

BINARY IMAGES RECONSTRUCTION FOR DISCRETE TOMOGRAPHY USING TRIANGULAR GRID

Teză destinată obținerii
titlului științific de doctor inginer
la
Universitatea Politehnica Timișoara
în domeniul CALCULATOARE ȘI TEHNOLOGIA
INFORMAȚIEI
de către

Inf. Mat. Elisa Valentina Oneț

Conducător științific:
Referenți științifici:

prof.univ.dr.ing. Vladimir-Ioan Crețu
prof.univ.dr.ing. Mircea Stelian Petrescu
prof.univ.dr.ing. Dumitru Dan Burdescu
prof.univ.dr.ing. Ștefan Holban

Ziua susținerii tezei: 21 noiembrie 2014

Seriile Teze de doctorat ale UPT sunt:

- | | |
|---|--|
| 1. Automatică | 9. Inginerie Mecanică |
| 2. Chimie | 10. Știința Calculatoarelor |
| 3. Energetică | 11. Știința și Ingineria Materialelor |
| 4. Ingineria Chimică | 12. Ingineria sistemelor |
| 5. Inginerie Civilă | 13. Inginerie energetică |
| 6. Inginerie Electrică | 14. Calculatoare și tehnologia informației |
| 7. Inginerie Electronică și Telecomunicații | 15. Ingineria materialelor |
| 8. Inginerie Industrială | 16. Inginerie și Management |

Universitatea Politehnica Timișoara a inițiat seriile de mai sus în scopul diseminării expertizei, cunoștințelor și rezultatelor cercetărilor întreprinse în cadrul Școlii doctorale a universității. Seriile conțin, potrivit H.B.Ex.S Nr. 14 / 14.07.2006, tezele de doctorat susținute în universitate începând cu 1 octombrie 2006.

Copyright © Editura Politehnica – Timișoara, 2014

Această publicație este supusă prevederilor legii dreptului de autor. Multiplicarea acestei publicații, în mod integral sau în parte, traducerea, tipărirea, reutilizarea ilustrațiilor, expunerea, radiodifuzarea, reproducerea pe microfilme sau în orice altă formă este permisă numai cu respectarea prevederilor Legii române a dreptului de autor în vigoare și permisiunea pentru utilizare obținută în scris din partea Universității Politehnica Timișoara. Toate încălcările acestor drepturi vor fi penalizate potrivit Legii române a drepturilor de autor.

România, 300159 Timișoara, Bd. Republicii 9,
Tel./fax 0256 403823
e-mail: editura@edipol.upt.ro

Cuvânt înainte

Teza de doctorat a fost elaborată pe parcursul activității mele în cadrul Departamentului de Calculatoare al Universității Politehnica Timișoara.

Aș dori să exprim sincere mulțumiri și întreaga mea recunoștință conducătorului științific al acestei teze, domnului prof. univ. dr. ing. Vladimir-Ioan Crețu, pentru oportunitatea pe care mi-a oferit-o, sprijinul constant, și pentru îndrumările și încurajările profesionale de care am avut parte pe parcursul întregii perioade de cercetare și elaborare a prezentei teze de doctorat.

Mulțumesc domnului prof. Benedek Nagy pentru răbdarea, ajutorul și sfaturile competente date pe parcursul elaborării tezei, și sub a cărui îndrumare, am participat la numeroase manifestări științifice din țară și străinătate, unde am prezentat lucrări științifice publicate ulterior în reviste de specialitate.

Trebuie să mulțumesc cu acest prilej conducerii Universitatii unde îmi desfășor activitatea, colegilor și prietenilor mei în mod deosebit, prof. dr. ing. Cornelia-Emilia Gordan pentru răbdarea, ajutorul și sfaturile competente date pe parcursul elaborării tezei.

Nu în ultimul rând, doresc să mulțumesc părinților, tatălui meu și mamei mele pentru educația sprijinul material și moral acordat, precum și soțului meu pentru înțelegerea și suportul acordat pe parcursul realizării acestei lucrări.

Acknowledgements

I would like to express my most sincere gratitude to Prof. Dr. Eng. Vladimir-Ioan Crețu for the offered opportunity, constant support, and for his guidance and encouragements which I had part on the entire period of research and realization of this thesis.

Also, I acknowledge help and I am grateful to Professor Benedek Nagy for his patience, guidance and competent advices given during the entire period of research and realization of the present thesis, and under hoes guidance I participated to numerous national and international scientific events, where we presented scientific papers ulterior published in high proceedings.

I would like to thank to the management of the University where I work, to my colleagues and friends, especially to Prof. Dr. Eng. Cornelia-Emilia Gordan for her patience, help and competent advices given during realization of this thesis.

Last, but not least, I would like to thank to my parents, my father and my mother, for their education, moral and material support, and my husband for his understanding, patience and companionship during this part of my life.

Timișoara, noiembrie 2014

Elisa Valentina Oneț

Soțului, tatălui și mamei mele.

"Adevărata plăcere constă în descoperire, nu în cunoaștere."
(Isaac Asimov)

Oneț, Elisa Valentina

Binary Images Reconstruction for Discrete Tomography using Triangular Grid

Teze de doctorat ale UPT, Seria 14, Nr. 21, Editura Politehnica, 2014, 141 pagini, 79 figuri, 15 tabele.

ISSN: 2069-8216

ISSN-L: 2069-8216

ISBN: 978-606-554-884-8

Cuvinte cheie: tomografia discetă, tomografia binară, reconstrucții de imagini, rețeaua de celule triunghiulare, algoritm memetic, problema fluxului maxim de cost minim

Rezumat: Scopul prezentei teze este de a rezolva problema tomografiei binare folosind reprezentarea unei imagini digitale utilizând rețeaua de celule triunghiulare. Sunt prezentate proprietățile și structura rețelei de celule triunghiulare precum și reprezentarea imaginilor digitale folosind rețeaua de celule triunghiulare. De asemenea este definit conceptul de imagine binară având formă hexagonală. Se extinde suportul matematic referitor la transformata Radon, având ca bază rețeaua de celule triunghiulare. Teza propune reducerea problemei reconstrucției imaginilor binare reprezentate pe rețeaua de celule triunghiulare la problema flux maxim de cost minim într-un graf, folosind trei unghiuri diferite pentru măsurarea proiecțiilor. Folosind această modelare sunt propuși doi algoritmi de reconstrucție a imaginilor binare reprezentate folosind rețeaua de celule triunghiulare, și anume un algoritm memetic și unul iterativ.

CONTENTS

Abstract	7
List of Figures	8
List of Tables	11
List of Pseudocodes	12
List of Acronyms	13
1. Introduction	14
1.1. Motivation	14
1.2. Thesis objectives	15
1.3. Thesis structure	16
2. Discrete Tomography on the Square Grid. An Overview	18
2.1. Introduction	18
2.2. Radon transform	20
2.3. Image reconstruction problem for parallel-beam projections	22
2.4. Reconstruction as a network flow problem	25
2.4.1. Two projection reconstruction	25
2.4.2. Iterative reconstruction using more than two projections	28
2.5. Reconstruction as an optimization problem	30
2.5.1. Overview of memetic algorithm approach	30
2.6. Conclusions	32
3. Triangular Grid	33
3.1. Introduction	33
3.2. Features of triangular grid	34
3.3. Triangular grid image representation	36
3.4. Topology of the triangular grid	37
3.5. Lanes and diamond-chains on the triangular grid	38
3.6. Capturing and displaying of triangular sampled images	39
3.7. Triangular grid image reconstruction problem using projections	40
3.8. Conclusions	43
4. Radon Transform on Triangular Grid Basis	44
4.1. Introduction	44
4.2. The Radon transform	44
4.3. Conclusions	48
5. Modeling the Reconstruction Problem on Triangular Grid from Three Projections as Minimum Cost Maximum Flow Problem	49
5.1. Introduction	49
5.2. Preliminaries	50
5.3. Three projection reconstruction	52
5.3.1. Parallel-parallel to coordinate axes projection	55
5.3.2. Orthogonal-orthogonal to coordinate axes projection	58
5.3.3. Parallel-orthogonal to coordinate axes projection	62
5.4. Conclusions	64

6. Memetic Algorithm for Image Reconstruction in Discrete Tomography on the Triangular Grid from Six Projections	65
6.1. Introduction	65
6.2. Preliminaries.....	66
6.3. Algorithm description.....	67
6.3.1. Initial population	69
6.3.2. Fitness function.....	69
6.3.3. Crossover operator	69
6.3.4. Guided mutation operator.....	71
6.3.5. Compactness operator	72
6.3.6. Minimal hill climb operator.....	73
6.3.6.1. Rhombus switching component.....	73
6.3.6.2. Hourglass switching component	74
6.3.6.3. Rotation switching component	76
6.4. Experimental results	78
6.4.1. HV-convex polyominoes	79
6.4.2. Generic regular hexagon-shaped images	83
6.4.3. Non-convex regular hexagon-shaped images	87
6.5. Performance analysis	97
6.6. Conclusions	101
7. Iterative Reconstruction Algorithm Based on Minimum Cost Maximum Flow Algorithm on the Triangular Grid	103
7.1. Introduction	103
7.2. Algorithm description.....	104
7.2.1. Projection angles selection.....	105
7.2.2. Pixel weight function on the triangular grid	105
7.2.3. Stopping criterion	106
7.3. Experimental results	107
7.3.1. HV-convex polyominoes	108
7.3.2. Generic regular hexagon-shaped images	112
7.3.3. Non-convex regular hexagon shaped images	115
7.4. Conclusions	125
8. Conclusions and Future Work.....	126
8.1. Conclusions	126
8.2. Summary of contributions	127
8.3. Research perspectives	128
9. Bibliography.....	129
Appendix A	135
Appendix B	137
List of Publications	139

ABSTRACT

Binary tomography is hard and challenging task in image processing. The usage of non-traditional grids such as the triangular grid may have several benefits. This thesis focuses on the problem of reconstruction of binary hexagon-shaped images, represented on the triangular grid, from projections measured along few different angles. The directions along which these projections are measured are the ones corresponding to the natural directions of a triangular grid. We have explored two different approaches for reconstruction of binary hexagon-shaped images, represented on the triangular grid. The proposed approaches use the formulation of the reconstruction problem on the triangular grid using three projections as the minimum cost maximum flow problem. We offer two new solutions, a memetic algorithm and an iterative one, both based on the representation of images on the triangular grid.

As the first solution we propose a memetic algorithm. In this algorithm the initial population is generated using the minimum cost maximum flow algorithm for the case of hexagon-shaped images. We introduced new mutation and crossover operators for hexagon-shaped images. Also a new compactness operator and a minimal hill climb operator, based on the switching components are defined. The compactness and the switching components are two important components of discrete tomography.

The second solution is an iterative one. In each iteration, of the proposed algorithm, a new triplet of projection angles is selected. The reconstruction problem that depends on the selected triplet of projection angles and on the reconstruction from the previous iteration is solved. This reconstruction problem is equivalent with the problem of finding a flow with minimum cost in a graph. In our approach, we used the minimum cost maximum flow algorithm in order to solve the reconstruction problem.

We implemented the two solutions and tested them using a variety of binary hexagon-shaped test images. The test images were chosen of three types: hv-convex polyominoes, generic regular hexagon-shaped images and non-convex regular hexagon-shaped images resampled with different dimension. All test images were resampled from images represented on the square grid. The evaluation results show that the proposed two algorithms are robust enough, producing good quality reconstruction results in a relatively short time. The results of the memetic algorithm were compared with the results obtained with simulated annealing algorithm. The memetic algorithm results are comparable to that obtained with the simulated annealing, in quality, for the test cases which satisfy some topological or geometrical properties. However, the main advantage of the proposed memetic algorithm is the computational saving time.

