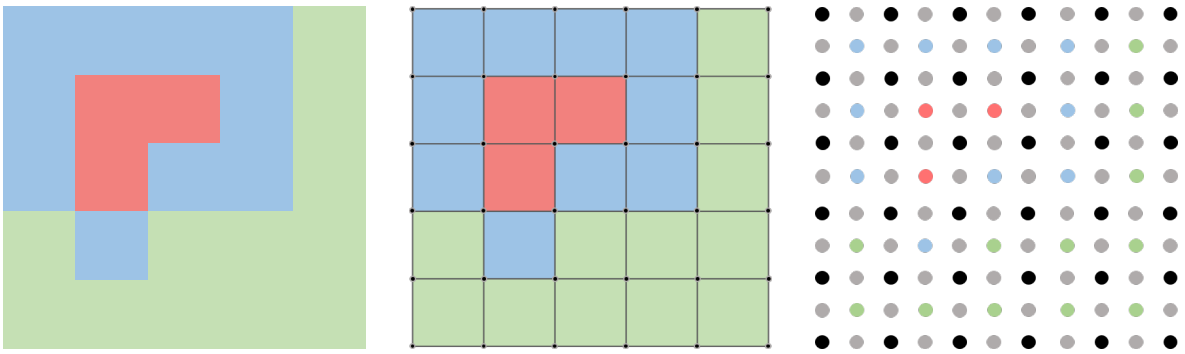
underlying physical level where the input multispectral images and intermediate and output Cartesian complexes are maintained. This abstraction level supports the logical level of the Cartesian complex topology (CCT), which provides an axiomatic locally finite space of complex cells with the required topological elements described in Sections 2.1.1- 2.1.10. This topology level supports the logical level of the Cartesian complex geometry (CCG), which provides the required geometric elements described in Sections 2.1.11– 2.1.13. Next, there is a layer of oriented matroids, where are located the conceptual elements in terms of matroids, that allow encoding topological and geometric properties relevant to the segmentation of the multispectral images, described in Section 2.4. Finally, there is a segmentation task layer (STL) gathering all the tasks required to accomplish the image segmentation described in Section 2.3. The use of GDAL facilitates transparently accessing the underlying imagery dataset without concern for its format. Software implemented in Octave (Eaton et al., 2017) provided the tools for solving the generalized eigenvector matrices.

The following sections address the proposed conceptual model developed in this thesis in agreement with the layered arrangement displayed in Figure 3.1 from bottom to top. Section 3.1 addresses the conceptual elements that support the theoretical concepts presented in Section 2.1 and the manner in which the underlying GDAL layer was used to provide persistence to the topological space model of the CCT. Section 3.2 describes the Cartesian complex geometrical space layer built on top of the CCT layer. Section 3.3 first addresses the proposed combinatorial approach to central standard separating form (SSF) arrangement from the Cartesian complex geometry layer (Section 3.3.1) and then how central SSF arrangements are used to build the oriented matroids (Section 3.3.2). Finally, Section 3.4 presents the proposed reformulations for the segmentation tasks in terms of 1-cells, central SSF arrangements, and oriented matroids.

3.1 Cartesian complex topology (CCT)

Immediately above the GDAL layer is the CCT, used mainly for supporting the theoretical concepts presented in the Section 2.1. However, the CCT layer also provides the facilities for converting a conventional image to a Cartesian complex. Figure 3.2(a) displays a 5×5 pixel conventional image; Figure 3.2(b) and Figure 3.2(c) display the same image as a 2D Cartesian complex including inter-pixel space elements. A 2-cell in a Cartesian complex corresponds to a pixel of the conventional image; lower dimensional cells correspond to the inter-pixel elements that do not exist in the pixel conventional image. In Figure 3.2(b), a 2-cell is represented as a square area, each 1-cell is represented as a vertical or horizontal dark segment line corresponding to the inter-pixel element between the two respective pixels, and each 0-cell is represented as a black dot corresponding to the inter-pixel element in the center of four respective pixels. The finite space is completed with lower dimensional cells on its boundary. Although the representation of Figure 3.2(b) is intuitive, it suggests a nonexistent "density" in a locally finite space; for this reason, it is preferred that the representation provided in Figure 3.2(c) be used. In this figure, each cell of any dimension is represented as a dot and differentiated by its combinatorial coordinates; colors have been used in the figure, as in Figure 3.2(b), because the combinatorial coordinates are not indicated.

The topological elements described in Sections 2.1.1- 2.1.10 are provided by the CCT layer (through Cartesian complex space finally available). The respective CCT conceptual elements are displayed in



(a) Color image

(b) Image with inter-pixel elements included

(c) Digital representation as abstract cell finite set

FIGURE 3.2: 2D image represented in 2D Cartesian complex

Figure 3.3. The *CartesianComplex* class represents the finite space for a 2-dimensional Cartesian complex where each element is a cell (*Cell* class). Each cell c has a vector $\mathbf{v}_c$ of m values (Equation 3.1).

$$\mathbf{v}_c = \left[v_1, v_2, \ldots, v_m\right]^T \ . \tag{3.1}$$

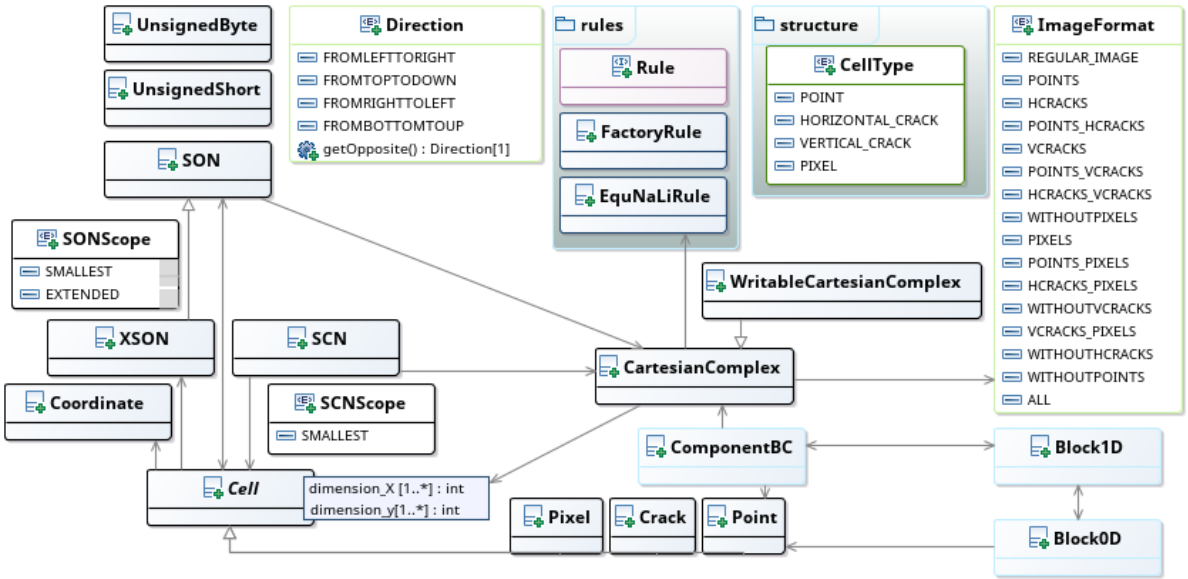


FIGURE 3.3: Conceptual elements in CCT layer

The type τ of a cell c is defined based on its combinatorial coordinate (x_c, y_c). When both of a cell's coordinate (*Coordinate* class) components are even, the cell is a 0-cell (*Point* class); when both of cell's coordinate components are odd, the cell is a 2-cell (*Pixel* class), and when one component is even and the other is odd, the cell is a 1-cell (*Crack* class). In fact, when the x axis component is odd and the

y axis component is even, the respective 1-cell is horizontal; in the opposite case, it is vertical. This is established by Equation 3.2.

$$\tau(c) = \tau(x_c, y_c) = \begin{cases} 0-cell\ , & x_c \text{ is even , } y_c \text{ is even ,} \\ \text{horizontal } 1-cell\ , & x_c \text{ is odd , } y_c \text{ is even ,} \\ \text{vertical } 1-cell\ , & x_c \text{ is even , } y_c \text{ is odd ,} \\ 2-cell\ , & x_c \text{ is odd , } y_c \text{ is odd} \end{cases} \tag{3.2}$$

The employment of Cartesian complexes to represent digital images would entail the use of four times more storage space than conventionally is required because of the additional inter-pixel elements (one 0- and two 1-cells) for each pixel (2-cell) (Kovalevsky, 2008). Nevertheless, it is not always necessary to save the cells from all the dimensions. When a type of cell becomes necessary, the rules for coordinate assignment (*Coordinate* class) and set membership (*EquNaliRule* class) can be suitability applied (Kovalevsky, 1989). Although this *EquNaliRule* rule was the only one used in this research, if required, it is possible to redefine the rule for establishing a cell's value. It is always possible to apply, dynamically, any pre-established rule (using the *FactoryRule* class and create a subclass from the *Rule* class) based on the application specific requirements.

Cells in the space are managed by the respective Cartesian complex (*CartesianComplex* class); hence, it is not possible to modify any of these, except by using this class. If a Cartesian complex is the result of some process (*WritableCartesianComplex* class), it can be saved on disk. To support a type of cell specific to an upper layer (as is the case of *WatershedCrack* on the *WatershedCartesianCompex* in Section 3.4.5), *CartesianComplex* allows any subclass of the three basic classes: *Point*, *Crack*, and *Pixel*. According to the above, when a *CartesianComplex* is created, it must specify the specific subclass required. Thereafter, the basic features available on the created space are provided transparently. A *cell* is obtained from the location given in the combinatorial coordinates; however, if at that location there is no cell, a new cell of the type requested by the subclasses is created there. *CartesianComplex* provides physical support for each cell type as if it were an additional data layer through GDAL in agreement with the possible element-type combinations established (*ImageFormat* enumeration).

The value type of each value in a cell value vector is based on the specific application domain; however, they are restricted to those types provided by the GDAL library, e.g., if the underlying dataset is a multispectral image, the value at a cell would be a vector with the available band value arrangement; however, if the underlying dataset is an segmentation image, the value could be a number corresponding to the respective superpixel membership. Thus, the upper layers can access the values transparently assuming they process their specific types. Conversely, the underlying layer GDAL is not aware of these specific types; hence, any requirement for processing must be developed in the upper layer where it is required. *UnsignedByte* and *UnsignedShort* classes are included for supporting the respective types recognized by the GDAL library and not provided by Java as primitive data types.

The Smallest Open Neighborhood (*SON*) class for a given cell c as defined in Section 2.1.1, along with two more neighborhood scope definitions required by the STL layer, are provided by *SONScope* as *SMALLEST*, *EXTENDED*, and *CONNECTED*. The *SMALLEST* scope neighborhood definition corresponds to the smallest open neighborhood in agreement with Equation 3.3.

$$SON(c) = \{c_i \mid B_2(c, c_i)\}\ . \tag{3.3}$$

where B_2 is defined based on Equation 2.2. The *EXTENDED* scope neighborhood definition corresponds to the next smallest open neighborhood in agreement with Equation 3.4.

$$XSON(c) = \bigcup_{c_i \in Cl(SON(c))} SON(c_i). \tag{3.4}$$

$Cl(t)$ is the t set closure as defined in Section 2.1.2; however, T is the complete Cartesian complex ($Cl(t,T)$). The *CONNECTED* neighborhood scope definition corresponds to SON extended to include the SON of each cell belonging to its closure in agreement with the Equation 3.5.

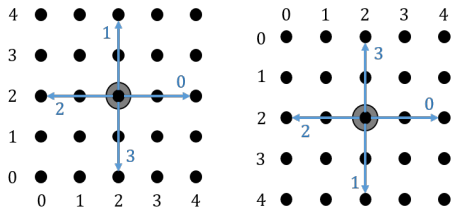
$$ConSON(c) = \bigcup_{c_i \in Cl(c)} SON(c_i). \tag{3.5}$$

$Cl(c)$ is the $\{c\}$ set closure as defined in Section 2.1.2; however, T is the complete Cartesian complex ($Cl(\{c\},T)$). The smallest closed neighborhood (*SCN* class) topological concept is defined in agreement with Equation 3.6.

$$SCN(c) = Cl(c)\ . \tag{3.6}$$

$Cl(c)$ is defined as in Equation 3.5.

For tracing purposes, *Direction* defines the possible movement directions supported. The combinatorial coordinates system can go from bottom to top (*FROMBOTTOMTOUP*) and from left to right (*FROMLEFTTORIGHT*) in a mathematical manner (Figure 3.4(a)), or from top to down (*FROMTOPTODOWN*) and from left to right in a graphical manner (Figure 3.4(b)). The numbers in the figures indicate the valid turn sequence. CCG and STL upper layers assume the graphical approach.



(a) Mathematical (b) Graphical

FIGURE 3.4: Coordinate orientation

3.2 Cartesian complex geometry (CCG)

Standard separating form (*SSF* class) and digital straight segment (*DSS* class) geometrical concepts as defined in Section 2.1.13 are included in the CCG layer. As indicated in Figure 3.5, the CCG layer uses conceptual elements from the CGT layer. Specifically, an SSF, in addition to its a,b and l parameters,

requires a starting 0-cell (*Point*). In the same manner, a *DSS* requires a set of 2-cells (*Pixels*) for both its positive and negative bases, its two forbidden directions (*Direction*), and a supporting SSF.

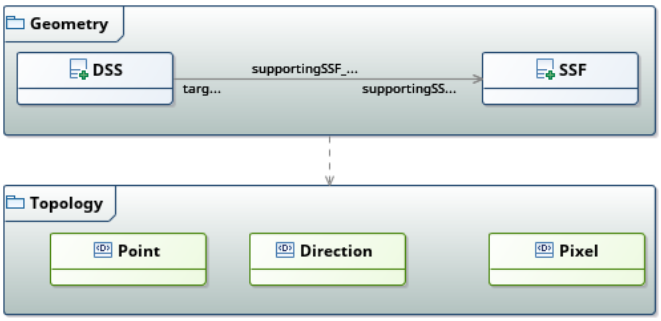


FIGURE 3.5: Conceptual elements in CCG layer

When an SSF is used, the space is subdivided into three subsets: (1) inter=pixel elements on that SSF, (2) space elements on one side of the SSF, and (3) space elements on the other side. These three subsets can be established using the $H(x,y)$ SSF definition given in Section 2.1.13. Equation 3.7 divides a Cartesian complex as indicated in Figure 3.6. In doing so, however, not all the elements of the SFF (indicated as red dots) remain in the same set. In fact, for the Cartesian complex snippet displayed in the figure, only three are included in the subset of those that are on the SFF (red dots included in a marginally larger circle).

$$subset(c) = \begin{cases} \text{on one side}(SSF^{-}), & H(x_c,y_c) < 0 \\ \text{on the SSF}(SSF^{0}), & H(x_c,y_c) = 0 \\ \text{on other side}(SSF^{+}), & H(x_c,y_c) > 0 \end{cases}, \tag{3.7}$$

where c is a Cartesian complex space element.

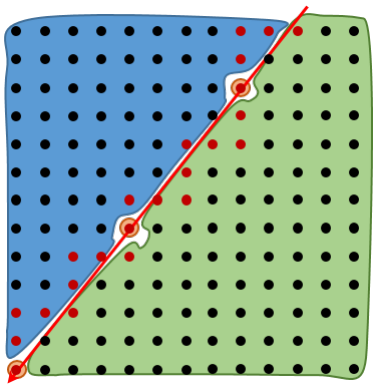


FIGURE 3.6: Subsets produced by conventional SSF definition

Although $H(x,y) = ax + by + l = 0$ for all cells on $H(x,y)$, the H value after shifting is established by Equation 3.8.

$$H(x+k, y+l) = ak + bl \ . \tag{3.8}$$

When Equation 3.7 is used to divide a Cartesian complex, the elements of the SFF (Figure 3.6)do not all remain in the same set. To ensure that all the elements of an SFF remain in the same set, it is necessary to establish a threshold that depends on the cell type. Because this research is 1-cell oriented, in the following, only the threshold for 1-cells is established.

Figures 3.7 and 3.8 display a Cartesian complex snippet including an SSF (clear green dots with a darker green outer circle and linked with green lines) and the original line $H(x,y) = 0$ (the blue line). To illustrate the procedure followed to establish the threshold that a 1-cell must meet to be considered part of an SSF that crosses two 1-cells (dots with outer red circle). To achieve this, it is necessary to establish the threshold based on the SSF a and b parameters. In this example, the b parameter is greater than the a parameter.

As can be observed in Figures 3.7 and 3.8, around each b/a occurs a side change of the SSF with respect to the $H(x,y) = 0$. If the initial 1-cell is vertical, its bounding points are the farthest SSF cells from $H(x,y) = 0$ and are at an exact distance of b/a, and therefore any SSF 1-cell between them and the $H(x,y) = 0$, will be at a distance $< b/a$. If $a = 1$, the farthest point is at a distance b; hence, the furthest 1-cell is less than b. If $a > 1$, b/a can be expressed as the mixed number $g + h/a$ with $1 \leq h \leq a-1$. There are as many as $a-1$ SSF side changes regarding $H(x,y)$, and for each h/a is accumulated to the next until, at the end, adding the h units finally attains $ag + h = b$. The membership threshold is established based on the two extreme cases: (1) when $h = 1/a$, i.e., the situation where the distance between the 1-cell that does not belong to the SSF and $H(x,y) = 0$ is minimized (Figure 3.7) and (2) when $h = (a+1)/2$, i.e., the situation where h accumulates to such magnitude that the point of the SSF of change to the singular direction is the farthest possible from $H(x,y) = 0$, and therefore also the furthest 1-cell (Figure 3.8).

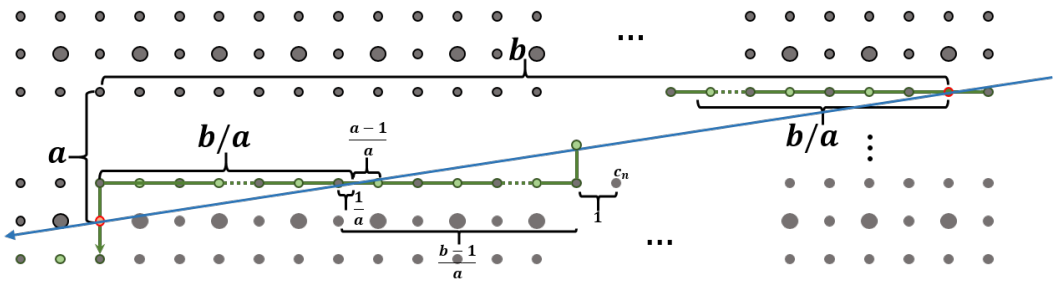


FIGURE 3.7: Subsets produced by SSF definition - nearest non-SSF 1-cell case

For the Cartesian complex snippet in Figure 3.7, $b > a$, which means the singular direction is on the vertical 1-cells. On the non-singular direction (horizontal 1-cells) near of each b/a there should be an SSF change of side with respect to the reference line $H(x,y) = 0$ (the blue line). The nearest 1-cell that is not a member of the SSF occurs when $h = 1$; hence, $g = (b-1)/a$. These values are indicated in the figure with the nearest 1-cell (labelled c_n) at a distance of "1" from the SSF. Therefore, the nearest 1-cell is at a distance of $\frac{b-1}{a} - \frac{1}{a} + 1$, i.e., $\frac{b-2+a}{a}$. By Equation 3.8, $H(c_n) = H(x + \frac{b-2+a}{a}, y) =$

$a(\frac{b-2+a}{a}) = b-2+a$. The least value is obtained when $a = 3$. For the nearest 1-cell, $H(c_n) = b$, hence the value at c 1-cell on the SSF meets Inequality 3.9.

$$-|b| < H(x_c, y_c) < |b| \ . \tag{3.9}$$

For the Cartesian complex snippet in Figure 3.8, although $b > a$, in this case h is sufficiently large to demand that the next SSF side change be prolonged. This occurs when $h = (a+1)/2$, therefore $g = \frac{2b-a-1}{2a}$.

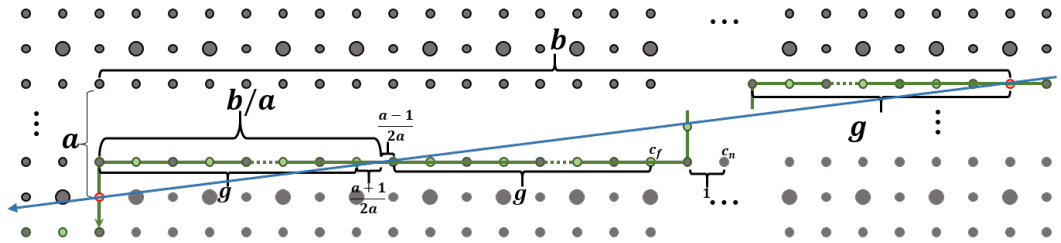


FIGURE 3.8: Subsets produced by SSF definition - farthest on-SSF 1-cell case

The situation illustrated in Figure 3.8 allows the establishment of the H threshold value at the farthest on-SFF 1-cell (labelled c_f). The greatest H value is obtained when from the previous SSF side changes the most is kept to the next SSF side change, i.e., as much as $(a-1)/2a$ and the farthest 1-cell is at distance g from there as indicated in the figure. Hence, the farthest on-SFF 1-cell is at a distance of $g + \frac{a-1}{2a}$, i.e., $\frac{2b-a-1+a-1}{2a} = \frac{b-1}{a}$. By Equation 3.8, $H(c_n) = H(x + \frac{b-1}{a} + 2, y) = a(\frac{b-1}{a} + 2) = b-1+2a$, which confirms Inequality 3.9 because there is no on-SSF 1-cell with an H value greater than b. The same analysis was performed for the SSF vertical 1-cells, the singular direction, allowing the establishment of Inequality 3.10.

$$-|a| < H(x_c, y_c) < |a| \ . \tag{3.10}$$

This is summarized in Equation 3.11, which allows the division of a Cartesian complex into three subsets in the manner indicated in Figures 3.7 and 3.8, where the SSF 1-cells are represented with a green outer circle, the elements on one side with a black outer circle, and the elements on the other side without an outer circle.

$$subset(c) = \begin{cases} \text{on one side}(SSF^-), & H(x_c, y_c) < -|m| \\ \text{on other side}(SSF^+), & H(x_c, y_c) > |m| \\ \text{on the SSF}(SSF^0), & \text{otherwise} \end{cases} , \tag{3.11}$$

where $H(x,y) = ax + by + l$ (Kovalevsky, 2008) and $m = a$ if c is a vertical 1-cell or b if c is a horizontal 1-cell.

Equation 3.11 can be written in terms of value sign of $H(x,y)$ (δ) in agreement with Equation 3.12; hence, $c \in SSF^{\delta(c)}$. This approach is provided to the sibling *OrientedMatroid* class and upper STL layer.

$$\delta(c) = sign(c) = \begin{cases} 0, & -|m| \leq H(x_c, y_c) \leq |m| \\ sign(H(x_c, y_c)), & \text{otherwise} \end{cases} . \tag{3.12}$$

3.3 Oriented matroid layer (OML)

One of the most important and the greatest challenge that required attention during the research was how to link the topological and geometric properties required in the segmentation stage using the associated oriented matroids. To achieve this, the conceptual model displayed in Figure 3.9 was defined. As established in Section 2.4.5, the concept of central SSF arrangement (*SSFCentralArray* class) was determined.

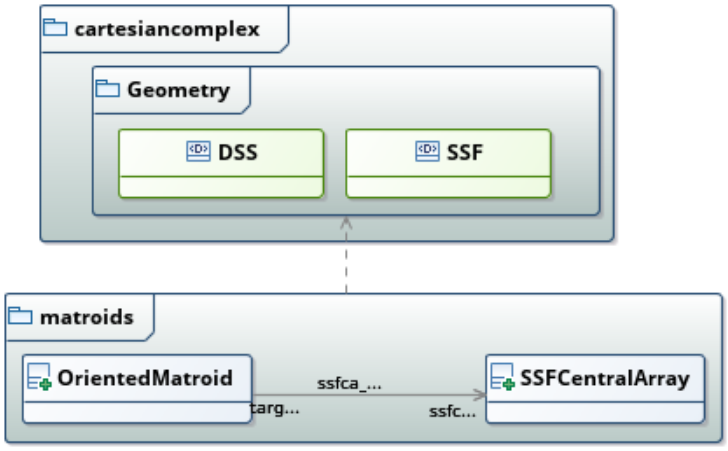


FIGURE 3.9: Conceptual elements in the OML layer

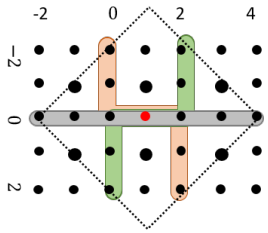
3.3.1 Central SSF arrangement

Because the upper STL layer tasks are 1-cell oriented, a scheme for obtaining SSFs based on that type of space element was established. Based on a number of orientations, this is defined as an central SSF arrangement having a central 1-cell as the common origin. A Cartesian complex can have only a restricted number of SSFs centred at a 1-cell as they require mutual prime numbers as their parameters (Kovalevsky, 2008). To the extent that the number of 1-cells increases, before and after the central 1-cell, similarly does the number of SSFs that can be defined (see Figure 3.10(a) and 3.10(b)); thus, they can be confined to a Manhattan distance-based ball (Worboys and Duckham, 2004). The a, b, and l SSF parameters can be established from a cell's relative combinatorial coordinates regarding the central 1-cell. Because, in the underlying CCT layer, the combinatorial coordinates of a cell determine its type, the relative combinatorial coordinate of a central 1-cell cannot be placed at the conventional $(0,0)$, otherwise this would correspond to a 0-cell. It must be displaced one unit in the horizontal direction if the 1-cell is horizontal

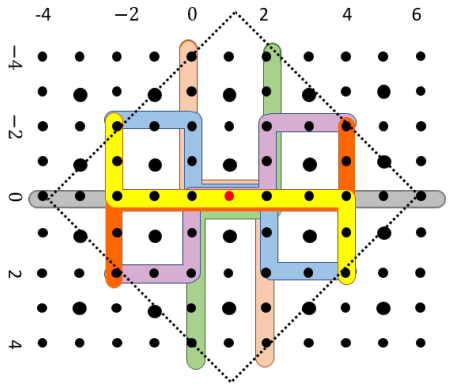
or one unit in the vertical direction if the 1-cell is vertical. In this research, by convention, the central 1-cell c combinatorial coordinate is placed based on Equation 3.13.

$$(x_c, y_c) = \begin{cases} (1,0)\,, & c \text{ is horizontal} \\ (0,1)\,, & c \text{ is vertical} \end{cases}\,, \tag{3.13}$$

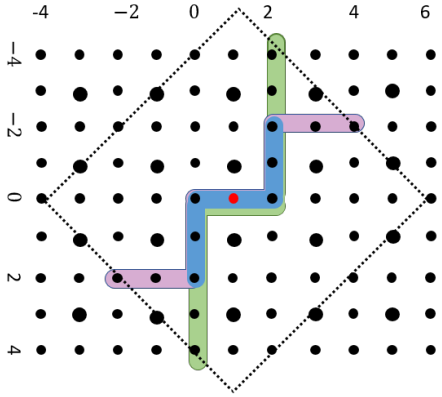
where c is the central 1-cell.



(a) One 1-cell in addition to central 1-cell



(b) Two 1-cells in addition to central 1-cell



(c) Inner SSF contained by two outer

FIGURE 3.10: Number of SSFs in central SSF arrangement

The maximum number of SSFs can be combinatorically calculated based on the number of space elements inside the Manhattan distance-based ball centered at a 1-cell. Equation 3.14 corresponds to the number of orientations based on the number of outer 0-cells contained in a closed Manhattan distance-based ball. However, as can be observed in Figure 3.10(b), up to two SSFs can cross a 0-cell; however, with its combinatorial coordinates, it is possible to establish only one of these.

$$maxSSFs = n(n+1)+1\,, \tag{3.14}$$

where n is the radius in number of additional 1-cells in addition to the central 1-cell.

It is must be considered that an inner SSF can be completely overlapped by outer SSFs. Figure 3.10(c) displays this situation for three SSFs, the blue SSF is completely contained by the other two. Based on the ending 0-cells (points) of the inner SSF, their a and b parameters have the values "2" and "1", respectively, which means that their path inside the ball (diamond with dashed black line) will continue

along the green SSF, i.e., inside the ball boundary. These two SSFs are equal; hence, there is no value to analyzing them separately. From the above, the only different SSFs, within the boundary of a given ball, are those obtained in such a manner that each one crosses both an external 1-cell and the central 1-cell. The outer 1-cell combinatorial coordinate establishes the a and b SSF parameters and the ending 0-cell establishes the l SSF respective parameter. Because each 1-cell is included in at least one valid SSF, each SSF is defined based on an outermost 1-cell to avoid any overlapping SSF. Therefore, the maximum number of SSFs is obtained combinatorically from Equation 3.15.

$$maxSSFs = 2(2(n-1)+1)+1 = 4n-1 \ , \tag{3.15}$$

where n is the radius in number of additional 1-cells in addition to the central 1-cell.

The outer 1-cell set for an n number of additional 1-cells in addition to a central 1-cell is established by Equation 3.16 based on the ball definition (Worboys and Duckham, 2004).

$$B(c,2n) = \{c_i | d(c,c_i) = 2n, \exists SSF_k, H_k(c) = H_k(c_i) = 0\} \ , \tag{3.16}$$

where n is the radius in number of additional 1-cells in addition to the central 1-cell, and $d(c,c_i)$ is the Manhattan distance between the central 1-cell c and the outer 1-cell c_i.

Hence, each SSF_i $H_i(x,y)$ in the family of SSFs is defined based on Equation 3.17. SSF_i $(2 \leq i \leq 2n)$ is the central 1-cell symmetric to SSF_{4n-i+1}.

$$H_i(x,y) = a_i x + b_i x + l_i, 1 \leq i \leq maxSSFs \ . \tag{3.17}$$

The a_i, b_i, and l_i parameters are established from the relative combinatorial coordinates of the c and c_i 1-cells. If the most horizontal SSF is SSF_1, vertical $\triangle y$ and horizontal $\triangle x$ differences depend on i in agreement with Equations 3.18 and 3.19, respectively, in such a manner that $d(c,c_i) = \triangle x_i + \triangle y_i = 2n$.

$$\triangle y_i = i - 1 \ . \tag{3.18}$$

$$\triangle x_i = 2n - \triangle y_i \ . \tag{3.19}$$

For $1 \leq i \leq 2n$, the a_i and b_i parameters are defined based on Equations 3.20 and 3.21, respectively. Dividing by gcd (greatest common divisor) guarantees that a_i and b_i are mutual primes.

$$a_i = - \triangle y_i / gcd(\triangle x_i, \triangle y_i) \ . \tag{3.20}$$

$$b_i = \triangle x_i / gcd(\triangle x_i, \triangle y_i) \ . \tag{3.21}$$

As SSF_i is a central 1-cell symmetric to SSF_{4n-i+1}, for $2n+1 \leq i \leq 4n-1$, the a_i and b_i parameters are defined based on Equations 3.22 and 3.23, respectively.

$$a_{4n-i+1} = a_i \ . \tag{3.22}$$

$$b_{4n-i+1} = -b_i \ . \tag{3.23}$$

Because $H(c) = H(x_c, y_c) = 0$, the l_i parameter is defined based on Equation 3.24 established from Equation 3.13.

$$l_i = \begin{cases} -a_i \ , & 1 \leq i \leq 2n, c \text{ is horizontal} \\ -b_i \ , & 1 \leq i \leq 2n, c \text{ is vertical} \end{cases} \ , \tag{3.24}$$

where c is the central 1-cell.

Applying the previous equations, each SSF in the arrangement displayed in Figure 3.10(b) can be completely differentiated from the others by the family of SSFs presented in Table 3.1.

TABLE 3.1: SSF family for the central SSF arrangement in Figure 3.10(b)

i	a_i	b_i	l_i	x_{p_i}	y_{p_i}	$SSF_i = H_i(x,y)$
1	0	1	0	6	0	$H_1(x,y) = y$
2	-1	3	1	4	2	$H_2(x,y) = -x+3y+1$
3	-1	1	1	4	2	$H_3(x,y) = -x+y+1$
4	-3	1	3	2	4	$H_4(x,y) = -3x+y+3$
5	-3	-1	3	0	4	$H_5(x,y) = -3x-y+3$
6	-1	-1	1	-2	2	$H_6(x,y) = -x-y+1$
7	-1	-3	1	-2	2	$H_7(x,y) = -x-3y+1$

p_i is the SSF_i starting 0-cell.

When the central 1-cell c is horizontal, as in the case of Figure 3.11, to move from the most horizontal to the most vertical outer 1-cells, it is necessary to use Equation 3.25 rather than Equation 3.18.

$$\triangle y_i = 2n - i \ . \tag{3.25}$$

Applying the equations previously presented, each SSF in the arrangement displayed in Figure 3.11 can be completely differentiated from the others by the family of SSFs presented in Table 3.2.

To maintain the central 1-cell symmetry in an SSF family and the geometrical similarity between the horizontal and vertical central SSF arrangements, the horizontal SSF in a horizontal central SSF arrangement (first row in Table 3.1) and the vertical SSF in a vertical central SSF arrangement (fourth row in

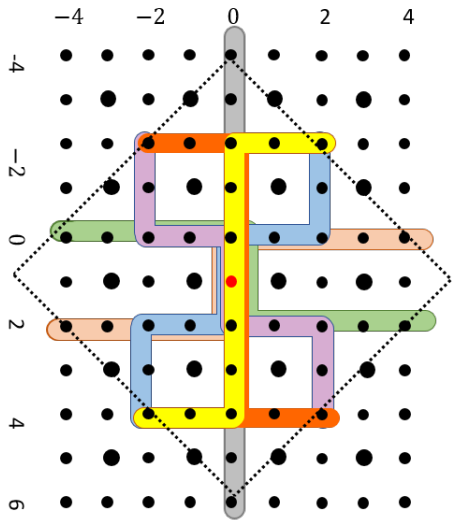


FIGURE 3.11: Vertical central SSF arrangement

TABLE 3.2: SSF family for vertical central SSF arrangement in Figure 3.11

i	a_i	b_i	l_i	x_{p_i}	y_{p_i}	$SSF_i = H_i(x,y)$
1	-1	3	-3	4	2	$H_1(x,y) = -x+3y-3$
2	-1	1	-1	2	4	$H_2(x,y) = -x+y-1$
3	-3	1	-1	2	4	$H_3(x,y) = -3x+y-1$
4	-1	0	0	0	6	$H_4(x,y) = -x$
5	-3	-1	1	-2	4	$H_5(x,y) = -3x-y+1$
6	-1	-1	1	-2	4	$H_6(x,y) = -x-y+1$
7	-1	-3	3	-4	2	$H_7(x,y) = -x-3y+3$

p_i is the SSF_i starting 0-cell.

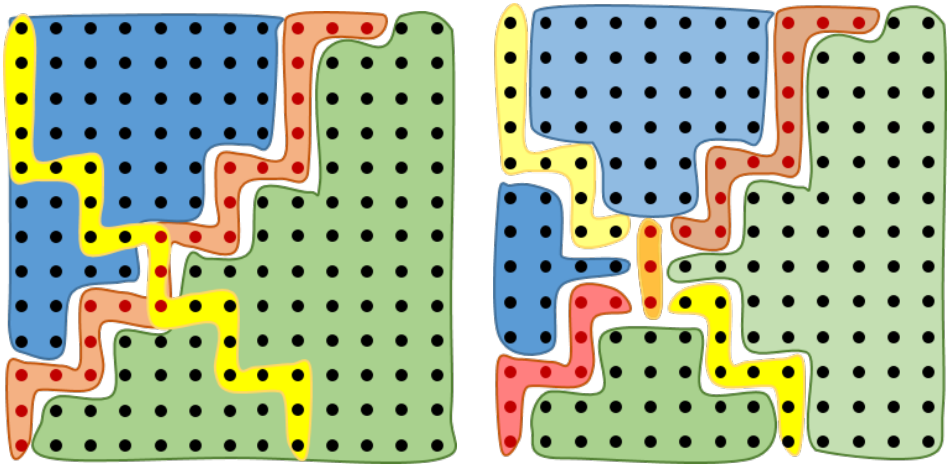
Table 3.2) are not used. Hence, given a number of orientations $1 \leq o \leq 4n-2$, o SFFs are chosen based on the sampling rate established by Equation 3.26.

$$\rho = \frac{4n-2}{o} \,. \tag{3.26}$$

3.3.2 Oriented matroid

The central SSF arrangement geometry is transferred to associated combinatorics through a oriented matroid (*OrientedMatroid* class) using the concept of position vector (De Loera, Rambau, and Santos, 2010). As indicated in Figures 3.7 and 3.8, an SSF divides the containing Cartesian complex into three subsets, which can be established by Equation 3.12. Each time an SSF is added that shares the same central 1-cell, each space that it crosses is divided into at least three new subspaces including the subspace that contains the central 1-cell. Figure 3.12(a) displays an additional SSF (the dot set in the yellow background) to be added to the arrangement in Figure 3.6 producing the arrangement in Figure 3.12(b).