LIST OF FIGURES

2.1. Left: (a) Parallel beam tomography; Right: (b) Fan beam tomography.....	19
2.2. Components of a parallel beam tomography (X-ray CT) system [Abb08]	20
2.3. Radon transform of a function $f(x, y)$	21
2.4. (a) Shepp-Logan brain phantom (b) the sinogram $p(r, \theta)$ of (a) with 180 discrete radial and angular samples. Taken from [Neh12].	22
2.5. (a) <i>Left</i> : Parallel beam tomography using line integrals; (b) <i>Right</i> : Parallel beam tomography using strip integrals	23
2.6: Parallel beam geometry.	24
2.7: Basic structure of the associated graph [Bat08].	27
3.1. Euclidean plane tiling with regular polygons (a) square (b) triangular (c) hexagonal.....	34
3.2. Rotation symmetry of (left) square and (right) triangular grid nodes.	35
3.3. Neighborhood relations in the triangular grid of regions and the hexagonal grid of nodes.	35
3.4. Coordinate system of the triangular grid.	36
3.5. The topological coordinate system for a segment of the triangular grid, from [Nag12].....	38
3.6. Examples of (a) lines orthogonal to coordinate axes and (b) diamond-chains parallel to coordinate axes.	39
3.7. Hartman and Tanimoto resampling scheme	40
3.8. Coordinate representation of a hexagon-shaped image of size $m=3$	41
3.9. 2D binary hexagon-shaped image of six pixels, represented on the triangular grid.....	41
3.10. (a) First projection at angle $\pi/6$ degrees with the x-axis; (b) Second projection at angle $\pi/2$ degrees with the x-axis; (c) Third projection at angle $5\pi/6$ degrees with the x-axis.	42
3.11. Three projections of a hexagon-shaped image represented on the triangular grid.....	42
4.1. Parallel beam, L , going through an object at an angle θ	45
4.2. The differential element of the line L	46
4.3. A projection line - the line integral of $f(x, y, z)$ on line L	47
4.4. Radon transform integrals	47
5.1. Radon transform representation	50
5.2. Parallel beam geometry which contains the projection lines.	51
5.3. General representation of a graph G for the minimum cost maximum flow problem using three projection angles.	53
5.4. Example for diamond-chains, parallel to the coordinate axes.	56
5.5. A $2 \times 2 \times 2$ hexagon-shaped image with its projections corresponding to the projection angles $\theta_1 = 0$, $\theta_2 = \frac{\pi}{3}$, $\theta_3 = \frac{2\pi}{3}$	57
5.6. The corresponding graph associated to the hexagon-shaped image from Figure 5.5. The lines represent flow 1.	58
5.7. Examples for orthogonal projection to coordinate axes.....	59
5.8. A $2 \times 2 \times 2$ hexagon-shaped image with its projections corresponding to the projection angles: $\theta_1 = \frac{\pi}{6}$, $\theta_2 = \frac{\pi}{2}$, $\theta_3 = \frac{5\pi}{6}$	61

5.9. The corresponding graph associated to the hexagon-shaped image from Figure 5.8. The black lines represent flow 1 and dashed lines represent flow 2.	62
5.10: Two direction combinations (a) parallel-orthogonal same axes direction; (b) parallel-orthogonal different axes direction;	63
6.1. A hexagon-shaped binary image (a) and its representation (b), with its projections measured along the orthogonal to the axes directions.....	67
6.2. Two parents (first row) and their $\gamma_{ }$ and γ_{\perp} offsprings (second, third row).....	70
6.3. The mutated image (left) and the parent image (right).	71
6.4. Two isolated pixels (a white and a black) and their neighborhoods. Image after applying compactness operator (right).	72
6.5. Rhombus switching operation for direction orthogonal to coordinate axis y	74
6.6. Subhexagons of the hourglass switching for the three combinations from equation (6.7).	75
6.7. Example for the hourglass switching operation.	76
6.8. Possible positions of the initial triplets in the rotation switching component for a hexagon of size 6 ($r = 3$).	77
6.9. Average results for hv-convex polyominoes of dimension $26 \times 26 \times 26$	80
6.10. Five $26 \times 26 \times 26$ hv-convex polyomino (left) and there reconstruction (right).	81
6.11. Fitness results for hv-convex polyominoes of dimension $26 \times 26 \times 26$	82
6.12. Error results for hv-convex polyominoes of dimension $26 \times 26 \times 26$	82
6.13. Computation time results for hv-convex polyominoes of dimension $26 \times 26 \times 26$	83
6.14. Average results for 30 generic hexagon-shaped images of dimension $26 \times 26 \times 26$	84
6.15. Five $26 \times 26 \times 26$ generic hexagon-shaped images (left) and their reconstruction (right).	85
6.16. Fitness results for generic hexagon-shaped images of dimension $26 \times 26 \times 26$	86
6.17. Reconstruction error results for generic hexagon-shaped images of dimension $26 \times 26 \times 26$	86
6.18. Computation time results for generic hexagon-shaped images of dimension $26 \times 26 \times 26$	87
6.19. Five examples of $26 \times 26 \times 26$ non-convex regular hexagon-shaped images (left) and their reconstruction (right).....	88
6.20. Average results for 10 non-convex regular hexagon-shaped images of dimension $4 \times 4 \times 4$	89
6.21. Average results for 10 non-convex regular hexagon-shaped images of dimension $8 \times 8 \times 8$	90
6.22. Average results for 10 non-convex regular hexagon-shaped images of dimension $13 \times 13 \times 13$	91
6.23. Average results for 10 non-convex regular hexagon-shaped images of dimension $26 \times 26 \times 26$	92
6.24. Results for non-convex regular hexagon-shaped mages of dimension $4 \times 4 \times 4$	93
6.25. Results for non-convex regular hexagon-shaped mages of dimension $8 \times 8 \times 8$	94
6.26. Results for non-convex regular hexagon-shaped mages of dimension $13 \times 13 \times 13$	95
6.27. Results for non-convex regular hexagon-shaped mages of dimension $26 \times 26 \times 26$	96
6.28. Images used in tests.	98
6.29. Reconstruction results for the three test images using simulated annealing.	99
6.30. Reconstruction results for the three test images using memetic algorithm.	100

6.31. Reconstruction results for the three test images.	101
7.1. Hv-convex polyominoes with dimension of 26x26x26.....	109
7.2. Results for the 10 hv-convex polyominoes.....	110
7.3. Generic regular hexagon-shaped images with dimension of 26x26x26.	111
7.4. Reconstruction results for the 10 generic regular hexagon-shaped images...	113
7.5. Generic hexagon-shaped images reconstruction examples.....	114
7.6. Non-convex hexagon-shaped test images of dimension 50x50x50.....	115
7.7. Reconstruction results for non-convex hexagon-shaped images with dimension 4x4x4.	116
7.8. Reconstruction results for non-convex hexagon-shaped images with dimension 8x8x8.	117
7.9. Reconstruction results for non-convex hexagon-shaped images with dimension 13x13x13.	119
7.10. Reconstruction examples for non-convex hexagon-shaped images with dimension 13x13x13.....	119
7.11. Reconstruction results for non-convex hexagon-shaped images with dimension 26x26x26.....	121
7.12. Reconstruction examples for non-convex hexagon-shaped images with dimension 26x26x26.....	122
7.13. Reconstruction results for non-convex hexagon-shaped images with dimension 50x50x50.....	123
7.14. Reconstruction examples for non-convex hexagon-shaped images with dimension 50x50x50.....	124
A. 1. Triangular grid coordinate and Cartesian grid coordinate systems.....	135
B. 1. Triangular and rotated triangular coordinate	137

LIST OF TABLES

6.1. HV-convex polyominoes with dimension $26 \times 26 \times 26$	80
6.2. Generic hexagon-shaped images with dimension $26 \times 26 \times 26$	83
6.3. Non-convex regular hexagon-shaped images with dimension $4 \times 4 \times 4$	89
6.4. Non-convex regular hexagon-shaped images with dimension $8 \times 8 \times 8$	90
6.5. Non-convex regular hexagon-shaped images with dimension $13 \times 13 \times 13$	91
Table 6.6. Non-convex regular hexagon-shaped images with dimension $26 \times 26 \times 26$	92
6.7. Reconstruction results for the test images	100
7.1. Reconstruction results for the hv-convex polyominoes case.	110
7.2. Reconstruction results for PH1-PH5 generic regular hexagon-shaped images.	112
7.3. Reconstruction results for PH6-PH10 generic regular hexagon-shaped images.	113
7.4. Reconstruction results for non-convex hexagon-shaped images with dimension $4 \times 4 \times 4$	116
7.5. Reconstruction results for non-convex hexagon-shaped images with dimension $8 \times 8 \times 8$	117
7.6. Reconstruction results for non-convex hexagon-shaped images with dimension $13 \times 13 \times 13$	118
7.7. Reconstruction results for non-convex hexagon-shaped images with dimension $26 \times 26 \times 26$	120
7.8. Reconstruction results for non-convex hexagon-shaped images with dimension $50 \times 50 \times 50$	122

LIST OF PSEUDOCODES

2.1. Basic steps of the algorithm [Bat07a]	29
2.2. Steps of a generic memetic algorithm [Gar09]	31
5.1. Algorithm for computing graph G	55
6.1. Basic steps of the memetic algorithm.....	68
6.2. Main steps for the crossover operator	71
6.3. Main steps for the guided mutation operator.....	72
6.4. Main steps for the compactness operator.....	73
6.5. Main steps of a generalized minimal hill climb operator	78
6.6. Simulated Annealing algorithm [Luk12].....	98
7.1. Basic steps of the iterative reconstruction algorithm	105

LIST OF ACRONYMS

Binary Tomography (BT).....	8
Computer Tomography (CT).....	9
Digital Geometry (DG).....	27
Discrete Tomography (DT).....	8
Genetic Algorithm (GA).....	24
Memetic Algorithms (MA).....	24
Positron Emission Tomography (PET)	9
Radon transform (RT).....	14
Simulated Annealing (SA)	91

1. INTRODUCTION

1.1. Motivation	14
1.2. Thesis objectives.....	15
1.3. Thesis structure	16

In this thesis, I am concerned with the development of discrete tomography reconstruction methods for images represented on the triangular grid using few projections. Taking into consideration the different ways the plane can be covered with various figures, we can make use of the triangular grid, which became an important tool utilized in digital image processing, digital geometry and also in other related fields. The advantage of using triangular grid is that it has better symmetric properties than the square grid. Tomography is used in several fields in practice, when data about the inner structure of the object is needed without breaking it. It is applied in different areas of daily life, from scientific phenomena and industrial applications to medical applications.

With this purpose, the focus of this work is to design efficient methods for solving the discrete tomography problem of reconstruction of objects from a few numbers of projections on triangular grid. In this context, the “minimum cost-maximum flow” algorithm is employed in different phases of the proposed methods. Main theory and related algorithms are developed on the framework of binary tomography on the triangular grid.

1.1. MOTIVATION

In digital image processing, the triangular grid can be considered of real interest. Triangular grids have been of interests to humans from some time. Triangles have been studied from a long time and have been found to have special properties. These properties include the fact that triangles are part of the three regular polygons with which a plane can be tiled with. The other two are the square and the hexagon. The triangular grid plays an increasing role in geometric modeling. The human retina is often modeled by a Delauney triangulation and many 3D-scanners also produce triangulations. Also computer graphic algorithms use the triangular grid [Deu72][Shi81].

Digital signal processing plays an increasing role in the field of image processing. An important imaging method is the so called tomography. Tomography deals with reconstruction of images from projections. Mathematically, the image corresponds to a function and the problem is to reconstruct this function from its integrals or sums over subsets of its domain. In general, the topographic problem can be continuous or discrete. In discrete tomography (DT) the domain of the function is discrete, and the range of the function is a finite set of real numbers, usually nonnegative numbers. In practice, discrete tomography deals with the reconstruction of objects from a few numbers of projections [Haj01]. The simplest case of discrete tomography is the binary tomography (BT) that deals with the problem of the reconstruction of a binary image (i.e., an image containing only

black and white pixels represented by 0's and 1's) from few projections. These problems are considered hard to solve due to the fact that their solution is usually undetermined thus means that there is more than one solution for every case. The same problem is NP-hard when three or more than three projections are considered. The field of tomography is still an active area of research, being implemented in numerous domains, where data about the inner structure of the object is needed without breaking it. External radiation sources and sensors are used to collect data that will be utilized to reconstruct or predict the shape of the object using various algorithms. These techniques are applied in metal industry, and in medical applications. Some examples are: Positron Emission Tomography (PET), Computer Tomography (CT) [Her99][Kub99][Her07].

The basic problem of tomography is the following: having a set of 1-D projections and the corresponding angles at which these projections were taken, how do we reconstruct the original 2-D image considering the projections that were taken. Due to the hardware limitation the image generated via reconstruction from a sinogram is always defined on a square grid. Because the sinogram contains information about a continuous or discrete structure, this can be back projected onto any grid. The first studied grids in discrete tomography, traditionally, were the square grid and the cubic grid, because of the widespread usage of the Cartesian coordinate system. In 1957, Ryser [Rys57] proposed an algorithm for solving the problem of reconstruction of a binary matrixes from its row and column sums. After that, Gale [Gal57] placed the two-projection reconstruction problem in the context of flows in networks.

Like stated before, there are two other regular 2D grids used in digital geometry [Ros86][Ros89] and image processing: the hexagonal and the triangular grid. There is a wide literature on binary tomography using the square [Kon87][Mel91] and the hexagonal grids [Her07][Mid05][Mat98]. The advantage of using a non-traditional grid is that it has better symmetric properties than the square grid. Due to its symmetry, the triangular grid can elegantly be described by the help of three coordinate axes [Nag03a][Nag04b]. Using three coordinate values one can define lanes and diamond-chains (somewhat similar concepts to rows and columns on the square grid). A lane contains a set of pixels that have a fixed coordinate value; it is orthogonal to one of the coordinate axes. A diamond-chain contains a set of pixels according to a line that is parallel to one of the coordinate axes. It is known that two directions are not enough to identify a pixel, and thus (at least) three projection directions are necessary to be used in reconstructions on the triangular grid. Apart from the three lane-directions, three diamond-chain directions are also natural. In [Her95][Nag04][Nag03b][Wut91] a connection of the cubic, hexagonal and triangular grids can be found.

1.2. THESIS OBJECTIVES

The aim of this thesis is to study the effect of changing the sampling grid from square to triangular one and to find efficient methods for solving the binary tomography reconstruction problem. Based on its geometry, the triangular grid has some advantage over the square grid which can help in processing images defined on this grid. The main advantage of the triangular grid is that it has better

symmetric properties, a rotation by $\frac{2\pi}{3}$ moves the grid to itself.

The problem we propose to solve is the reconstruction of binary hexagon-shaped images represented on the triangular grid, where the object is represented in white and the background in black, using projections of the image from few different angles. In order to solve this problem, the following objectives of the thesis have been established.

The first objective of the thesis is to develop fundamental bases, such as data structures, definitions and properties for the triangular grid and digital image representation using the triangular grid.

The binary tomography reconstruction presumes two main phases: firstly, the generation of the set of projections and secondly the combination of images to reconstruct an approximation of the original image.

The first phases of the binary tomography reconstruction problem, the generation of the set of projections of a hexagon-shaped image measured along the projection angles, mathematically, is accomplished using the Radon transform. In this context, the second objective of the thesis is to extend the mathematical support of the Radon transform to the case of using a triangular grid bases.

The second phase of the binary tomography reconstruction problem, mathematically, consists of image reconstruction algorithms.

Our first step towards developing algorithms to reconstruct hexagon-shaped images on the triangular grid is to reduce the binary tomography reconstruction problem of hexagon-shaped images on triangular grid, from two projections using a third projection as a restriction, to the problem of minimum cost maximum flow in a graph. This represents the third objective of the present thesis.

The fourth objective of the thesis is to develop a hexagon-shaped binary image reconstruction algorithm based on an optimization method, in which the initial population is generated using the minimum cost maximum flow algorithm for the case of hexagon-shaped images.

The fifth objective of the present thesis is to develop an iterative method for solving the binary tomography reconstruction problem for hexagon-shaped images on the triangular grid.

1.3. THESIS STRUCTURE

This thesis is divided into eight chapters and two appendices.

Chapter 2 provides an overview of the relevant background in the field of discrete tomography. It focuses on square grid image reconstruction methods using network flow and evolutionary algorithm and summarizes the work that has been done in the literature until now.

Chapter 3 is concerned with presenting the aspects related to the triangular grid and the triangular grid image representation, addressing and processing. Fundamental aspects of discrete tomography reconstruction problem using projections are described. The main point of this chapter is the definition of a hexagon-shaped binary image on the triangular grid, its representation and addressing, as this is used in the remaining chapters of the thesis. This chapter fulfills the first objective of the thesis.

Chapter 4 is concerned with developing the mathematical framework for projection representation and acquisition on the triangular grid. Mathematically, projections are generated applying the Radon transform. This chapter extends the formulation of the Radon transform to the case of triangular grid bases, based on

the triangular grid representation of a digital image, achieving the second objective of the thesis.

Chapter 5 provides a method in which binary tomography reconstruction problem of hexagon-shaped images on triangular grid is reduced to the problem of minimum cost maximum flow in a graph. This chapter fulfills the third objective of the present thesis.

Chapter 6 provides a solution for the problem of binary tomography by proposing a memetic algorithm, in which the initial population is generated using the minimum cost maximum flow algorithm on the triangular grid. A detailed description of its steps is included. Also experimental results are presented. In this way the fourth objective of the thesis is achieved.

Chapter 7 provides an iterative algorithm for solving the discrete tomography reconstruction problem from few projections in case in which images are sampled using the triangular grid, fulfilling the fifth objective of the thesis. The proposed solution uses in each iteration the minimum cost maximum flow algorithm. Experimental results and conclusions are also included.

The final chapter provides conclusions and a discussion of the future works related to discrete tomography problem on triangular grid.

Appendix A provides mathematical formulas used for converting triangular grid coordinates into corresponding square grid coordinates. Appendix B provides the derivation for the rotation transformation on the triangular grid.

2. DISCRETE TOMOGRAPHY ON THE SQUARE GRID. AN OVERVIEW

2.1. Introduction	18
2.2. Radon transform	20
2.3. Image reconstruction problem for parallel-beam projections	22
2.4. Reconstruction as a network flow problem	25
2.4.1. Two projection reconstruction	25
2.4.2. Iterative reconstruction using more than two projections.....	28
2.5. Reconstruction as an optimization problem	30
2.5.1. Overview of memetic algorithm approach	30
2.6. Conclusions	32

This chapter presents an overview of the discrete tomography problem using the square grid representation of an image. This will be used as a starting point in different parts of this thesis, and in particular for the description of the binary tomography approach and for the reconstruction of a 2D discrete image from its projections on the triangular grid.

2.1. INTRODUCTION

Tomography is an imaging process where the shape and dimensional information of an object are determined from its projections. The projections are created by beams or rays that are emitted from a source, transmitted across the object and detected by an array of detectors. The pixels corresponding to the projections represent the total absorption of the beams along the lines between the source and the detector. Mathematically, the object corresponds to a function and the problem is to reconstruct this function from its integrals or sums over subsets of its domain. Using Beers' law, and considering the X-ray case, the projections can be measured as:

$$I = I_0 \cdot e^{-\int_0^d \mu(x) dx} \quad (2.1)$$

where I_0 is the intensity of the X-ray source, I is the detected X-ray intensities, μ is the absorption coefficient of the object and d is the distance between the source and the detector.

Discrete tomography (DT) is a research area of tomography that is applied for the reconstruction of objects that are discrete (such as crystals) or homogenous (such as metal). Mathematically this means that the domain of the function that corresponds to an object is discrete, and the range of the function is a finite set of real numbers, usually nonnegative numbers. In practice, discrete tomography deals usually with the reconstruction of objects from a small number of projections

[Haj01]. The objects that have to be reconstructed consist of a few discrete or homogenous materials. Incorporating this information about the object to be reconstructed, simple objects can be reconstructed from a smaller number of projections than more complex objects. This is the reason why discrete tomography is important in applications where the object is simple and lots of projections cannot be acquired or it is too costly to do it. Such area of application of discrete tomography is in medicine, electron microscopy and non-destructive testing.

The simplest case of discrete tomography is the binary tomography (BT) that deals with the problem of the reconstruction of a binary image (i.e., an image containing only black and white pixels represented by 0's and 1's) from a small number of projections.

The necessary projections can be acquired basically in two ways, depending on the way in which the beams or rays are emitted, parallel or non-parallel (fan-beam/cone-beam) to each other. Figure 2.1 shows the parallel (a) and fan beam acquisition technique (b). In this thesis only the parallel beam case is considered and discussed.

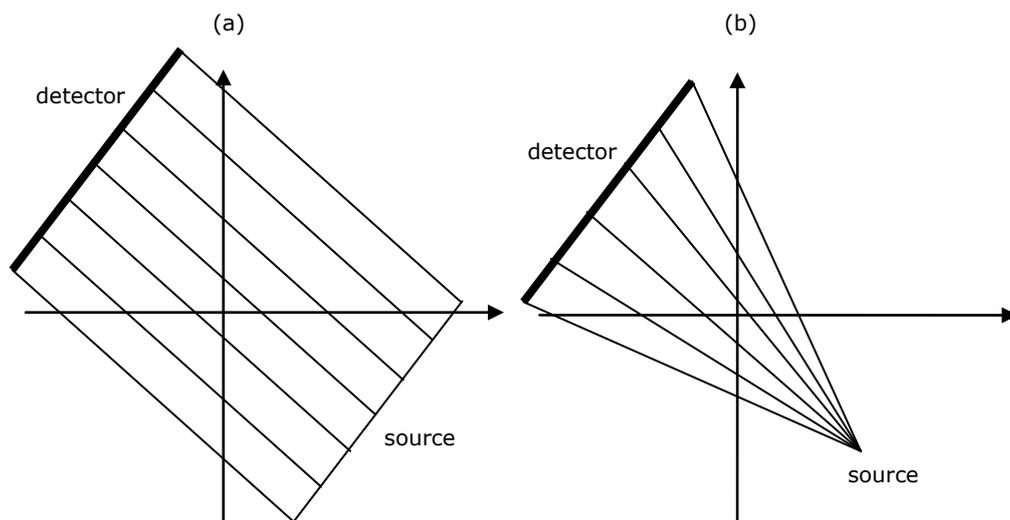


Figure 2.1. Left: (a) Parallel beam tomography; Right: (b) Fan beam tomography.

The technique of image processing using the parallel beam tomography works as follows: a source is used to emit/transmit parallel beams, called X-rays in computer tomography (CT), that pass through an object which attenuates the beam. The intensity of the attenuated beam is detected and recorded as an image on a parallel array detector surface situated behind the object.