Now, each cell has two signs, (δ_1 and δ_2), one for each SFF in the arrangement, whose combination as a position vector allows the differentiation of each subset (Fukuda, 2004).



(a) (b)

FIGURE 3.12: Central SSF arrangement formation

The position vector takes a geometric coordinate in combinatorial coordinates and places it in the sign vector space of an oriented matroid associated with the respective central SSF arrangement (see Figure 1.5). Space elements belonging to the same subspace must produce an identical sign vector. Equation 3.12 is rewritten as Equation 3.27 to establish the sign for each space element in regards to a specific SSF_i. The position vector of a cell c is defined by $(\delta_1(c), \delta_2(c), \ldots, \delta_n(c))$, where n is the number of SSFs in the central SSF arrangement.

$$\delta_i(c) = \begin{cases} 0, & -|m_i| \leq H_i(x_c, y_c) \leq |m_i| \\ sign(H_i(x_c, y_c)), & \text{otherwise} \end{cases}, \tag{3.27}$$

where c is a 1-cell, $H_i(c)$ is the ith SSF value at the combinatorial coordinate of c, and $m_i = a_i$ if the central 1-cell is a vertical 1-cell or b_i if it is a horizontal 1-cell.

For indexing each subspace, a position vector is encoded as though each sign value was a 3 base value according to Equation 3.28 with respect to each SSF_i. The position value vector of a cell c is defined by Equation 3.29, where n is the number of SSFs in the central SSF arrangement.

$$\upsilon_i(c) = \begin{cases} 0, & \delta_i(c) = - \\ 1, & \delta_i(c) = 0 \\ 2, & \delta_i(c) = + \end{cases}. \tag{3.28}$$

$$\overrightarrow{\upsilon}(c) = \left[\upsilon_1(c), \upsilon_2(c), \ldots, \upsilon_n(c)\right]^T. \tag{3.29}$$

Figure 3.13 displays a Cartesian complex snippet with an central SSF arrangement containing 7 SSFs (each one represented as a thick line of different color) along with the respective reference lines (thinner

lines with the same color of the respective SSF). Based on the reference lines, 29 subspaces (including the central subspace) are produced, each of which is numbered from "1" to 28. In the figure only, some are displayed with the respective position value vectors obtained by applying Equation 3.28 (i increasing from right to left).

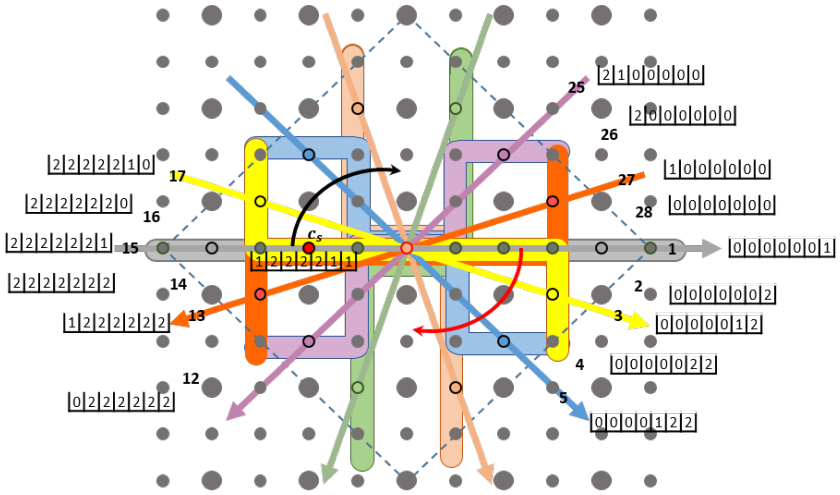


FIGURE 3.13: SSF arrangement subspace indexing

As can be observed in Figure 3.13, each SSF uses a position in the position vector starting from the most horizontal SSF, which corresponds to the rightmost vector position, and continuing in the order given by the clockwise direction to their positive sides (displayed as a curved red arrow), i.e., from right to left on the position vector. When a 1-cell c is on the negative side of an SSF_i ($c \in SSF_i^-$), the respective position on the vector takes the value of 0. When c is on the SSF_i ($c \in SSF_i^0$), the value is "1", and when c is on the positive side ($c \in SSF_i^+$), the value is "2". Therefore, for subspaces between the negative subspace of the initial SSF (the most horizontal labelled with 28 in the figure) and the positive subspace of the last SSF (labeled with 14), the value at the i position of the vector provides the order of each of the three subspaces formed by the respective SSF_i. Furthermore, the order of the three subspaces of an SSF is displaced exactly two subspaces with respect to the subspace order of the previous SSF because the positive subspace of the previous SSF, whose value is "2", corresponds to the negative subspace of the next SSF, whose value is "0"; this is established by Equation 3.30 when $c \in SSF_j^0$ and 3.31 when $c \in SSF_j^+$.

$$\upsilon_i(c) = \begin{cases} 2, & 1 \leq i < j \\ 1, & i = j \\ 0, & i > j \end{cases}, c \in SSF_j^0, \upsilon_1(c) > \upsilon_n(c) \; . \tag{3.30}$$

$$\upsilon_i(c) = \begin{cases} 2, & 1 \leq i \leq j \\ 0, & i > j \end{cases}, c \in SSF_j^+, \upsilon_1(c) > \upsilon_n(c) \; . \tag{3.31}$$

However, when the value of the vector corresponding to the first SFF is less than that corresponding to the last SSF, i.e., $\upsilon_1(c) < \upsilon_n(c)$, the respective subspace is after the positive subspace of the last SSF (marked 14 in the figure) and the cumulative process is no longer fulfilled in the same manner. These subspaces

are defined by the same SSFs starting again with the more horizontal; however, now in the direction of their negative sides (displayed in Figure 3.13 as a curved black arrow). Hence, the values in the position vectors vary in the reverse order "2", "1", and "0". The above means that the respective order can be established by determining the complement to "2" of each value of the vector plus the displacement that was already provided. This corresponds to the positive subspace of the last SSF whose value corresponds to the position vector with all their positions having a value of "2"; this is the case of the subspace labelled with 14 in the figure. Therefore, in Figure 3.13, the vector $(2,2,2,2,2,2,1)$ of subspace "15" is expanded to $(0,0,0,0,0,0,1,2,2,2,2,2,2,2)$. This has already been established by Equations 3.32 when $c \in SSF_j^0$ and 3.33 when $c \in SSF_j^+$. In this case, Equation 3.34 defines the position value vector of a cell c.

$$\upsilon_i(c) = \begin{cases} 2, & 1 \leq i < j+n \\ 1, & i = j+n \\ 0, & i > j+n \end{cases}, c \in SSF_j^0, \upsilon_1(c) < \upsilon_n(c) \ . \tag{3.32}$$

$$\upsilon_i(c) = \begin{cases} 2, & 1 \leq i \leq j+n \\ 0, & i > j+n \end{cases}, c \in SSF_j^+, \upsilon_1(c) < \upsilon_n(c) \ . \tag{3.33}$$

$$\overrightarrow{\upsilon}(c) = \left[\upsilon_1(c), \upsilon_2(c), \ldots, \upsilon_{2n}(c)\right]^T \ . \tag{3.34}$$

That is, to reference on-SSF subspaces (SSF^0), the order (index) of a subspace ss to which a 1-cell c belongs can be established by adding the values of its respective position vector in agreement with Equation 3.35.

$$ss(c) = \begin{cases} \sum_{i=1}^{n} \upsilon_i(c), & \upsilon_1(c) \geq \upsilon_n(c) \\ 2n + \sum_{i=1}^{n} (2 - \upsilon_i(c)), & \text{otherwise} \end{cases} , \tag{3.35}$$

where n is the number of SSFs in the central arrangement.

Notwithstanding the above, as indicated in Figure 3.13, in addition to the central 1-cell, it is possible that other 1-cells can also be shared by more than one SSF. This condition can be established when within the position vector the value "1" appears more than once, which is the case of the vector $(1,2,2,2,2,1,1)$ of the 1-cell c_s in the figure (red dot with a black outer circle). This means that for SSFs, there are additional subspaces to be considered and the Equation 3.35 must be redefined because it can obtain ambiguous values. For this, a position vector value can be made unique taking it as a 3 base value by Equation 3.36 because $2\sum_{i=0}^{n} 3^i = 3^{n+1} - 1$.

$$val(c) = \begin{cases} \sum_{i=1}^{n} 3^{i-1} \upsilon_i(c), & \upsilon_1(c) \geq \upsilon_n(c) \\ 3^n - 1 + \sum_{i=1}^{n} 3^{i-1} (2 - \upsilon_i(c)), & \text{otherwise} \end{cases} , \tag{3.36}$$

where n is the number of SSFs in the central arrangement.

For the special cases established by Equations 3.30 to 3.33, Equation 3.36 can be written as Equations 3.37 and 3.38.

$$val(c) = \begin{cases} 2 \times 3^{j-1} - 1, & \upsilon_1(c) \geq \upsilon_n(c) \\ 2 \times 3^{j-1+n} - 1, & \text{otherwise} \end{cases}, c \in SSF_j^0 \ . \tag{3.37}$$

$$val(c) = \begin{cases} 3^{j} - 1, & \upsilon_1(c) \geq \upsilon_n(c) \\ 3^{j+n} - 1, & \text{otherwise} \end{cases}, c \in SSF_j^+ \ . \tag{3.38}$$

TABLE 3.3: Subspace values for central SSF arrangement on Figure 3.13

$val(c)$	Index	values vector	$s_j=\{c\|val(c)=j\}$ 1-cell coordinates	$\|\{i \mid \upsilon_i(c) = 1, 1 \leq i \leq n\}\|$
1	1	[0, 0, 0, 0, 0, 0, 1]	(5,0)	1
5	3	[0, 0, 0, 0, 0, 1, 2]	(5,2), (4,1)	1
17	5	[0, 0, 0, 0, 1, 2, 2]	(4,3), (5,4), (3,2)	1
26	6	[0, 0, 0, 0, 2, 2, 2]	(3,4)	0
53	7	[0, 0, 0, 1, 2, 2, 2]	(2,3)	1
80	8	[0, 0, 0, 2, 2, 2, 2]	(1,2), (1,4)	0
161	9	[0, 0, 1, 2, 2, 2, 2]	(0,3)	1
242	10	[0, 0, 2, 2, 2, 2, 2]	(-1,4)	0
485	11	[0, 1, 2, 2, 2, 2, 2]	(-3,4), (-2,3), (-1,2)	1
1457	13	[1, 2, 2, 2, 2, 2, 2]	(-2,1), (-3,2)	1
4373	15	[2, 2, 2, 2, 2, 2, 1]	(-3,0)	1
13121	17	[2, 2, 2, 2, 2, 1, 0]	(-2,-1), (-3,-2)	1
39365	19	[2, 2, 2, 2, 1, 0, 0]	(-3,-4), (-2,-3), (-1,-2)	1
59048	20	[2, 2, 2, 2, 0, 0, 0]	(-1,-4)	0
118097	21	[2, 2, 2, 1, 0, 0, 0]	(0,-3)	1
177146	22	[2, 2, 2, 0, 0, 0, 0]	(1,-4), (1,-2)	0
354293	23	[2, 2, 1, 0, 0, 0, 0]	(2,-3)	1
531440	24	[2, 2, 0, 0, 0, 0, 0]	(3,-4)	0
1062881	25	[2, 1, 0, 0, 0, 0, 0]	(3,-2), (4,-3), (5,-4)	1
3188645	27	[1, 0, 0, 0, 0, 0, 0]	(5,-2), (4,-1)	1
44	–	[0, 0, 0, 1, 1, 2, 2]	(2,1)	2
404	–	[0, 1, 1, 2, 2, 2, 2]	(0,1)	2
733	–	[1, 0, 0, 0, 0, 1, 1]	(3,0)	3
1453	–	[1, 2, 2, 2, 2, 1, 1]	(-1,0)	3
98414	–	[2, 2, 2, 1, 1, 0, 0]	(0,-1)	2
885734	–	[2, 1, 1, 0, 0, 0, 0]	(2,-1)	2

Table 3.3 displays, for the central SSF arrangement in Figure 3.13, the subspace position vector value $val(c)$ for each 1-cell that is from the central 1-cell at a Manhattan distance of "2" (broken line diamond

in the figure). Each subspace s_j is defined based on Equation 3.39.

$$s_j = \{c | val(c) = j\} \ . \tag{3.39}$$

If $|\{i \mid \upsilon_i(c) = 1, 1 \leq i \leq n\}| > 1$, then the respective s_j will be a subset shared by several SSFs. The last six rows correspond with the six subspaces that share 1-cells. In addition to the central 1-cell, the rows with $val(c)$ values of 733 and 1453 correspond to the most shared 1-cells whose position vectors contains three values of "1" ($|\{i \mid \upsilon_i(c) = 1\}| = 3$). Therefore, Equation 3.40 provides SSF_k as a subset. Table 3.4 presents the SSFs in the central SSF arrangement in Figure 3.13 defined in terms of the respective subsets in Table 3.3.

$$SSF_k = s_{2\times 3^{k-1}-1} \cup s_{2\times 3^{k-1+n}-1} \cup \{c_c | c_c \text{ is the central 1-cell}\} \cup \bigcup_{\substack{c\in s_j \\ \upsilon_k(c)=1 \\ |\{i|\upsilon_i(c)=1\}|>1}} s_j = \bigcup_{\substack{c\in s_j \\ \upsilon_k(c)=1}} s_j \ . \tag{3.40}$$

TABLE 3.4: SSFs defined in terms of subspaces of Table 3.3

SSF_k	$\bigcup_{\substack{c\in s_j \\ \upsilon_k(c)=1}} s_j$
1	$s_1 \cup s_{733} \cup s_{4373} \cup s_{1453}$
2	$s_5 \cup s_{733} \cup s_{13121} \cup s_{1453}$
3	$s_{17} \cup s_{44} \cup s_{39365} \cup s_{98414}$
4	$s_{53} \cup s_{44} \cup s_{118097} \cup s_{98414}$
5	$s_{161} \cup s_{404} \cup s_{354293} \cup s_{885734}$
6	$s_{485} \cup s_{404} \cup s_{1062881} \cup s_{885734}$
7	$s_{1457} \cup s_{1453} \cup s_{3188645} \cup s_{733}$

Figure 3.14 displays the case when the central 1-cell is vertical. As can be observed, the most horizontal initial subspace (that with position value vector $(0,0,0,0,0,0,1)$ in the figure) is not parallel to the x axis, hence, in this case, the SSF_k, as a subset, is conformed as Table 3.5 indicates.

TABLE 3.5: SSFs defined in terms of subspaces of Figure 3.14

SSF_k	$\bigcup_{\substack{c\in s_j \\ \upsilon_k(c)=1}} s_j$
1	$s_1 \cup s_4 \cup s_{4373} \cup s_{10934}$
2	$s_5 \cup s_4 \cup s_{13121} \cup s_{10934}$
3	$s_{17} \cup s_{125} \cup s_{39365} \cup s_{275561}$
4	$s_{53} \cup s_{125} \cup s_{118097} \cup s_{275561}$
5	$s_{161} \cup s_{125} \cup s_{354293} \cup s_{275561}$
6	$s_{485} \cup s_{1214} \cup s_{1062881} \cup s_{2657204}$
7	$s_{1457} \cup s_{1214} \cup s_{3188645} \cup s_{2657204}$

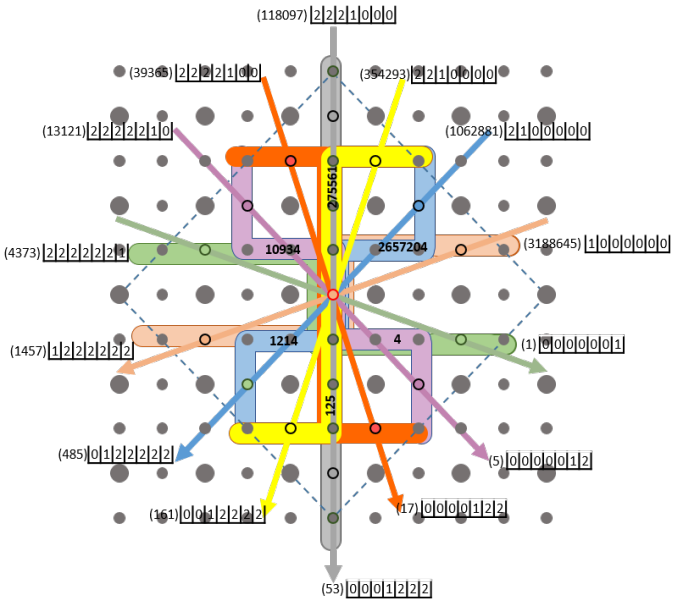


FIGURE 3.14: Vertical 1-cell centered SSF arrangement subspaces indexing

3.4 Layer of segmentation tasks (STL)

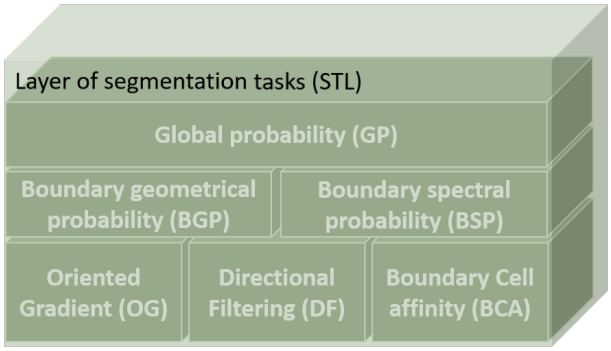


FIGURE 3.15: Subcomponents of the layer of segmentation tasks

The model for multispectral image segmentation must address all conceptual elements that participate in the segmentation process, specifically elements required to support the functionalities indicated in Figure 2.22 for conducting, first, a geometrical edge detection task and then, a spectral edge detection task following (Arbelaez et al., 2011), however using a Cartesian complex space rather than a conventional image space. The STL includes the conceptual subcomponents displayed in Figure 3.15. Directional filtering (DF) supports 1-cell oriented filtering and convolution. Orient gradient (OG) provides the gradient disks required to produce the maximum gradient images. Boundary cell affinity (BCA) supports the spectral affinity analysis between the 1-cells.

3.4.1 Directional filtering (DF)

Image segmentation focuses on boundary arrangements those are not areal; rather they are linear and hence the directional filtering must focus on that structure. 1D linear structures are not available in a conventional image analysis context. Rather, a Cartesian complex provides 1-cells that are used as the filtering building block. The edge detection process proposed by Arbelaez et al. (2011) is based on the ability to detect the contrasts and absence of variation in the neighborhoods close to each spatial location of the elements used to represent the image, which is achieved by applying filters as tools of analysis of convolution. Because the research that was undertaken is based on using Cartesian complexes to represent the image, each of the filtering processes was redefined using a 1-cell as the central element. Figure 3.16 displays the conceptual elements used in the filtering task. As can be observed in the figure, all DF conceptual elements are supported by the CCT (Section 3.1), CCG (Section 3.2), and OML (Section 3.3) underlying layers. The core concept is the filter kernel (*FilterKernel* class), which corresponds to a small 2-dimensional neighborhood whose settings can be set in a Prewitt (Prewitt, 1970) (*DirectionalFilter* class) or Gaussian derivative arrangement in agreement with an orientation set (*FilterBank* class). *DirectionalFilterProcessor* applies a filter kernel to a *CartesianComplex* to produce a filtered image (*FilteredCartesianComplex* class).

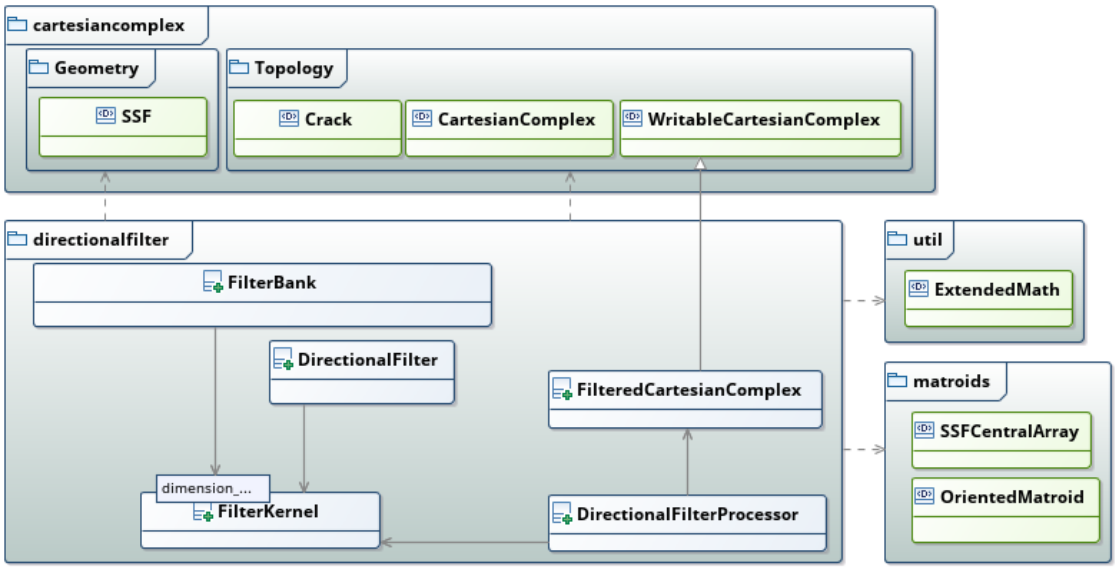


FIGURE 3.16: Directional filter conceptual elements

Filter kernel

A *FilterKernel* corresponds to a small neighborhood of 1- or 2-cells, each of which has an associated value and a position relative to the combinatorial coordinates of a reference 1-cell on which it will be applied. A kernel κ can be thought of as small Cartesian complex whose cells have their combinatorial

coordinates relative to a c central 1-cell (Equation 3.41). Each of these cells has a kernel value that is used in the convolution processing.

$$\kappa = \{c_i | \nu(c, c_i)\} \ , \tag{3.41}$$

where $\nu(c, c_i)$ is the neighborhood relationship condition that c central 1-cell and c_i cell must satisfy.

Prewitt directional filtering is supported to be used in texture analysis tasks during the image classification stage (Valero, Dallos, and Lizarazo, 2018). As Figure 3.17 indicates, a directional Prewitt filter kernel is defined based on an SSF_i along a horizontal central 1-cell (Figure 3.17(a)) or a vertical central 1-cell (Figure 3.17(b)) in a neighborhood given by Equation 3.42. Equation 3.43 sets the kernel value at a 2-cell (pixel) p, following Equation 3.1. The figure displays a directional Prewitt kernel for SSF_i $H_i(x,y) = x + y - 1$ and $n = 2$.

$$\kappa_n = \{p | p \in SON(c_i), d(c, c_i) \leq 2n, \tau(p) = 2\text{−cell}\} \ , \tag{3.42}$$

where n is the radius in number of additional 1-cells in addition to the central 1-cell,
$d(c, c_i)$ is the Manhattan distance between the c central 1-cell and a c_i 1-cell, and
$\tau(p)$ is given in agreement with Equation 3.2.

$$\mathbf{v}_p = \left[\delta_i(p)\right] \ , \tag{3.43}$$

where $\delta_i(p)$ is given by Equation 3.27.

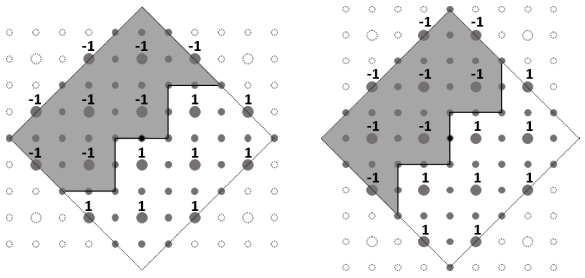


FIGURE 3.17: Directional Prewitt kernel

To support the segmentation tasks displayed in Figure 2.22, the implementation proposed in (Arbelaez et al., 2011) was reformulated using a 1-cell as the central space element in calculating the Gaussian and center-surround layouts. A 2D Gaussian filter kernel is built from two 1D filter kernels, one vertical and the other horizontal, each one based on $g(x;\sigma) = \exp\left(-\frac{x^2}{(2\sigma^2)}\right)$, although $g'(x)$ and $g''(x)$ are also supported with an optional Hilbert transform. When the central 1-cell is horizontal, the horizontal kernel $\mathbf{h}$ is composed of $2n+1$ 2-cells (pixels) and the vertical kernel $\mathbf{v}$ of $2n$. When the central 1-cell is vertical, the horizontal kernel is composed of $2n$ 2-cells and the vertical of $2n+1$ (n is the number of 1-cells from the central 1-cell). Following that established by Equations 3.2 and 3.13, when the central 1-cell is horizontal, the $\mathbf{v}$ and $\mathbf{h}$ vector kernels are respectively defined by Equations 3.44 and 3.45. When the

central 1-cell is vertical, Equation 3.44 must be extended such that $0 \leq i \leq 2n$ and Equation 3.45 must be contracted such that $0 \leq j \leq 2n-1$; Equation 3.46 must be subject to same considerations.

$$\mathbf{v} = (g(y_0), g(y_1), \ldots, g(y_{2n-1})), y_j = 2j - 2n + 1 \ . \tag{3.44}$$

$$\mathbf{h} = (g(x_0), g(x_1), \ldots, g(x_{2n})), x_i = 2i - 2n + 1 \ . \tag{3.45}$$

$$\begin{aligned} \kappa_n = \{p| \quad & x_p \in \{x_1, x_2, \ldots, x_i, \ldots, x_{2n}\}, x_i = 2i - 2n + 1, \\ & y_p \in \{y_1, y_2, \ldots, y_j, \ldots, y_{2n-1}\}, , y_j = 2j - 2n + 1\} \ , \end{aligned} \tag{3.46}$$

where n is the radius in number of additional 1-cells in addition to the central 1-cell.

$$\mathbf{v}_p = \left[g(x_p; \sigma_x) g(y_p; \sigma_y) \right] \ . \tag{3.47}$$

The kernel can be rotated based on the SSF a and b parameters. Figure 3.18 displays a rotated Gaussian filter kernel (largest dots are 2-cells, smallest dots are inter-pixel elements, and the black dot is the central 1-cell). This figure illustrates that the initial kernel (horizontal along central 1-cell) is rotated and the new kernel is extended and completed with zeroes such that it contains the rotated evaluation kernel (filled dots).

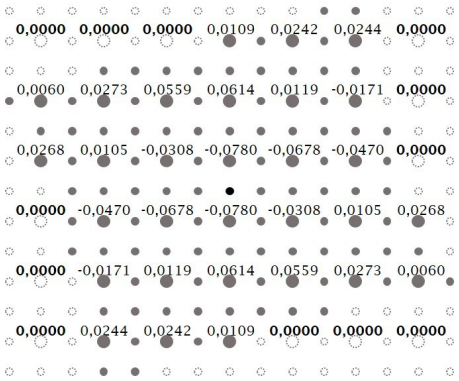


FIGURE 3.18: $\frac{\pi}{10}$ rotated second derivative Gaussian filter kernel

The center-surround kernel is built as in (Arbelaez et al., 2011) based on a Gaussian difference; the center at 1-cell c is in agreement with Equation 3.48.

$$\mathbf{v}_p = \left[g_1(p) - g_2(p) \right], \sigma_1 > \sigma_2 \ , \tag{3.48}$$

where $g_i(p) = g(x_p; \sigma_i) g(y_p; \sigma_i)$.

Once the filter kernel has been defined based on n 1-cells in addition to the central 1-cell and o orientation, it can be applied in the same manner regardless of the approach followed for its creation (Prewitt or

Gaussian). Equation 3.49 provides the 2D kernel application at a 1-cell c of a Cartesian complex C for some position in a c value vector.

$$K_{n,o,b}(c) = K_{n,o,b}(x_c, y_c) = \sum_{p \in \kappa_n} \mathbf{v}_{p_c}(b)\mathbf{v}_p(1), \quad c, p_c \in C \,, \tag{3.49}$$

where $\mathbf{v}_p(1)$ is the value at position "1", the only one, in the value vector of kernel pixel p,
$\mathbf{v}_{p_c}(b)$ is the value at position b in the value vector of pixel p_c of Cartesian complex C.
If c is horizontal, $x_{p_c} = x_c - x_p + 1$, otherwise $x_{p_c} = x_c - x_p$;
if c is horizontal, $y_{p_c} = y_c - y_p$, otherwise $y_{p_c} = y_c - y_p + 1$.

The application of Equation 3.49 for each value of value vector between positions b_s and b_f produces the value vector established by Equation 3.50.

$$\mathbf{v}_{c_{K_{n,o}}} = \begin{bmatrix} K_{n,o,b_s}(x_c, y_c) \\ \vdots \\ K_{n,o,b_f}(x_c, y_c) \end{bmatrix} \,, \tag{3.50}$$

where $K_{n,o,b_s}(x_c, y_c), \ldots, K_{n,o,b_f}(x_c, y_c)$ are evaluations of Equation 3.49 between $\mathbf{v}_{p_c}(b_s)$ and $\mathbf{v}_{p_c}(b_f)$, $1 \leq b_s \leq b_f \leq m$.

Bank of Directional filters

A directional filter allows analysis in a single direction; however, a particular feature can be in any direction, hence, it is necessary to apply several filters in different directions to determine a specific feature (Leung and Malik, 1996). As was discussed in Section 3.3.1, in a locally finite space the number of orientations that can be evaluated is also finite and can be determined by the maximum number of SSFs in an central SSF arrangement. A bank of filters can be produced from an central SSF arrangement for a o orientation and n 1-cells in addition to the central 1-cell. In fact, in a Cartesian complex, two filters are required for each orientation in a filter bank: one with a horizontal 1-cell as the central 1-cell and the other with a vertical 1-cell as the central 1-cell.

In the case of Prewitt filters, a filter bank is produced assigning to each 2-cell the respective sign based on Equation 3.43 for each SSF_i in the central arrangement. This is displayed in Figure 3.19 for $n = 1$ as the number of 1-cells in addition to the central 1-cell and $o = 3$ as the number of orientations, the maximum in this case. The figure displays the filter banks when the central 1-cell is horizontal (top row) and when the central 1-cell is vertical (bottom row). The figure indicates "6" filters in each bank, although there are "3" orientations; for an orientation, it is possible to have two layouts, the positive and negative on each orientation SSF side. This is established by Equation 3.51.

$$\mathbf{v}_p = \begin{cases} (\delta_i(p)) & i \leq o, \\ (-\delta_{i\%o}(p)) & \text{otherwise} \end{cases} \,, \tag{3.51}$$

where $i\%o$ is the remainder of dividing i by o.

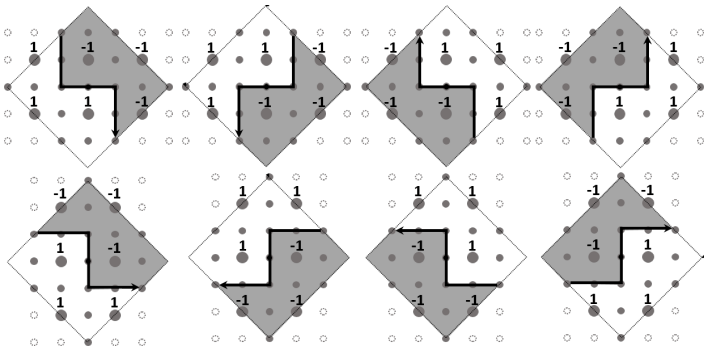
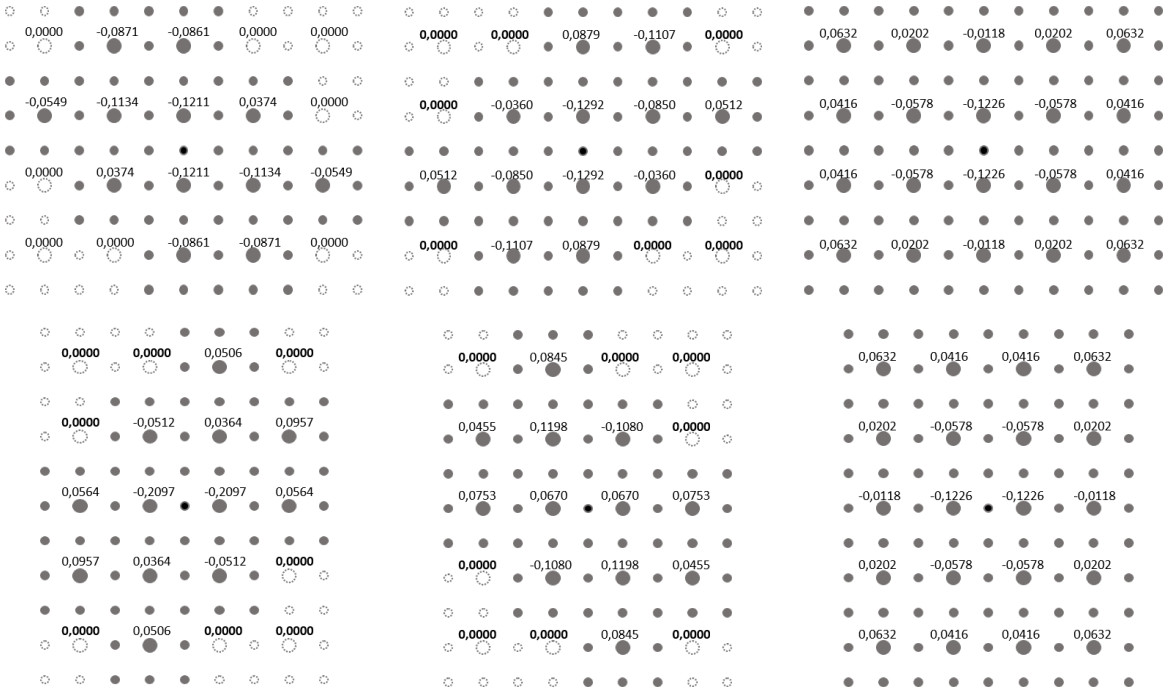


FIGURE 3.19: Prewitt directional filter bank for $n = 1$ and $o = 2$

When the kernels are Gaussian, given o orientations and n 1-cells in addition to the central 1-cell, a filter bank (Leung and Malik, 1996) is composed of 2D Gaussian kernels that are Gaussian along the central 1-cell and Gaussian second derivative with Hilbert transform perpendicular to the central (Arbelaez et al., 2011). The filter bank is completed with a center-surround filter kernel. These filters can be produced to different scales, i.e., different numbers of 1-cells in addition to the central 1-cell.



(a) Gaussian/Gaussian derivative for $n = 1$ and $o = 2$ (b) Center surround for $n = 2$

FIGURE 3.20: Gaussian directional filter bank

Figure 3.20 displays a Gaussian filter bank subset for $n = 1$ and two orientations ($o = 2$). The figure indicates, on the top row, the filter bank when the central 1-cell is horizontal; the bottom row displays the filter bank when the central 1-cell is vertical. Figure 3.20(a) displays the Gaussian/Gaussian derivative kernels and Figure 3.20(b) is a center-surround kernel.

When a filter bank is applied, a 1-cell produces a vector with as many values as $m_F \times o$, where m_F is the number of values in the value vector in agreement with Equation 3.50 and o is the number of filter kernels in the filter bank. Hence, the value vector $\mathbf{v}_{c_{K_{n,1\ldots o}}}$ produced by applying a filter bank is the concatenation of each filter appliance; this is established by Equation 3.52.

$$\mathbf{v}_{c_{K_{n,1\ldots o}}} = \mathbf{v}_{c_{K_{n,1}}} \,||\, \mathbf{v}_{c_{K_{n,2}}} \,||\, \ldots \,||\, \mathbf{v}_{c_{K_{n,o}}} \;, \tag{3.52}$$

where $||$ implies vector concatenation.

Cartesian complex $F_{n,1\ldots o}$ produced by applying a filter bank, with o orientations and n 1-cells in addition to the central 1-cell, at each 1-cell in a Cartesian complex C ($c \in C$), is given by Equation 3.53.

$$F_{n,1\ldots o} = \{c_{K_{n,1\ldots o}} | x_{c_{K_{n,1\ldots o}}} = x_c, y_{c_{K_{n,1\ldots o}}} = y_c \forall c \in C\} \;, \tag{3.53}$$

where $\mathbf{v}_{c_{K_{n,1\ldots o}}}$, as given by Equation 3.52, is the value vector at $c_{K_{n,1\ldots o}}$ 1-cell.

3.4.2 Oriented gradient (OG) Cartesian complex

Oriented gradient is a task prior to the task of establishing the boundary probability at a space element (Martin, Fowlkes, and Malik, 2004). The oriented gradient $G(c;\theta)$ is calculated as in (Arbelaez et al., 2011); however, at 1-cells by placing a Manhattan distance-based ball centered in x_c, y_c and divided into two half-balls by an SSF_i based on its θ_i in agreement with the respective a_i and b_i parameters. The gradient calculates histogram values on each disk halve and calculates the gradient as the chi squared distance between both sides.

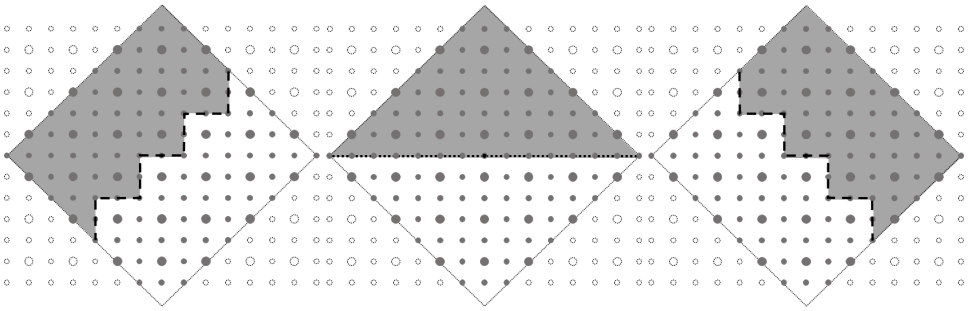


FIGURE 3.21: Oriented ball division definition for calculating gradient Cartesian complex for central SSF arrangement with three SSFs

Figure 3.21 displays three disk half arrangements for three SSFs in an central SSF arrangement. If the gradient calculates the histogram in each disk half for each oriented diameter, it would actually recalculate three times the six disk sectors displayed in the figure. Rather, a ball is segmented into as many subspaces as defined by the central SSF arrangement as described in Section 3.3.2, which are then properly combined to calculate the gradients.

The OG conceptual elements are displayed in Figure 3.22. The core concept is a gradient disk (*GradientDisk* class) that first calculates each histogram half-ball subspace (*HistoSector* class), then combines

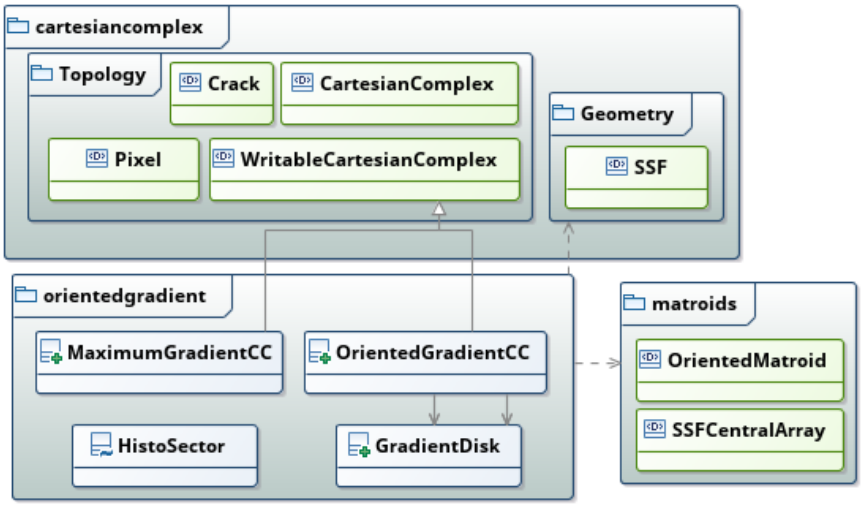


FIGURE 3.22: Oriented gradient conceptual elements

them to properly calculate the histogram of each half-ball subset, and finally calculates each oriented gradient for each orientation present in an central SSF arrangement. The oriented gradients are calculated at each 1-cell of a Cartesian complex to produce, first, an oriented gradient Cartesian complex (*OrientedGradientCC* class), and then the respective maximum gradient at each 1-cell (*MaximumGradientCC* class). As the figure indicates, the supporting conceptual elements are in the underlying CCT, CCG, and OML layers.

For a c central 1-cell in a Cartesian complex C, the oriented gradient calculation starts with the subspace histogram calculation for each s_j subspace linked to an central SSF arrangement with o orientations and n 1-cells in addition to the central c through its oriented matroid. Equation 3.39 is applied to each 1-cell in a neighborhood, established by Equation 3.54.

$$\kappa_n = \{c_i | d(c, c_i) \leq 2n\} \ , \tag{3.54}$$

where n is the radius in number of additional 1-cells in addition to the central 1-cell, and
$d(c, c_i)$ is the Manhattan distance between the c central 1-cell and a c_i 1-cell.

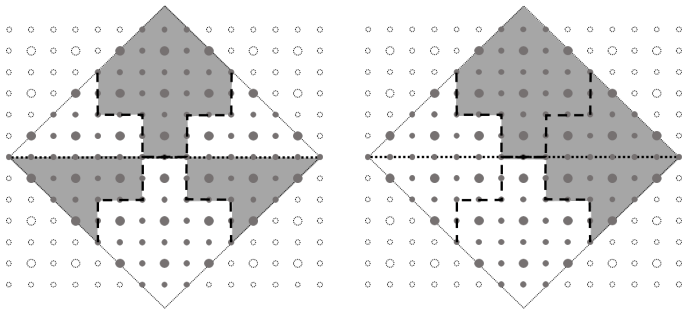
Hence, the histogram is calculated from partitions established by Equation 3.55 using Equation 3.56. Because the subspaces on each SSF side must be used, the central subspace is never used and its histogram calculation is not necessary.

$$f_{n,s_j,\mathbf{V}_h} = \{c_i | val(c_i) = j, \mathbf{V}_{c_i} = \mathbf{V}_h, c_i \in \kappa_n\} \ , \tag{3.55}$$

where $val(c_i)$ is given by Equation 3.36,
$\mathbf{V}_{c_i}$ is the value vector of c_i 1-cell, and
$\mathbf{V}_h$ is a histogram beam corresponding to one of the unique values of $\mathbf{V}_{c_i}$ in κ_n.

$$H_n(s_j, \mathbf{V}_h) = |f_{n,s_j,\mathbf{V}_h}| \ . \tag{3.56}$$

Figure 3.23(a) displays a thirteen-subspace partition for an central SSF arrangement with three orientations and three 1-cells in addition to the central 1-cell. In this case, twelve histograms must be calculated.



(a) (b)

FIGURE 3.23: Oriented ball division definition for calculating gradient Cartesian complex for central SSF arrangement with three SSFs

Once the subspace histograms are calculated, they must be suitably combined to calculate the oriented gradients using the respective oriented matroid. For a given SSF_i orientation in an central SSF arrangement with o orientations, there are $2o-1$ subspaces, in addition to those shared by the other SSFs, on each SSF_i side. Furthermore, an on-SSF subspace is two subspaces after the previous on-SSF subspace. Hence, the subspace s_j assignment to an SSF_i sides $S_{SSF_i}(s_j)$ is given by Equation 3.57.

$$S_{SSF_i}(s_j) = \begin{cases} + & ,2\times 3^{i-1}-1 < j < 2\times 3^{o+i-1}-1, \\ - & \text{otherwise} \end{cases} , j \neq 2\times 3^{i-1}-1, j \neq 2\times 3^{o+i-1}-1, \tag{3.57}$$

where, by Equation 3.37 and $j \neq 2\times 3^{i-1}-1, j \neq 2\times 3^{o+i-1}-1$,
$1 \leq i \leq o$.

Figure 3.23(b) displays a subspace combination for the second SSF, in this case, the SSFs do not share a 1-cell. However, as established in Section 3.3.2, in a locally finite space, several SSFs can share other 1-cells in addition to the central 1-cell, which are given by Equation 3.40. Hence, the histogram for each SSF_i side is given by Equations 3.58 and 3.59.

$$H_n^+(S_{SSF_i}, \mathbf{V}_h) = \sum_{\substack{S_{SSF_i}(s_j)=+ \\ s_j \not\subseteq SSF_i}} H_n(s_j, \mathbf{V}_h)\ . \tag{3.58}$$

$$H_n^-(S_{SSF_i}, \mathbf{V}_h) = \sum_{\substack{S_{SSF_i}(s_j)=- \\ s_j \not\subseteq SSF_i}} H_n(s_j, \mathbf{V}_h)\ . \tag{3.59}$$

The gradient $g(c,S_{SSF_i})$ oriented based on SSF_i at c 1-cell is given by Equation 3.60 as a χ^2 distance (Arbelaez et al., 2011).

$$g_n(c,SSF_i) = \frac{1}{2}\sum_{\mathbf{V}_h} \frac{(H_n^+(S_{SSF_i},\mathbf{V}_h) - H_n^-(S_{SSF_i},\mathbf{V}_h))^2}{H_n^+(S_{SSF_i},\mathbf{V}_h) + H_n^-(S_{SSF_i},\mathbf{V}_h)} \ , \tag{3.60}$$

where SSF_i is in the central SSF arrangement centered at c 1-cell.

The gradient calculation for all SSFs in an central SSF arrangement produces the value vector given by Equation 3.61.

$$\mathbf{v}_{c_{gn}} = \begin{bmatrix} g_n(c,SSF_1) \\ \dots \\ g_n(c,SSF_o)) \end{bmatrix} \ . \tag{3.61}$$

Cartesian complex $G_{n,o}$, produced by calculating the oriented gradients for o orientations and n radius at each c 1-cell in a Cartesian complex C ($c \in C$), is given by Equation 3.62.

$$G_{n,o} = \{c_g | x_{c_g} = x_c, y_{c_g} = y_c \forall c \in C\} \ , \tag{3.62}$$

where $\mathbf{v}_{c_{gn}}$, as given by Equation 3.61, is the value vector of the c_g 1-cell.

Oriented gradients can be calculated based on the value at each value vector position of 1-cells extending Equations 3.58 to 3.61 in the manner provided by Equations 3.63 to 3.66.

$$H_n^+(S_{SSF_i},\mathbf{V}_h(b)) = \sum_{\substack{S_{SSF_i}(s_j)=+ \\ s_j \not\subseteq SSF_i}} H_n(s_j,\mathbf{V}_h(b)) \ . \tag{3.63}$$

$$H_n^-(S_{SSF_i},\mathbf{V}_h(b)) = \sum_{\substack{S_{SSF_i}(s_j)=- \\ s_j \not\subseteq SSF_i}} H_n(s_j,\mathbf{V}_h(b)) \ . \tag{3.64}$$

$$g_n(c,SSF_i,b) = \frac{1}{2}\sum_{\mathbf{V}_h(b)} \frac{(H_n^+(S_{SSF_i},\mathbf{V}_h(b)) - H_n^-(S_{SSF_i},\mathbf{V}_h(b)))^2}{H_n^+(S_{SSF_i},\mathbf{V}_h(b)) + H_n^-(S_{SSF_i},\mathbf{V}_h(b))} \ , \tag{3.65}$$

where SSF_i is in the central SSF arrangement centered at the c 1-cell.

$$\mathbf{v}_{c_{gn}} = \left[g_n(c,SSF_1,1),\dots,g_n(c,SSF_o,1),\dots,g_n(c,SSF_1,m),\dots,g_n(c,SSF_o,m)\right]^T \ , \tag{3.66}$$

where the 1 to m interval corresponds to the value vector positions of the c 1-cells (Equation 3.1).

The maximum oriented gradient in an orientation is calculated in agreement with Equation 3.67.

$$\mathbf{v}_{mg} = \left[\max_j\{g_n(c,SSF_j,1)\},\dots,\max_j\{g_n(c,SSF_j,m)\}\right]^T \ . \tag{3.67}$$

Equation 3.68 provides the Cartesian complex MG produced by calculating the maximum oriented gradient from the $G_{n,o}$ Cartesian complex.

$$MG = \{mg | x_{mg} = x_{c_g}, y_{mg} = y_{c_g} \forall c_g \in G_{n,o}\} \,, \tag{3.68}$$

where $\mathbf{v}_{mg}$, as it is given by Equation 3.67 is the value vector at the mg 1-cell and $G_{n,o}$ is given by Equation 3.62.

3.4.3 Boundary cell affinity (BCA)

As reviewed in Section 2.3 (Equation 2.20), the maximal oriented gradient Cartesian complex is valued along a line connecting two 2-cells (pixels) for its affinity using a sparse symmetric matrix W. Rather than using 2-cells, it uses an affinity measure for each pair of 1-cells c_i, c_j within a fixed radius n considering other 1-cells on the SSF defined between them. Once W_{ij} and D_{ii} obtain their calculated values, they are passed to the underlying layer to obtain the generalized eigenvectors for a given number of smallest eigenvalues.

The BCA conceptual model elements are displayed in Figure 3.24. The areas for evaluating the 1-cell pairs for a given action radius (*AffinitySearchArea* class) are based on Manhattan distance balls. These affinity areas are applied at each 1-cell for a given Cartesian complex to produce the respective 1-cell pair in the affinity matrix (*AffinityMatrixW* class) and the cell global affinity matrix. The solution of the complete matrix equation using a given eigenvalue set, which the underlying layer provides, is used to create the generalized eigenvector Cartesian complex (*GeneralizedEigenVectorsCC* class). Each 1-cell has an eigenvector value for each eigenvalue used.

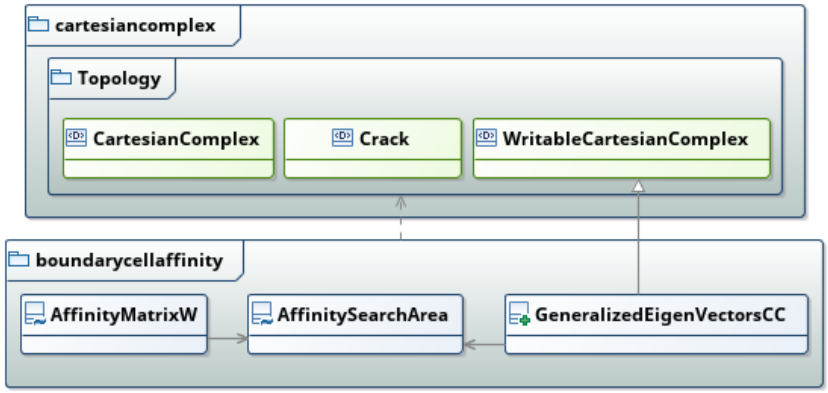


FIGURE 3.24: Conceptual elements in BCA component

An affinity analysis is performed between pairs of 1-cell elements in an affinity area. The affinity area is established using a Manhattan distance-based ball in agreement with Equation 3.54. To avoid comparing each pair of 1-cells in the search area κ, the search starts with the smallest ball, i.e., only one 1-cell in addition to the central 1-cell ($n = 1$), calculating and updating if necessary at each c_j outer 1-cell the affinity between it and the c_i central 1-cell. Next, the search proceeds in an iterative process expanding the ball by a 1-cell in all directions each time in agreement with Equation 3.69. Then, it calculates the

affinity only between each now external 1-cell and its previous 1-cell that is in the same SSF_j as the central 1-cell. Figure 3.25 displays the affinity area definition for $n = 3$; on each iteration, the outer 1-cells are represented with fewer and fewer dark arrows that cross them.

$$\kappa_r = \{c_j | d(c_i, c_j) = 2r\} \ , \tag{3.69}$$

where $1 \leq r \leq n$ is the radius in number of additional 1-cells on each r iteration, and
$d(c_i, c_j)$ is the Manhattan distance between the c_i central 1-cell and a c_j outer 1-cell.

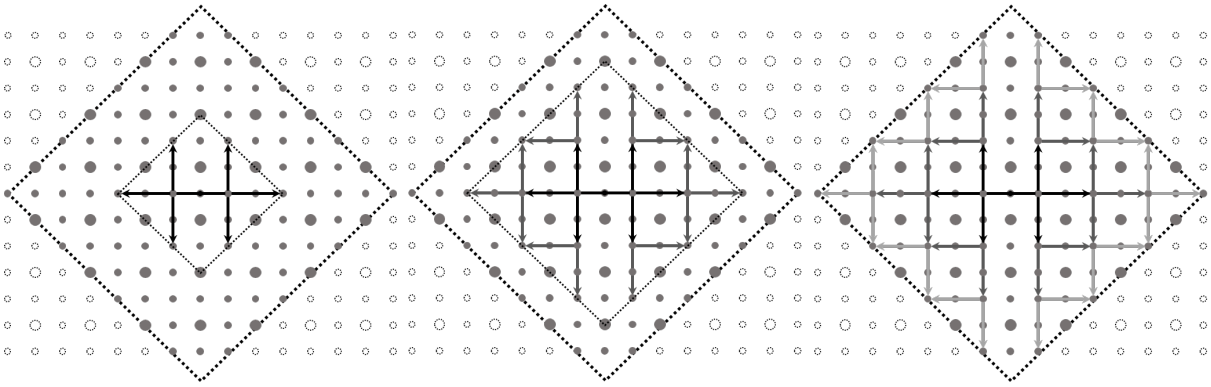


FIGURE 3.25: Affinity search area defined using Manhattan distance-based ball

To the extent that the ball grows from the smallest with $r = 1$ to the largest with $r = n$, W and D sparse matrices are progressively calculated. The affinity measurement w_{ij} between the c_i central and c_j outer 1-cells based on the value at position b of the value vector is given by Equation 3.70

$$w_{ij} = e^{\left(-\max_{c_k \in SSF_{ij}}\{\mathbf{v}_{c_k}(b)\}/\rho\right)} \ , \tag{3.70}$$

where c_i is the central 1-cell,
c_j is an outer 1-cell, i.e., $c_j \in \kappa_r$,
c_k is some 1-cell between c_i and c_j,
SSF_j crosses c_i, c_k, and c_j, and
SSF_j crosses c_i and c_j.

The a_j, b_j, and l_j parameters of $H_j(x,y)$, associated with SSF_{ij}, are established respectively by Equations 3.71, 3.72, and 3.24.

$$a_j = -\frac{\Delta y}{gcd(\Delta x, \Delta y)} \ , \tag{3.71}$$

where $\Delta x = \max(x_{c_i}, x_{c_j}) - \min(x_{c_i}, x_{c_j})$, and
$\Delta y = \max(y_{c_i}, y_{c_j}) - \min(y_{c_i}, y_{c_j})$,
gcd is the greatest common divisor.

$$b_j = \frac{\Delta x}{gcd(\Delta x, \Delta y)} \ , \tag{3.72}$$

where $\Delta x = \max(x_{c_i}, x_{c_j}) - \min(x_{c_i}, x_{c_j})$,
$\Delta y = \max(y_{c_i}, y_{c_j}) - \min(y_{c_i}, y_{c_j})$, and
gcd is the greatest common divisor.

From c_j to c_i, there is a previous 1-cell from two possible others, c_t turning or c_l continuing by the even axis. The previous 1-cell is at a Manhattan distance of "2" from the c_j to c_i direction. The displacement direction (x_d, y_d) from c_j to c_i is established by Equation 3.73.

$$\begin{bmatrix} x_d \\ y_d \end{bmatrix} = \begin{bmatrix} sign(x_{c_j} - x_{c_i}) \\ sign(y_{c_j} - y_{c_i}) \end{bmatrix} . \tag{3.73}$$

Hence, c_t and c_l are at locations given by Equations 3.74 and 3.75, respectively. Only one of c_t and c_l is on SSF_j, i.e., for only one of them $\delta_j(c) = 0$ (in agreement with Equation 3.27).

$$\begin{bmatrix} x_{c_t} \\ y_{c_t} \end{bmatrix} = \begin{bmatrix} x_{c_j} - x_d \\ y_{c_j} - y_d \end{bmatrix} , \tag{3.74}$$

$$\begin{bmatrix} x_{c_l} \\ y_{c_l} \end{bmatrix} = \begin{bmatrix} x_{c_j} - 2(x_{c_j} \% 2) x_d \\ y_{c_j} - 2(y_{c_j} \% 2) y_d \end{bmatrix} , \tag{3.75}$$

where $\%2$ is the remainder of dividing by two.

After determining what c_j is the previous 1-cell, the respective w_{ij} can be calculated by Equation 3.76 rather than Equation 3.70.

$$w_{ij} = \min\left(e^{\left(-\mathbf{v}_{c_j}(b)/\rho\right)}, w_{iprev} \right) , \tag{3.76}$$

where w_{iprev} is the affinity between the c_i and c_{prev} previous to c_j, with respect to SSF_{ij}, and
$c_{prev} \in \kappa_{r-1}$ while $c_j \in \kappa_r$.

After W and D sparse matrices are built, they are used to solve the respective generalized eigenvalue problem for a given number of eigenvalues. Each final eigenvector corresponds with a position in the value vector of a generalized eigenvector (Equation 3.77) Cartesian complex S_n. This process is calculated for a n radius at each c 1-cell in a Cartesian complex C ($c \in C$) in agreement with Equation 3.78. The octave (Eaton et al., 2017) underlying layer using the decimated procedure proposed in (Pont-Tuset et al., 2015) is used for this function.

$$\mathbf{v}_{ev} = \begin{bmatrix} ev_1 \\ \vdots \\ ev_m \end{bmatrix} , \tag{3.77}$$

where $ev_1, \ldots, ev_m$ is the value vector of the eigenvector Cartesian complex produced after solving the respective generalized eigenvalue problem for m smallest eigenvalues.

$$EV_n = \{ev | x_{ev} = x_c, y_{ev} = y_c \forall c \in C\} , \tag{3.78}$$

where $\mathbf{v}_{ev}$ as it is given by Equation 3.77 is the value vector of the ev 1-cell.

3.4.4 Boundary probability

Figure 3.26 displays the conceptual model for boundary probability that uses the concepts from virtually all the underlying layers. The main purpose, with the geometric and affinity (spectral) analysis discussed in Sections 3.4.2 and 3.4.3, is to serve as an instrument to establish the geometrical (*BoundaryOrientedMultiscaleGeometricPbCC* class) and spectral (*BoundaryOrientedSpectralPbCC* class) strength measurements (probability) that each 1-cell belongs to the boundary between two homogeneous segments of an image (Arbelaez et al., 2011).

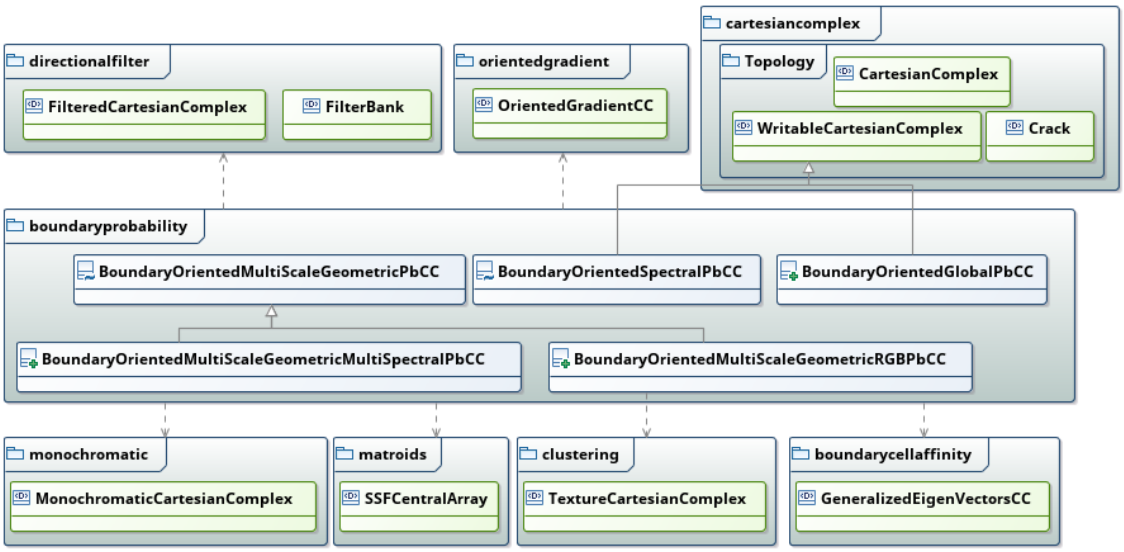


FIGURE 3.26: Conceptual elements for boundary probability

The first step is to create a *GS* gray scale Cartesian complex (*MonochromaticCartesianComplex* class) from an *I* multispectral Cartesian complex using the procedure proposed in (Valero, Dallos, and Lizarazo, 2018). This gray scale Cartesian complex is used to calculate the geometrical and spectral boundary strength measurements. In the following, first, a description of how this is performed for these two approaches is provided; then, the manner in which a unified measure of boundary strength (*BoundaryOrientedGlobalPbCC* class) is produced from these two is described.

Multiscale Boundary geometrical probability (MBGP)

After applying the procedure described in Section 3.4.1, each 1-cell c in the *GS* gray scale Cartesian complex obtains its value filtered by applying the oriented Gaussian filter banks for texture recognition (Leung and Malik, 1996). Each filter bank is composed of Gaussian derivative filter kernels for o orientations, s radii n_i, in 1-cells around the central 1-cell, at several granularity scales. Hence, the value vector $\mathbf{v}_{c_{K_{n_1\ldots n_s,1\ldots o}}}$ produced by applying a multiscale filter bank is the concatenation of each scale filter bank application; this is established by Equation 3.79.

$$\mathbf{v}_{c_{K_{n_1\ldots n_s,1\ldots o}}} = \mathbf{v}_{c_{K_{n_1,1\ldots o}}} \,||\, \mathbf{v}_{c_{K_{n_2,1\ldots o}}} \,||\, \ldots \,||\, \mathbf{v}_{c_{K_{n_s,1\ldots o}}} \,, \tag{3.79}$$

where $||$ implies vector concatenation.

The Cartesian complex $M_{n_1\ldots n_s,1\ldots o}$ produced by applying a multiscale filter bank, with o orientations and s scales, at each 1-cell in a Cartesian complex C ($c \in C$) is given by Equation 3.80.

$$M_{n_1\ldots n_s,1\ldots o} = \{c_{F_{n,1\ldots o}} | x_{c_{F_{n,1\ldots o}}} = x_c, y_{c_{F_{n,1\ldots o}}} = y_c \forall c \in C\} \,, \tag{3.80}$$

where $\mathbf{v}_{c_{K_{n_1\ldots n_s,1\ldots o}}}$ as it is given by Equation 3.79 is the value vector of the $c_{F_{n,1\ldots o}}$ 1-cell.

After multiscale Gaussian filter bank analysis, each 1-cell receives a response vector with as many values as the number of filter kernels in the multiscale filter bank. These response vectors are grouped using the k-means clustering technique proposed in (Valero, Dallos, and Lizarazo, 2018) to produce a T Cartesian complex of texture classes (*TextureCartesianComplex* class in Figure 3.26).

Multiscale oriented gradient Cartesian complexes I and T are calculated from both input and texture-produced Cartesian complexes using Equations 3.63 to 3.66 and 3.62, respectively. The oriented gradient Cartesian complexes calculated for each scale based on several texture radii $n_{tx_1}, n_{tx_2}, \ldots, n_{tx_s}$ and multispectral radii $n_{ms_1}, n_{ms_2}, \ldots, n_{ms_s}$ from the finest to the coarsest are linearly combined along with the texture multiscale gradient Cartesian complex to produce the oriented boundary geometrical probability using Equation 3.81.

$$\mathbf{v}_{gp} = \sum_{1\leq j\leq s} \beta_j \left(\gamma_{m+1} \begin{bmatrix} \mathbf{v}_{t_{g_{n_{tx_j}}}}(1) \\ \vdots \\ \mathbf{v}_{t_{g_{n_{tx_j}}}}(o) \end{bmatrix} + \sum_{1\leq l\leq m} \gamma_l \begin{bmatrix} \mathbf{v}_{i_{g_{n_{ms_j}}}}(1+(l-1)o) \\ \vdots \\ \mathbf{v}_{i_{g_{n_{ms_j}}}}(o+(l-1)o) \end{bmatrix} \right) \,, \tag{3.81}$$

where n_{tx_j} and n_{ms_j} are texture and multispectral radii at a j scale ($n_{tx_1} < \ldots < n_{tx_s}$ and $n_{ms_1} < \ldots < n_{ms_s}$),
l indexes multispectral bands,
$\mathbf{v}_{i_{g_{n_{ms_j}}}}$ is the value vector for $i_{g_{n_{ms_j}}}$ 1-cell in I multispectral gradient ($i_{g_{n_{ms_j}}} \in I$) at j scale,
$\mathbf{v}_{t_{g_{n_{tx_j}}}}$ is the value vector for $t_{g_{n_{tx_j}}}$ 1-cell in T texture gradient ($t_{g_{n_{tx_j}}} \in T$) at j scale,
γ_l is the weight of the relative contribution of each band,
(γ_{m+1}) is the weight of the relative contribution of the texture Cartesian complex, and
β is the weight of the contribution of each scale.

Equation 3.82 gives the Cartesian complex *ogPb* produced by calculating the oriented geometric probability for o orientations at each c 1-cell in a Cartesian complex.

$$ogPb = \{gp | x_{gp} = x_c, y_{gp} = y_c \forall c \in C\} , \tag{3.82}$$

where $\mathbf{v}_{gp}$ as it is given by Equation 3.81 is the value vector of a gp 1-cell.

Boundary spectral probability (BSP)

The maximum geometrical probability in an orientation is calculated in agreement with Equation 3.83.

$$\mathbf{v}_{gp_{max}} = \left[\max_j \{\mathbf{v}_{gp}(j)\} \right] . \tag{3.83}$$

Equation 3.84 provides the Cartesian complex *MP* produced by calculating the maximum geometrical probability from the *ogPb* Cartesian complex (Equation 3.82).

$$MP = \{gpmax | x_{gpmax} = x_{gp}, y_{gpmax} = y_{gp} \forall gp \in ogPb\} , \tag{3.84}$$

where $\mathbf{v}_{gpmax}$, as it is given by Equation 3.83 is the value vector at the $gpmax$ 1-cell, and *ogPb* is given by Equation 3.82.

The EV_n generalized eigenvector Cartesian complex (Equation 3.78), which is produced by the boundary cell affinity analysis described in Section 3.4.3 on *MP* Cartesian complex, is filtered with a filter bank of oriented first derivative Gaussian kernels. Then, the filtered eigenvector (given by Equations 3.52 and 3.53) are linearly combined by orientation as a boundary spectral probability measurement to produce the oriented affinity image.

$$\mathbf{v}_{sp} = \begin{bmatrix} \sum_{l=1}^{m} \frac{1}{\sqrt{\lambda_l}} \mathbf{v}_{ev_{K_{n,1\ldots o}}}(l) \\ \sum_{l=1}^{m} \frac{1}{\sqrt{\lambda_l}} \mathbf{v}_{ev_{K_{n,1\ldots o}}}(l+m) \\ \sum_{l=1}^{m} \frac{1}{\sqrt{\lambda_l}} \mathbf{v}_{ev_{K_{n,1\ldots o}}}(l+2m) \\ \vdots \\ \sum_{l=1}^{m} \frac{1}{\sqrt{\lambda_l}} \mathbf{v}_{ev_{K_{n,1\ldots o}}}(l+(o-1)m) \end{bmatrix} , \tag{3.85}$$

where l indexes m eigenvalues,
$\mathbf{v}_{ev_{K_{n,1\ldots o}}}$ is the value vector at $ev_{K_{n,1\ldots o}}$ 1-cell in the EV_n eigenvector Cartesian complex, and
$\frac{1}{\sqrt{\lambda_l}}$ is the weight of the relative contribution of the eigenvector motivated by the physical interpretation of the generalized eigenvalue (Belongie and Malik, 1998).

The Cartesian complex *osPb* produced by calculating the oriented geometric probability for o orientations at each ev 1-cell in a Cartesian complex is given by Equation 3.82.

$$osPb = \{sp | x_{sp} = x_c, y_{sp} = y_c \forall c \in C\} , \tag{3.86}$$

where $\mathbf{v}_{sp}$ as it is given by Equation 3.85 is the value vector of the sp 1-cell.

Boundary global probability (BGP)

The oriented geometrical and spectral probability Cartesian complexes are linearly combined according to orientation to produce the oriented boundary global probability using Equation 3.87.

$$\mathbf{v}_{bp} = \begin{bmatrix} \mathbf{v}_{gp}(1) \\ \vdots \\ \mathbf{v}_{gp}(o) \end{bmatrix} + \varphi \begin{bmatrix} \mathbf{v}_{sp}(1) \\ \vdots \\ \mathbf{v}_{sp}(o) \end{bmatrix} . \tag{3.87}$$

Equation 3.88 gives a Cartesian complex oPb produced by calculating the oriented boundary global probability for o orientations based on each pair of gp and sp 1-cells in the oriented boundary geometrical and spectral probability Cartesian complex.

$$oPb = \{pb | x_{pb} = x_{gp}, y_{pb} = y_{gp} \forall gp \in ogPb\} , \tag{3.88}$$

where $\mathbf{v}_{bp}$ as it is given by Equation 3.87 is the value vector at bp 1-cell.

The global boundary probability is produced from the maximum oriented boundary probability in some direction in agreement with Equation 3.89.

$$\mathbf{v}_{gbp} = \left[\max_{l} \{\mathbf{v}_{bp}(l)\} \right] . \tag{3.89}$$

Equation 3.90 gives the Cartesian complex BP produced by calculating the maximum oriented boundary global probability for o orientations based on each pair of gp and sp 1-cells in the oriented boundary geometrical and spectral probability Cartesian complex.

$$BP = \{gbp | x_{gbp} = x_{bp}, y_{gbp} = y_{bp} \forall bp \in oPb\} , \tag{3.90}$$

where $\mathbf{v}_{gbp}$ as it is given by Equation 3.89 is the value vector at gpb 1-cell.

3.4.5 Watershed transform

The maximum of the boundary oriented global probability Cartesian complex BP (Equation 3.90) is used as the input to the watershed transform implementation to produce the final superpixels or image-objects. The Vincent-Soille algorithm is implemented (Vincent and Soille, 1991), however on the 1-cells (Valero, Dallos, and Lizarazo, 2018). For exploring an eventual basin expansion, the 1-cells in the union of the SON of elements in the central 1-cell closure (Equation 3.91) are processed, except for the central 1-cell (Figure 3.27), rather than a 4- or 8-connected conventional pixel neighborhood.

$$\nu(c) = \bigcup_{c_i \in Cl(\{c\})} SON(c_i) , \tag{3.91}$$

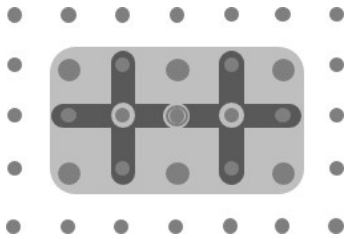


FIGURE 3.27: Cracks (*dots in darker area*) in the union of SON (*clearer area*) of elements in central 1-cell closure (*all circled dots*), except the central 1-cell, used in exploring basin expansion from the central 1-cell (*central circled dot*).

where $Cl(\{c\})$ is the $\{c\}$ set closure (Section 2.1.2) with T as the complete Cartesian complex, c_i is a 1-cell, and c is the central 1-cell.

The watershed transform output can be the over-segmentation image or a multilevel segmentation with a given number of levels. A multilevel segmentation is obtained from a given number of l of levels, each of them with $2^{i+1}+1$ slots, where i is between 1 and l. The range of probability values present in the maximum of the boundary oriented global probability Cartesian complex is explored to distribute, until $2^{i+1}+1$, evenly spaced slots if there are sufficient values for the l levels; otherwise, only the number of available levels is returned.