As the beams pass through the object, their intensity attenuates. The value for the intensity of a beam depends on the materials from which the object is made of, and on the length of the intersection between the beam and the object. The attenuation of a beam occurs due to the fact that different types of materials absorb different amount of radiation.

In parallel beams tomography, thin parallel beams are transmitted across a section of an object at a specific angle. When the beams hit the detector on the other side of the object, the flux is recorded. This flux represents the projection of the cross section at the specified angle. Incrementing the angle another projection of the same object section is obtained. In this way a number of projections from different angles are generated. These projections are used to reconstruct a 2D image of a section of the object. Figure 2.2 illustrates the way in which the projection data is acquired using a parallel beam source.

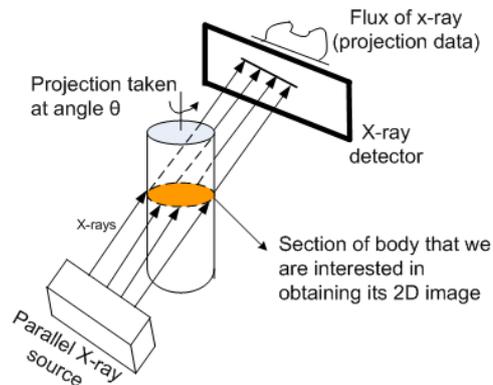


Figure 2.2. Components of a parallel beam tomography (X-ray CT) system [Abb08]

There are two main phases in tomography. The first phase is the generation of the set of projections and the second one is the combination of images to reconstruct an approximation of the original 2D image. Mathematically speaking, the first phase of tomography is accomplished by using the Radon transform and the second one using image reconstruction algorithms.

2.2. RADON TRANSFORM

All tomography types use Radon transform (RT) as the basic principal for generating the projections. The transformation was named after the Australian mathematician Johann Radon, who introduced it in 1917 [Rad17]. The Radon transform is the integral of a function over straight lines at different angles. In his work, Johann Radon showed how a function can be described in terms of its projections. The Radon transform is the mapping of the function on the projections. The inverse Radon transform is used to reconstruct images from their projections [Nat01][Nat011].

The Radon transform (RT) of a function $f(x, y)$ is (see Figure 2.3):

$$R(f(x, y)) = p(r, \theta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(x \cos \theta + y \sin \theta - r) dx dy \quad (2.2)$$

Where $p(r, \theta)$ is the integral of function $f(x, y)$ at angle θ for all x and y which satisfy the equation $x \cdot \cos \theta + y \cdot \sin \theta - r = 0$, and r is the perpendicular distance of a line from the origin. δ is the Dirac delta function [Dir27], which is a generalized function that can be defined as the limit of a class of delta sequences.

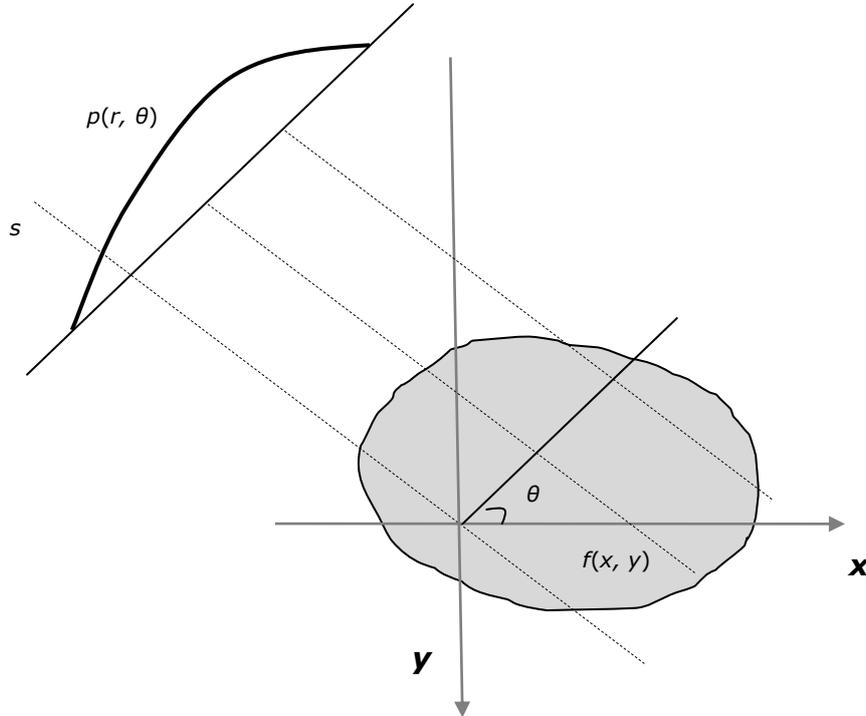


Figure 2.3. Radon transform of a function $f(x, y)$

Interpreting the Radon transform as projection of function $f(x, y)$ at angle θ onto the axis s at a discrete value r , the coordinate (r, s) are related to (x, y) by a rotation matrix [Neh12]:

$$\begin{bmatrix} r \\ s \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (2.3)$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} r \\ s \end{bmatrix} \quad (2.4)$$

Using the relationship from equations (2.3) and (2.4), equation (2.2) can be written as:

$$p(r, \theta) = \int_{-\infty}^{\infty} f(r \cos \theta - s \sin \theta, r \sin \theta + s \cos \theta) ds \quad (2.5)$$

Sampling this projection function, the discrete sinogram $p(r_i, \theta_k)$ is obtained (see Figure 2.4). The term sinogram comes from the fact that the Radon transform of an impulse function is a sine wave. In practice, the projection $p(r, \theta)$ is a discrete function obtained from the mapping of the function $f(x, y)$ to a distance r and angle θ .

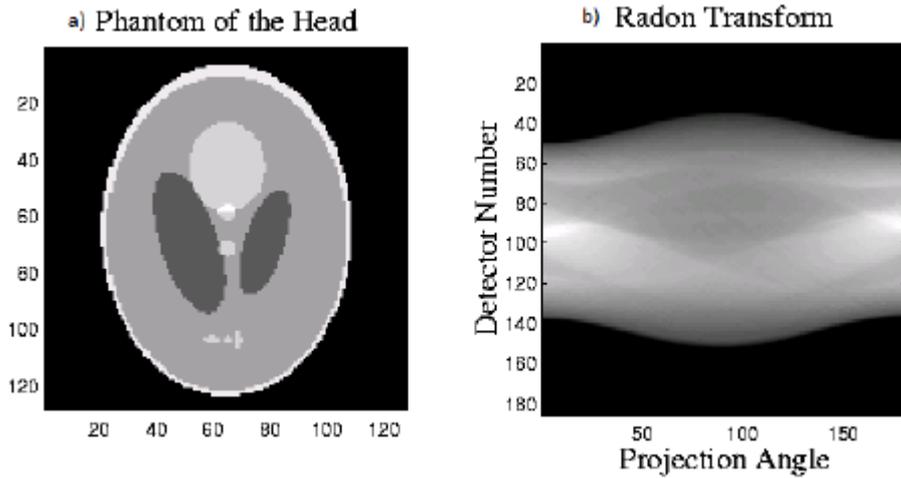


Figure 2.4. (a) Shepp-Logan brain phantom (b) the sinogram $p(r, \theta)$ of (a) with 180 discrete radial and angular samples. Taken from [Neh12].

2.3. IMAGE RECONSTRUCTION PROBLEM FOR PARALLEL-BEAM PROJECTIONS

In tomography, the image reconstruction problem is an inverse problem, where for a given projection $p(r, \theta)$, one has to recover $f(x, y)$. In the next paragraph the image reconstruction problem for parallel beam projections is presented. The theoretical aspects considered for the image reconstruction problem using the case scenario of parallel beam projections, are similar with those presented in [Bat06a] and [Cie11].

Considering r as being the source points and v a unit vector in direction $\theta \in [0, 2\pi)$, the integral of f along a line starting from r in direction v can be written as:

$$Rf(r, \theta) = \int_0^{\infty} f(r + t \cdot v) dt \quad (2.6)$$

Given a set of source points r , the reconstruction problem using parallel beam projections can be stated in the following way:

PB(r) RECONSTRUCTION PROBLEM

INPUT: A function $p: r \times [0, 2\pi) \rightarrow \mathbb{R}$

TASK: Construct a function f such that $Rf(r, \theta) = p(r, \theta)$ for all r corresponding to the angle $\theta \in [0, 2\pi)$.

In this chapter, the reconstruction of a special type of f function from parallel beam projections is considered. Further it is assumed that f is a $m \times n$ lattice with constant values at each 1×1 square of the lattice, that could be either 0 or 1. That means that f can be represented as a binary matrix or a binary image on the

square grid. The binary matrix can also be considered as a binary vector $x \in \{0,1\}^{mn}$. In practice, the projections are acquired from a finite number of angles θ_k , $k=1,\dots,d$. The i^{th} projection b_i for the angle θ_k can be described as a linear equation:

$$\sum_{j=0}^{mn} a_{ij}x_j = b_i, \quad i = 1, 2, \dots, I \tag{2.7}$$

where a_{ij} is the length of the intersection of the i^{th} line with the j^{th} unit square and I is the number of lines for the projection angles. Using equation (2.7) the projections are obtained by measurements. Using the information about the positions of the squares on the square grid and of the lines starting from the source points, one can easily compute the elements of the matrix (a_{ij}) , $i=1,\dots,I$, $j=1,\dots,mn$.

In the case of the parallel beam tomography, the sources r_k are located on the circle with center in the origin O , $C_R = \{(x, y) \mid x^2 + y^2 = R^2\}$. The radius R is big enough so that the image to be covered with the beams starting from the source. Usually in discrete tomography, a small number of projections are available, the reconstruction being influenced by the position of the sources and the number of beams.

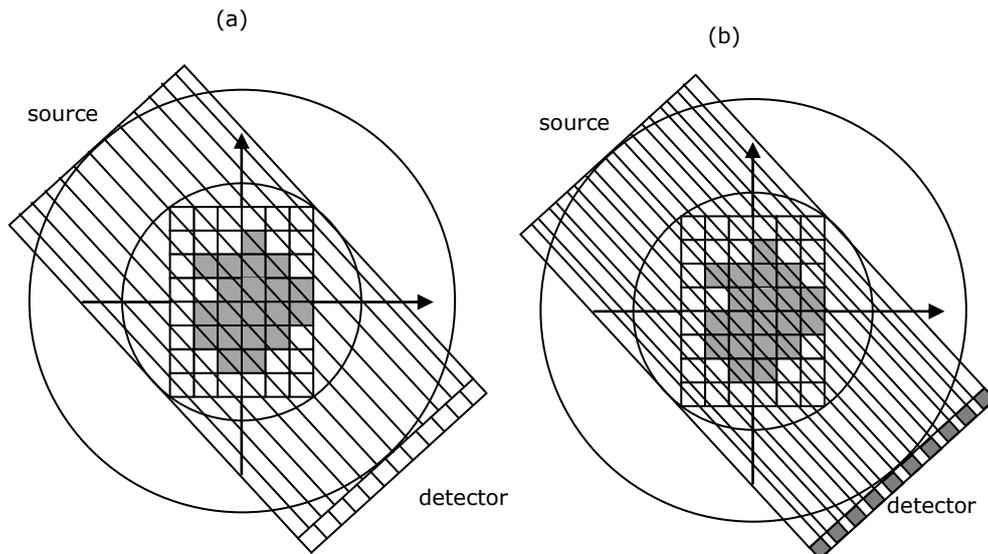


Figure 2.5. (a) *Left*: Parallel beam tomography using line integrals; (b) *Right*: Parallel beam tomography using strip integrals

Depending on the way in which the projections are measured, using the line or strip integrals, the values of a_{ij} are computed in different ways. In the case of line integrals (Figure 2.5a) the value of a_{ij} is the length of the intersection of line I with unit square j . In the case of strip integrals Figure 2.5.b) the values of a_{ij} is the common area between the beam i and unit square j . In the following sections it is considered that $a_{ij}=1$.