Chapter 4

Computational framework based on ALFS and oriented matroid

A computational framework was built based on the conceptual model established in Chapter 3. Figure 4.1 displays the defined package hierarchy that allows the demonstration of the applicability of Cartesian complex space representation and central SSF arrangements linked with oriented matroids.

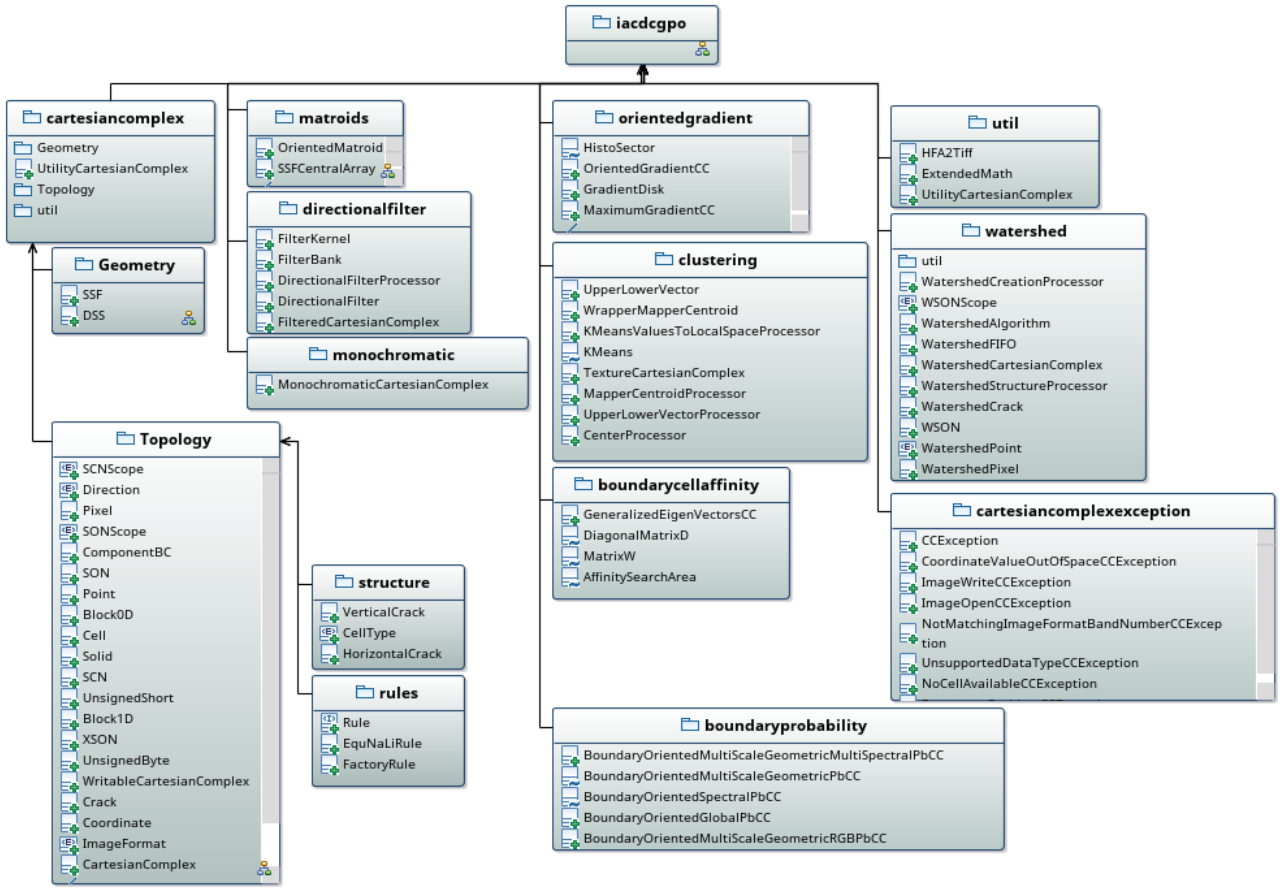


FIGURE 4.1: Package hierarchy of conceptual elements

The Cartesian complex package (*cartesiancomplex*) supports both the representation model based on the Cartesian complexes and the topological and geometric concepts built upon this. The matroids package (*matroids*) supports the geometric to matroid transformation required in the segmentation tasks. The additional packages provide support for each of these tasks, in addition to their conceptual elements, of the segmentation process. The clustering package (*clustering*) provides support for k-means clustering (Valero, Dallos, and Lizarazo, 2018), which is required after filtering in the texture analysis and spectral probability tasks. The watershed package (*watershed*) provides support for the hierarchical watershed transform (Valero, Dallos, and Lizarazo, 2018). Support is also provided to the management of exceptions (*cartesiancomplexexception*) and utilitarian tasks (e.g., format conversion) (*util*).

Every algorithm has been defined in agreement with the Java Fork/Join framework for parallel computing. In the following, each package is described in agreement with the layer arrangement illustrated in Figure 3.1 from the Cartesian complex topology, the lowest, to segmentation task layer, the uppermost. All diagrams are presented in terms of an oriented object paradigm using UML (Unified Modelling Language). The implementation code documentation for implemented functions can be found in Appendix A.

4.1 Cartesian complex

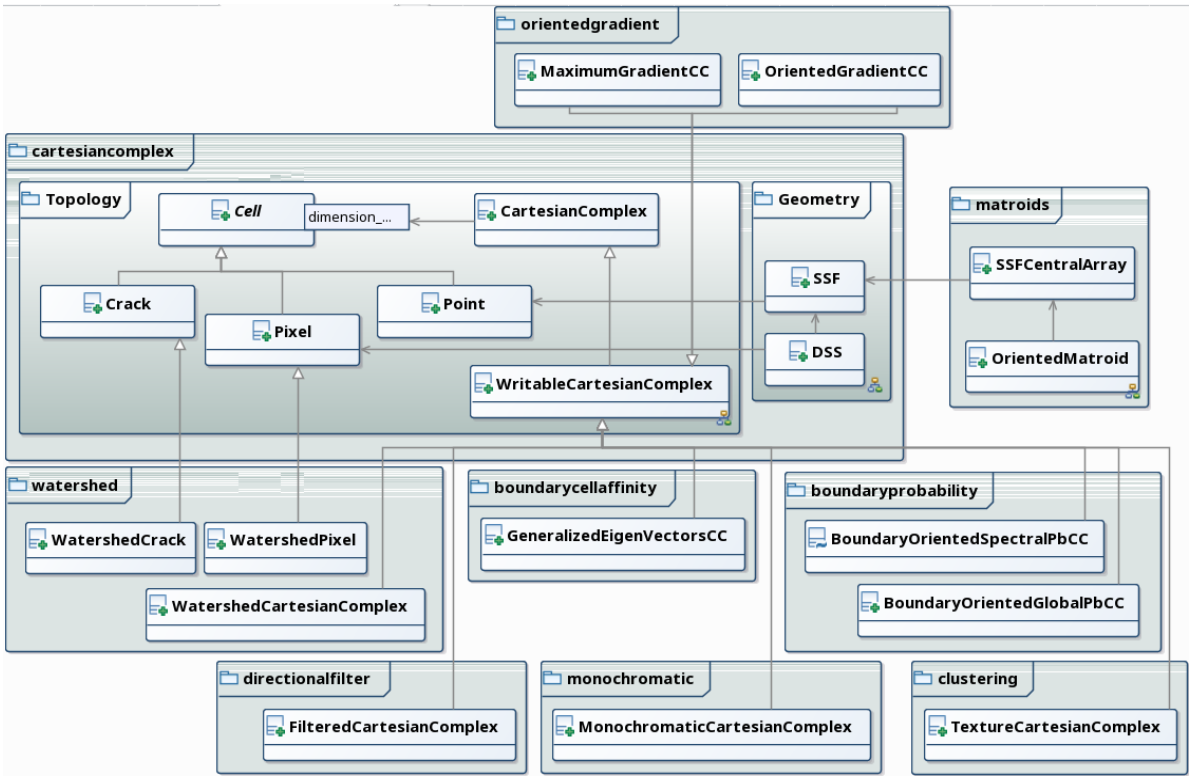


FIGURE 4.2: Cartesian complex package elements

As Figure 4.2 indicates, the package for the conceptual elements supporting a Cartesian complex provides the other packages with the elements and functionalities required according to the conceptual model

presented in Chapter 3, through its two sub-packages for Cartesian complex topology and geometry that are described in the following. The figure displays only the cell (*Cell* class), Cartesian complex (*CartesianComplex*, and *WritableCartesianComplex* classes) core concepts, which provide support to the core concepts in the other packages.

4.1.1 Topology

Conceptual topology elements are displayed in Figure 3.3. The cell is the core concept (*Cell* class) whose structure is displayed in Figure 4.3. A cell is located at a combinatorial coordinate (*Coordinate* class) that is tested for validity in agreement with Equation 3.2 (*checkCellTYpe*). A cell value is a vector of values that can be of any number type supported by the *GDAL* underlying layer. Each cell has two neighborhoods: (1) a smallest open neighborhood, defined based on Equation 3.3 (*SON* class), and (2) extended smallest open neighborhood, based on Equation 3.4 (*XSON* class). Both are defined in a combinatorial manner from *x* and *y* axes intervals (*fromRow*, *toRow*, *fromColumn*, and *toColumn*) depending on the cell type (*Point*, *Crack*, or *Pixel*) (Kovalevsky, 2008). A cell neighborhood takes its cells from a Cartesian complex based on a local positioning, which is translated to the respective Cartesian complex in a relative manner to the central cell coordinate. With membership purposes, SON and XSON can calculate, respectively, the highest and narrow strip values (Kovalevsky, 1989).

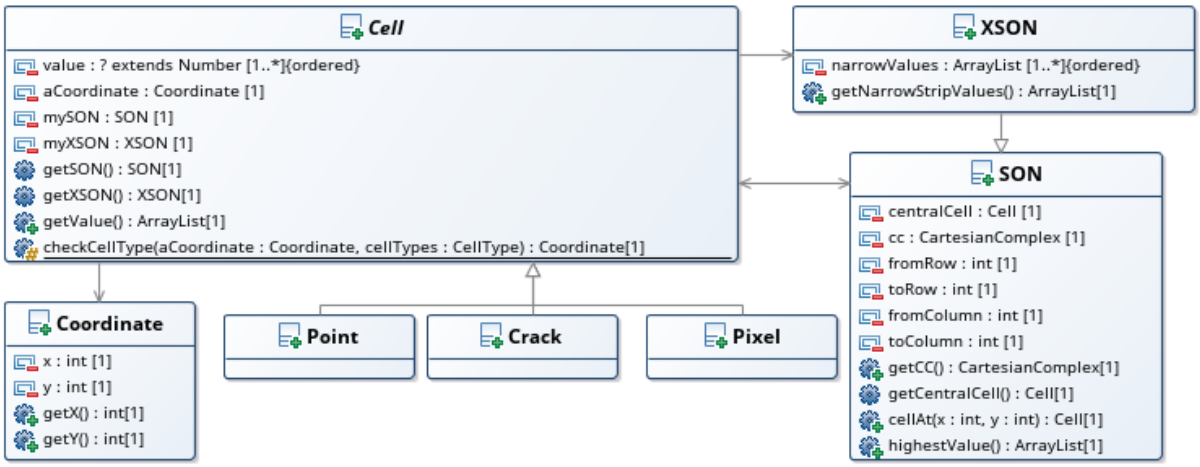


FIGURE 4.3: Cell structure

A 2D Cartesian complex (*CartesianComplex* class) as displayed in Figure 4.4 supports the finite set of cells corresponding to an image. A finite Cartesian complex space can be created from an image based on the facilities provided by the GDAL underlying layer (*Dataset* class).

An image can correspond to a conventional pixel one or to a combination of different types of Cartesian complex space elements, which is indicated by an image format parameter (*ImageFormat* enumeration). When a combination of more than one type of Cartesian complex space element band is the case, they are assumed to be arranged one alongside the other and the respective order to be from low to high dimensionality. There must be one band for each type of Cartesian complex space element indicated by

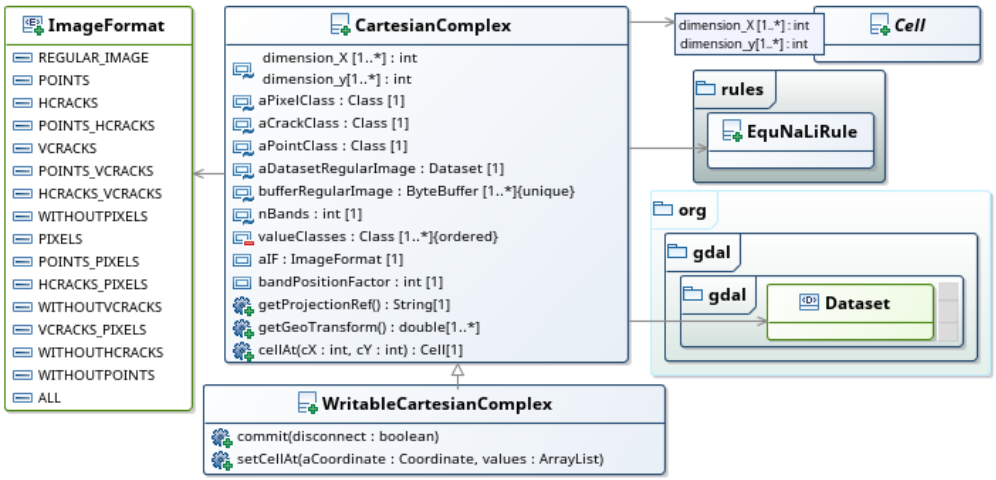


FIGURE 4.4: Cartesian complex structure

the image format parameter, i.e., the band number in the dataset (*Dataset* class) must be a multiple of the number of space element types in the combination; if this is not correct, a NotMatchingImageFormat-BandNumberCCException exception is thrown. When a GDAL dataset is open, on each read operation the respective data are placed in an intermediate buffer (ByteBuffer) for each band in the dataset. Hence, the number of bands divided by the type of space elements indicates the number of values in a value vector for any cell.

When a cell is to be retrieved, if it had not been retrieved previously, it is created with the respective combinatorial coordinate and its value vector is filled with the data in the respective position of the byte buffer. Each created cell is placed in a 2D array whose positions are indexed based on the cell's combinatorial coordinate. The data type is obtained from the metadata GDAL dataset and placed in the value class vector linked with the Cartesian complex. The projection reference and geometric transforms parameters are also retrieved from the GDAL dataset.

When it is necessary to create a new Cartesian complex, as is the case for all packages (except the matroids package) displayed in Figure 4.2, a writable Cartesian complex (*WritableCartesianComplex* class) allows writing the data in the byte buffer onto disk through GDAL, considering the creation of a GDAL data layer for each Cartesian complex space element type, in agreement with the established space element combination (*ImageFormat*).

When the dataset corresponds to a conventional image (*REGULAR_IMAGE* or *PIXELS* image format), inter-pixel elements are produced applying the membership rule (*EquNaLiRule* class) to establish its value vector (Valero, Dallos, and Lizarazo, 2018). If the GDAL underlying layer reports any problem opening an image, an ImageOpenCCException exception is thrown. Only GDT_Byte, GDT_Float32, GDT_Float64, GDT_Int16, and GDT_Int32 GDAL data types are now supported; hence, if any other type is used an UnsupportedDataTypeCCException is thrown.

4.1.2 Geometry

As Figure 4.5 indicates, the geometry sub-package contains two supporting elements: the SSF (*SSF* class) and digital straight segment (*DSS* class). An SSF is defined based on a starting point (*Point* class), whose concept is taken from the topology package, and a, b, and l parameters. The SSF length in number of 1-cells (nc) is also retained.

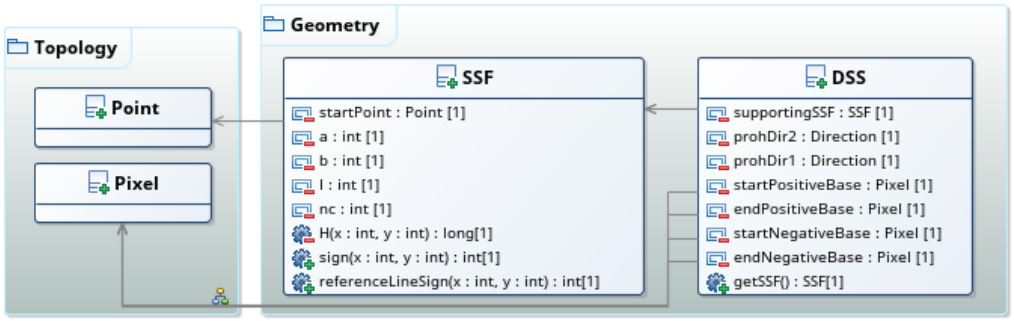


FIGURE 4.5: Geometry elements structure

The δ sign value as defined by Equation 3.12 at an (x,y) Cartesian complex location is provided by $(sig(x,y))$ and produced based on Equation 3.7 $(refrenceLineSign(x,y))$. The DSS is supported with virtually all features defined in (Kovalevsky, 2008) for tracing purposes, including start and end positive and negative bases (pixels).

4.2 Matroid package

The Matroid package provides support for central SSF arrangement (*SSFCentralArray* class) and its linked oriented matroid (*OrientedMatroid* class) concepts. As can be observed in Figure 4.6, an SSF-CentralArray is defined based on a given radius and number of orientations. The number of orientations must be less than or equal to the maximum established by Equation 3.15 to not trigger an exception condition.

As many SSFs as the number of orientations are built following the construction process described in Section 3.3.1 taking a central 1-cell (*HORIZONTAL_CRACK* or *VERTICAL_CRACK*) as the relative origin shifted by one to meet the combinatorial coordinate-based cell type definition (Equation 3.2) and crossing an SSF by it and by each external (outer) 1-cell. When an SSFCentralArray is built, it can be reused as many times as necessary, i.e., two SSFCentralArrays (one horizontal 1-cell centered and the other vertical 1-cell centered) can be used to process all cells in one or more Cartesian complexes.

An oriented matroid (*OrientedMatroid* class) is defined based on a central arrangement of SSFs from which all necessary elements are constructed to transfer the geometric characteristics of the central arrangement to the oriented matroid. For each space element in the Manhattan ball linked with the central SSF arrangement, the (x,y) to subspace mapping is calculated and stored as part of the oriented matroid

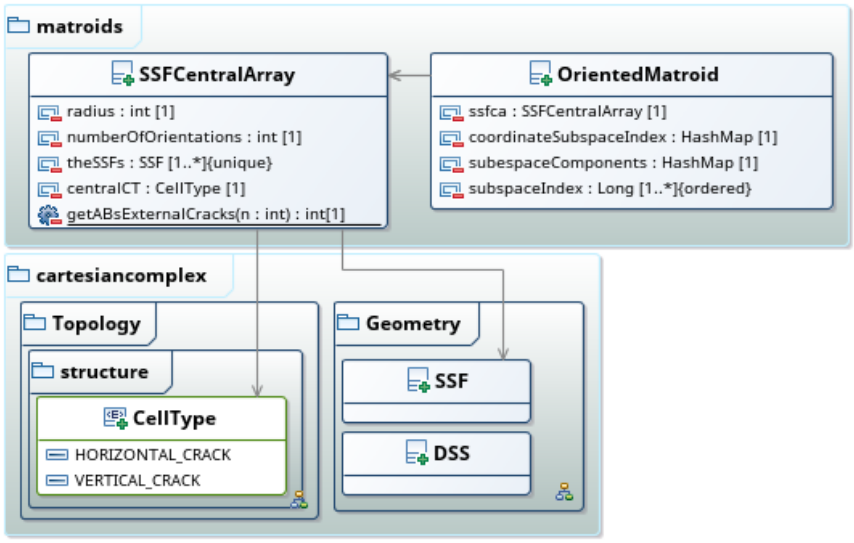


FIGURE 4.6: Matroids package elements

(*coordinateSubespaceIndex*), applying Equations 3.27 to 3.33 for each SSF in the central SSF arrangement and finally Equation 3.36. When a subspace is shared, the SSFs sharing it are updated in agreement with Equation 3.40 (*subspaceComponents*). When a subspace is not shared, the matroid order position to subspace mapping is calculated, and stored as part of the oriented matroid (*subespaceIndex*), using Equations 3.37 and 3.38.

This calculation is performed only once when an oriented matroid is built; subsequently, an oriented matroid can be used as many times as necessary. Al least two oriented matroids must be built, one based on a horizontal 1-cell centered SSFCentralArray and the other based on a vertical 1-cell centered SSFCentralArray.

4.3 Segmentation task layer (STL)

The segmentation task layer provides support for all conceptual elements defined and described in Section 3.4. Virtually all these concepts relate directly with the steps that must be executed to perform the segmentation of a multispectral image represented as a Cartesian complex. To use all the available computing power, a parallel computing architecture was established and used whenever it was possible to perform the tasks simultaneously. In the following, first a description of the parallelism scheme used and then, of each of the packets of the STL layer is presented.

4.3.1 Parallelism scheme

Remembering that the number of elements of the space on which it is necessary to perform the different tasks that are part of the segmentation process, is considerably large and that in the majority of the

cases several computer processing units (CPU) are available, a simple subdivision scheme of a task was established considering these two variables.

For a Cartesian complex consisting on $r \times c$ 2-cells and p CPUs, 2-cells were grouped in a row two-dimensional ordering fashion, i.e., left to right first and then top to bottom (Figure 4.7). There are up to two group sizes given by equations 4.1 and 4.2.

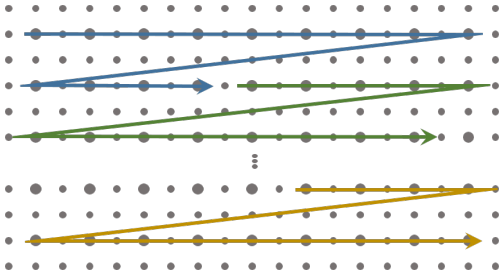


FIGURE 4.7: Space element grouping by CPU

$$g_1 = \frac{(r \times c) - ((r \times c)\%p)}{p} , \tag{4.1}$$

where $\%p$ is the remainder of dividing by p.

$$g_2 = g_1 + 1 . \tag{4.2}$$

Hence, as many as $p - ((r \times c)\%p)$ CPUs can have g_1 2-cells assigned and as many as $(r \times c)\%p$ CPUs can have g_2 2-cells. Once the groups are built and assigned to the available CPUs, the respective task can be simultaneously performed by p CPUs. For each 2-cell, its respective own inter-pixel 1-cells must be accessed as required (Kovalevsky, 2008). This parallelism scheme can be used only if a 2-cell's own space element outcome is independent of the outcome of the others. Specifically, the watershed transform, k-means, and gray scale image calculation can use this scheme only in parts of their processes (Valero, Dallos, and Lizarazo, 2018).

4.3.2 Monochromatic package

Figure 4.8 displays the elements included in the monochromatic package and used to produce a gray scale Cartesian complex from a multispectral Cartesian complex. To obtain a single gray scale image from the multispectral Cartesian complex image, a linear combination of all bands is computed based on the level of information provided (*WeightsCalculator* class) by each band using the covariance matrix (Tso and Mather, 2009).

A weight calculation is performed using the parallelism scheme described in Section 4.3.1. In the first round, after calculating a per group average vector, each CPU outcome is consolidated with the others (*VectorCalculatorAverageProcessor* class). In a second round, the covariance matrix is calculated in a similar manner (*VarianceMatrixProcessor* class). Once the weights are calculated, they are used

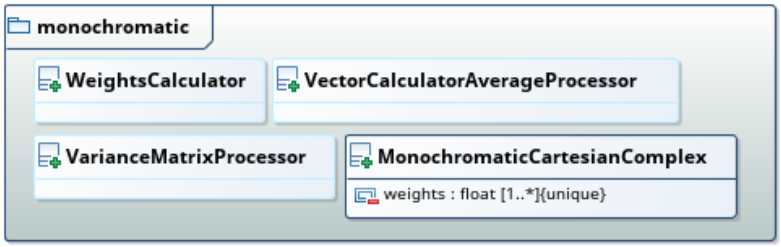


FIGURE 4.8: Utilitarian package structure

for creating the gray scale Cartesian complex (*MonochromaticCartesianComplex* class), also using the established parallelism scheme.

4.3.3 Directional filtering package

The core element of the directional filter package structure is the filter kernel (*FilterKernel* class). As can be observed in Figure 4.9, it provides support to three types of filters: the Gaussian derivative filters (*FilterBank* class), directional Prewitt filters (*DirectionalFilter* class), and filters for pre-designed generic kernels.

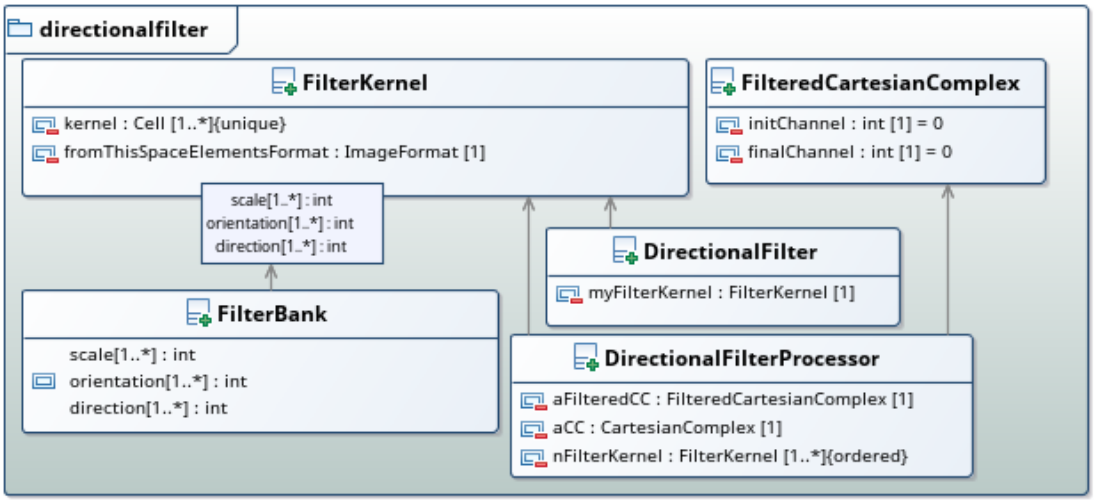


FIGURE 4.9: Filtering package structure

In agreement with the definition given in Section 3.4.1, a filter kernel is considered as a small neighborhood of cells; however, it can have several layouts. Hence, independent of the layout, each of the cells is placed in any order in a 1D array (*kernel*) and then, when required, it is linearly obtained from there and processed because each cell contains the information of its (x,y) location relative to a central cell. Although the default cell type of cells in a kernel is *Pixel* (2-cell), the cell type (*fromThisSpaceElementFormat*) can be established when a filter kernel is created. At creation time, support is given for building Gaussian and Prewitt filters based on a radius (in number of 1-cells) and an orientation (θ angle or an

SSF index in an central SSF arrangement) and pre-designed kernel (as a 2D array). Always, there are at least two filter kernels in a bank: one horizontal 1-cell centered and the other one vertical 1-cell centered.

Both Gaussian filter bank (*FilterBank* class) and Prewitt directional filter (*DirectionalFilter* class) produce filter banks creating an central SSF arrangement for a radius (in number of 1-cells) and a number of orientations; however, in the case of the Gaussian, it can be performed in a multiscale manner (i.e., using several radii and numbers of orientation pairs). When a Gaussian filter bank is created, each SSF in the central arrangement is used to establish the θ angle whose value is used by the filter kernel constructor to rotate the respective kernel. When a bank of Prewitt filters is created, the cell sign vector of the linked matroid regarding each SSF is used to generate the respective kernel.

After building a filter bank, this is applied at each input Cartesian complex cell (*DirectionalFilterProcessor* class), between two given channels (i.e., cell value vector positions), to produce a filtered Cartesian complex (*FilteredCartesianComplex* class). The filter bank is applied using the parallelism scheme described in Section 4.3.1, i.e., the respective filter Kernel set is created only once and shared by all available CPUs. Hence, all input Cartesian complex cells are processed with the same filter kernel set.

4.3.4 Oriented gradient Cartesian complex package

The oriented gradient calculation from an input Cartesian complex is based on the procedure established in Section 3.4.2. The elements used are displayed in Figure 4.10. The process begins building the necessary combinatorial structures based on two central SSF arrangements (one horizontal 1-cell centered and the other vertical 1-cell centered) for a radius (in number of 1-cells) and number of orientations. The central arrangements are used to build two oriented matroids (Section 4.2), each linked to a ball (*GradientDisk* class) defining the calculation neighborhood around a central cell.

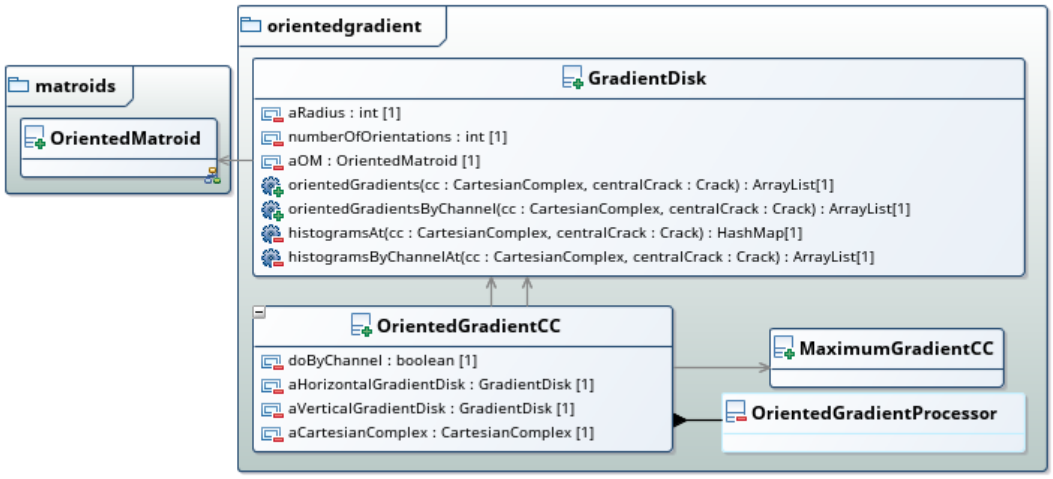


FIGURE 4.10: Oriented gradient package structure

The gradient disks calculated are used to calculate first, the oriented histogram and then, the respective oriented gradient at each 1-cell of an input Cartesian complex. Support is provided to calculate both oriented histograms and gradients based on the central 1-cell value vector as a whole or individually for

each of its channels (i.e., positions). Only the same gradient disk pair is used to produce all 1-cells of the output oriented gradient Cartesian complex sharing them among the available CPUs using the parallelism scheme described in Section 4.3.1 (*OrientedGradientProcessor* class). The maximum oriented gradient is supported (OrientedGradient class) using Equations 3.67 and 3.68

4.3.5 Boundary cell affinity package

Affinity analysis involves calculating the diagonal scattered matrices D and W according to the area of calculation for a given radius (Arbelaez et al., 2011). Figure 4.11 displays the elements used to support the affinity procedure as established in Section 3.4.3.

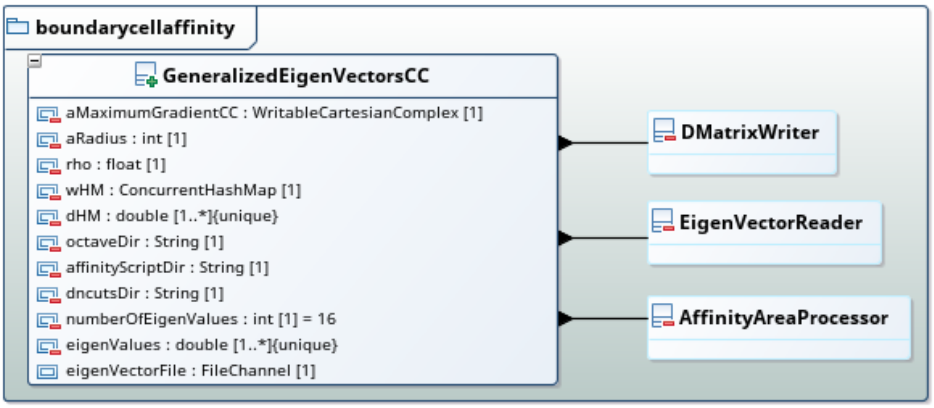


FIGURE 4.11: Affinity analysis package structure

The analysis is performed between each cell pair, which means that even though the W matrix is sparse, the number of positions that must be indexed overflows the indexing capabilities not only of the Java execution environment (JRE) but also the software tools based on 32-bit indexing such as Matlab® and Octave (open source). Hence, although the process of W and D matrix building is performed using the parallelism scheme described in Section 4.3.1, to solve the generalized eigenvalue problem, a customized *Octave* implementationit had to be compiled to support 64-bits indexing.

While W and D are being calculated (*AffinityAreaProcessor* class), W is written to an exchange mat file (i.e., a version "4" Matlab® file). Once the W is calculated and written to disk, the D matrix is also written to the same exchange file using the established parallelism scheme (*DMatrixWriter* class). Then, the subsequent process of solving the $(D-W)v = \lambda Dv$ system (Martin, Fowlkes, and Malik, 2004), for a given number of eigenvalues, is executed in the previously customized *Octave* software. When this process terminates, the set of eigenvalues and respective eigenvectors are written to another exchange file and control is returned. While the eigenvalues/eigenvector exchange file is read using the parallelism scheme (*EigenVectorReader* class), the generalized eigenvectors Cartesian complex (*GeneralizedEigen-VectorsCC* class) is being built.

4.3.6 Boundary probability package

This package gathers the necessary elements to provide support for the calculation of the boundary strength (i.e., probability) whose structure is displayed in Figure 4.12.

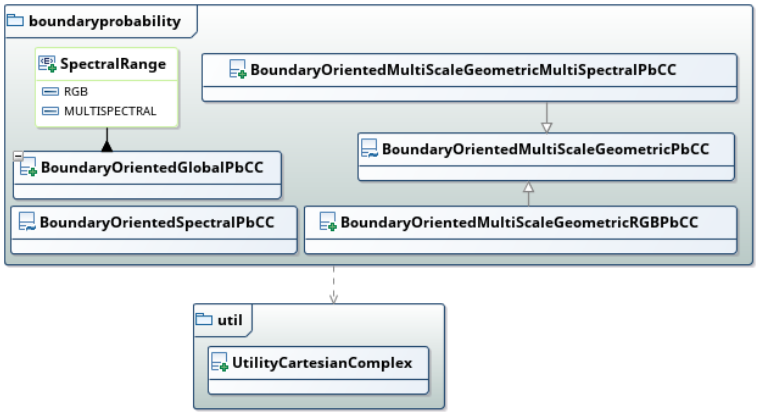


FIGURE 4.12: Boundary probability package structure

The calculation of boundary strength measurements established by Equations 3.81, 3.85, 3.87, and 3.89 is provided using the parallelism scheme described in Section 4.3.1 through the utilitarian Cartesian complex from utility package (Section 4.3.9).

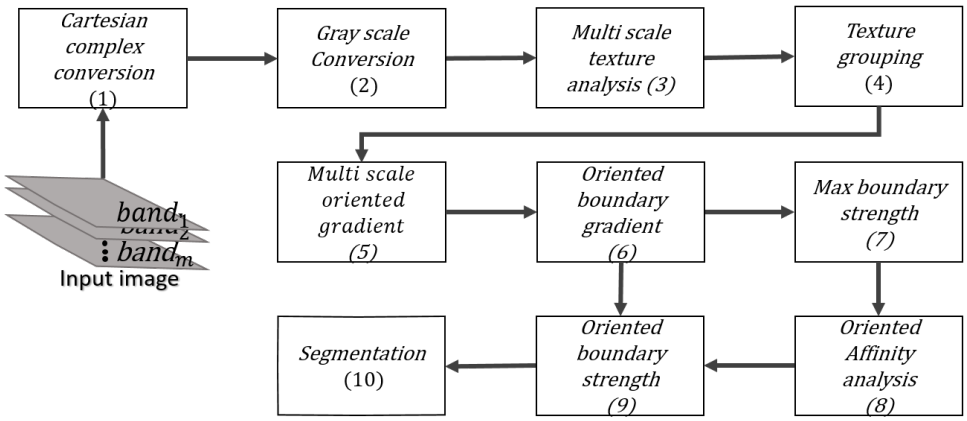


FIGURE 4.13: Boundary probability process

Figure 4.13 indicates the steps performed to calculate the boundary probability measurement. The process begins by converting from conventional image representation to Cartesian complex (Section 4.1) (1) and after, obtaining a gray scale Cartesian complex (Section 4.3.2) (2). Then, a multiscale (using several radii an orientation pairs) Gaussian filter bank (3) is applied on the gray scale Cartesian complex to produce a filtered Cartesian complex whose cells are clustered using k-means (Section 4.3.7) considering each cluster as a texture class (4). Then, on both the input multispectral Cartesian complex (transformed into CIE Lab color space (Szeliski, 2010) based on the procedure proposed in (Arbelaez et al., 2011) when it is an RGB image) and the produced texture Cartesian complex are calculated multiscale oriented gradients (5),

which are linearly combined using Equation 3.81 to obtain the boundary oriented geometrical probability Cartesian complex (6).

Then, Equation 3.89 is used to produce a global boundary geometrical probability Cartesian complex (7) that is used as input to the affinity analysis to produce a generalized eigenvector Cartesian complex (8). Then, an oriented first derivative Gaussian filter bank is applied to each eigenvector (i.e., position in value vector of eigenvector Cartesian complex cells) whose outcome is linearly combined by orientation along with the oriented boundary geometrical probability (obtained in step 6) using Equations 3.85 and 3.87 to produce the boundary oriented global probability (9). Finally, the boundary global probability Cartesian complex (produced using 3.89) is used as input to the segmentation by the watershed transform (10) (Section 4.3.8).

4.3.7 Clustering package

As the final step in texture analysis, the clustering approach proposed in (Lloyd, 1982) is supported using the parallelism scheme described in Section 4.3.1. The main clustering package elements are displayed in Figure 4.14.

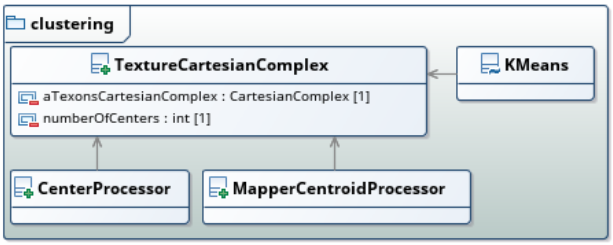


FIGURE 4.14: Clustering package structure

In a training stage, the clustering (*KMeans* class) starts with k cluster centers, whose location vectors can be generated either by uniformly distributing them along the greater diagonal of the minimum parallelepiped that contains the complete set of hyper-dimensional points being clustered, or by selecting them randomly from the same set of hyper-dimensional points that are going to be grouped. Then, using the established parallelism scheme iteratively, first, the center assignment (*MapperCentroidProcessor* class) and then, new center calculation (*CenterProcessor* class) tasks are performed until convergence or until a percentage of no change is obtained. In a test stage, cluster centers previously calculated (*KMeans* class) are used for assigning (*MapperCentroidProcessor* class) the Cartesian complex cells to them.

4.3.8 Watershed package

The watershed transform procedure established in Section 3.4.5 is supported through the element set displayed in Figure 4.15. The segmentation process ends with a watershed transform whose outcome can be an over-segmentation or a multilevel segmentation (Valero, Dallos, and Lizarazo, 2018)(*WatershedCartesianComplex* class). The algorithmic approach proposed in (Vincent and Soille, 1991) is implemented (WatershedAlgorithm class); however, using only 1-cells (*WatershedCarck* class) to test the eventual watershed expansions using Equation 3.91 (*WSON* class).

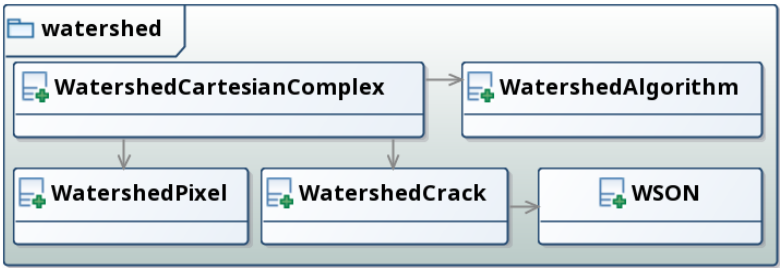


FIGURE 4.15: Watershed package structure

4.3.9 Utility package

A set of general tasks that are required to support the others are grouped as general purpose tools in a utilitarian package (*util*). Figure 4.16 displays the tool set provided by this package.

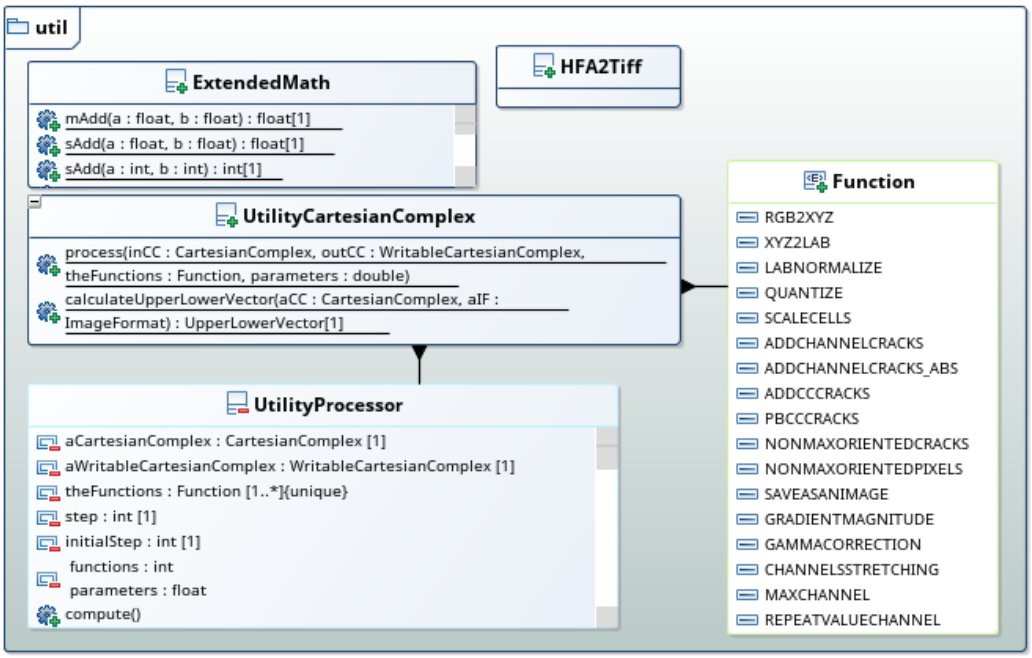


FIGURE 4.16: Utilitarian package structure

To avoid having to import an entire library, some vector and matrix operations (e.g., matrix addition and product) not available in the programming environment are provided (*ExtendedMath* class) together with an utility to change a Cartesian complex to a "tiff" format for Matlab® software portability (*HFA2Tiff* class).

A basic support for map algebra between Cartesian complexes is provided (*UtilityCartesianComplex* class) using the parallelism scheme described in Section 4.3.1. Several functions (*Function* enumeration) can be simultaneously performed at each cell of an input Cartesian complex and its outcome is placed at the same location on an output writable Cartesian complex.

4.4 Discussion of the computational framework implementation

This section discusses aspects of the prototype computational implementation with the objective of providing a clearer vision of how it was performed and a greater certainty of what can (or cannot) be expected from it. Hopefully, this can serve as a guide for what should be addressed in future implementations. This discussion addresses three general aspects: (1) representation of a multispectral image as a Cartesian complex, (2) link between a central arrangement of standard separating forms and their associated oriented matroid, and (3) multispectral image segmentation tasks based on the 1-cells of a Cartesian complex.

4.4.1 Representation of a multispectral image as a Cartesian complex

The use of the GDAL library simplified the process of accessing the data sets of a multispectral image, reducing it to allow the use of the provided API. Owing to this, it was not necessary to perform a computational development for this task. However, whereas using GDAL, support was available to access the wide range of storage formats supported by this tool, it is also true that its inclusion as input data structures for the different processes was not transparent. The main problem was that some of the data types of the images supported by GDAL were not directly supported in the Java programming language that was used for the implementation. This difficulty was overcome with the use of Java wrapper classes for the types that were not supported, e.g., unsigned bytes and integers. This enabled access to this type of data, which, once loaded, was converted to float type and adjusted to what was required by the different tasks to be performed. This strategy simplified the implementation process because no additional support was necessary for the wrapper classes introduced. However, if the computational implementation obtained as part of the research presented here is used in the future for tasks that require data types not directly recognized by Java, it would be necessary to perform the computational implementation of the additional support required in the wrapper classes included.

The functionality provided by GDAL was also used to support the storage of the different types of elements of the Cartesian complex space. As described in Section 3.1, each element type in the complex Cartesian space, (i.e., 0-cell, horizontal 1-cell, vertical 1-cell, and 2-cell), was represented as an additional band of the multispectral image supported by GDAL. As mentioned in Section 3.1, the 4 types of space elements were not always required; in fact, during the research presented herein, only 2-cells or 1-cells were used in the Cartesian complexes. The conversion between an element type of the Cartesian complex space and the respective band of the underlying multispectral image in the GDAL layer was performed by the current implementation of the *CartesianComplex* class when reading from or writing to the multispectral image maintained in disk storage through GDAL. Therefore, the possible future use of any combination of element types of the Cartesian complex space that could be required could be provided transparently, only specifying the combination of element types required at the time of the Cartesian complex creation.

In addition to support for the 4 types of elements in a Cartesian complex, the future use of any object that is an instance of a class found in the inheritance hierarchy whose common root corresponds to any of them is also possible. The only action required is to specify for each of the basic element classes, (i.e., Point for 0-cell, Crack for 1-cell, and Pixel for 2-cell), the required subclass, providing the respective instance of the *Class* class at the Cartesian complex creation time. This conceptual extension scheme was used during the research presented herein for the implementation of the watershed transform using the *WatershedCrack*

subclass rather than the *Crack* base class. However, it should be considered that any data that is part of an instance of the subclass used must be transferred to the respective value vector of each cell for the purpose of being stored on disk by the Cartesian complex through GDAL. Further, to consider performance, the current computational implementation uses a primitive array of cells to support the Cartesian complex space representation with the premise that the supported parallelism scheme (Section 4.3.1) is only used when the processing at a cell does not depend on the processing output at any other. However, any future use must consider that there is no guarantee of consistency when performing concurrent access to this type of data structure.

Cell block complexes (Section 2.1.8) were not implemented as they were not required during the research presented herein; hence, only the concepts of blocks of 0-cells (Block0D), 1-cells (Block1D), and components of the block complex (ComponentBC) were included with the basic support. Therefore, any future use of these must address the additional computational implementation that could become necessary.

4.4.2 Link between an central SSF arrangement and its associated oriented matroid

Regarding the concept of digital straight line segment (DSS), the current prototype computational implementation provides only the basic support necessary for the implementation of the concept of SSF, which is the core concept in establishing the link between the geometry present in a central arrangement of SSFs and the combinatorics of its associated oriented matroid. Therefore, any future implementation that uses the DSS concept must address the implementation of the additional support that could be required and consider that in the current implementation, in addition to the underlying SSF, the concepts associated with the positive and negative bases of a DSS have been included without support.

As mentioned in Section 3.3.2, the determination of one of the subsets where a central arrangement of SSFs divides the Cartesian complex uses Equation 3.36. Because this equation is based on an exponential of the number of SSFs, to the extent that the number of orientations is increasing, the value obtained becomes extremely large. Therefore, any future use should consider that when using sufficiently large orientations numbers (> 38), it is possible that memory overflow could arise; hence, this problem should be addressed with an implementation alternative to avoid this occurrence.

4.4.3 Multispectral image segmentation tasks based on 1-cells

As mentioned in Section 4.3.7, the image segmentation process described herein includes the implementation of the k-means algorithm, which is used to define the texture classes after the filtering phase. This implementation provides two mechanisms for obtaining the initial class centers: (1) by distributing them along the diagonal between the minimum and maximum vertices of the minimum bounding box and (2) obtaining them in a random manner from the set of points to be grouped. However, in the first case, to the extent that the set of points has an orthogonal grouping pattern to the diagonal, the number of classes obtained decreases, becoming considerably less than the number of classes indicated at the beginning of the grouping. In the second case, randomness is introduced that makes the set of classes obtained for the same set of points vary each time the grouping is performed. In either case, the final segmentation obtained depends, to some extent, on these non-controllable factors. Therefore, a research should be conducted with the purpose of determining a better strategy.

As described in Section 4.3.4, the implementation of the oriented gradient calculation task takes as a basis the oriented matroid associated with the central SSF arrangement using Equations 3.37 and 3.38 to transfer the geometry of the central arrangement to the combinatoric of the associated matroid. In this process, it is necessary to establish to which subspace each cell of the space belongs; to achieve this, Equation 3.7, rather than Equation 3.12, is used. The reason for this is that to avoid any bias, the data of the 1-cells, which belong to the SSF used as orientation, are not included in the calculation of the respective oriented gradient. When using Equation 3.12, the number of cells that are disregarded is noticeably greater than those not considered when using Equation 3.7. This effect becomes more noticeable to the extent that the Manhattan distance-based ball size that defines the oriented gradient calculation area becomes smaller. A future implementation could use either of the two that it deems most appropriate for its particular purposes.

The segmentation process established in (Arbelaez et al., 2011) provides a task for adjusting cylindrical parabolas to an elliptical patch to counteract the phantom edge effect produced by the oriented gradient calculation. Although the prototype computational implementation performed as part of the research presented herein provides support for this task, the results were not satisfactory and therefore it was not included in the segmentation process. A future computational implementation should provide adequate support.

With respect to the affinity analysis task (Section 4.3.5), in the current prototype computational implementation it was impossible to directly provide support because the affinity matrix overflowed the maximum 32-bit indexing capacity provided by the Java language. Therefore, it was necessary to use an external open source platform that did not present this limitation and that could provide support to solve the problem of generalized eigenvalues. The open source tool *Octave* was the selected platform because it was necessary to have the source code and perform a recompilation of the entire code to support 64-bit indexing. This means that to execute the segmentation process with the current prototype computational implementation, it is necessary to have, in addition to the Java virtual machine, a customized version of the *Octave* software that supports 64-bit indexing.

Chapter 5

Results and Validation of proposed model

This chapter presents the results obtained from the computational implementation of the proposed digital image segmentation process (Chapter 3), which was carried out as part of the present investigation and described in Chapter 4. The results are presented by first showing examples of the outcomes from the execution of each of the tasks of the segmentation process (Figure 4.13) and then the outputs of the proposed model validation tests. The choice of the dataset to be used became a challenging task since, although it is true that a large number of remote sensor images of the earth's surface are currently available online with ground true to validate classifications, the truth is that it was impossible to find one with ground truth to segmentation assessment. To take advantage of the available images with ground truth for classification, two strategies were proposed. In the following, Section 5.1 describes the dataset used in both task execution and evaluation tests. Section 5.2 presents the outcomes for each main task and Section 5.3 describes the validation of proposed segmentation model.

5.1 Dataset

During the assessment test design, it was realized that there was a scarcity of earth surface hyper and multispectral datasets with segmentation ground truth. As an alternative, dataset for classification along with their ground truth were used instead. In order for the assessments to be meaningful for the dataset, the evaluation process was performed using two strategies: (1) a segment scale generalization (Section 5.3.1) to bring segments to the scale of the available ground truth, and (2) a boundary detection benchmarking (Section 5.3.2) from very high spatial resolution images taking their classification ground truth as a segmentation ground truth. Since the first evaluation strategy performs a segment generalization to group them and bring them to the coarse scale of the available classification ground truth, a dataset with lower spatial resolution was selected. On the other hand, a very high resolution dataset was used for the second validation strategy looking for the available classification ground truth to be closer to a segmentation ground truth.

For the evaluation based on segment generalization (Section 5.3.1), a hyper-spectral image of Pavia (Italy) was used, which is described in Section 5.1.1. To produce outcomes of each segmentation task (Section 5.2) and validation based on benchmarking (Section 5.3.2), a set of images of the cities of Vaihingen and Potsdam (Germany) was used, which is described in Section 5.1.2.

5.1.1 Pavia dataset

A subset of Hyperspectral Remote Sensing Scenes website (`http://goo.gl/0zKFTx`) was used. The selected scene is an image acquired by the ROSIS sensor over center of Pavia, northern Italy. ROSIS (Reflective Optics System Imaging Spectrometer) is a compact airborne imaging spectrometer, which had been developed jointly by Daimler-Chrysler Aerospace AG, the GKSS Research Centre (Institute of Hydrophysics) and the German Aerospace Center based on an original design for a flight on ESA's EURECA platform. ROSIS features include a radiometric quantization of 14 bits, a pixel size at 3 km flight altitude of 1.7×1.7 m^2, and 115 spectral channels (Müller, Gege, and Cocks, 2001).

The chosen 1096×1096 pixel image is composed of 102 spectral bands. After removing the "without data area", it was a 1096×715 pixel image with spatial resolution of "1.3" meters and radiometrics of 14 bits. The ground truth image (see Figure 5.1 right) differentiates "9" classes: Water, Trees, Asphalt, Self-Blocking Bricks, Bitumen, Tiles, Shadows, Meadows, and Bare Soil.



FIGURE 5.1: Dataset used covers Pavia Centre (Italy) (*first principal component of hyperspectral image on left and ground truth image on right*)

Before being used, principal component analysis (PCA) was applied to Pavia center image to reduce the spectral resolution from 102 to "5" bands. The main components were selected so as to ensure that the total variance explained by them was at least 99.5%. Once the principal component analysis was applied to the Pavia center image, it was obtained that this condition was fulfilled by the first five components since the variations that each one explained were (ordered from the first to the fifth component) 70.24%, 26.07%, 2.81%, 0.32%, and 0.12%.

5.1.2 Multispectral image dataset

A subset of the airborne image dataset provided by the International Society for Photogrammetry and Remote Sensing (ISPRS) were used in the accuracy assessment. The used subset is part of the dataset that were provided when ISPRS launched the "semantic labeling contest". These sources were selected

because along with the image dataset, they provided a ground truth, which could be used as valid segments given their high resolution.

Vaihingen dataset

ISPRS provides very high resolution true orthophoto (TOP) tiles of Vaihingen, a small village area in Germany, derived from dense image matching techniques. The digital aerial images are a part of the high-resolution Intergraph/ZI DMC block (Cramer, 2010). They were acquired using an Intergraph / ZI DMC by the company RWE Power on 24 July and 6 August 2008. In total, the block consisted of five overlapping strips with two additional cross strips at both ends of the block. The TOPs are 8-bit TIFF files with a spatial resolution of "8" centimeters and near infrared, red, and green bands delivered by the camera. The images cover urban scenes that have been classified manually into five land cover classes: Building, Low vegetation, Tree, Car, and Impervious surfaces. 16 tiles of the Vaihingen dataset, which had their respective ground truth available, were used. A 1000×1000 pixel window was extracted from each tile, whose location parameters are listed in Table 5.1, for both the multispectral and ground truth dataset.

TABLE 5.1: Vaihingen windows extracted dataset

Tile	From column	From row
top_mosaic_09cm_area37.tif	243	595
top_mosaic_09cm_area34.tif	322	1
top_mosaic_09cm_area32.tif	238	1
top_mosaic_09cm_area30.tif	887	289
top_mosaic_09cm_area28.tif	12	3
top_mosaic_09cm_area26.tif	1205	294
top_mosaic_09cm_area23.tif	702	1107
top_mosaic_09cm_area21.tif	860	311
top_mosaic_09cm_area17.tif	500	200
top_mosaic_09cm_area15.tif	749	705
top_mosaic_09cm_area13.tif	1159	886
top_mosaic_09cm_area11.tif	786	638
top_mosaic_09cm_area7.tif	445	664
top_mosaic_09cm_area5.tif	1	1047
top_mosaic_09cm_area3.tif	284	390
top_mosaic_09cm_area1.tif	745	960

Potsdam dataset

BSF Swissphoto (`https://goo.gl/LXtyCE`) launched the innovation Mobile Semantic Navigation and Information (MSNI) (`http://www.m-s-n-i.de/`) project in February 2012. The project MSNI aimed to support regional companies in the development of novel content applications for the mobile market. As part of the project, BSF Swissphoto carried out a complete aerial survey of Potsdam, a historic city in

Germany, with vertical images with a spatial resolution of "5" cm using an Aerial Oblique System (AOS) consisting of three modular high-quality measuring digital cameras at a height of 800 meters. From the vertical images digital orthophoto mosaics were calculated. A subset of this very high resolution airborne image dataset was provided by ISPRS as TOP tiles over Potsdam.

The TOP tiles were 8-bit TIFF files with four bands: RGBIR (R-G-B-IR). The images cover urban scenes that have been classified manually into six land cover classes: *Building*, *Low vegetation*, *Tree*, *Car*, *Impervious surfaces*, and *Clutter/background*. 24 tiles from the Potsdam dataset, which had their respective ground truth available, were used in this research. Four 1500×1500 pixel windows were extracted from each tile whose location parameters are presented in Table 5.2, for both multispectral and ground truth datasets.

TABLE 5.2: Potsdam windows extracted dataset

Tile file	Window 1		Window 2		Window 3		Window 4	
	Column	Row	Column	Row	Column	Row	Column	Row
top_potsdam_7_12_RGBIR.tif	1	1	1500	1500	3000	3000	4500	4500
top_potsdam_7_11_RGBIR.tif	1	1	1500	1500	3000	3000	4500	4500
top_potsdam_7_10_RGBIR.tif	1	1	1500	1500	3000	3000	4500	4500
top_potsdam_7_9_RGBIR.tif	1	1	1500	1500	3000	3000	4500	4500
top_potsdam_7_8_RGBIR.tif	1	1	1500	1500	3000	3000	4500	4500
top_potsdam_7_7_RGBIR.tif	1	1	1500	1500	3000	3000	4500	4500
top_potsdam_6_12_RGBIR.tif	1	1	1500	1500	3000	3000	4500	4500
top_potsdam_6_11_RGBIR.tif	1	1	1500	1500	3000	3000	4500	4500
top_potsdam_6_10_RGBIR.tif	1	1	1500	1500	3000	3000	4500	4500
top_potsdam_6_9_RGBIR.tif	1	1	1500	1500	3000	3000	4500	4500
top_potsdam_6_8_RGBIR.tif	1	1	1500	1500	3000	3000	4500	4500
top_potsdam_6_7_RGBIR.tif	1	1	1500	1500	3000	3000	4500	4500
top_potsdam_5_12_RGBIR.tif	1	1	1500	1500	3000	3000	4500	4500
top_potsdam_5_11_RGBIR.tif	1	1	1500	1500	3000	3000	4500	4500
top_potsdam_5_10_RGBIR.tif	1	1	1500	1500	3000	3000	4500	4500
top_potsdam_4_12_RGBIR.tif	1	1	1500	1500	3000	3000	4500	4500
op_potsdam_4_11_RGBIR.tif	1	1	1500	1500	3000	3000	4500	4500
top_potsdam_4_10_RGBIR.tif	1	1	1500	1500	3000	3000	4500	4500
top_potsdam_3_12_RGBIR.tif	1	1	1500	1500	3000	3000	4500	4500
top_potsdam_3_11_RGBIR.tif	1	1	1500	1500	3000	3000	4500	4500
top_potsdam_3_10_RGBIR.tif	1	1	1500	1500	3000	3000	4500	4500
top_potsdam_2_12_RGBIR.tif	1	1	1500	1500	3000	3000	4500	4500
top_potsdam_2_11_RGBIR.tif	1	1	1500	1500	3000	3000	4500	4500
top_potsdam_2_10_RGBIR.tif	1	1	1500	1500	3000	3000	4500	4500

To produce the computational outcomes of segmentation tasks presented in Section 5.2, a window $((3000, 5000)$ to $(4000, 6000))$ of tile 40_10 (Figure 5.2) of Potsdam dataset was used.



FIGURE 5.2: Input image window to segmentation tasks

5.2 Computational outcomes of segmentation tasks

This section presents examples of the results obtained from the application of each task involved in the segmentation process of multispectral images represented as Cartesian complexes described in Section 3.4 using the computational framework developed as described in Section 4.3. To produce the results presented here, each task of the complete segmentation process presented in Figure 4.13 was applied to the Potsdam dataset window displayed in Figure 5.2. In the following, the outcomes of each task are presented.

5.2.1 Cartesian complex conversion (1)

Figure 5.3 displays a pictorial representation of the Cartesian complex conversion outcome. In Figure 5.3(a) is displayed the original conventional image; Figure 5.3(b) displays the respective Cartesian complex including the inter-pixel cells as additional pixels. As can be observed on the right, for each pixel in the original image, there are four cells in the respective Cartesian complex: the pixel itself, two 1-cells (1-cells) and one point (0-cell). It must be remembered that inter-pixel cells take their value vector from pixels in their SONs in agreement with the *EQUNALI* rule (Kovalevsky, 2008). Here, the complete Cartesian complex is indicated only to present a visual illustration; however, it should be remembered from Section 3.1, that cells are only created when they are required and that they are stored in storage using GDAL (GDAL Development Team, 2016), according to the established Cartesian complex format.



(a) Original image

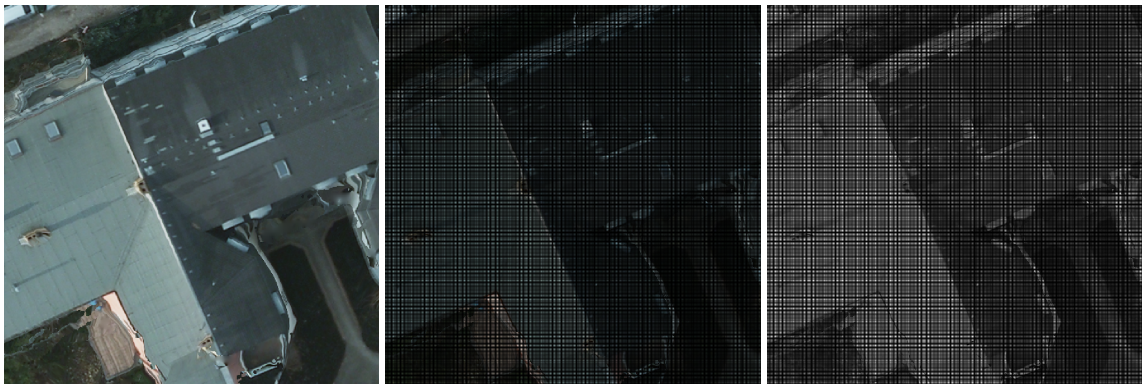


(b) Cartesian complex as pixel image

FIGURE 5.3: Cartesian complex conversion

5.2.2 Gray scale conversion (2)

Figure 5.4 displays a gray scale Cartesian complex production outcome using the procedure described in Section 4.3.2 (Valero, Dallos, and Lizarazo, 2018). For the input image displayed in Figure 5.4(a), the respective Cartesian complex is indicated in Figure 5.4(b), whose inter-pixel cells have no value because they are not required for this task. The black grid effect is used to illustrate this situation. The weight vector calculated from the four band input Cartesian complex is $[0.27042267, 0.26960182, 0.2710394, 0.18893614]^T$, which was used to produce the gray scale Cartesian complex displayed in Figure 5.4(c).



(a) Input image (b) Input Cartesian complex (c) Gray scale Cartesian complex

FIGURE 5.4: Gray scale conversion

5.2.3 Multiscale texture analysis (3)

As described in Section 4.3.6 (Figure 4.13), a multiscale (using several radius and orientation pairs) Gaussian filter bank is applied to a gray scale Cartesian complex to produce a filtered Cartesian complex. Figure 5.5 displays the filtered images obtained using a one-scale Gaussian filter bank ("1" 1-cell) for "2" orientations and "10" filter kernels. The filtering task was performed as in (Arbelaez et al., 2011); however, on inter-pixel 1-cells; hence, there are two kernels: one for the horizontal 1-cells (first row) and another for the vertical 1-cells (third row).

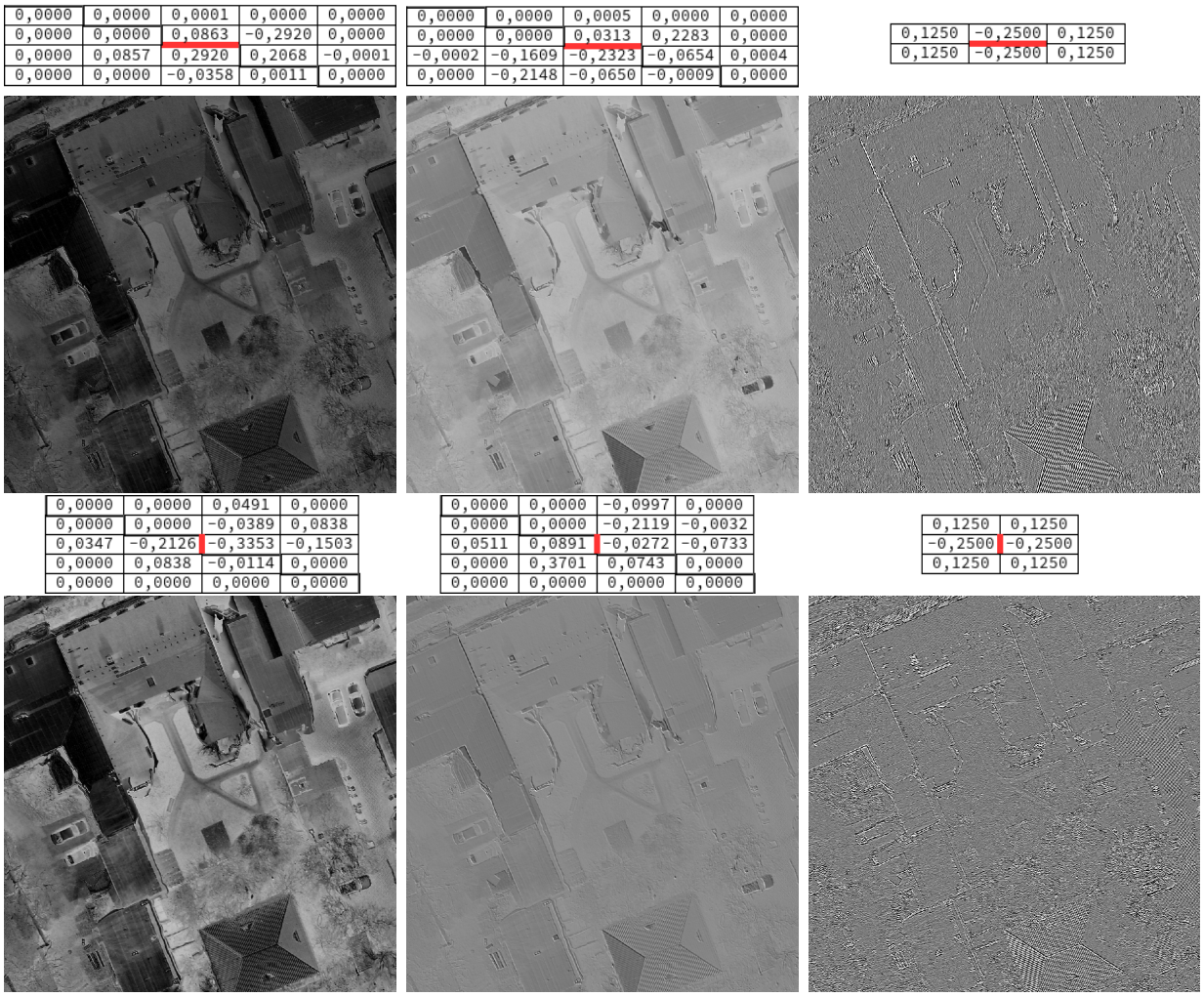


FIGURE 5.5: Gaussian filtering-based texton analysis

As established in Section 3.4.1, (Equations 3.44 to 3.48), the kernel values displayed in the figure correspond to 2-cells and the kernel application outcome corresponds to the respective central 1-cell (indicated as a red segment line). In the figure, x non-derivative, y second derivative filtering (on the left), y Hilbert transformation filtering (center), and a center surround filtering (on the right) for $\pi/4$ orientation are displayed. Horizontal filtering applications are indicated on the second row and vertical applications on the

bottom row as pixel images to avoid the black grid effect; however, they correspond to 1-cell filtering outcomes.

5.2.4 Texture grouping (4)

Figure 5.6 displays the texture grouping obtained by applying K-means grouping of 256 texture classes on a two scale Gaussian filter bank ("3" 1-cells and "5" 1-cells) for "8" orientations, i.e., 34 filter kernels on each 1-cell. The figure displays the texture grouping outcome for the horizontal 1-cells (on the left) and the texture grouping outcome for the vertical 1-cells (on the right).

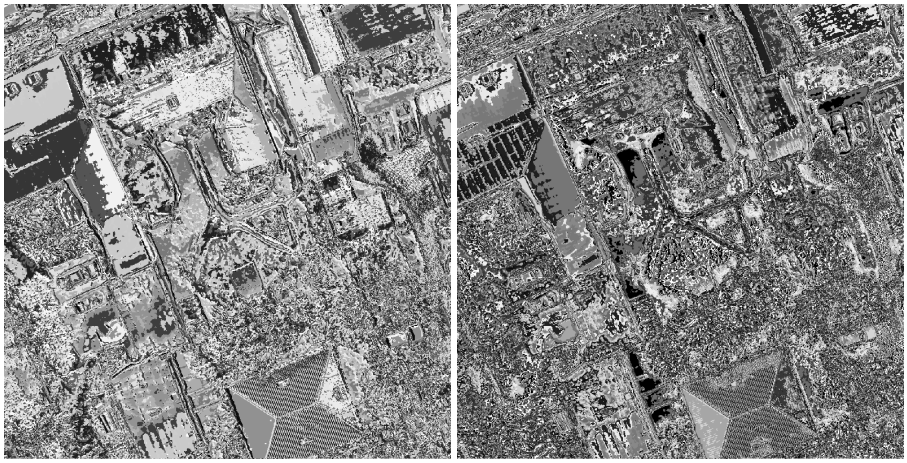


FIGURE 5.6: Texture grouping using k-means

5.2.5 Multiscale oriented gradient (5)

Figure 5.7 displays, from the leftmost column to the fourth column, the maximum oriented gradient outcome of applying the procedure established in Sections 3.4.2 and 4.3.4 to the multispectral Cartesian complex presented in Figure 5.2. The lighter the color of a 1-cell, the greater the gradient value associated with it. Only the horizontal 1-cell image representations are indicated because the vertical 1-cell image representations are virtually the same. The columns correspond to each spectral band and row, from the top to bottom, to the radii of "3", "5", and "10" 1-cells. To the extent that the scale becomes coarser, less detail is preserved. This can be observed in the figure exploiting the fact that the scale becomes thicker from the top to bottom. The spectral band gradients are similar to each other.

The figure also displays, in the column to the far right, the maximum oriented gradient outcome of applying the oriented gradient procedure to the texture Cartesian complex presented in Figure 5.6.