Now let $r = \{r_0, \dots, r_N\}$ be a set of real numbers in the interval $\left(-\frac{\pi R}{2}, \frac{\pi R}{2}\right)$ satisfying $r_0 < r_1 < \dots < r_N$ called the parallel beam parameters. The equation of a line L which depends on r , the distance L from the origin, and θ is:

$$L(r, \theta) = \{(x, y) \in \mathbb{R}^2 : x \cos \theta + y \sin \theta = r\} \quad (2.8)$$

and the equation of the strip $S_\theta(r, r')$ is:

$$S(r, r', \theta) = \{(x, y) \in \mathbb{R}^2 : r \leq x \cos \theta + y \sin \theta \leq r'\} \quad (2.9)$$

Any point (x, y) on the line L , that depends on a specified r and θ , satisfies the equation $x \cdot \cos \theta + y \cdot \sin \theta = r$.

Figure 2.6 shows the parallel beam geometry. Line $x \cdot \cos \theta + y \cdot \sin \theta = r$ has the angle θ with the x -axis and has distance r to the origin.

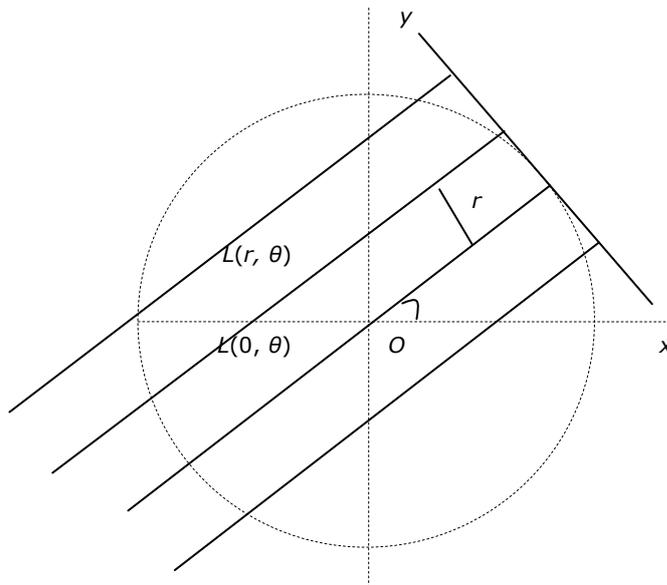


Figure 2.6. Parallel beam geometry.

In real situation the projections are usually acquired with an error. The noise projection data can be added to the computed projections generating a Gaussian noise.

The solution of the PB(r) reconstruction problem in the parallel beam model is equivalent to finding a solution of the linear equation system

$$Bx = b \quad (2.10)$$

where x is a binary vector, B is the projection matrix and b is the vector of prescribed line sums.

Because of the measurement errors, equation (2.10) has no exact solution. This is why it is better to find a binary vector x which satisfies at least

approximately equation (2.10). Actually, a possible way of solving (2.10), at least approximately, is to reformulate it as an optimization problem and as an iterative reconstruction problem that uses the two-projection network flow problem. In the next sections the reconstruction will be tackled as a network flow problem and as an optimization problem, more exactly as an evolutionary algorithm.

2.4. RECONSTRUCTION AS A NETWORK FLOW PROBLEM

There exists a correspondence between the reconstruction problem of binary images represented on square grid using two of its projections and the problem of finding a maximum flow in a certain graph. In this chapter we describe how the network flow algorithms can be used to solve the image reconstruction problem on square grids. Firstly we describe the network flow algorithm for two projections of the image. Then we describe an algorithm for reconstructing binary images from more than two projections.

The first to describe the two-projection reconstruction problem in the context of flows in network was Gale, in 1957 [Gal57]. In 1956 Ford and Fulkerson [For56], in their seminar paper presented an algorithm for computing a maximum flow in a network, which can be used to solve the two-projection reconstruction problem. Starting from their description, Batenburg [Bat06b][Bat08] proposed an iterative algorithm for reconstructing binary images from a small number of discrete and continuous X-rays. The algorithm considers that the reconstruction problem for two projections on the square grid can be efficiently solved, as a case of the network flow problem in graph, in polynomial time. The proposed reconstruction procedure is an iterative one, and in each iteration a new reconstruction is obtained using two projections together with the result obtained in the previous reconstruction. Section 2.4.1 presents the two projection reconstruction problem in the context of network flows. Using the network flow algorithm to resolve the two projection problem, in section 2.4.2, an iterative algorithm that uses a small number of projections is described.

2.4.1. TWO PROJECTION RECONSTRUCTION

Firstly the two projection reconstruction algorithm based on weighted network flow approach was presented in [Slu82][Bat06b][Bat08]. The approach of reconstruction of binary images from two projections as a network flow method was studied by several authors [Gal57][For56][Slu82][Ans83]. Next, the two-projection reconstruction problem is presented.

A binary two projection image is considered to be a mapping $\{1, \dots, m\} \times \{1, \dots, n\} \rightarrow \{0, 1\}$ which assigns the value 0 or 1 to each grid cell of a two-projection grid. Let X be a binary two-projection image, and $|X|$ the cardinality of the image A . Let $\theta_1, \theta_2 \in [0, 2\pi)$ be two projection angles, and $p_1 : L_1 \rightarrow \mathbb{N}$ and $p_2 : L_2 \rightarrow \mathbb{N}$ two given functions (the projections associated to the projection angles θ_1, θ_2), where $L_1 = (L_{1,1}, \dots, L_{1,|L_1|})$ and $L_2 = (L_{2,1}, \dots, L_{2,|L_2|})$ are the set of line projections associated to angle θ_1, θ_2 . The problem that has to be solved is the construction of an image X such that $P_1 = p_1$ and $P_2 = p_2$, where $P_k, k=1,2$ are the projections of image X along the projection angles θ_1 and θ_2 .

The representation of a binary image as a two-projection image is an approximated one because there are cases when there is no two-projection image corresponding to the scenario – the value of the projections equals the measured

projections. That is why it is necessary to be found a solution that corresponds approximately to the acquired projection values. So, let $T_k = \sum p_k$, $k=1,2$, be the projection sum for projection angle θ_k . One has to construct a binary image X such that $T_k = |X|$ and $|P_1 - p_1|_1 + |P_2 - p_2|_1$ is minimal, where $|\cdot|_1$ means the sum-norm.

The network flow problem in a particular graph can be efficiently used to solve the previously presented reconstruction problem, by incorporating prior knowledge about the unknown image in the reconstruction using a weight function [Slu82][Bat08]. In this way considering two line projection values p_1 and p_2 that correspond to the two projection angles θ_1 and θ_2 respectively, considering T_k , $k=1,2$ as being the projection sum and $W: X \rightarrow \mathbb{R}$ as a weight map, one has to construct X such that $P_1 = p_1$ and $P_2 = p_2$ and the total weight $\sum_{x \in X} W(x)$ is maximal [Bat07a]. The weighted network flow approach was used first in [Slu82] for the reconstruction of a medical image of the left ventricle using a prior information. This weights map is often utilized for reconstruction of an image making use of more than two projections. Such an algorithm that uses more than two projections will be presented in the next section.

For describing the image reconstruction problem in the context of flows in a graph, with the triplet (X, θ_1, θ_2) a direct graph $G = (V, E)$, where V is the set of nodes or vertices and E is the set of edges or lines, is associated. Figure 2.7 shows the basic structure of such a graph. The set V contains a nodes, the source, a node t , the sink, one node for each strip projection angles θ_1, θ_2 . The so called line nodes $n_{k,i}$ corresponds to $L_{k,i}$. The set of edges E of G contains the pixels edges, the connections between every pair $(n_{1,i}, n_{2,j})$ of nodes, and the lines that connect the source s to the line nodes $n_{1,i}$ and the line nodes $n_{2,j}$ to the sink t .

The book [Ahu93] offers an accurate introduction to the network flow problem.

The capacity function $C : E \rightarrow \mathbb{R}_{\geq 0}$ assigns to each edge $e \in E$ a positive real value as follows:

$$\begin{aligned} C((n_{1,i}, n_{2,j})) &= 1 \quad \text{for } 1 \leq i \leq |L_1|, 1 \leq j \leq |L_2| \\ C((s, n_{1,i})) &= p_{1i} \quad \text{for } 1 \leq i, j \leq |L_1|, \\ C((n_{2,j}, t)) &= p_{2j} \quad \text{for } 1 \leq i, j \leq |L_2|. \end{aligned} \tag{2.11}$$

The flow in G is a mapping $F : E \rightarrow \mathbb{R}_{\geq 0}$ such that $F(e) \leq C(e)$ for all $e \in E$ and such that for all $v \in V \setminus \{s, t\}$: $\sum_{w: w, v \in E} F((w, v)) = \sum_{w: (v, w) \in E} F((v, w))$. The size of

a flow F is equal to

$$|F| = \sum_{1 \leq i \leq |L_1|} F((s, n_{1,i})) = \sum_{1 \leq j \leq |L_2|} F((n_{2,j}, t)) = \sum_{\substack{1 \leq i \leq |L_1| \\ 1 \leq j \leq |L_2|}} F((n_{1,i}, n_{2,j})).$$

A flow is an integral flow if $F(e) \in \mathbb{N}_0$ for all $e \in E$. For any integral flow F in the associated graph, for all $e \in E \setminus \{(s, n_{1,i}), (n_{2,j}, t)\}$, $F(e) \in \{0, 1\}$, because the capacity of all edges $(n_{1,i}, n_{2,j})$ is 1.

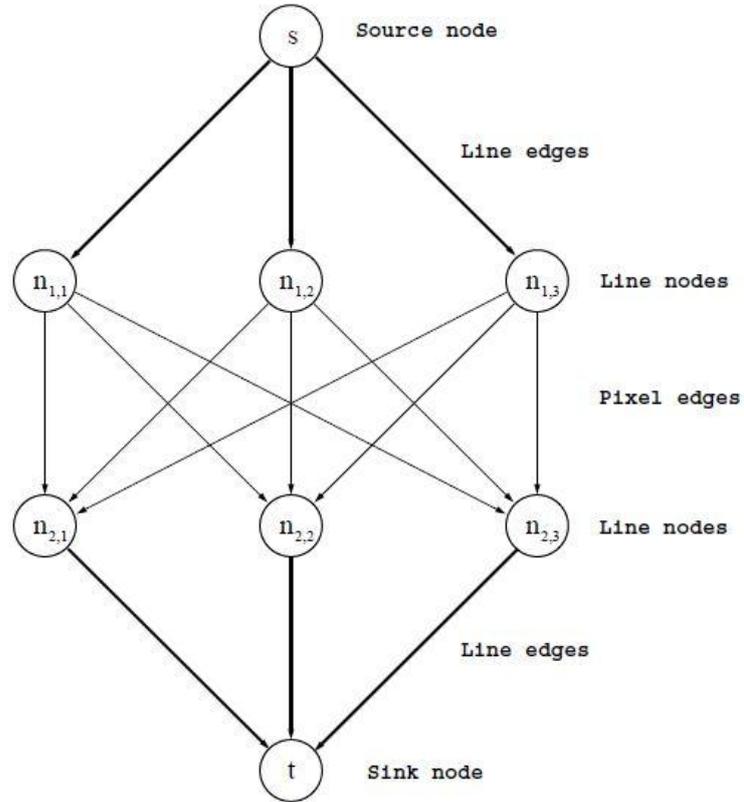


Figure 2.7. Basic structure of the associated graph [Bat08].