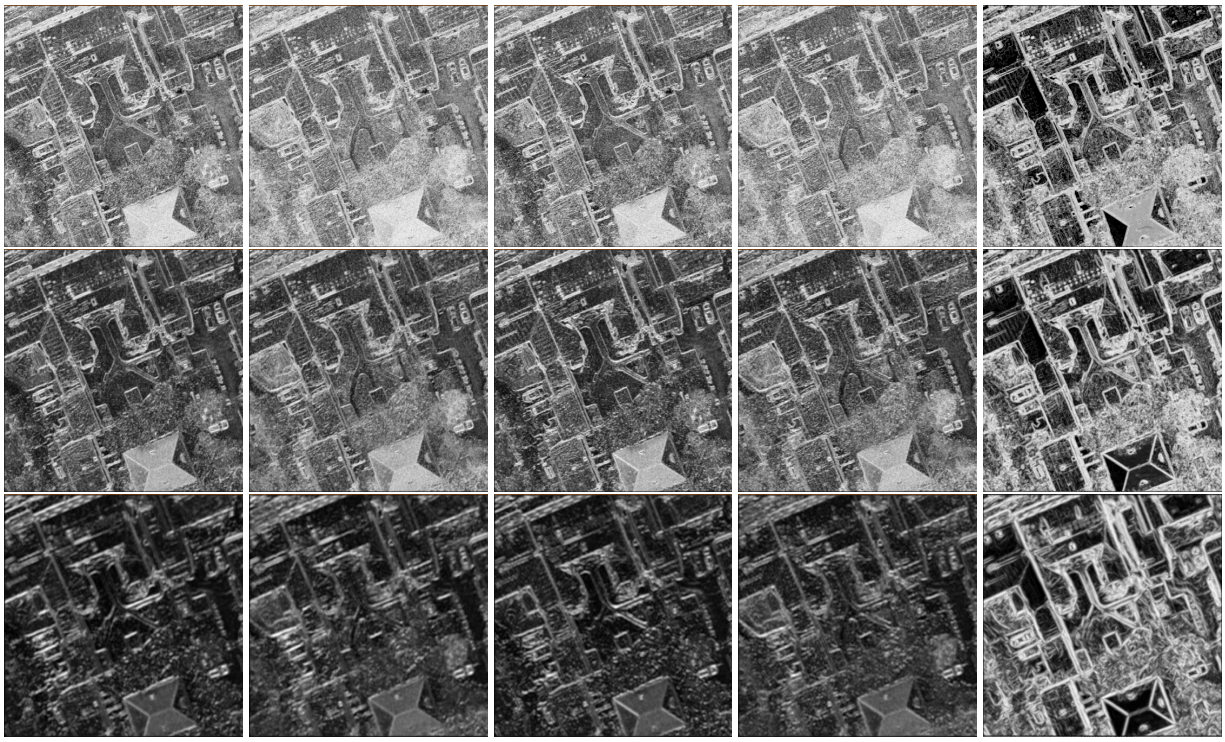


FIGURE 5.7: Maximum oriented multispectral and texture gradient outcomes

5.2.6 Boundary Oriented (6) and maximum (7) geometrical strength

As established in Sections 3.4.4 (Equations 3.81 and 3.82) and 4.3.6, the multiscale spectral and texture gradients indicated in Figure 5.7, scaled to be between 0 and 1, are linearly combined by orientation to produce the boundary geometric strength. Figure 5.8 displays, in the first four columns (from left to right), the boundary geometric strength for four orientations: $\pi/60$, $2\pi/7$, $3\pi/5$, and $59\pi/60$.

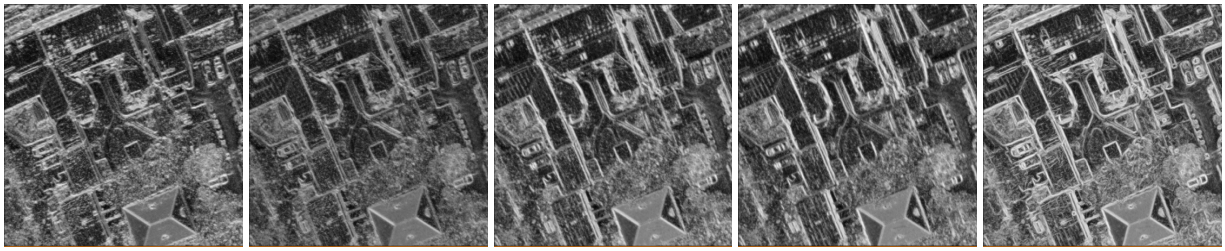


FIGURE 5.8: Boundary oriented geometrical strength

In agreement with Section 3.4.4 and using Equation 3.89, from the boundary oriented geometric strength outcome, the maximum boundary strength indicated in Figure 5.8 (in the column to the far right) is obtained. As Figure 5.8 indicates, the boundary probability (strength) for the test window (Figure 5.2) is greater in orientations $3\pi/5$, and $59\pi/60$, which coincides with the geometric arrangement of element edges present in the window; hence, the maximum of all of them (on the left) manages to gather the best

boundary definition. This tendency is maintained in the boundary oriented global probability displayed in Figure 5.11.

5.2.7 Oriented affinity analysis (8)

Maximum boundary geometrical strength is used as the affinity analysis input (Sections 3.4.3 and 4.3.5). Figure 5.9 displays four of the 16 used eigenvector images in the eigenvector Cartesian complex produced by the generalized eigenvalue problem solver Octave implementation corresponding to the respective eigenvalues of "$1.09E-4$", "$1.01E-4$", "$9.80E-5$", and "$8.90E-5$".

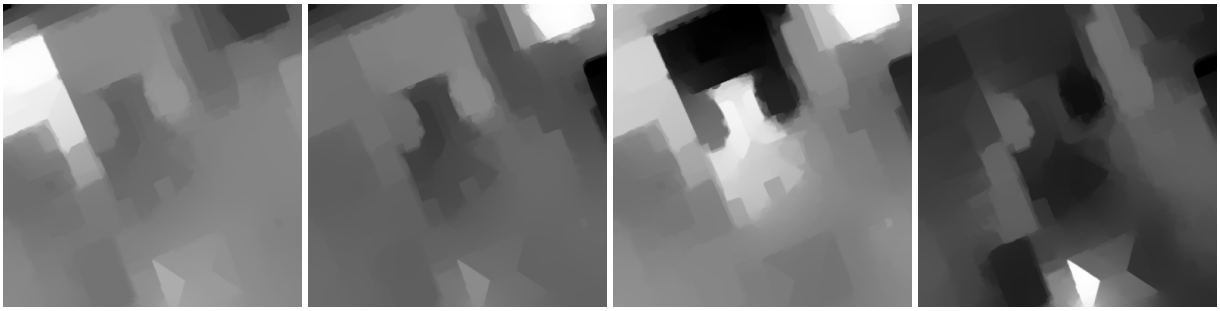


FIGURE 5.9: Outcome affinity analysis eigenvectors

The produced Cartesian complex, with 16 eigenvectors (first convolved with oriented derivative filter kernels) and their respective first largest eigenvalues, is used to produce the boundary oriented spectral strength in agreement with the procedure established in Section 3.4.4 (Equations 3.85 and 3.86). Figure 5.10 displays (from left to right), the boundary spectral strength for three orientations: $\pi/60$, $2\pi/5$, and $59\pi/60$, and the maximum boundary spectral strength.

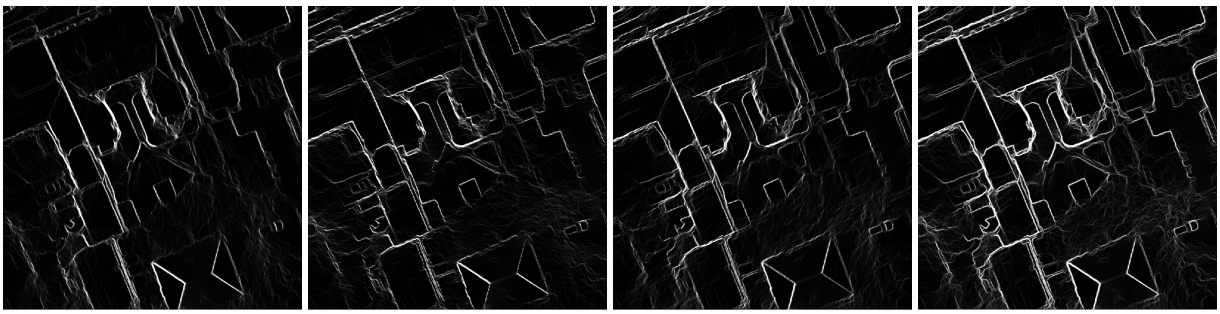


FIGURE 5.10: Boundary oriented spectral probability

5.2.8 Boundary oriented global strength (9)

Applying the procedure established in Section 3.4.4, Equations 3.87 and 3.88 were used to linearly combine the oriented geometrical and spectral probability Cartesian complexes and to produce the boundary oriented global probability Cartesian complex displayed in Figure 5.11. The boundary global strengths

(from left to right) are indicated for three orientations: $\pi/60$, $2\pi/5$, and $59\pi/60$, and the maximum boundary global strength.

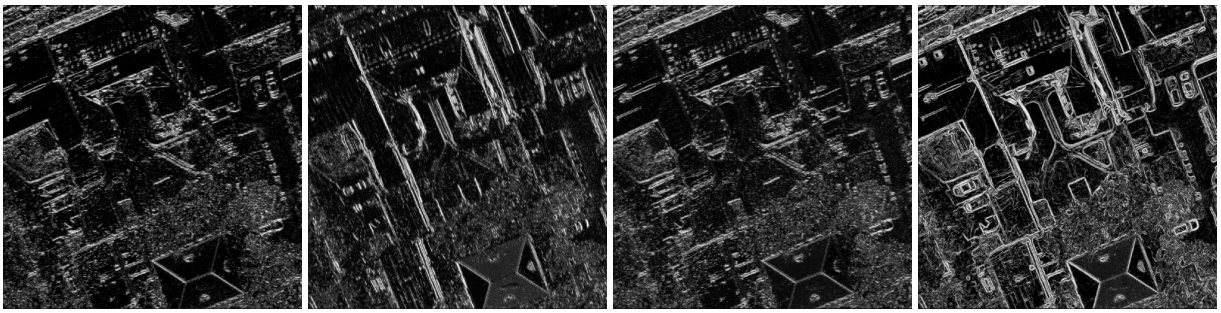
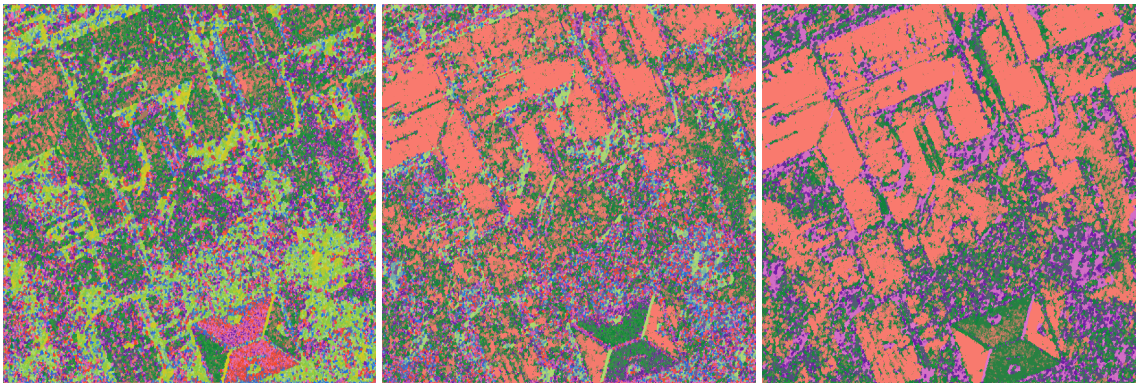


FIGURE 5.11: Boundary oriented global probability

5.2.9 Hierarchical segmentation (10)

The segmentation task outcome obtained by applying the hierarchical watershed procedure established in Section 3.4.5 of the maximum boundary global probability Cartesian complex is displayed in Figure 5.12. The figure indicates the over-segmentation and two more coarse levels of segmentation with respectively 84833, 70242, and 39822 segments each.



(a) Over-segmentation (b) Level 1 of two level segmentation (c) Level 2 of two level segmentation

FIGURE 5.12: Cartesian complex segmentation outcome

5.3 Validation of proposed segmentation model

Several evaluation tests were conducted to establish the validity and effectiveness of the segmentation process from the spatial representation in terms of Cartesian complex based on their topological and geometric models and in the procedural framework adopted. During the assessment test design, it was realized that there was a scarcity of earth surface hyper and multispectral datasets with segmentation ground truth. Therefore, as mentioned in Section 5.1, the evaluation process was performed using two

strategies: (1) segment scale generalization (Section 5.3.1) to format segments to the scale of the available segmentation ground truth and (2) regular boundary detection benchmarking (Section 5.3.2).

In both segmentation assessment strategies, the evaluation framework proposed in (Martin et al., 2001) and its update in (Arbelaez et al., 2011) was used because this provides an empirical basis for validating research on image segmentation and boundary detection. The goal of the benchmark is to produce a score for boundary detection outcomes, allowing the comparison of different algorithms. This methodology can be applied to any boundary dataset. The benchmarking framework assumed that any algorithm output was a soft boundary map with one pixel-wide boundaries valued from zero to one, where high values signify greater confidence in the existence of a boundary.

The benchmarking framework task was to determine how well this soft boundary map approximates the ground truth boundaries. What is usually done is to "binarize" the boundary map by choosing any threshold. There are two problems with thresholding a boundary map: (1) the optimal threshold depends on the application, and it would be preferred the benchmark to be useful across different applications and (2) thresholding a low-level feature such as boundaries is likely to be unacceptable for the majority of applications because it destroys considerable information. Nevertheless, the benchmarking framework does require thresholding the boundary map to compare it to the ground truth boundaries and it does this at many levels. At each level, it computes two quantities, ***precision*** and ***recall***, and in this manner produces a precision-recall curve for the algorithm. *Precision* is the probability that a machine-generated boundary is a true boundary; it is a measure of how much noise is in the output of the detector. *Recall* is the probability that a true boundary is detected; it is a measure of how much of the ground truth is detected. The curve indicates the inherent trade-off between these two quantities, the trade-off between misses and false positives, as the detector threshold changes.

The performance of an algorithm is summarized into a single number. The summary statistic that it uses is a measure of the distance from the origin to the precision-recall curv. It is the F-measure, which is the harmonic mean of the precision and recall. The F-measure is defined at all points on the precision-recall curve. The maximum F-measure value across an algorithm's precision-recall curve is its summary statistic. In the update presented in (Arbelaez et al., 2011), three additional region-based metrics are included: the best F-measure on the dataset for a fixed scale (ODS), the aggregate F-measure on the dataset for the best scale in each image (OIS), and the average precision (AP) on the full recall range (equivalently, the area under the precision-recall curve).

5.3.1 Segment scale generalization

A regular segmentation assessment could not be performed in this assessment because the ground truth available corresponded to a classification, not to a segmentation. For this validation strategy, previously, an over-segmented Cartesian complex was produced using the computational prototype implementation (Chapter 4) following the pipeline established in Figure 2.22. First, principal component analysis (PCA) was applied for reducing the spectral resolution from 102 to "5" bands. Next, a contour detection was performed. Then, watershed transform-based segment production and segmentation assessment were conducted. As part of this test, an oriented gradient calculation execution time comparison between the Matlab® and computational prototype implementations was performed (Table 5.5).

The validation process was conducted in two segment geometry assessments using (1) segment and ground truth image intersection and (2) the BSDS segmentation assessment benchmark (Arbelaez et al., 2011; Martin, Fowlkes, and Malik, 2004). In the first segment geometry assessment (*spAss*), it compared the ground truth image boundaries against segments intersecting them using Equation 5.1. Figure 5.13 displays the multilevel segmentation produced from the over-segmented to the most general, using "5" boundary strength ranges.

$$spAss = \frac{1}{\sum_{sp \in Bint} area(sp)} \sum_{sp \in Bint} \max_{gt}(area(int(sp, gt)) \ , \tag{5.1}$$

where *Bint* is the ground truth boundary intersecting segment set,
gt is a connected ground truth area, and
$int(sp, gt)$ is a set of intersections between a segment sp and any gt.

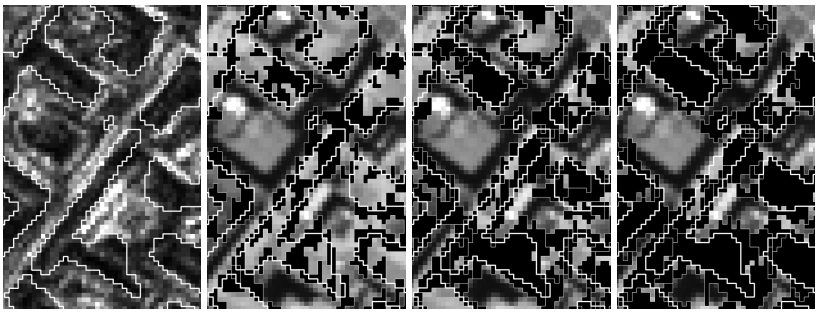


FIGURE 5.13: Boundary strength image (*left*) and multilevel segmentation image (*on right: over, "9" height ranges and "5" height range segmentations*)

TABLE 5.3: Segment intersecting boundary ground truth geometry assessment includes several segmentations (*one by row*)

Ranges	Segments	Intersecting	$\sum area(sp)$	$\sum \max(area(int(sp, gt))$	spAss
All	145638	13157	161865	128577	0.7943
513	144265	13096	163223	129645	0.7943
129	140970	12956	164996	130918	0.7935
33	130845	12469	172235	136181	0.7907
9	102694	9433	209932	165522	0.7885
5	74956	5321	286630	224402	0.7829

Table 5.3 presents the assessment outcomes of the segment intersecting ground truth boundaries. The first column is the number of boundary strength ranges used, the second is the total number of segments, the third is the number of intersecting segments, the fourth is the total intersecting area (*in number of pixels*), the fifth is the total maximum intersecting segment area, and the last is the segment geometry assessment. The segment geometry assessment was conducted on each of "9" levels in the multilevel segmentation produced by the proposed watershed transform implementation. Both over-segmentation and 8-level multilevel segmentation images were produced. The first row in Table 5.3 corresponds to the

over-segmentation image segment geometry assessment, the other rows assess the segment geometry for the seventh and eighth segmentation levels, i.e., using "9" and "5" boundary strength values, respectively.

In the BSDS segmentation assessment benchmark, the suitability of segments for downstream recognition applications was evaluated. For this purpose, it was assumed that all segments were correctly classified by assigning to each segment the ground truth region label most frequently included inside it. Then, each so produced segmentation was assessed using the BSDS benchmarks, taking the semantic annotation boundaries of the hyperspectral image as ground truth. Table 5.4 presents the OIS boundary and region quality measures for the multilevel segmentation. Table 5.5 displays the oriented gradient execution time comparison between the Berkeley Computer Vision Group Matlab® and the computational prototype implementations using three radio measurements in pixels for the conventional pixel and 1-cells for the Cartesian complex representations.

TABLE 5.4: Boundary assessment reports precision, recall, and F-measure; region quality reports covering of ground-truth (*GT covering*)

Boundary			Region
Precision	Recall	F-measure	GT covering
0.99	0.42–0.60	F(0.60, 1.00) = 0.75	0.92

TABLE 5.5: Execution time measured for three radii (*in pixels for Matlab and cracks for proposed*) and several orientations (*always the maximum*)

Radius	1	2		3		
Orientations	3	3	7	3	8	15
Matlab	23.94	23.54	44.35	23.12	56.81	99.75
Cartesian complex implementation	23.33	27.33	35.33	26.33	43.33	47.67

5.3.2 Boundary detection benchmarking

This assessment strategy considered the complete boundary detection process through the evaluation of both approaches, the one proposed in (Arbelaez et al., 2011) and the one implemented as part of the present research (Chapter 4). To perform a meaningful evaluation, two high geometric resolution datasets with their ground truth were selected. The main assessment process was based on the boundary detection images produced by both the implementation from the pixel conventional representation provided in (Arbelaez et al., 2011) and the computational prototype from the Cartesian complexes obtained as part of the present research. The assessment pipeline, which is displayed in Figure 5.14, was composed of "6" tasks.

A window reshaping task (1) was performed to ensure that the two used datasets (Vaihingen and Potsdam) had the same scale resolution and could be segmented with the same multiscale parameters. The 16 Vaihingen windows were reshaped taking only one of each two pixels, thus producing 500×500 pixel windows with a pixel size of 16 meters. 96 Potsdam windows were reshaped taking only one of each three pixels, thus producing 500×500 pixel windows with a pixel size of 15 meters. Although the computational prototype produced as part of this research can function on any type of image (RGB, gray scale,

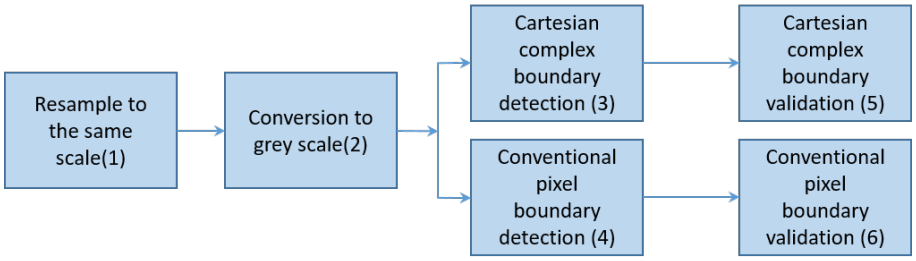


FIGURE 5.14: Boundary detection assessment pipeline

multi and hyperspectral), the available version provided in (Arbelaez et al., 2011) is oriented to RGB / gray scale images. Therefore, the conversion task to images of gray levels (2) was performed to ensure that the results derived from input images with the same conditions, in this case, the gray scale images.The boundary detection task was performed for both Cartesian complexes using the prototype computational implementation produced as part of this research (3) and conventional pixels using the implementation provided in (Arbelaez et al., 2011) (4). The Final task of boundary detection assessment was made for both Cartesian complex outcomes (5) and conventional pixels using the assessment framework provided in (Arbelaez et al., 2011) (6). In both cases, pixel and Cartesian complex representations, the assessment framework employed thins the probability boundary to ensure boundaries are standard thickness using the *bwmorph* Matlab® function. The results of this evaluation for multispectral gradient calculation, texture gradient calculation, geometric and spectral (based on affinity analysis) boundary strength, and global boundary probability are described below.

Multispectral gradient validation

TABLE 5.6: Multispectral multi-scale oriented gradient assessment summary.

Dataset	Radius	ODS		OIS		AP	
		Pixels	CC	Pixels	CC	Pixels	CC
Potsdam	3	0.52	0.52	0.54	0.53	0.43	0.43
	5	0.51	0.54	0.52	0.55	0.37	0.45
	10	0.43	0.52	0.44	0.53	0.24	0.38
Vaihingen	3	0.58	0.59	0.58	0.59	0.41	0.46
	5	0.55	0.60	0.56	0.60	0.33	0.48
	10	0.46	0.57	0.47	0.58	0.21	0.39

Table 5.6 presents the oriented multispectral multi-scale oriented gradient assessment summary information including ODS and OIS F-measure values as well as the average precision for Potsdam (top row) and Vaihingen (bottom row) dataset. As table shows, ODS and OIS optimal F-measures obtained from Cartesian complex representation (columns headed with “CC”) were greater or equal than those obtained from conventional pixel representation(columns headed with “Pixels”). Cartesian complex representation always produced a better average precision (AP column).

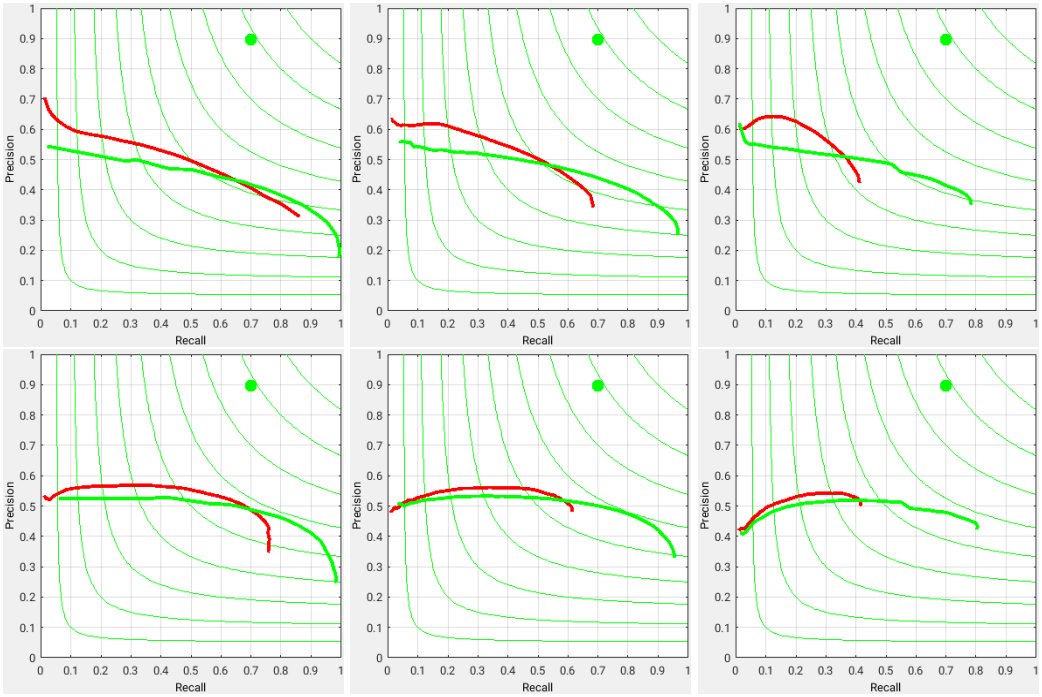


FIGURE 5.15: Multispectral multi-scale oriented gradient assessment (*from left to right radii of "3", "5" and "10" 1-cells/pixels*) for Potsdam (top) and Vaihingen (bottom).

Figure 5.15 illustrates the assessment outcomes for the multispectral multi-scale oriented gradient calculation process applied to Potsdam (top row) and Vaihingen (bottom row) dataset. For the three used scales (radii of "3", "5" and "10" 1-cells for Cartesian complex representation or pixels for conventional pixel representation), the oriented gradient calculations from conventional pixels (red curve) produced better precision, while the oriented gradient calculations from Cartesian complex (green curve) produced better recall. In the figure it can be seen how, to the extent that the radius increases, so does the precision (i.e., the curve falls less in the values of greater recall), while the recall decreases.

Texture gradient validation

Table 5.7 presents the oriented multi-scale oriented texture gradient assessment summary outcomes including ODS and OIS F-measure values as well as the average precision for Potsdam (top row) and Vaihingen (bottom row) dataset. As table shows, ODS and OIS optimal F-measures obtained from Cartesian complex representation (columns headed with "CC") were greater than those obtained from conventional pixel representation(columns headed with "Pixels") at coarser scales (radii of "10" and 20 1-cells for Cartesian complex representation or pixels for conventional pixels). ODS and OIS optimal F-measures obtained from conventional pixel representation were greater than those obtained from Cartesian complex representation at the finest scale (radii of "5"). Cartesian complex representation always produced a better average precision (AP column).

Figure 5.16 displays the assessment outcomes for the multi-scale oriented texture gradient calculation process applied to Potsdam (top row) and Vaihingen (bottom row) dataset for the three used scales (radii

TABLE 5.7: Texture multi-scale oriented gradient assessment summary.

Dataset	Radius	ODS		OIS		AP	
		Pixels	CC	Pixels	CC	Pixels	CC
Potsdam	5	0.51	0.51	0.53	0.52	0.34	0.41
	10	0.42	0.52	0.44	0.53	0.21	0.38
	20	0.29	0.47	0.30	0.48	0.11	0.24
Vaihingen	5	0.57	0.55	0.58	0.55	0.35	0.43
	10	0.48	0.57	0.49	0.57	0.23	0.37
	20	0.35	0.50	0.36	0.50	0.14	0.25

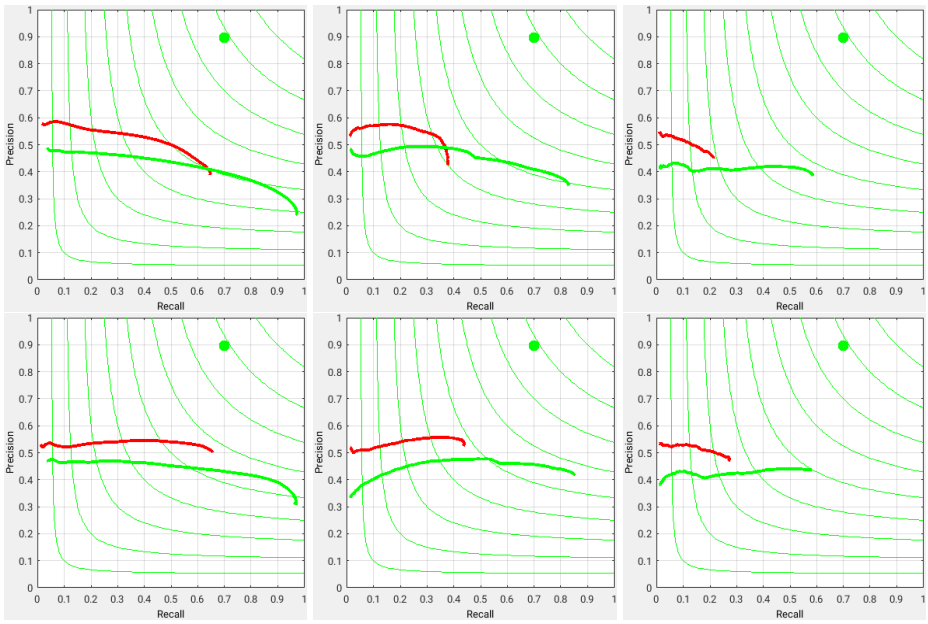


FIGURE 5.16: Oriented Texture multi-scale oriented gradient assessment(*from left to right radii of "5", "10" and* 20 1*-cells/pixels*).

of "5", "10" and 20 1-cells for Cartesian complex representation or pixels for conventional pixel representation). As in the case of the multispectral gradient, while oriented gradient calculation based on conventional pixels (red curve) produced better precision, oriented gradient calculation from Cartesian complex (green curve) produced better recall. In the figure it can be seen how, to the extent that the radius increases, so does the precision (i.e., the curve falls less in the values of greater recall), while the recall decreases.

Geometric and spectral boundary strength validation

Table 5.8 presents summary outcomes of geometric and spectral boundary assessments for Potsdam and Vaihingen dataset including ODS and OIS F-measure values as well as the average precision. Geometric

boundary corresponds with the outcome of sixth task in Figure 4.13 described as Multiscale Boundary geometrical probability (MBGP) in Section 3.4.4 while spectral boundary corresponds with the outcome of seventh task described as Boundary spectral probability (BSP) in Section 3.4.4. As table shows, ODS and OIS optimal F-measures obtained from Cartesian complex representation (columns headed with "CC") were greater than or equal to those obtained from conventional pixel representation(columns headed with "Pixels"). In the case of spectral boundary probability, the assessment outcomes show that the average precision (AP column) obtained from Cartesian complex representation (columns headed with "CC") were less than those obtained from conventional pixel representation, i.e., its actual implementation is not good enough yet.

TABLE 5.8: Geometric and spectral boundary detection assessment summary.

Dataset		ODS		OIS		AP	
		Pixels	CC	Pixels	CC	Pixels	CC
Potsdam	geometric	0.47	0.54	0.49	0.55	0.28	0.45
	spectral	0.52	0.53	0.55	0.55	0.35	0.30
Vaihingen	geometric	0.54	0.59	0.55	0.59	0.29	0.46
	spectral	0.59	0.59	0.59	0.59	0.41	0.34

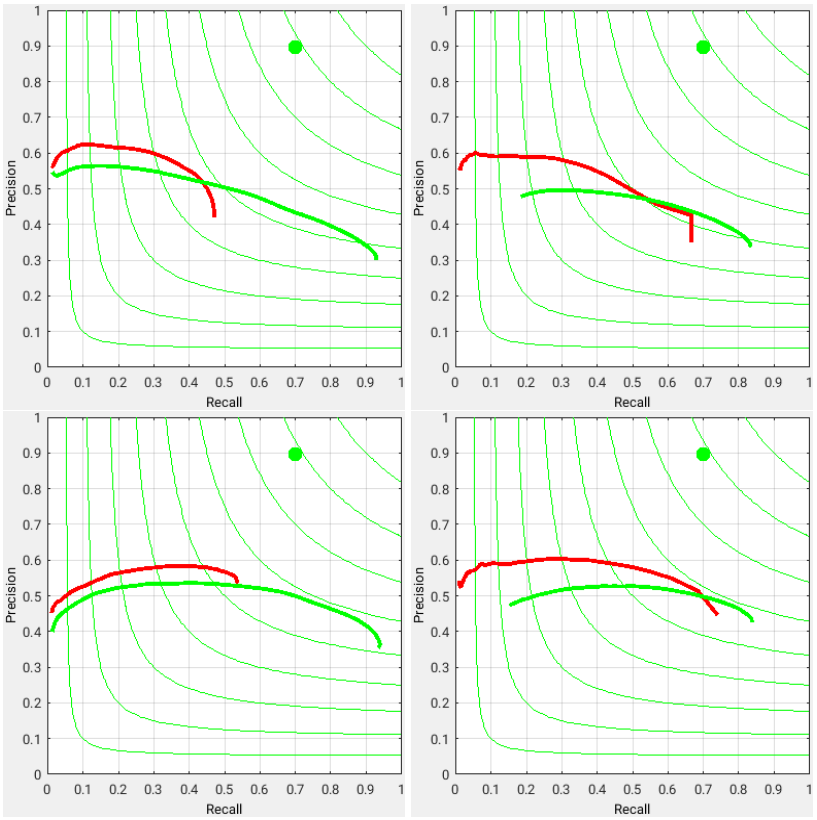


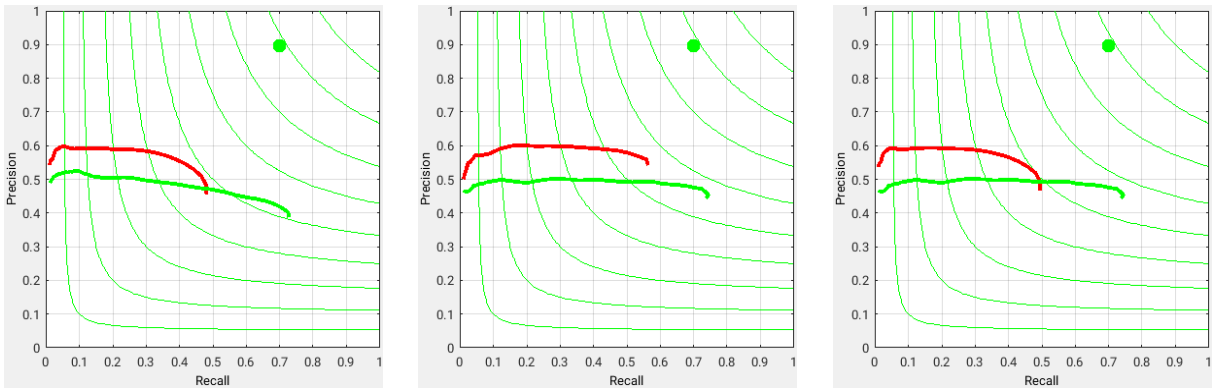
FIGURE 5.17: Geometric (left) and spectral (right) boundary detection assessment.

Figure 5.17 displays the assessment outcomes for geometric (left) and spectral (right) boundary processes

applied to Potsdam (top) and Vaihingen (bottom) dataset. The geometric boundary probability from conventional pixels (red curve) produced better precision, while that from Cartesian complex (green curve) produced better recall. However, in the case of the spectral boundary probability, the figure indicates that the current implementation from the Cartesian complex does not yet have any advantage, but it only a slightly better recall.

Global boundary probability

Figure 5.18 displays the assessment outcomes for the boundary detection process applied to Potsdam (5.18(a)), Vaihingen (5.18(b)) and both (5.18(c)) dataset. As can be observed, for Potsdam (5.18(a)) and Vaihingen (5.18(b)) dataset, boundary detecting based on conventional pixels (red curve) produced better precision, while boundary detecting from Cartesian complex (green curve) produced better recall and average precision.



(a) (b) (c)

FIGURE 5.18: Boundary detection assessment for Potsdam (left), Vaihingen (middle) and both (right) dataset.

Figure 5.18(c) illustrates the assessment outcome for the boundary detection process applied to entire (Potsdam and Vaihingen) dataset. As can be seen, in general, boundary detecting based on conventional pixels (red curve) produced better precision, while boundary detecting from Cartesian complex (green curve) produced better recall.

TABLE 5.9: Boundary detection assessment summary for Potsdam and Vaihingen dataset.

Dataset	ODS		OIS		AP	
	Pixels	CC	Pixels	CC	Pixels	CC
Potsdam	0.49	0.53	0.51	0.54	0.28	0.35
Vaihingen	0.56	0.57	0.57	0.58	0.33	0.36
Both	0.50	0.57	0.52	0.58	0.28	0.36

Table 5.9 presents the boundary assessment summary outcomes for Potsdam, Vaihingen, and entire (Potsdam and Vaihingen) dataset including ODS and OIS F-measure values as well as the average precision.

For Potsdam and Vaihingen dataset (first and second rows), as well as for entire dataset (last row), boundary detecting based on Cartesian complex always produced better ODS and OIS optimal F-measure values and average precision (AP column).

Chapter 6

Discussion

This chapter discusses the manner in which the objectives that were identified at the start of research were met and the fashion in which the hypothesis was contrasted, in addition to relevant aspects of the results obtained from tasks implemented in the computational prototype that was developed. Section 6.1 presents a discussion of the research question and objectives, Section 6.2 discusses the hypothesis, and Section 6.3 discusses aspects of the results obtained by the developed computational prototype.

6.1 Objective achievement

Conventional pixel image representation is used in several image segmentation approaches based on edge detection, which include to model edges as image intensity strong transitions (Canny, 1986), image responses to a family of multi-scale steerable filters (Freeman and Adelson, 1991), and color and texture variations (Dollar, Tu, and Belongie, 2006; Martin, Fowlkes, and Malik, 2004). In contrast, conducting the research presented here succeeded in attaining the main goal (Section 1.4.1) of developing an alternative representation of space for image segmentation by utilizing the 1-cells provided by Cartesian complexes.

In particular, this allowed specific objective "**To devise a conceptual model for multispectral image representation based on Cartesian complexes of abstract cells that considers topological and geometric properties**" to be achieved because this alternative image representation facilitated the improvement of the segmentation process because each of its tasks could be defined based on an image space representation that meets the topological space separation axiom T_0, as is the case of ALFSs provided by Cartesian complexes (Section 2.1). Hence, it was possible to reformulate the procedural framework proposed by Arbelaez et al. (2011) in terms of the 1-cells available in Cartesian complexes. This was accomplished as discussed in the conceptual model proposed in Chapter 3.

The specific objective "**To design a mechanism for encoding topological and geometric features present in combinatorial half-spaces using oriented matroids to reduce algorithmic complexity in multispectral image segmentation**" also was achieved since, conducting the research demonstrated the ability to link oriented matroids with the topological and geometrical properties of the ALFSs provided by Cartesian complexes (Section 1.2). Specifically, the definition of the central SSF arrangement geometric concept based on the respective underlying concept of SSF defined by Kovalevsky (2008) allowed the definition of an oriented matroid using half-space sign vectors in the manner described in Section 3.3.2.

This offered an improvement in several tasks that could be defined based on SSF and their linked oriented matroid, e.g., the filter kernel definition and gradient calculation.

Regarding the specific objective "**To build and evaluate a computational framework that enables the implementation of Cartesian complexes to adequately represent the topological and geometric image-object properties used in the segmentation of multispectral images**", as part of the research and with the purpose of improving the efficiency of the solution, the design of the computational prototype was developed based on a parallel computing framework. In this structure, each of the tasks of the segmentation process can be divided into as many subtasks as the number of processing units available following the procedure established in Section 4.3.1. This ensures that to process a given number of 1-cells, the processing time can be decreased as the number of assigned processing units is increased. The proposed digital image segmentation model validation described in Section 5.3 is discussed in Section 6.2.

The multispectral image segmentation improvement raised by the initial research question set (Section 1.2) was achieved in agreement with the findings discussed in the following.

6.1.1 Filtering stage

Having inter-pixel elements allows the results of operations oriented to detect the boundaries to be associated unambiguously with 1D elements, which are in accordance with the dimensionality of the boundaries to be detected. Conversely, in the conventional pixel model, the result must be associated with a pixel, which is 2D. In the particular case of filter application, this situation is illustrated with the calculation of Sobel filters for edge detection. Figure 6.1 displays the Sobel filter kernel definition (Szeliski, 2010) in a conventional pixel space compared to that in a Cartesian complex. The conventional Sobel kernel definition uses two 3×3 pixel arrangements to produce the convolution for the central pixel and then to calculate the gradient. In a Cartesian complex, it is necessary to use a 2×3 pixel arrangement for the central horizontal 1-cell, a 3×2 pixel arrangement for the central vertical 1-cell to produce the convolution, and then only the absolute value of the filtered Cartesian complex can be calculated. In a Cartesian complex, the 1D pattern (the boundary) found by the kernel is retained in a 1D space element, the central 1-cell.

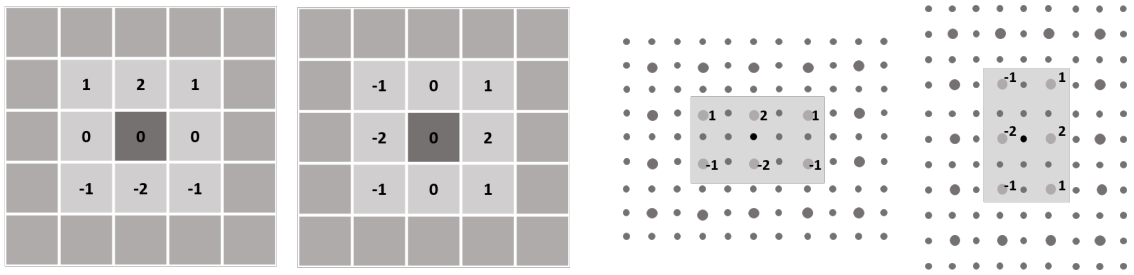
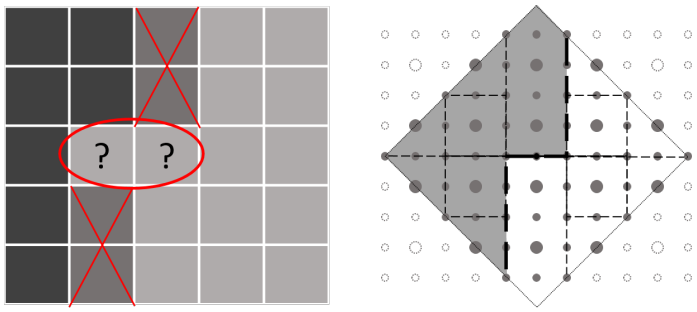


FIGURE 6.1: Conventional versus Cartesian complex Sobel filter kernel definition

Although the process of construction of filter kernels from 2-cells whose result is associated with a central 1-cell has been solved and is used in the texture analysis task, the filter creation to counteract the effect of phantom edges, which is presented during the calculation of oriented gradients, is not yet complete.

6.1.2 Gradient half-subspace definition

In the case of the calculation of the oriented gradient, in a Cartesian complex, each side of the diameter line is represented in an unambiguous manner because the diameter is defined based on the concept of SSF, which is 1D. Therefore, the histograms of each side can be properly calculated. Conversely, in a conventional pixel representation, the diameter must be represented by a linear array of pixels, which must also be ignored in the final calculation to not bias the histogram of either side of the diameter line. Figure 6.2 displays this in a conventional pixel space; in addition to the connection ambiguity, at least "5" pixels (if 4-connected pixels are used) representing the diameter line must be ignored. Conversely, in a Cartesian complex, none of the pixels (2-cells) are ignored because none of them are used in representing the diameter line; only 1-cells belonging to the diameter SSF are used.



(a) (b)

FIGURE 6.2: Conventional versus Cartesian complex gradient half-space definition

The subspace definition based on SSF permits unambiguously establishing the half-space membership. However, the use of Equation 3.12 of membership to an SSF imposes a restriction on the use of several elements of the space that comply with it, which are included in the calculations when Equation 3.7 is used in its place. This last strategy is recommended when the area of application is small, that is, it is less than "3" 1-cells of radius.

6.1.3 1-cell affinity

In the case of affinity analysis, in the conventional pixel space, the determination of the line that connects two pixels is made by determining the linear array of pixels based on the Bresenham line algorithm (Arbelaez et al., 2011). At line breakpoints, this algorithm faces the dilemma of whether to use 4-connected or 8-connected pixels, whatever choice is made, it is known that it will be ambiguous. However, in Cartesian complexes representation, the affinity analysis is made from 1-cells, not from pixels (2-cells). Therefore, the connecting line corresponds to the SSF that contains them and its definition is established in an unambiguous manner. Figure 6.3 illustrates the situation dilemma for two breakpoints, which in general, is resolved according to the convenience of the particular case. However, in a Cartesian complex, the two 1-cells, whose affinity is being analyzed, can be linked by the SSF calculated from their combinatorial coordinates according to the procedure established in Section 3.3.1.

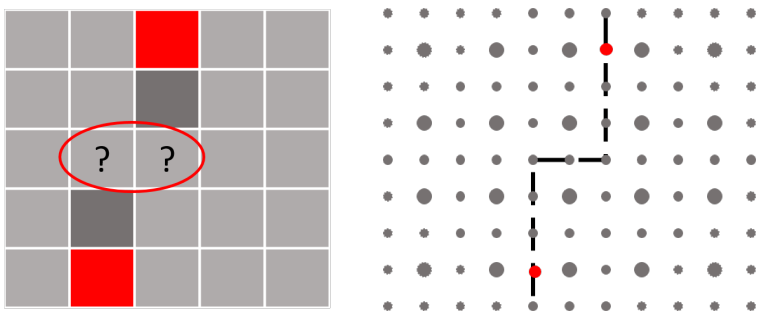
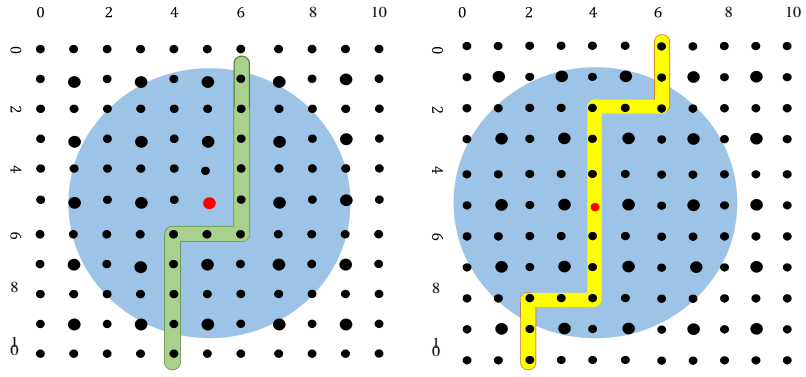


FIGURE 6.3: Conventional versus Cartesian complex connecting line in affinity analysis

In a Cartesian complex, if the radius were centered on a pixel, the dividing SSF would always be asymmetric and ambiguous, as indicated in Figure 6.4(a); therefore, measures would be inaccurate. Conversely, with a 1-cell centered radius, the measures are accurate because of the symmetry and non-ambiguity (Figure 6.4(b)).



(a) In a 2-cell centered radius, dividing DSS is never symmetric and always ambiguous

(b) In a 1-cell centered radius, DSS is always symmetric and unambiguous

FIGURE 6.4: Affinity between pair of cells (*red cells are centers*)

Although the affinity analysis process can be performed, the Cartesian complex of boundary probabilities produced from the Cartesian complex of eigenvectors presents a small number of high outliers that currently reduce the contribution of this spectral probability Cartesian complex.

6.1.4 Oriented matroid-based gradient calculation

The oriented gradient calculation was based on the approach proposed in (Martin, Fowlkes, and Malik, 2004); however, rather than using evenly spaced orientations, the SSF orientations available in an SSF central array were used, which do not produce evenly spaced orientations. Figure 6.2 displays a comparison between a conventional pixel and a Cartesian complex spatial arrangements for the calculation of the oriented gradients for a particular orientation (displayed as darkest dashed line in Figure 6.2(b)). The

configuration of the sectors (pie slice) is performed in accordance with the subspaces formed by the SSFs in the central array, each of which is identified based on the elements of the associated oriented matroid. In this manner, the calculation from the elements of the associated matroid can be completed regardless of whether the sectors (pie slice) are evenly spaced or not.

Owing to the fact that the boundaries between one sector and another are established from the SSFs, there is no ambiguity when determining to what subspace a given space element belongs. Therefore, the respective histogram is calculated in a suitable fashion.

6.1.5 Watershed calculation

It was possible to define the process of segment construction based on the watershed transformation from the 1-cells. For this, the exploration of an eventual basin expansion was performed based on the non-ambiguous topological definition established by Equation 3.91, which is unlike the situation faced by the approach proposed in (Vincent and Soille, 1991) based on a conventional pixel space, where the choice must be made between an exploration scheme based on 4-connected or 8-connected pixels. Figure 6.5 illustrates this situation for the conventional pixel space and Cartesian complex representations.

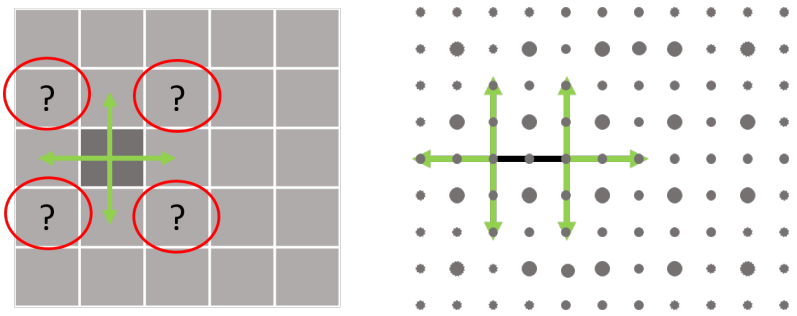


FIGURE 6.5: Conventional versus Cartesian complex basin expansion

Notwithstanding the topologically correct definition for the exploration of basin expansion, the method to address the problem of plateaus is not adequate, which is evident in the irregular definition of the edges between the defined segments.

6.1.6 Parallel computing scheme

For the case of the tasks of the segmentation process, for each 2-cell, the calculation must be performed on its two own 1-cells; hence, the number of space elements to be processed is doubled. Therefore, when the calculation performed on a 1-cell is independent of the result of the calculation of another 1-cell, the parallel computational framework established in (Valero, Dallos, and Lizarazo, 2018) and described in Section 4.3.1 provides a performance improvement. Figure 6.6 illustrates parallel execution scenarios. Each scenario is established based on the number of central processing units (CPU) and the number of space elements that must be processed. The total number of space elements to be processed is divided into as many groups as the number of available processors. Therefore, as the figure indicates, the more CPUs available, the fewer space elements that are assigned to each CPU; hence, the less time each will require

to complete the task, which on average will be the same for the complete Cartesian complex process, because the tasks are executed simultaneously.

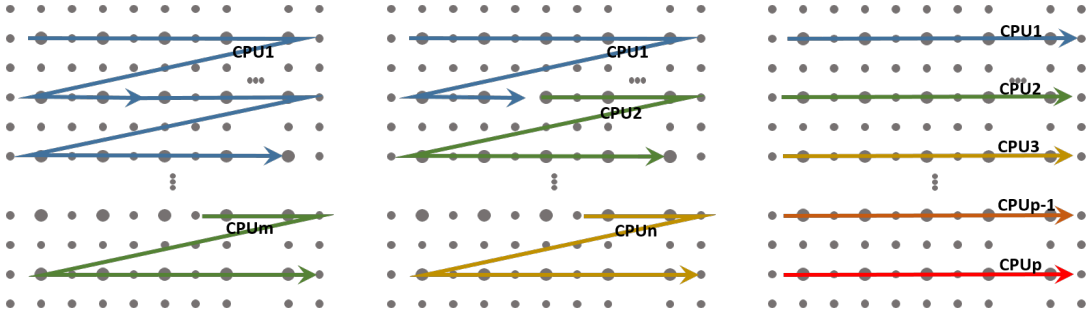


FIGURE 6.6: Parallel execution scenarios ($m < n < p$)

The proposed parallel computing framework provides an important instrument for increasing the performance of the segmentation process, that is, a decrease in the execution time to the extent that more CPUs are available. However, in (Arbelaez et al., 2011), an implementation based on graphical processing units (GPU) that accomplishes the same parallelization purpose is reported. However, providing computational support for GPUs requires the reimplementation of the tasks that are part of the segmentation process because, unlike CPUs, which are general purpose, GPUs are purpose-specific for graphic processing, and code must be adapted to the application program interface (API) provided by them.

6.1.7 Specific objectives achievement

After conducting the research, it was also possible to achieve the specific objectives established in Section 1.4.2. A conceptual model was devised, which is presented in Chapter 3, for multispectral image representation based on Cartesian complexes. Further, both topological and geometric elements were used to reformulate the tasks of the image segmentation process in terms of available 1-cells. It was designed using the mechanism established in Section 3.3 for encoding topological and geometric features present in combinatorial half-spaces using oriented matroids, which provided reduced the execution time in multispectral image segmentation, specifically in the oriented gradient calculation used several times along the segmentation process pipeline.

Finally, a computational prototype, described in Chapter 4, allowed Cartesian complexes to adequately represent topological and geometric image-object properties used in the segmentation of multispectral images. Computational implementation outcomes are presented in Section 5.2. The complete segmentation process was assessed in the manner presented in Chapter 5.

6.2 Testing the hypothesis

The validation of the hypothesis involves conceptual and empirical aspects. With respect to the first, the fact of using, as a basis, a space representation model that meets the axioms of separation, established by the topological spaces T_0, guarantees that the ambiguities described in Section 1.1 do not exist. Therefore, the situations by which the conventional algorithms would be impeded from making adequate decisions

have been avoided. Further, because in the Cartesian complex definition, the topological space was defined first and then, on this, the geometric space, it is guaranteed that the topological properties present in the first are maintained in the geometric space.

The fact that the topological properties described in Section 2.1 are established solely on the basis of comparing combinatorial coordinates between cells of the space means that the requirement for more elaborated mathematical calculations is avoided, which results in more simple algorithms and reduced processing loads that can degrade performance. Further, the evaluation of subspace membership from the associated oriented matroid implies the verification established by Equation 3.12; however, according to the respective computational implementation established in Section 4.2, the matroid is calculated only once together with all the required mappings from the combinatorial coordinates, meaning that the evaluation of Equation 3.12 is performed only once for each element of the space defined by the associated SSF and not during its use to perform a required calculation on each 1-cell of the image space. Because the number m of 1-cells in the image space is considerably larger than the number n of 1-cells in the central SSF arrangement linked to the oriented matroid (i.e., $m >> n$), the degradation of performance by the evaluation of Equation 3.12 is small.

The hypothesis empirical testing involved first the design and construction of a computational prototype, which implements all the proposed concepts in Chapter 3, followed by the execution of Cartesian complex segmentation. Finally, outcome assessments were performed (Chapter 5) against three datasets. The first fact revealed during the design stage of the assessment test was the scarcity of earth surface hyper and multispectral datasets with ground truth segmentations for performing the evaluation of segmentation tests using this type of image. In the following are discussed the assessment outcomes presented in Chapter 5.

6.2.1 Segment scale generalization

The hyperspectral image segmentation test executed on the Pavia dataset (Section 5.3.1) demonstrated the use of the implemented hierarchical segmentation to obtain a measure of segment geometry quality and how to evaluate it using the image segmentation assessment framework used.

The results of this assessment allowed corroborating the acceptable quality of the probability Cartesian complex from which the hierarchical segmentation derived. This is evidenced by the fact that as the scale became coarser, the segments that remained were those along the lines of the ground truth. This can be deduced from Table 5.3, where the coincidence of these segments with the ground truth at different scales can be confirmed. This table displays how the border precision remained above 78% for the segments intersecting the ground truth connected region boundaries independently of the segmentation level. Figure 5.13 displays that the highest boundary strengths tended to follow the ground truth borders.

A ground truth covering report (see Table 5.4) indicated that if it could classify all of the segments correctly, it would have achieved a 0.92 semantic segmentation metric (Everingham et al., 2008) because of the effective boundary definition provided by the actual segmentation implementation.

6.2.2 Boundary detection benchmarking

Boundary detection assessment demonstrated the usability of the constructed computational prototype, with the completion of all the tasks established in Figure 2.22 along the entire segmentation pipeline, from the conventional pixel to Cartesian complex multispectral image conversion, until the final generation of the boundary probability Cartesian complexes (boundary detection). The boundary detection assessment based on benchmarking allowed to evaluate both the proposed model based on Cartesian complexes and their associated matroids as well as their computational implementation as an alternative to conventional pixel-based segmentation.

Regarding the orientated gradient calculation, in contrast to (Maire, 2009) which is based on conventional pixels, the use of 1-cells provides some advantages. Assessment outcomes for both the multispectral and textured images show that the use of 1-cells provides a better recall than the recall obtained with the use of the conventional pixels, and that the precision is only slightly lower. The lowest precision may be because the conventional pixel-based implementation performs a further fitting cylindrical parabola to elliptical patches process to counteract the phantom boundary effect of the gradient calculation, which improves precision (Maire, 2009). Although the computational prototype built during this research included this extra process, its results are not yet suitable, so it was not used. Despite the foregoing, the satisfactory results mentioned above make it worthwhile to address in the future the proper implementation of the fitting to counteract the phantom boundary effect in the Cartesian complex to improve current results.

Regarding the assessment of geometric and spectral boundary probabilities (strengths), in the case of geometrical probability the results show the same pattern of assessment outcomes of the gradient calculation because this probability is obtained from the linear combination of the gradients following the procedure established in Section 3.4.4. However, in the case of the boundary spectral probability the assessment outcomes show that the spectral probability obtained from the model proposed only provides a slightly higher recall (Figure 5.17 right).Results also show as for conventional pixel images the spectral probability, which is calculated by the analysis of affinity from the geometrical, is better in recall than the geometrical. However, for the case of Cartesian complex images, the results show that the opposite occurs. The above may be due to two facts: (1) the affinity analysis algorithm and (2) the final stage of histogram expansion.

Affinity analysis from a conventional pixel image attempts to find changes between regions (boundaries) by detecting the less-similar pixel along the line that connect to two others. These less-similar pixel is considered a member of some pixel linear array not parallel to the searching line (Shi and Malik, 2000; Leung and Malik, 1998). However, when this affinity analysis is performed from a Cartesian complex image, it is also attempted to detect the less-similar 1-cell between two others along a SSF, but in this case its membership to some 1-cell linear arrangement non-parallel to the searching SSF may not be entirely natural since the 1-cell detected is located along the SSF, i.e., parallel to it. A further task, after obtaining the eigenvalues and the respective eigenvector Cartesian complex, a directional filtering stage based on a Gaussian kernel (first derivative) is performed whose current definition may be preventing better results from being obtained.

As far as the spectral probability final stage of the histogram expansion is concerned, although it counteracts the effect of the few higher atypical values, it can also propitiate, on the one hand, the presence of false positives in high probabilities which causes the precision to decrease in the areas where it would expect it to be higher (higher precision and lower recall in the precision/recall curves), or on the other

hand, the absence of true positives in low probabilities which causes the recall to decrease in the zones in which it would be expected to be larger (lower precision and higher recall in the precision/recall curves).

The results of the global boundary probability assessment (Figure 5.18 and Table 5.9), show that the F-measures ODS and OIS were always better for Cartesian complex images. Furthermore, seeing all assessment outcomes can be evidenced that until before the spectral probability task, the boundary detection based on Cartesian complex images had better recall and ODS and OIS F-measures and the precision was only slightly lower. On the other hand, while the inclusion of the spectral probability had a positive effect on the conventional pixel representation, on the Cartesian complex representation was negative. Notwithstanding the foregoing, the prototype computational implementation produced as part of the present research allowed that the average precision for Potsdam, Vaihingen and complete dataset (Potsdam and Vaihingen) was better for the representation of the image as complex Cartesian.

As can be observed in Figures 5.15 to 5.18, the assessment process of proposed model based on benchmarking from 112 windows, corresponding to the dataset used, confirmed that although it is true that the precision produced from Cartesian complexes does not yet exceed that obtained from the conventional approach based on pixels, the proposed approach does achieve a superior recall and average precision. This confirms that the model proposed and implemented as part of this research constitutes a reliable alternative for the segmentation of multispectral images.

6.3 Discussion of computational prototype outcomes

In this section, aspects related to the execution of the segmentation process tasks and their results (Section 5.2) are discussed. Specifically, in boundary detection, it was possible to link filtering outcomes to 1D space elements (Figure 5.5) because they were available as inter-pixel 1-cells. The use of central SSF arrangements made it possible to constrain the filter and gradient orientations to only those available in the digital space. Because the maximum number of orientations depends on the established radius, in the subsequent texture grouping task, this parameter was set at “3” and “5” 1-cells at each scale to provide sufficient orientations to produce the detail displayed in Figure 5.6.

The calculation of oriented gradient Cartesian complexes benefits from not only the half-ball combinatorial formation from the subspaces described in Section 3.4.2 but also does not require searching the subspace that contains a particular 1-cell; rather, the 1-cell subspace is based on a subspace indexing from the position vector value introduced. Because a gradient ball (Equation 3.54) is created once along with its combinatorial structure, i.e., mapping from space element location to subspace, insofar as the radius increases, the improvement over the conventional implementation becomes more evident (see Table 5.5).

In agreement with Figure 5.7, the texture gradient provides an improved boundary definition compared to the multispectral gradient. To the extent that the scale becomes coarser, the boundary definition does as well and hence, some detail is lost. However, the loss of detail also leads to a reduction in the presence of false boundary definitions, whereas at coarser scales, only true boundary definitions remain, though, with the risk that some could be lost.

Regarding the definition of the boundary spectral probability (strength) based on the affinity analysis established in Sections 3.4.3 and 3.4.4, Figure 5.9 displays that the eigenvector Cartesian complex produced provides uniform areas, yet maintains the edges between image elements as reported in (Arbelaez et al.,

2011). Therefore, the subsequent first derivative Gaussian filter application manages to detect the edges with the oriented spectral probability (strength) displayed in Figure 5.10.

The use of ALFSs for representing the underlying space, which meets a topological space T_0, allowed the definition of connectivity relationships based on unambiguous topological concepts. This was the case of eventual basin expansion explorations in the watershed transform, used to produce the final hierarchical segmentation displayed in Figure 5.12. In this case, it was possible to evaluate each 1-cell in the subset of those that belonged to the SON union of the 1-cells included in the closure of the central 1-cell being evaluated (Figure 3.27), unlike the ambiguous situation to which the algorithms based on pixels are subject.

Chapter 7

Conclusions

Conducting the research presented herein demonstrated that the representation of images based on the ALFSs provided by Cartesian complexes not only leads to a space representation free of topological ambiguities but also confirms that the availability of inter-pixel space elements facilitates the definition of boundary detection based on the 1D elements (1-cells) available and the topological relationships present in that underlying representation.

As for the specific purpose "**To devise a conceptual model for multispectral image representation based on Cartesian complexes of abstract cells that considers topological and geometric properties**", this research managed to devise a conceptual model based on the Cartesian complexes (Chapter 3) through which the segmentation framework, proposed in (Arbelaez et al., 2011), was reformulated in terms of 1-cells. This allowed to define the tasks of the segmentation process using the available inter-pixel elements and to exploit the capabilities that the underlying model has to represent in an unambiguous way the topological and geometric properties. This reformulation demonstrated that there are topologically correct alternative methods of achieving results such as those achieved with conventional pixel representation.

Validation benchmarking results confirm that the proposed approach to boundary detection is similar to that proposed in (Arbelaez et al., 2011) and that the benefits of the inclusion of an underlying space based on Cartesian complex manage to be significant. Although the spectral probability results obtained from the representation of the space based on Cartesian Complexes are not always better than those obtained from the conventional representation of pixels, the progress made in the current research by providing benefits in processing and guaranteeing the topological model are evidence that it is worthwhile advancing further research in this area. The benefits of having a topologically correct representation of space and using combinatorial features through the association of oriented matroids highlights the interest of extending the implemented framework in the future to other processes of image analysis.

Regarding the specific objective "**To design a mechanism for encoding topological and geometric features present in combinatorial half-spaces using oriented matroids to reduce algorithmic complexity in multispectral image segmentation**", the definition of the central arrangement of SSFs became one of the most relevant concepts of research, because it served not only as an instrument to link the respective geometric subspaces with an oriented matroid but also became the construction block of the 1-cell-based definitions of several tasks in the segmentation process. The use of central SSF arrangements and their oriented matroids allows improving the oriented gradient computation because the respective

histograms are calculated once by sector and, in the final calculation, combined in agreement with each of the specific orientations.

The availability of an underlying Cartesian complex space provides a topologically correct oriented gradient calculation. In a conventional space, when a disk is defined, it is ambiguous to use the pixel linear arrangement that meets the diameter along the orientation because it does not lay in either of the two sides, rather in the orientation. In a Cartesian complex space there are inter-pixel elements that allow representing this situation properly by modeling the dividing line based on SSFs (i.e., from 1-cells) and not on pixels.

Moreover, the inter-pixel space element availability allows the application of filter kernels at 1D elements (1-cells), which agrees with the boundary feature dimensionality to be detected. Regarding multispectral and texture analysis, both in the calculation of the oriented gradient and in the multi-scale filtering, the results of the assessment based on benchmarking indicate that the proposed approach based on central 1-cells improves the average precision of the detected boundaries.

The boundary spectral probability assessment results show that the spectral probability obtained from the model proposed only gives a slightly higher recall. The results also show that for the proposed representation the spectral probability, which is calculated by the analysis of affinity from the geometric, is worse in recall than the geometric. However, the improvement in geometric probability achieved based on the proposed approach makes it worthwhile to continue with the research of better alternatives based on Cartesian complexes for affinity analysis and to improve the calculation of spectral probability.

On the other hand, the topological features available in underlying ALFS provided by Cartesian complexes allows the implementation of an alternative watershed transform procedure that can unambiguously make connectedness tests, e.g., basin expansions.

As regards the specific objective "**To build and evaluate a computational framework that enables the implementation of Cartesian complexes to adequately represent the topological and geometric image-object properties used in the segmentation of multispectral images**", although the computational solution obtained demonstrated that it is possible to build such a system, it is the result of an initial implementation and is open to improvements in light of clearer, deeper, and new knowledge. The proposed segmentation approach achieves improved recall and superior average accuracy, despite the limitations discussed in Sections 6.1.1 to 6.1.6. This constitutes evidence that the proposed research line based on Cartesian complexes must be continued to resolve the limitations raised and thus improve the results obtained with future improved versions of the computational implementation.

Chapter 8

Main contributions

The present research is original because it addresses a new subject, the reformulation of a computational framework for image segmentation, in terms of the inter-pixel elements available in the alternative representation model based on locally axiomatic finite spaces provided by Cartesian complexes. Further, it makes possible to link the oriented matroids to central SSF arrangements using the alternative space provided. This research constitutes a contribution to knowledge in the area of spatial information science, in relation to the representation models of digital images in general, and with image segmentation in particular.

Furthermore, from the previous achievements, the research contributes to the computing area with image segmentation algorithms, based on a topologically correct representation model, that can overcome the difficulties related to situations of ambiguity that face the algorithms that function on the conventional representation based on pixels. These algorithms were constructed and organized in a computational prototype, which can become an application interface, whose functions can be used together, or individually, for future developments, both in the specific field of image segmentation or in other areas of digital image analysis, as discussed in Section 8.2. The computational prototype construction produced an API written in Java™ whose source code can be found in Appendix A.

Although the research was undertaken to use the representation model based on Cartesian complexes including the 1-cells available, the concepts of central SSF arrangement and its associated oriented matroid can also be applied in the conventional model of pixels. Specifically, the generation of Prewitt filter banks and calculation of gradients oriented in conventional space can be implemented using these concepts.

The research presented here has provided contributions at both the conceptual and implementation levels. Further, the conducted research has contributed with papers and works, which are the result of numerous social interactions. In the following, Sections 8.1 and 8.2 describe the conceptual and implementation contributions. Section 8.3 describes the socialization contributions, and finally Section 8.4 describes the academic contribution.

8.1 Conceptual model

The main contributions of the research presented here have a conceptual and empirical scope; these are listed below.

- Conceptual model based on Cartesian complexes by means of which, from an alternative representation of image space, the tasks of the segmentation process are reformulated from the combinatorial coordinates of inter-pixel 1D elements;
- Conceptual model to link oriented matroids to central SSF arrangement based on the number of orientations and through which to encode the respective subspaces and alternatively to define the tasks of filtering and calculation of gradient oriented;
- Computational prototype based on the parallel computing framework implemented for the complete process of image segmentation. This prototype implements all the conceptual elements established by the conceptual models listed above.

8.2 Image analysis

Although the main purpose of the research was to provide an alternative method for the segmentation of multispectral images, having adopted the procedural framework proposed in (Arbelaez et al., 2011) (Section 2.3), demonstrates that it possible for several generic tasks to be reformulated in terms of one-dimensional inter-pixel space elements, i.e., 1-cells, which are summarized in the following.

8.2.1 Support for Cartesian complex space representation

The development of the research presented herein has allowed the construction of a framework for the transparent management of Cartesian complexes based on GDAL (GDAL Development Team, 2016). Hence, all that is necessary is to link a conventional image and start accessing the inter-pixel elements as required. If a Cartesian complex must be saved, it is only necessary to create the complex and specify the types of elements of the space that must be saved; the framework will complete the function.

This framework provides the topological and geometric elements described in Sections 3.1 and 3.2 and allows defining elements at more specific levels from the elements available here. For example, the referenced section was developed for the implementation of the watershed transform (Section 3.4.5).

8.2.2 1-cell filtering

Sections 3.4.1 and 4.3.3, respectively, describe the directional filtering logical model and computing implementation, which provides a "parameterizable" tool set for creating and applying 1-cell-centered directional filters in an individual manner or in a filter bank fashion. Prewitt, Gaussian, and customized kernel types are supported. The following tools are provided:

- Individual Prewitt and Gaussian kernel definition from given radius (in number of cracks) and orientation;
- Filter bank definition from given radii and central SSF arrangement-based number of orientations;
- Gaussian kernel definition based on no, first, or second derivative along any x or y axis;
- Multiscale Gaussian filter banks for texture analysis;

- Cartesian complex channel range filtering.

8.2.3 Oriented gradient calculation

Cartesian complex 1-cell-centered gradient calculation, whose logical model and computer implementation are, respectively, described in Sections 3.4.2 and 4.3.4, can be obtained for each channel or for the entire value vector from the radii and number of orientation-based central SSF arrangement and its oriented matroid. This calculation produces an oriented gradient channel for each orientation and two additional channels for the maximum oriented gradient and its respective orientation.

8.2.4 Affinity analysis

1-cell-based affinity analysis as described in Sections 3.4.3 (logical model) and 4.3.5 (computer implementation) is provided to be applied in a Cartesian complex based on a given radius and number of eigenvalues. The outcome is an eigenvector Cartesian complex with its respective eigenvalue set.

8.2.5 Boundary strength calculation

1-cell-oriented boundary strength measurements can be obtained from an input Cartesian complex using the boundary strength calculation provided for:

- Boundary-oriented multiscale geometric probability for RGB or multispectral images;
- Boundary-oriented spectral probability;
- Boundary-oriented global probability.

Boundary strength calculation tools are provided in agreement with those established in Sections 3.4.4 and 3.4.4.

8.2.6 Segmentation calculation

A 1-cell-oriented watershed calculation as described in Sections 3.4.5 (logical model) and 4.3.8 (computer implementation) is provided to be applied in a Cartesian complex based on providing a number of levels to obtain a hierarchical segmentation; when the number of levels is one, the outcome is the over-segmentation.

8.2.7 Octave software support

As mentioned in Section 4.3.5, a compiled version of the software *Octave* was generated to support the indexation of arrays based on 64-bit integer values. In this version of the software, the components for solving the eigenvector generalized problem were compiled and usable.

8.3 Socialization of research results

As part of the development of the research presented herein, several participations were made in international congresses and publications in national and international journals. The main contributions in this field are listed in chronological order below.

- An initial exploratory paper regarding the possible uses of oriented matroids in the context of geographic space representation models, which was titled *Modelo de representación del espacio geográfico mediante matroides orientados*, and published in the journal *UD y la Geomática* (Valero, 2013);
- A paper describing the implementation of a prototype REST web service for triangulation of point sets collected by mobile GPS receivers titled *A REST Service for Triangulation of Point Sets Using Oriented Matroids* and published in *the International Journal of Interactive Multimedia and Artificial Intelligence (IJIMAI)* (Valero and Lizarazo, 2014);
- A paper titled *Topological challenges in multispectral image segmentation* published in the *Tecnura* journal discussing the initial findings of the literature review and the state of the art of the research (Valero, Lizarazo, and Elsner, 2014);
- Participation in the international congress *XVI Simposio Internacional Selper 2014* at Medellín, Colombia, with a work titled *Componente para convertir una imagen binaria a un espacio axiomático finito localmente (AFS)*;
- Participation in the international congress *GEOBIA 2016 SOLUTIONS & SYNERGIES conference* at Enschede, Netherlands, presenting the work titled *Multispectral image segmentation based on Cartesian complexes and their associated oriented matroids*, which evaluates whether an alternative representation of digital images based both on Cartesian complexes and oriented matroids can improve multispectral image segmentation by enforcing topological and geometric properties and subsequently be used in the classification stage (Valero, Lizarazo, and Arbelaez, 2016);
- Participation in *Computer Vision – ACCV 2016 Workshops: ACCV 2016 International Workshops* in Taipei, Taiwan, presenting the work titled *A Combinatorial Approach for Hyperspectral Image Segmentation* that was published in *Lecture Notes in Computer Science* Journal of Springer International Publishing (Valero, Arbeláez, and Lizarazo, 2017);
- A paper titled *Un nuevo enfoque para la clasificación de imágenes multiespectrales basado en complejos cartesianos* and published in *DYNA* journal proposing a new approach to classification of images using Cartesian complexes as a representation of space, free of ambiguity in topological relationships (Valero, Dallos, and Lizarazo, 2018);
- Acceptance of application for participation in the international congress *GEOBIA 2018 GEOBIA IN A CHANGING WORLD conference* in Montpellier, France, presenting the work titled *Multispectral image classification from axiomatic locally finite spaces-based segmentation* regarding segmentations conducted using a novel approach based on axiomatic locally finite spaces (ALFS) and their linked oriented matroids.