Let ϕ be the edge to point mapping of G .

The flow corresponding to a binary image X is defined as a mapping $F_X : E \rightarrow \{0, 1\}$ which has the values:

$$F_X((n_{1,i}, n_{2,j})) = \begin{cases} 1 & \text{if } \phi((n_{1,i}, n_{2,j})) \in X \\ 0 & \text{otherwise} \end{cases} \quad (2.12)$$

The cost function assigned for each edge determines the cost of sending a certain amount of flow through that edge. The cost function depends on the weight map W and has the values:

$$\begin{aligned}
U((n_{1,i}, n_{2,j})) &= -W(\Phi((n_{1,i}, n_{2,j}))) && \text{for } 1 \leq i \leq L_1, 1 \leq j \leq L_2 \\
U((s, n_{1,i})) &= 0 && \text{for } 1 \leq i \leq L_1, \\
U((n_{2,j}, t)) &= 0 && \text{for } 1 \leq j \leq L_2.
\end{aligned} \tag{2.13}$$

Using the fact that the two-projection reconstruction problem can be solved efficiently with the network flow method using two projections, an iterative algorithm can be considered, to reconstructed images using more than two projections. In the iterative algorithm, in each iteration, a new reconstruction is obtained using two projections and the result obtained in the previous reconstruction. In this way, the obtained reconstruction becomes closer to the original image. Next section describes briefly such an iterative algorithm for the case of more than two projections.

2.4.2. ITERATIVE RECONSTRUCTION USING MORE THAN TWO PROJECTIONS

The previous section presents the formulation of the two projection reconstruction problem as a network flow problem that leads to a good binary solution in the case of parallel beam projections. Considering more than two projections, let $\theta_1, \dots, \theta_d$ be the given different projection angles and $p_k: L_k \rightarrow \mathbb{N}$ given functions. The reconstruction problem consists of constructing an image X such that $P_k = p_k$ for $k=1, \dots, d$.

In [Bat06b][Bat07a] an iterative algorithm to solve the binary image reconstruction problem using more than two projections, is presented. The proposed approach uses the fact that the two projection problem can be effectively solved as a network flow problem. In each iteration the algorithm solves the two projection problem using a new pair of projection angles each time. Also the concept of the weight map is used, that is computed using the reconstruction from the previous iteration, in this way the new reconstruction incorporates the previous one. A similar approach was also described in [Gri00] but in this approach the smoothness assumption is not used. The basic steps of the algorithm proposed by Batenburg are presented in the Pseudocode sequence 2.1 [Bat07a].

The first step of the algorithm is the generation of a starting solution. The starting solution can be generated in different way and it has to offer a good approximation of the image to be reconstructed. The first step is to find a real-valued solution. One way in which a real-valued solution can be found, is by using a Simultaneous Iterative Reconstruction Algorithm (SIRT) [Kak01]. Another solution can be to solve a real-valued relaxation of the binary tomography problem, using an iterative method considering a fix number of iterations [Tan71]. Besides these algorithms, the initialization of the initial image with equal values [Bat06a] can also be considered as a good solution. After finding a real-valued starting solution the binary starting solution is computed by solving a min-cost max-flow problem for the selected pair of projections for the first iteration. The values of the real-valued solution are used as pixel weights for the corresponding network flow problem.

Pseudocode sequence 2.1. Basic steps of the algorithm [Bat07a]

Compute the real-valued start solution $X^* = (x_{ij}^*)$

Compute the binary start solution X^0 by solving a min-cost max-flow problem for angles ϑ_1 and ϑ_2 , using $w_{ij} = x_{ij}^*$

$i := 0$;

While (stop criterion is not met) do

$i := i + 1$;

Select a new pair of angles ϑ_a and ϑ_b ($1 \leq a < b \leq k$),

Compute a new solution X^i by solving the min-cost max-flow problem for directions ϑ_a and ϑ_b , using a weight function;

End While

The pixel weights for all the pixels in the image are computed, in each iteration, depending on the value of the pixel obtained in the reconstruction from the previous iteration and on the values of the pixels in a neighborhood. There can be several ways in which one can compute the pixel weight. In [Bat06b][Bat07a][Bat08], the author proposed different possibilities in which the weight function can be defined.

In every iteration a new pair of projection angle different from the one used in the previous iteration is selected. When the number of projections is small there are considered all the possible pairs of two projection angles, in order to avoid cycling. If there is a bigger number of projection angles not all the projection angles are suitable for solving the two-projection problem. There can be several criteria for choosing the projection pair in a new iteration. In [Bat07a][Bat08] there are some criteria presented for choosing the new pair of projections in the case of discrete and continuous X-rays.

The stopping criteria used imply the computation of the distance between the prescribed projection and the values of the projections in the current reconstructed image:

$$\Delta(X) = \sum_{k=1}^d |P_k - p_k| \quad (2.14)$$

The algorithm terminates if no improvements has been made in the projection distance during a constant number of iterations. Moreover if the projection distance becomes less than a certain constant the algorithm will terminate after a number of default iterations.

2.5. RECONSTRUCTION AS AN OPTIMIZATION PROBLEM

As described in the previous section, the solution of the reconstruction problem is equivalent to find a solution that minimizes as much as possible the distance between the prescribed projections and the values of the projections in the reconstructed image.

The iterative method provides a binary solution but a big number of projection values are needed in order to obtain the solution.

In discrete tomography applications, an important property of binary images is that the number of projections is usually much less than the number of unknowns. This means that several solutions exist. A possible way to solve the reconstruction problem using a small number of projections, at least approximately is to reformulate it as an optimization problem. Optimization techniques search the best of the solution, by measuring the quality of the solution using an evaluation function.

An evaluation function $f : D \rightarrow \mathbb{R}$ is defined over a domain D with values in the set of real numbers. The best solution means finding a value $x \in D$ which gives the minimum of the function f :

$$f_{min}(x) = \min_{x \in D} f(x) \quad (2.15)$$

If the evaluation function does not reflect the goodness of the solution, optimization algorithms cannot guarantee finding a good solution to the problem. On the square grid, the reconstruction problem was efficiently solved by using genetic algorithm [Nak94][Val08], naïve genetic algorithm [DiG10], or memetic algorithm [Bat05][DiG08]. In the proposed solutions, the authors proposed more types of evaluation functions. In the case of the memetic algorithm, one should find the minimum of the objective function defined as the distance between the prescribed projections and the values of the projections in the reconstructed image. In the following section the memetic algorithm approach is considered.

2.5.1. OVERVIEW OF MEMETIC ALGORITHM APPROACH

Memetic algorithms (MA) [Mos89] were introduced by Moscato in his technical report in 1989. MA's are inspired by Darwinian principals of natural evolution [Daw83], that provides a unifying framework governing the evolution of any complex system, and Dawkins' notion of a meme [Daw76] that is "the basic unit of cultural transmission, or imitation". A memetic algorithm can be considered as a combination between population-based hybrid genetic algorithm (GA) and individual learning procedure that performs local search. A presentation of the memetic algorithm is included in [Gar09].

In the memetic algorithm the population is initialized randomly or using a heuristic. For selecting the new population a selection algorithm identical to those from genetic algorithms is used. Also in the generation of new individuals from two selected parents, classical operators of crossover and mutation are used. For improving the fitness each individual makes local search using a local hill climbing. Using the local search in memetic algorithm, the local optimum is more efficiently found. Pseudocode sequence 2.2 presents the steps of the memetic algorithm, and respectively shows the basic steps for a hill climbing local search [Gar09].

Memetic algorithms search for the best solution using a population of agents, not a single agent. In each step a population of solutions is processed, considering agents from many points in the search space, in this way being assured a better chance to find the global optimum. The generation of the new population with better solutions depends on the values of the evaluation function called fitness in genetic algorithms. Better agents, which represent better solutions, have a higher probability to survive the evolution process than agents with worse fitness. In the case of reconstruction methods using memetic and genetic algorithms the (approximate) global optimum of the fitness function is found but it is not sure that the evaluation function is relevant to the problem, describing the goodness of a particular agent.

Pseudocode sequence 2.2. Steps of a generic memetic algorithm [Gar09]

```
Encode solution space;
Set population_size, generation_size, generation=0;
Set crossover_rate and mutation_rate;
Initialize population;
While(generation < generation_size) do
    Evaluate fitness;
    For(i=1 to population_size)
        Select parents;
        If(random(0,1) <= crossover_rate) then
            Child = crossover(parents);
        End if
        If(random(0,1) <= mutation_rate) then
            Child = mutation();
        End if
        Repair child if necessary;
        While (termination condition is not satisfied) do
            New solution = neighbors(best solution);
            If new solution is better than actual solution then
                Best solution = actual solution
            End if
        End while
    End for
    Add offspring to new generation
    Generation=generation+1;
End while
```

As in the case of reconstruction of binary images, the measured projections are known, it is advantageous to generate the initial population of agents by using this information so that the agents to be as fit as possible and the search for the

optimum to be easier. Among the algorithms used are the network flow algorithm [Gal57] and Ford-Fulkerson algorithm [For56].

The operators used in the case of memetic algorithms are the one specific for the genetic algorithm, crossover and mutation operators which lets the solution evolve toward an optimal or close to optimal solution. Beside these operators local search is performed by using the switch and compactness operators that improve the quality of the reconstructed images from a generation.

The selection operator leads the evolution to an optimal solution by assuring the survival of the fittest solutions. The operator creates a new population containing agents with the best fitness values. This operation is usually a probabilistic one. The probability that an agent will be part of the new population, will survive, is given of its relative fitness in the current population. In the new population there may exists agents that have the same fitness value.

For the crossover operator there are many possibilities. The basic idea is to combine randomly selected pairs of agents choosing different parts from each of the agents. In the end not all the agents from the population are used in the crossover process. Usually a random number of pairs of agents are selected for the crossover process.

Selection together with crossover is the main power of genetic algorithms. However there is another operator called mutation which is used to protect the lost of local configurations of elements of an agent as a result of selection and crossover. The main principal of the mutation operator is to randomly change one element of an agent.

Memetic algorithms perform a local search by using a hill climb operator based, in the case of discrete tomography, on the compactness and switching operators.

The criteria used for stopping the evolution process is to use besides the verification if the solution is found, a generation size, or a condition in which it is verified if the best fitness in the population improved substantially in a number of previous generations.

Description of such algorithms used in the discrete tomography can be found in [Bat05][DiG08][DiG10]. Descriptions and related techniques are included such as implementation of crossover and mutation operators, hill climbing operator generation of the initial population, etc.

2.6. CONCLUSIONS

The chapter presents an overview of the problem of discrete tomography for images represented on the square grid. The first studied grid in discrete tomography was the square grid, because it uses Cartesian coordinate system. Optimization algorithms and iterative reconstruction based on network flow algorithm are two common methods in reconstructing images from the acquired projections presented above. In the following chapter another grid that can be used efficiently for the discrete tomography problem, the triangular grid is presented.

3. TRIANGULAR GRID

3.1. Introduction	33
3.2. Features of triangular grid	34
3.3. Triangular grid image representation	36
3.4. Topology of the triangular grid	37
3.5. Lanes and diamond-chains on the triangular grid	38
3.6. Capturing and displaying of triangular sampled images	39
3.7. Triangular grid image reconstruction problem using projections	40
3.8. Conclusions	43

There are three possible regular grids, in the Euclidean space. Among these, the triangular grid can be used, in an efficient way, for image representation. In this chapter we describe the triangular grid and the image representation on the triangular grid, with their structure and properties. Subsequently, we illustrate what we mean by projections of an image represented on the triangular grid. Also we describe the binary tomography reconstruction problem for images represented on the triangular grid.