8.4 Academic contribution

Part the research was the achievement of a master research work entitled *Modelo para clasificación de imágenes multiespectrales a partir de los complejos de células abstractas*. The main purpose of this work was to propose a multispectral image classification model that included the topological component provided by the complexes of abstract cells. From this work, the initial version of the following functions were obtained:

- EquNaLi membership rule implementation;
- Gray scale Cartesian complex conversion;
- Prewitt filtering for seven orientations, one scale, and two 1-cells of radius;
- K-means grouping with centers at long of minimum bounding box;
- Over-segmented watershed.

Chapter 9

Future work and recommendations

As part of the conducted research, the image segmentation process and its related tasks were the only ones approached from an image represented as a Cartesian complex, demonstrating the benefits of having a topologically correct digital space representation. However, the usefulness of this representation in other areas of image analysis must also be evaluated. Because the underlying representation space is valid for spatial analysis in images in general and the earth's surface in particular, a wide range of possibilities exist.

During the assessment and test design stage, it was realized that there was a scarcity of earth surface hyper and multispectral datasets with ground truth segmentations for performing evaluation tests of segmentation processes applied to this type of image. Therefore, it is necessary to undertake an effort to produce datasets from hyper and multispectral earth surface images with ground truths sufficient for use in segmentation benchmarking.

During the definition of the k-means grouping algorithm, the determination of the initial class center was a topic of great interest. The current implementation supports two approaches: (1) obtain the group centers by distributing them along the hyper-diagonal between the minimum and maximum of the hyper-dimensional minimum box and (2) selecting the initial centers randomly from the dataset to be grouped. As a future work, the problem of determining the hyper-dimensional line of best fit to the data to be grouped and distributing the initial class centers along this should be addressed.

The use of strategies to obtain gray images based on alternative techniques such as wavelet transform from a Cartesian complex space representation, would contribute to a better initial task of gray scale conversion for later filtering.

The computational implementation was performed using a parallel computing framework, which allowed the performance to be improved to the extent that a greater number of processing units (CPU) were available. In this regard, it is recommended to conduct research aimed at proposing the functions that can be performed by graphical processing units (GPUs) and thus significantly increasing the possibilities of more parallel processes. Further, a distributed implementation on a weakly coupled architecture, e.g., using grid and cloud computing, must be explored.

The conducted research focused on the evaluation of Cartesian complexes as an alternative representation of geometric space. It is recommended that research be initiated to extend the Cartesian complexes to also represent feature spaces, e.g., spectral bands.

In (Arbelaez et al., 2011), a gradient calculation method based on integral images using pie slice orientation evenly spaced is proposed. It is recommended that this be further explored for use with the oriented matroid-based method to perhaps achieve the best of the two, in only one method.

Options to reformulate several processing intermediate tasks of fitting cylindrical parabolas to elliptical patches at each 1-cell after computing an oriented gradient to counteract the phantom border effect (i.e., to avoid producing borders which do not exist), should be explored (Malik et al., 2001).

The current implementation of the spectral probability does not yet provide an improvement in the boundary detection, so a research should be undertaken to improve the definition of the affinity analysis and filtering based on 1-cell Gaussian kernel phases. It is also necessary to address the problem of outliers that appear in the Cartesian complex of spectral boundary probability. In this respect, an initial solution to be evaluated is to create a Cartesian complex histogram expansion with a suitable highest saturation value.

It is recommended to undertake research to develop a Cartesian complex viewer, which suitably represents the inter-pixel space elements.

Because the concepts of the central SSF arrangement and its associated oriented matroid produced in this research can also be applied in the conventional pixel representation, conducting research for exploring its use in that conventional representation is recommended, specifically in the generation of Prewitt filter banks and calculation of gradients oriented.

Bibliography

Arbelaez, P. et al. (2009). "From contours to regions: An empirical evaluation." In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 2294–2301. DOI: 10.1109/CVPR.2009.5206707.

Arbelaez, Pablo (2006). "Boundary extraction in natural images using ultrametric contour maps." In: *In Proc. POCV*.

Arbelaez, Pablo et al. (2011). "Contour Detection and Hierarchical Image Segmentation." In: *IEEE Trans. Pattern Anal. Mach. Intell.* 33.5, pp. 898–916. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2010.161. URL: http://dx.doi.org/10.1109/TPAMI.2010.161.

Baatz, M and A Schape (2000). "Multiresolution segmentation: An optimization approach for high quality multi-scale image segmentation." In: *In Angewandte Geographische Informationsverarbeitung XII. Beitrage zum AGIT-Symposium Salzburg 2000; Herbert Wichmann Verlag*, pp. 12–23.

Baatz, Martin and Arno Schäpe (1999). "Object-Oriented and Multi-Scale Image Analysis in Semantic Networks." In: *Proc. of the 2nd International Symposium on Operationalization of Remote Sensing*. Enschede, ITC.

Bachem, Achim and Walter Kern (1992). *Linear Programming Duality. An Introduction to Oriented Matroids*. Springer Berlin Heidelberg. URL: http://link.springer.com/book/10.1007/978-3-642-58152-6.

Belongie, Serge and Jitendra Malik (1998). "Finding Boundaries in Natural Images: A New Method Using Point Descriptors and Area Completion." In: *Proceedings of the 5th European Conference on Computer Vision-Volume I - Volume I*. ECCV '98. London, UK, UK: Springer-Verlag, pp. 751–766. ISBN: 3-540-64569-1. URL: http://dl.acm.org/citation.cfm?id=645311.649050.

Benz, Ursula C. et al. (2004). "Multi-resolution, object-oriented fuzzy analysis of remote sensing data for GIS-ready information." In: *{ISPRS} Journal of Photogrammetry and Remote Sensing* 58.34. Integration of Geodata and Imagery for Automated Refinement and Update of Spatial Databases, pp. 239–258. ISSN: 0924-2716. DOI: http://dx.doi.org/10.1016/j.isprsjprs.2003.10.002. URL: http://www.sciencedirect.com/science/article/pii/S0924271603000601.

Berg, Mark de et al. (2008). *Computational Geometry*. Third. Springer Berlin Heidelberg, pp. 151–171. URL: link.springer.com/book/10.1007/978-3-540-77974-2.

Bertrand, Gilles and Michel Couprie (1999). "A Model for Digital Topology." In: *Discrete Geometry for Computer Imagery*. Ed. by Gilles Bertrand, Michel Couprie, and Laurent Perroton. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 229–241. ISBN: 978-3-540-49126-2.

Beucher, S. and F. Meyer (1992). "The Morphological approach of segmentation: the watershed transformation." In: *Mathematical Morphology in Image Processing*. Ed. by E. Dougherty. Optical Science and Engineering. Taylor & Francis, pp. 433–481. ISBN: 9780824787240.

Blaschke, T. (2009). "Object based image analysis for remote sensing." In: *{ISPRS} Journal of Photogrammetry and Remote Sensing* 65.1, pp. 2 –16. ISSN: 0924-2716. DOI: http://dx.doi.org/10.

1016/j.isprsjprs.2009.06.004. URL: http://www.sciencedirect.com/science/article/pii/S0924271609000884.

Borovik, Alexandre V., I. M. Gelfand, and Neil White (2003). *Coxeter Matroids*. Birkh'üser Boston. URL: http://link.springer.com/book/10.1007/978-1-4612-2066-4.

Boykov, Yuri and Gareth Funka-Lea (2006). "Graph Cuts and Efficient N-D Image Segmentation." In: *International Journal of Computer Vision* 70.2, pp. 109–131. ISSN: 1573-1405. DOI: 10.1007/s11263-006-7934-5. URL: https://doi.org/10.1007/s11263-006-7934-5.

Brice, Claude R. and Claude L. Fennema (1970). "Scene analysis using regions." In: *Artificial Intelligence* 1.3, pp. 205 –226. ISSN: 0004-3702. DOI: http://dx.doi.org/10.1016/0004-3702(70)90008-1. URL: http://www.sciencedirect.com/science/article/pii/0004370270900081.

Brun, Luc, Jean-Philippe Domenger, and Myriam Mokhtari (2003). "Incremental modifications of segmented image defined by discrete maps." In: *Journal of Visual Communication and Image Representation* 14.3, pp. 251 –290. ISSN: 1047-3203. DOI: http://dx.doi.org/10.1016/S1047-3203(03)00023-3. URL: http://www.sciencedirect.com/science/article/pii/S1047320303000233.

Bunge, M. (2004). *La investigación científica: su estrategia y su filosofía*. Filosofía (Siglo XXI Editores). Siglo XXI. ISBN: 9789682322259. URL: https://books.google.com.co/books?id=iDjRhR82JHYC.

Canny, J. (1986). "A Computational Approach to Edge Detection." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8.6, pp. 679–698. ISSN: 0162-8828. DOI: 10.1109/TPAMI.1986.4767851.

Cheevasuvit, Fusak, Henri Maitre, and Daniel Vidal-Madjar (1986). "A robust method for picture segmentation based on a split-and-merge procedure." In: *Computer Vision, Graphics, and Image Processing* 34.3, pp. 268 –281. ISSN: 0734-189X. DOI: http://dx.doi.org/10.1016/S0734-189X(86)80042-1. URL: http://www.sciencedirect.com/science/article/pii/S0734189X86800421.

Chen, C., D. Freedman, and C. H. Lampert (2011). "Enforcing topological constraints in random field image segmentation." In: *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 2089–2096. DOI: 10.1109/CVPR.2011.5995503.

Chi, M. et al. (2016). "Big Data for Remote Sensing: Challenges and Opportunities." In: *Proceedings of the IEEE* 104.11, pp. 2207–2219. ISSN: 0018-9219. DOI: 10.1109/JPROC.2016.2598228.

Comaniciu, Dorin and Peter Meer (2002). "Mean Shift: A Robust Approach Toward Feature Space Analysis." In: *IEEE Trans. Pattern Anal. Mach. Intell.* 24.5, pp. 603–619. ISSN: 0162-8828. DOI: 10.1109/34.1000236. URL: http://dx.doi.org/10.1109/34.1000236.

Cracknell, A. P. (1998). "Review article Synergy in remote sensing-what's in a pixel?" In: *International Journal of Remote Sensing* 19.11, pp. 2025–2047. DOI: 10.1080/014311698214848. eprint: http://dx.doi.org/10.1080/014311698214848. URL: http://dx.doi.org/10.1080/014311698214848.

Cramer, Michael (2010). *The DGPF-Test on Digital Airborne Camera Evaluation Overview and Test Design*. http://www.ifp.uni-stuttgart.de/dgpf/DKEP-Allg.html. DOI: doi:10.1127/1432-8364/2010/0041. URL: http://www.ifp.uni-stuttgart.de/dgpf/DKEP-Allg.html.

De Loera, Jesus A., Jorg Rambau, and Francisco Santos (2010). *Triangulations: Structures for Algorithms and Applications*. 1st. Springer Publishing Company, Incorporated. ISBN: 3642129706, 9783642129704. URL: http://dl.acm.org/citation.cfm?id=1952022.

Delucchi, Emanuele (2003). "On generalizing oriented matroids to a complex setting." PhD thesis. ETH Zurich, p. 50.

Digabel, H. and Ch. Lantuéjoul (1978). "Iterative algorithms." In: 2^{eme} *European Symposium on Quantitative Analysis of Microstructures in Material Science, Biologie and Medecine*. Ed. by J-L. Chermant. Stuttgart, pp. 85–99.

Dollar, P., Zhuowen Tu, and S. Belongie (2006). "Supervised Learning of Edges and Object Boundaries." In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. Vol. 2, pp. 1964–1971. DOI: 10.1109/CVPR.2006.298.

Eaton, John W. et al. (2017). *GNU Octave version 4.4.0 manual: a high-level interactive language for numerical computations*. URL: https://www.gnu.org/software/octave/doc/v4.4.0/.

Everingham, M. et al. (2008). *The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results*. http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html.

Felzenszwalb, Pedro F. and Daniel P. Huttenlocher (2004). "Efficient Graph-Based Image Segmentation." In: *Int. J. Comput. Vision* 59.2, pp. 167–181. ISSN: 0920-5691. DOI: 10.1023/B:VISI.0000022288.19776.77. URL: http://dx.doi.org/10.1023/B:VISI.0000022288.19776.77.

Folkman, Jon and Jim Lawrence (1978). "Oriented matroids." In: *Journal of Combinatorial Theory, Series B* 25.2, pp. 199 –236. ISSN: 0095-8956. DOI: https://doi.org/10.1016/0095-8956(78)90039-4. URL: http://www.sciencedirect.com/science/article/pii/0095895678900394.

Freeman, W. T. and E. H. Adelson (1991). "The design and use of steerable filters." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13.9, pp. 891–906. ISSN: 0162-8828. DOI: 10.1109/34.93808.

Fukuda, K. (2004). *Lecture notes on oriented matroids and geometric computation*. Tech. rep. RO-2004.0621, course of Doctoral school in Discrete System Optimization, EPFL 2004. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.87.3924&rep=rep1\&type=pdf.

GDAL Development Team (2016). *GDAL - Geospatial Data Abstraction Library, Version 2.1.0*. Open Source Geospatial Foundation. URL: ?http://www.gdal.org.

Glasbey, C. A. and G. W. Horgan (1995). *Image Analysis for the Biological Sciences*. New York, NY, USA: John Wiley & Sons, Inc. ISBN: 0-471-93726-6.

Horowitz, Steven L. and Theodosios Pavlidis (1976). "Picture Segmentation by a Tree Traversal Algorithm." In: *J. ACM* 23.2, pp. 368–388. ISSN: 0004-5411. DOI: 10.1145/321941.321956. URL: http://doi.acm.org/10.1145/321941.321956.

Ishikawa, Hiroshi (2012). "Graph CutsCombinatorial Optimization in Vision." In: *Image Processing and Analysis with Graphs*. Digital Imaging and Computer Vision. CRC Press. DOI: 10.1201/b12281-3. URL: http://dx.doi.org/10.1201/b12281-3.

Johansen, Kasper et al. (2007). "Application of high spatial resolution satellite imagery for riparian and forest ecosystem classification." In: *Remote Sensing of Environment* 110.1, pp. 29 –44. ISSN: 0034-4257. DOI: http://dx.doi.org/10.1016/j.rse.2007.02.014. URL: http://www.sciencedirect.com/science/article/pii/S0034425707000818.

Jong, Stephanus Michaël de and Freek D. Van der Meer, eds. (2004). *Remote sensing image analysis : including the spatial domain*. Remote sensing and digital image processing. La couv. porte : CD-ROM included. Dordrecht, Boston: Kluwer Academic. ISBN: 1-402-02559-9. URL: http://opac.inria.fr/record=b1121779.

Kaewpijit, S., J. Le Moigne, and T. El-Ghazawi (2003). "Automatic reduction of hyperspectral imagery using wavelet spectral analysis." In: *IEEE Transactions on Geoscience and Remote Sensing* 41.4, pp. 863–871. ISSN: 0196-2892. DOI: 10.1109/TGRS.2003.810712.

Kass, Michael, Andrew Witkin, and Demetri Terzopoulos (1988). "Snakes: Active contour models." In: *International Journal of Computer Vision* 1.4, pp. 321–331. ISSN: 1573-1405. DOI: 10.1007/BF00133570. URL: https://doi.org/10.1007/BF00133570.

Khalimsky, Efim, Ralph Kopperman, and Paul R. Meyer (1990). "Computer graphics and connected topologies on finite ordered sets." In: *Topology and its Applications* 36.1, pp. 1 –17. ISSN: 0166-8641. DOI: https://doi.org/10.1016/0166-8641(90)90031-V. URL: http://www.sciencedirect.com/science/article/pii/016686419090031V.

"Axioms and Hulls" (1992). In: *Axioms and Hulls*. Ed. by Donald E. Knuth. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 1–98. ISBN: 978-3-540-47259-9. DOI: 10.1007/3-540-55611-7_1. URL: https://doi.org/10.1007/3-540-55611-7_1.

Kong, Chunfang, Kai Xu, and Chonglong Wu (2006). "Classification and Extraction of Urban Land-Use Information from High-Resolution Image Based on Object Multi-features." In: *Journal of China University of Geosciences* 17.2, pp. 151 –157. ISSN: 1002-0705. DOI: http://dx.doi.org/10.1016/S1002-0705(06)60021-6. URL: http://www.sciencedirect.com/science/article/pii/S1002070506600216.

Kong, T.Y. and A. Rosenfeld (1991). "Digital topology: a comparison of the graph-based and topological approaches. G.M. Reed, A.W. Roscoe, R.F. Wachter (Eds.), Topology and Category Theory in Computer Science, Clarendon Press, Oxford, Great Britain." In: pp. 273–287.

Kovalevsky, V. (2005). "Algorithms in Digital Geometry Based on Cellular Topology." In: *Combinatorial Image Analysis: 10th International Workshop, IWCIA 2004, Auckland, New Zealand, December 1-3, 2004. Proceedings*. Ed. by Reinhard Klette and Joviša Žunić. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 366–393. ISBN: 978-3-540-30503-3. DOI: 10.1007/978-3-540-30503-3_27. URL: http://dx.doi.org/10.1007/978-3-540-30503-3_27.

Kovalevsky, V. A. (1989). "Finite Topology as Applied to Image Analysis." In: *Computer Vision, Graphics, and Image Processing*, pp. 141–161.

Kovalevsky, Vladimir (2001). "Algorithms and Data Structures for Computer Topology." In: *Digital and Image Geometry: Advanced Lectures*. Ed. by Gilles Bertrand, Atsushi Imiya, and Reinhard Klette. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 38–58. ISBN: 978-3-540-45576-9. DOI: 10.1007/3-540-45576-0_3. URL: http://dx.doi.org/10.1007/3-540-45576-0_3.

– (2006). "Axiomatic Digital Topology." In: *J. Math. Imaging Vis.* 26.1-2, pp. 41–58. ISSN: 0924-9907. DOI: 10.1007/s10851-006-7453-6. URL: http://dx.doi.org/10.1007/s10851-006-7453-6.

Kovalevsky, Vladimir A (1984). "Discrete Topology and Contour Definition." In: *Pattern Recogn. Lett.* 2.5, pp. 281–288. ISSN: 0167-8655. DOI: 10.1016/0167-8655(84)90015-1. URL: http://dx.doi.org/10.1016/0167-8655(84)90015-1.

Kovalevsky, Vladimir A. (2008). "Geometry of Locally Finite Spaces." In: *International Journal of Shape Modeling* 14.02, pp. 231–232. DOI: 10.1142/S0218654308001178. eprint: http://www.worldscientific.com/doi/pdf/10.1142/S0218654308001178. URL: http://www.worldscientific.com/doi/abs/10.1142/S0218654308001178.

Kumar K., Abello J. Egecioglu O. (1995). "Visibility Graphs of Staircase Polygons and the Weak Bruhat Order, I: from Visibility Graphs to Maximal Chains." In: *Discrete & computational geometry* 14.3, pp. 331–358. URL: http://eudml.org/doc/131406.

Letscher, David and Jason Fritts (2007). "Image Segmentation Using Topological Persistence." In: *Computer Analysis of Images and Patterns: 12th International Conference, CAIP 2007, Vienna, Austria, August 27-29, 2007. Proceedings*. Ed. by Walter G. Kropatsch, Martin Kampel, and Allan Hanbury.

Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 587–595. ISBN: 978-3-540-74272-2. DOI: 10.1007/978-3-540-74272-2_73. URL: http://dx.doi.org/10.1007/978-3-540-74272-2_73.

Leung, Thomas and Jitendra Malik (1998). "Contour continuity in region based image segmentation." In: *Computer Vision — ECCV'98*. Ed. by Hans Burkhardt and Bernd Neumann. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 544–559. ISBN: 978-3-540-69354-3.

Leung, Thomas K. and Jitendra Malik (1996). "Detecting, Localizing and Grouping Repeated Scene Elements from an Image." In: *Proceedings of the 4th European Conference on Computer Vision-Volume I - Volume I*. ECCV '96. London, UK, UK: Springer-Verlag, pp. 546–555. ISBN: 3-540-61122-3. URL: http://dl.acm.org/citation.cfm?id=645309.648902.

Lézoray, O. and L. Grady (2012). *Image Processing and Analysis with Graphs: Theory and Practice*. Ed. by O. Lézoray and L. Grady. Digital Imaging and Computer Vision. CRC Press / Taylor and Francis, p. 562. URL: https://www.taylorfrancis.com/books/e/9781439855089.

Li, C. and Y. Sun (2010). "Active image: A shape and topology preserving segmentation method using B-spline free form deformations." In: *2010 IEEE International Conference on Image Processing*, pp. 2221–2224. DOI: 10.1109/ICIP.2010.5652762.

Listing, Johann Benedict (1862). "der Census räumlicher Complexe, oder Verallgemeinerung des Euler´schen Satzes von den Polyädern." In: *Abhandlungen der Königlichen Gesellschaft der Wissenschaften in Göttingen* 10, pp. 97–182. URL: http://eudml.org/doc/135737.

Liu, Peng et al. (2018). "Remote Sensing Big Data: Theory, Methods and Applications." In: *Remote Sensing* 10.5. ISSN: 2072-4292. DOI: 10.3390/rs10050711. URL: http://www.mdpi.com/2072-4292/10/5/711.

Lizarazo, I. and P. Elsner (2009). "Fuzzy segmentation for object-based image classification." In: *International Journal of Remote Sensing* 30.6, pp. 1643–1649. DOI: 10.1080/01431160802460062. eprint: http://dx.doi.org/10.1080/01431160802460062. URL: http://dx.doi.org/10.1080/01431160802460062.

Lizarazo, Ivan and Paul Elsner (2011). "Segmentation of Remotely Sensed Imagery: Moving from Sharp Objects to Fuzzy Regions." In: *Image Segmentation*. Ed. by Pei-Gee Ho. Rijeka: InTech. Chap. 13. DOI: 10.5772/15421. URL: https://doi.org/10.5772/15421.

Lloyd, S. (1982). "Least squares quantization in PCM." In: *IEEE Transactions on Information Theory* 28.2, pp. 129–137. ISSN: 0018-9448. DOI: 10.1109/TIT.1982.1056489.

Maire, Michael Randolph (2009). "Contour Detection and Image Segmentation." AAI3410970. PhD thesis. Berkeley, CA, USA. ISBN: 978-1-124-03088-3.

Malik, Jitendra et al. (2001). "Contour and Texture Analysis for Image Segmentation." In: *International Journal of Computer Vision* 43.1, pp. 7–27. ISSN: 1573-1405. DOI: 10.1023/A:1011174803800. URL: https://doi.org/10.1023/A:1011174803800.

Martin, D. et al. (2001). "A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics." In: *Proc. 8th Int'l Conf. Computer Vision*. Vol. 2, pp. 416–423.

Martin, David R., Charless C. Fowlkes, and Jitendra Malik (2004). "Learning to Detect Natural Image Boundaries Using Local Brightness, Color, and Texture Cues." In: *IEEE Trans. Pattern Anal. Mach. Intell.* 26.5, pp. 530–549. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2004.1273918. URL: http://dx.doi.org/10.1109/TPAMI.2004.1273918.

Mortensen, Eric N. and William A. Barrett (1995). "Intelligent Scissors for Image Composition." In: *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '95. New York, NY, USA: ACM, pp. 191–198. ISBN: 0-89791-701-4. DOI: 10.1145/218380.218442. URL: http://doi.acm.org/10.1145/218380.218442.

Müller, A., P. Gege, and T. Cocks (2001). "The Airborne Imaging Spectrometers Used in DAISEX." In: *The Digital Airborne Spectrometer Experiment (DAISEX)*. Ed. by M. Wooding and R. A. Harris. Vol. 499. ESA Special Publication, p. 3.

Munkres, J.R. (1999). *Topology; a First Course*. 2nd edition. Prentice-Hall. ISBN: 9780139254956. URL: https://books.google.com.co/books?id=LtEPAQAAMAAJ.

Najman, Laurent and Michel Schmitt (1996). "Geodesic Saliency of Watershed Contours and Hierarchical Segmentation." In: *IEEE Trans. Pattern Anal. Mach. Intell.* 18.12, pp. 1163–1173. ISSN: 0162-8828. DOI: 10.1109/34.546254. URL: https://doi.org/10.1109/34.546254.

Nunez-Iglesias, Juan et al. (2013). "Machine Learning of Hierarchical Clustering to Segment 2D and 3D Images." In: *PLoS ONE* 8.8, pp. 1–11. DOI: 10.1371/journal.pone.0071715. URL: http://dx.doi.org/10.1371%2Fjournal.pone.0071715.

Osher, Stanley and Nikos Paragios (2003). "Geometric Level Set Methods in Imaging, Vision, and Graphics." In: *Springer New York*.

Oxley, James (2003). "What is a matroid?" In: URL: https://www.math.lsu.edu/~oxley/survey4.pdf.

Oxley, James G. (2006). *Matroid Theory (Oxford Graduate Texts in Mathematics)*. New York, NY, USA: Oxford University Press, Inc. ISBN: 0199202508.

Pavlidis, T. (1977). *Structural Pattern Recognition*. Springer Series in Electronics and Photonics. Springer Berlin Heidelberg. ISBN: 9783642883040. URL: https://books.google.com.co/books?id=i-vrCAAAQBAJ.

Pfeifle, Julian and Jörg Rambau (2002). "Computing Triangulations Using Oriented Matroids." English. In: *Algebra, Geometry and Software Systems*. Ed. by Michael Joswig and Nobuki Takayama. Springer Berlin Heidelberg, pp. 49–75. ISBN: 978-3-642-05539-3. DOI: 10.1007/978-3-662-05148-1_3. URL: http://dx.doi.org/10.1007/978-3-662-05148-1_3.

Pitsoulis, Leonidas (2014). *Topics in Matroid Theory*. 1st. Springer New York. ISBN: 9781461489566, 9781461489573. URL: http://link.springer.com/book/10.1007/978-1-4614-8957-3.

Plaza, A. et al. (2005). "Dimensionality reduction and classification of hyperspectral image data using sequences of extended morphological transformations." In: *IEEE Transactions on Geoscience and Remote Sensing* 43.3, pp. 466–479. ISSN: 0196-2892. DOI: 10.1109/TGRS.2004.841417.

Pont-Tuset, Jordi et al. (2015). "Multiscale Combinatorial Grouping for Image Segmentation and Object Proposal Generation." In: *CoRR* abs/1503.00848. arXiv: 1503.00848. URL: http://arxiv.org/abs/1503.00848.

Prewitt, Judith MS (1970). "Object enhancement and extraction." In: *Picture processing and Psychopictorics* 10.1, pp. 15–19.

Ren, J. et al. (2014). "Effective Feature Extraction and Data Reduction in Remote Sensing Using Hyperspectral Imaging [Applications Corner]." In: *IEEE Signal Processing Magazine* 31.4, pp. 149–154. ISSN: 1053-5888. DOI: 10.1109/MSP.2014.2312071.

Richter-Gebert, Jürgen and Günter M. Ziegler (2004). "Oriented Matroids." In: *Handbook of Discrete and Computational Geometry*. Ed. by Jacob E. Goodman and Joseph O'Rourke. Boca Raton, FL, USA: CRC Press, Inc., pp. 129–151. ISBN: 1584883014. URL: http://www.crcpress.com/product/isbn/9781584883012.

Rinow, W. (1975). *Textbook of topology*. Hochschulbücher für Mathematik. Deutscher Verlag d. Wiss., VEB. URL: https://books.google.com.co/books?id=_IvQAAAAMAAJ.

Rosenfeld, Azriel (1970). "Connectivity in Digital Pictures." In: *J. ACM* 17.1, pp. 146–160. ISSN: 0004-5411. DOI: 10.1145/321556.321570. URL: http://doi.acm.org/10.1145/321556.321570.

Rubner, Yossi et al. (2001). "Empirical Evaluation of Dissimilarity Measures for Color and Texture." In: *Computer Vision and Image Understanding* 84.1, pp. 25 –43. ISSN: 1077-3142. DOI: https://doi.org/10.1006/cviu.2001.0934. URL: http://www.sciencedirect.com/science/article/pii/S1077314201909348.

Saeedi, Noushin (2012). "On fully characterizing terrain visibility graphs." PhD thesis. University of British Columbia. DOI: http://dx.doi.org/10.14288/1.0052141. URL: https://open.library.ubc.ca/cIRcle/collections/24/items/1.0052141.

Santos, Francisco (1997). *Triangulations Of Oriented Matroids*. URL: https://www.math.ucdavis.edu/~deloera/MISC/BIBLIOTECA/trunk/Santos/Santos9.pdf.

Savitzky A. & Golay, M.J.E (1964). "Smoothing and differentiation of data by simplified least squares procedures." In: *Analytical Chemistry* 36, pp. 1627–1639.

Schneider, Markus (1997). *Spatial Data Types for Database Systems: Finite Resolution Geometry for Geographic Information Systems*. Lecture Notes in Computer Science 1288. Springer Berlin Heidelberg. URL: http://link.springer.com/book/10.1007/BFb0028319.

Shi, Jianbo and Jitendra Malik (2000). "Normalized Cuts and Image Segmentation." In: *IEEE Trans. Pattern Anal. Mach. Intell.* 22.8, pp. 888–905. ISSN: 0162-8828. DOI: 10.1109/34.868688. URL: https://doi.org/10.1109/34.868688.

Smyth, Michael B. and Julian Webster (2007). "Discrete Spatial Models." In: *Handbook of Spatial Logics*. Ed. by Marco Aiello, Ian Pratt-Hartmann, and Johan Benthem. Springer Netherlands, pp. 713–798. ISBN: 978-1-4020-5586-7. DOI: 10.1007/978-1-4020-5587-4_12. URL: http://dx.doi.org/10.1007/978-1-4020-5587-4_12.

Stell, John and Julian Webster (2007). "Oriented matroids as a foundation for space in GIS." In: *Computers, Environment and Urban Systems* 31.4. Topology and Spatial Databases, pp. 379 –392. ISSN: 0198-9715. DOI: http://dx.doi.org/10.1016/j.compenvurbsys.2006.03.002. URL: http://www.sciencedirect.com/science/article/pii/S0198971507000178.

Stillwell, J. (1995). *Classical Topology and Combinatorial Group Theory*. Classical Topology and Combinatorial Group Theory. Springer. ISBN: 9780387979700. URL: https://books.google.com.co/books?id=265lbM42REMC.

Szeliski, Richard (2010). *Computer Vision: Algorithms and Applications*. 1st. New York, NY, USA: Springer-Verlag New York, Inc. ISBN: 1848829345, 9781848829343.

Tso, B. and P. Mather (2009). *Classification Methods for Remotely Sensed Data*. Environmental engineering. CRC Press, Taylor & Francis Group. ISBN: 9781420090741. URL: https://books.google.nl/books?id=oXpzROm11QMC.

Urquhart, Roderick (1982). "Graph theoretical clustering based on limited neighbourhood sets." In: *Pattern Recognition* 15.3, pp. 173 –187. ISSN: 0031-3203. DOI: http://dx.doi.org/10.1016/0031-3203(82)90069-3. URL: http://www.sciencedirect.com/science/article/pii/0031320382900693.

Valero, J.A., I. Lizarazo, and P.A. Arbelaez (2016). "Multispectral image segmentation based on Cartesian complexes and their associated oriented matroids." In: URL: http://proceedings.utwente.nl/442/.

Valero, José, Pablo Arbeláez, and Ivan Lizarazo (2017). "A Combinatorial Approach for Hyperspectral Image Segmentation." In: *Computer Vision – ACCV 2016 Workshops: ACCV 2016 International Workshops, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part I*. Ed. by Chu-Song Chen, Jiwen Lu, and Kai-Kuang Ma. Cham: Springer International Publishing, pp. 334–348. ISBN: 978-3-319-54407-6. DOI: 10.1007/978-3-319-54407-6_22. URL: https://doi.org/10.1007/978-3-319-54407-6_22.

Valero, José, Cristian Dallos, and Ivan Lizarazo (2018). "Un nuevo enfoque para la clasificación de imágenes multiespectrales basado en complejos cartesianos." In: *DYNA* 85.204, pp. 28–37. ISSN: 2346-2183. DOI: 10.15446/dyna.v85n204.66161. URL: https://revistas.unal.edu.co/index.php/dyna/article/view/66161.

Valero, Jose and Ivan Lizarazo (2014). "A REST Service for Triangulation of Point Sets Using Oriented Matroids." In: *International Journal of Interactive Multimedia and Artificial Intelligence* 2.6, pp. 33–40. ISSN: 1989-1660. DOI: 10.9781/ijimai.2014.264. URL: http://www.ijimai.org/journal/sites/default/files/files/2014/03/ijimai20142_6_4_pdf_14209.pdf.

Valero, José Antonio (2013). "Modelo de representación del espacio geográfico mediante matroides orientados." In: *UD y la Geomática* 0.7. ISSN: 2344-8407.

Valero, José, Iván Lizarazo, and Paul Elsner (2014). "Topological challenges in multispectral image segmentation." In: *Tecnura* 18.0, pp. 136–149. ISSN: 2248-7638. DOI: 10.14483/22487638.9250. URL: https://revistas.udistrital.edu.co/ojs/index.php/Tecnura/article/view/9250.

Vincent, L. and P. Soille (1991). "Watersheds in digital spaces: an efficient algorithm based on immersion simulations." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13.6, pp. 583–598. ISSN: 0162-8828. DOI: 10.1109/34.87344.

Visvalingam, Mahes and J. Duncan Whyatt (1990). "The Douglas-Peucker Algorithm for Line Simplification: Re-evaluation through Visualization." In: *Comput. Graph. Forum* 9.3, pp. 213–228. URL: http://dblp.uni-trier.de/db/journals/cgf/cgf9.html#VisvalingamW90.

Wang, S. and C. Wang (2015). "Research on dimension reduction method for hyperspectral remote sensing image based on global mixture coordination factor analysis." In: *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XL-7/W4, pp. 159–167. DOI: 10.5194/isprsarchives-XL-7-W4-159-2015. URL: https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XL-7-W4/159/2015/.

Whitney, H. (1935). "On the abstract properties of linear dependence." In: *American Journal of Mathematics* 57, pp. 509–533. URL: https://people.math.osu.edu/chmutov.1/wor-gr-su06/ref/Wh.pdf.

Winter, Stephan (1998). "Bridging Vector and Raster Representation in GIS." In: *Proceedings of the 6th ACM International Symposium on Advances in Geographic Information Systems*. GIS '98. Washington, D.C., USA: ACM, pp. 57–62. ISBN: 1-58113-115-1. DOI: 10.1145/288692.288704. URL: http://doi.acm.org/10.1145/288692.288704.

Worboys, Michael and Matt Duckham (2004). *GIS: A Computing Perspective, 2Nd Edition*. Boca Raton, FL, USA: CRC Press, Inc., pp. 90–113. ISBN: 0415283752. URL: http://dl.acm.org/citation.cfm?id=994026.

Yu, Qian et al. (2006). "Object-based detailed vegetation classification with airborne high spatial resolution remote sensing imagery." In: *Photogrammetric Engineering & Remote Sensing* 72.7, pp. 799–811.

Zahn, C. T. (1971). "Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters." In: *IEEE Trans. Comput.* 20.1, pp. 68–86. ISSN: 0018-9340. DOI: 10.1109/T-C.1971.223083. URL: http://dx.doi.org/10.1109/T-C.1971.223083.

Appendix A

Implementation source code

This appendix includes the implementation code documentation and the respective source code, which can be found in the folders *documentation* and *iacdcgpo* respectively, on the accompanying CD.