3.1. INTRODUCTION

In digital geometry (DG), the space consists of discrete points with integer coordinates. The Euclidean plane has three possible regular tiling or grid: square, triangular and hexagonal [Gru90] (see Figure 3.1). The square grids are widely used in digital geometry [Gon07][Bat06a][DiG10]. The hexagonal grid is also well-known [Her95]. In [Fay12] the hexagonal grid is described presenting the features of the hexagonal sampling scheme and the way in which hexagonal structures can be addressed to. Her [Her95], described the hexagonal grid using three coordinates of sum zero, because of its symmetric properties. The triangular grid is the third regular grid. Because of the triangular symmetry we will also refer to the triangular grid using three coordinates [Nag01][Nag03a][Nag02]. In the next sections the 2D triangular grid based on Nagy's description in [Nag01][Nag03a][Nag02][Nag03b] and Grunbaum's description in [Gru90] are described.

In this chapter an overview of the triangular grid and its properties is presented. First some features and the coordinate representation of the triangular grid are described, introducing then the concept of images on this grid. A topological description and definitions for lanes and diamond-chains on the triangular grid are also presented.

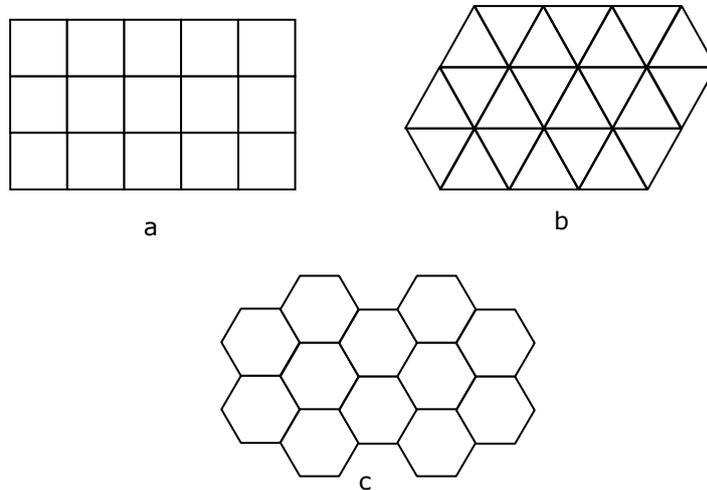


Figure 3.1. Euclidean plane tiling with regular polygons (a) square (b) triangular (c) hexagonal.

3.2. FEATURES OF TRIANGULAR GRID

A digital image may be considered as a binary representation of a two-dimensional image. Digital images can be mapped on a triangular grid, containing more information in the same image than in the case in which are mapped on the square grid.

The triangular grid is formed by tiling the plane regularly with equilateral triangles. Triangular grids began to play an important role in geometric modeling. Many 3D scanners produce generally non regular triangular grids which at high resolution become close to the regular triangular grid. Also human retina is often modeled using a triangulation, namely the Delaunay triangulation [Cel00][Pac05]. Many studies and algorithms in computer graphics are made on the triangular grid. Also in digital geometry [Nag01][Nag04] many studies are made using the triangular grid. Nagy presented some properties of the triangular grid and defined a distance based on neighborhood relations in this grid [Nag02][Nag03a].

In [Fon98][Fon01], Fontana and Rocchesso have demonstrated that a 2D medium modeled by a square grid can be also modeled by a triangular grid using 13.4% less samples per unit area. This means that triangular grid sampling requires a less number of samples to represent the same information than square grid, reducing the storage data space.

In the triangular grid the internal angle of the equilateral triangle is 60 degrees. In consequence six triangles at a point cover 360 degrees. The Schläfli symbol of the triangular tiling is $\{3, 6\}$, 6 triangles around every vertex. The triangular grid gives a tighter arrangement than the square tiling [Xia05]. The nodes of the triangular grid are the centers of the densest possible circle packing. The

packing density is $\frac{\pi}{\sqrt{12}}$ or 90.69%. Adjacent pixels in triangular grid are separated

by one hundred twenty degree instead of ninety degree which permits a better representation of curved images.

The triangular tiling has a 6-fold rotational symmetry which means that it has higher symmetry than the square grid [Fay12]. Figure 3.2 shows the rotational symmetry of square and triangular grid sampling. Due to additional rotational symmetry an image sampled on a triangular grid suffers less interpolation error under rotation as an image defined on a square grid. The elementary vectors of a regular triangular tiling are given by $\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1/2 \\ \sqrt{3}/2 \end{pmatrix}$.

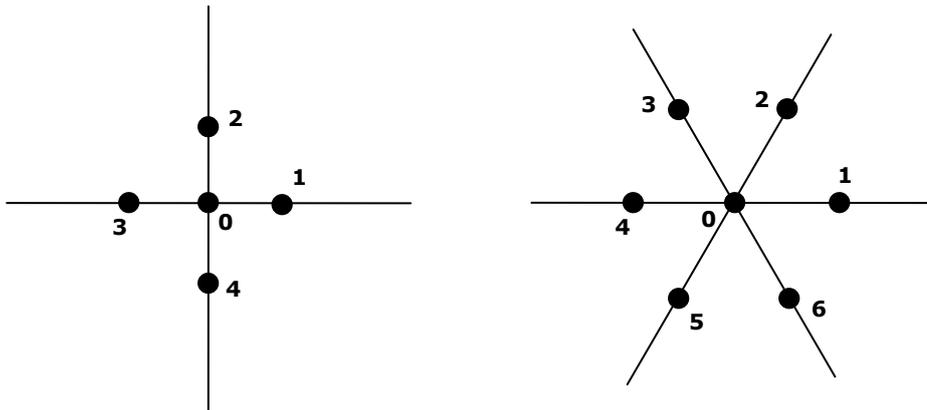


Figure 3.2. Rotation symmetry of (left) square and (right) triangular grid nodes.

The dual of the triangular grid is the hexagonal grid, the grid of triangular regions being equivalent with the grid of hexagonal nodes (see Figure 3.3). In the triangular grid there are three types of connectivity: three way connectivity, nine way connectivity and twelve way connectivity. One pixel has only one common edge and two common corners, having three 1-neighbors, nine 2-neighbors (the three 1-neighbors and six more 2-neighbors) and twelve 3-neighbors (nine 2-neighbors and three more 3-neighbors) [Nag03b]. In Figure 3.3 the dark gray points represent the 1-neighbors, the light gray points represent the 2-neighbors and the white points represent the 3-neighbors.

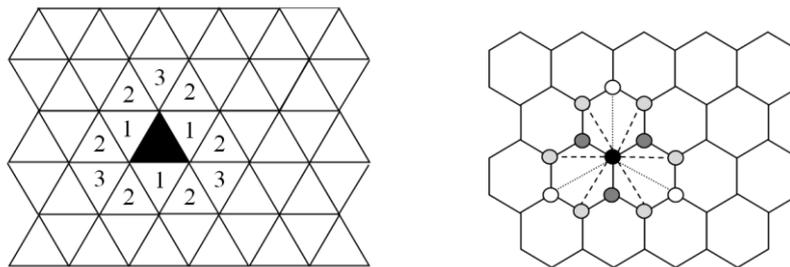


Figure 3.3. Neighborhood relations in the triangular grid of regions and the hexagonal grid of nodes.

In the triangular grid there are three types of distance measures; the distance between the pixel and the 1-neighbors, the distance between the pixel and the 2-neighbors which is $\sqrt{3}$ times than the distance between the pixel and the 1-

neighbors and the distance between the pixel and the 3-neighbors which is 2 times than the distance between the pixel and the 1-neighbors (see Figure 3.3).

3.3. TRIANGULAR GRID IMAGE REPRESENTATION

For the representation of points in the triangular grid three coordinates x, y, z are used. These coordinates depend one of the other since the point is part of a two dimensional plane. The coordinate system used for representing the triangular data is based on the coordinate system for the hexagonal grid denoted with $\ast\mathbb{R}^3$, proposed by Her [Her95]. The procedure for assigning the coordinate values to the points of the triangular grid is described by Nagy in [Nag01][Nag03a].

Procedure 3.1. (from [Nag03a]): Choose a point for the origin, whose coordinate values are $(0, 0, 0)$. Take the three lines through the center of the origin triangle, which are orthogonal to its sides. Fix these lines as the coordinate axes x, y and z , as shown in Figure 3.4. The coordinate values are assigned to the points inductively. Let the coordinate values of a triangle A be known. Consider a triangle B , which has not coordinate values yet and has a common side with A . This common side is orthogonal to one of the coordinate axes. According to the direction of this axis, the corresponding coordinate value of A is increased or decreased by 1 to get the corresponding coordinate of B . The other two values of A and B are equal.

In the triangular grid there are two types of grid-points, called triangle-pixels, based on the orientation of the triangle, even points \blacktriangle and odd points \blacktriangledown . For the even points the coordinate sum is zero and for the odd points the coordinate sum is one (see Figure 3.4).

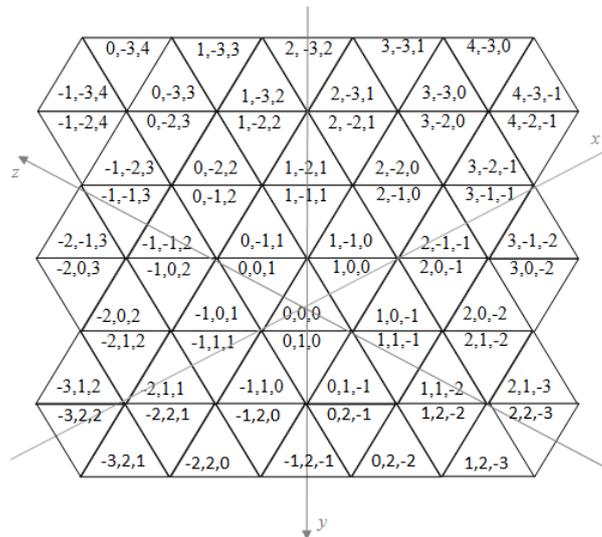


Figure 3.4. Coordinate system of the triangular grid.

In [Nag03a], Nagy gave the description of the neighborhood relation on the triangular grid by using the coordinate values.

Definition 3.1.: (from [Nag03a]): Let p and q be two points on the triangular grid. The i^{th} coordinate of the point p is indicated by $p(i)$ ($i=1,2,3$), and similarly for q . Then the points p and q are m -neighbors ($m=1,2,3$), if the following conditions hold:

$$\begin{aligned} |p(i) - q(i)| &\leq 1, \quad \text{for } 1 \leq i \leq 3, \\ |p(1) - q(1)| + |p(2) - q(2)| + |p(3) - q(3)| &\leq m. \end{aligned} \quad (3.1)$$

3.4. TOPOLOGY OF THE TRIANGULAR GRID

In digital geometry and image processing, topology plays an important role. The topological description of a grid contains more information about the image than the usual description. Using the topological description not only the pixels of the two dimensional image can be stored, but also the lower dimensional segments, the edges and the points separating the pixels. Nagy proposed in [Nag12] a description of cellular topology on the triangular grid. For topological calculations regarding the cell complex concept he made a modification for a two dimensional image, which contains not only two-dimensional pixels, but one dimensional edges and zero dimensional points. For addressing the zero and one dimensional cells of the triangular grid, beside the triangle pixels, the edges between two neighbor triangles and their endpoints are used. The elements of the triangular grid are addressed using coordinate triplets similar to the method given by Kovalevsky [Kov04][Kov08]. In order to define a coordinate system that will address the edges between two neighbor triangles and their endpoints, the (x, y, z) coordinates triplet is replaced by $(2x, 2y, 2z)$. In this coordinate system, the coordinate sum of the even-triangles (\blacktriangle) is zero and the coordinate sum of the odd-triangles (\blacktriangledown) is two (see Figure 3.5).

The coordinate triplet for an edge between two 1-neighbor triangles (x_1, y_1, z_1) and (x_2, y_2, z_2) is $\left(\frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2}, \frac{z_1 + z_2}{2}\right)$. In the triplet there are two even and one odd values and the sum of the coordinate values of every edge is one. Considering the gridlines orthogonal to the coordinate axis the edges are situated on these gridlines, in the way that for the edges orthogonal to axis x the odd coordinate is the first element, for the edges orthogonal to axis y the odd coordinate is the second element and for the edges orthogonal to axis z the odd coordinate is the third element [Nag12].

The nodes of a triangular grid, form a hexagonal grid and so they can be addressed with triplets that have a fixed coordinate sum [Nag03b][Nag08a][Nag08b]. For an endpoints, the nodes, coordinate triplet has only odd values and there sum is one [Nag12].

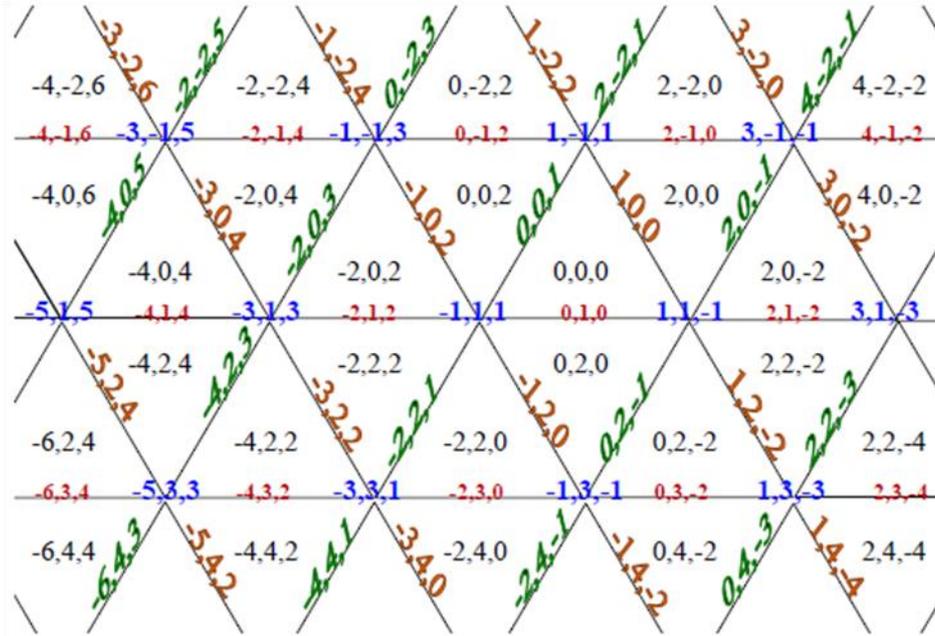


Figure 3.5. The topological coordinate system for a segment of the triangular grid, from [Nag12].

3.5. LANES AND DIAMOND-CHAINS ON THE TRIANGULAR GRID

The definition for the lanes on the triangular grid was given by Nagy in [Nag03a]. In this section, the definition for the diamond-chains on the triangular grid is also given.

Definition 3.2.: *The points, which have the same value for a corresponding coordinate, form a lane.*

Definition 3.3: *The points, for which two coordinate values, have a fixed difference, form a diamond-chain.*

A lane consists of even and odd triangle pixels alternately for which a coordinate value is fixed, e.g., $x = 2$ form a lane. Figure 3.6.a shows a lane in y orthogonal direction (green) and a lane in z orthogonal direction (pink). Also diamond-chains of y parallel direction (yellow) and diamond-chains of x parallel direction (blue) are shown in Figure 3.6.b.

The points of a lane, direction that is orthogonal to the coordinate axes, have one fix coordinate value and the other two coordinate values vary with ± 1 unit. The coordinates of the point from a diamond-chain, direction that is parallel to the coordinate axes, fulfill the condition that two coordinate values have a fixed

difference. For example, Figure 3.6.b the yellow diamond-chain has the points $\{(p(1), p(2), p(3)) | p(1)-p(3)=1\}$.

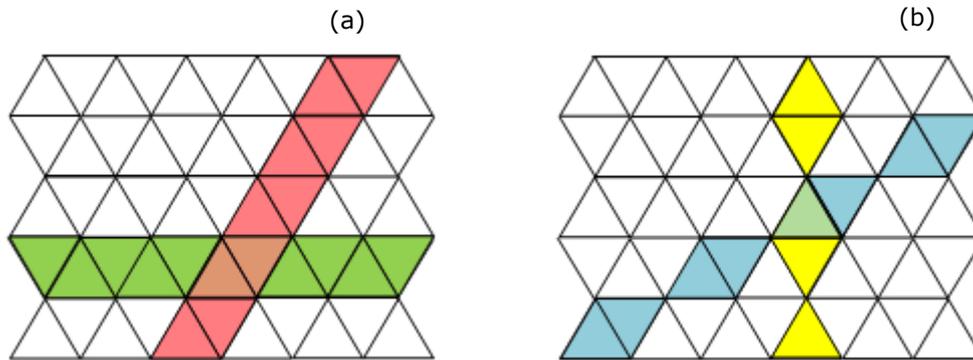


Figure 3.6. Examples of (a) lanes orthogonal to coordinate axes and (b) diamond-chains parallel to coordinate axes.

Considering the three lanes, for the directions orthogonal to the coordinate axes, there are two points in the intersections of two lanes (Figure 3.6.a), and only the third lane can distinguish these two points since two of their coordinate values are common. In the case of the three diamond-chains, for the directions parallel to the coordinate axes, there is one or no point in the intersection of two diamond-chains.

A lane of the triangular grid can be described using the pixel coordinates. A more easy way of addressing and constructing the lanes and diamond-chains of a triangular grid is by using the topological description from [Nag12], and address to the points that separate the pixels.

3.6. CAPTURING AND DISPLAYING OF TRIANGULAR SAMPLED IMAGES

Since devices usually capture square sampled images, these images have to be manipulated using software tools to obtain triangular sampled images. Resampling is the manipulation of data sampled on one grid to produce data sampled on a different grid. Resampling methods were proposed especially for the hexagonal grid. Watson and Ahumada [Wat89] proposed the hexagonal orthogonal-oriented pyramid method for resampling, in which the affine relationship between the square and hexagonal lattice points is used. Another approach was of Fitz and Green [Fit96] whom approximated the hexagonal lattice with a brick wall obtained by shifting by half a pixel the pixels in alternate rows. Her [Her94] proposed an interpolation method that halves the vertical resolution. Also other least squares approximation of splines methods were proposed for resampling square images onto hexagonal ones [Van02][Van03][Van04]. Since the triangular grid is the dual of the hexagonal grid these sampling methods can also be adapted for the triangular grid. Refer to Appendix A for detailed description of the formulas used for converting coordinates of pixels on a triangular grid into corresponding pixel coordinates on a square grid and vice-versa. Hartman and Tanimoto [Har84] approach resample in

the best way a triangular grid. In the approach a hexagonal lattice is constructed by generating triangular pixel from two square pixels that are vertically adjacent. The triangular pixels obtained in this way are almost perfectly equilateral triangles. The triangles have a base angle of 63.4° and a top angle of 53.2° . Figure 3.7 shows the resampling scheme proposed by Hartman and Tanimoto. The black squares indicate the square grid pixels.

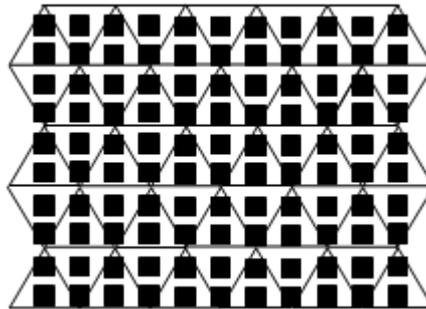


Figure 3.7. Hartman and Tanimoto resampling scheme

3.7. TRIANGULAR GRID IMAGE RECONSTRUCTION PROBLEM USING PROJECTIONS

In this section it is illustrated what means the hexagon-shaped image and how this image should be represented using the triangular grid. In order to understand the reconstruction problem of hexagon-shaped images on triangular grid, a brief presentation of the triangular grid projections is made.

A binary hexagon-shaped image represented on the triangular grid is a binary hexagonal set of size $m \times m \times m$. The number of pixels in such a hexagonal set is $6m^2$. Because discrete tomography and digital image processing are strongly related, the binary hexagonal sets are called images and the entries of the sets are called pixels with values black (0) and white (1). Using the triangular grid representation of the image, there are two types of pixels which depend on the orientation of the triangles, even (\blacktriangle) and odd (\blacktriangledown). Since the grid is a 2D one, the coordinate values are not independent of each other, but each triangle of the shape \blacktriangle is addressed with a triplet having sum 0, and each triangle of the shape \blacktriangledown is addressed with a triplet with sum 1. Figure 3.8 shows the coordinate representation of a hexagon-shaped image of size $m=3$ represented on the triangular grid.

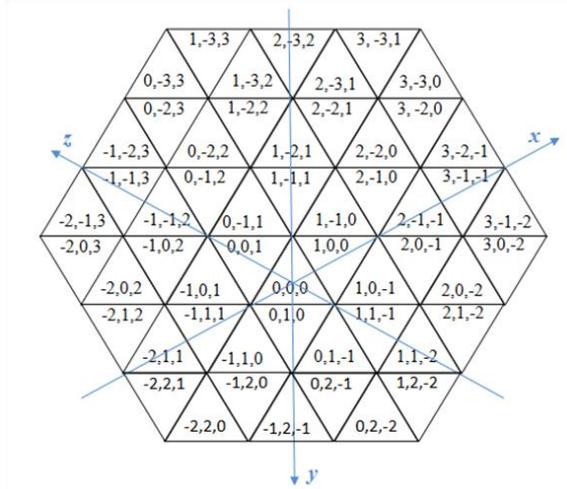


Figure 3.8. Coordinate representation of a hexagon-shaped image of size $m=3$.

A binary hexagon-shaped image is a set of pixels $A=\{a_{ijk}\}$, $i, k \in [-m+1,m]$, $j \in [-m,m-1]$ and i, j, k satisfy the following constrain $i+j+k=0$ or $i+j+k=1$, depending on the orientation of the triangle pixel.

As an example, a simple 2D binary hexagon-shaped image of six pixels and some values is considered (see Figure 3.9). The image shown in Figure 3.9 is a 2D binary hexagon-shaped image and represents a simplification of the cross section of an object when the scanner beams pass through.

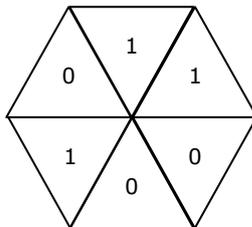


Figure 3.9. 2D binary hexagon-shaped image of six pixels, represented on the triangular grid.

Three projections for the considered image are obtained pointing the beam source at three angles and generating the projections as shown in the Figure 3.10. In Figure 3.10, the three projection angles are $\pi/6$ (a), $\pi/2$ (b) and $5\pi/6$ (c) degree with the x-axis.

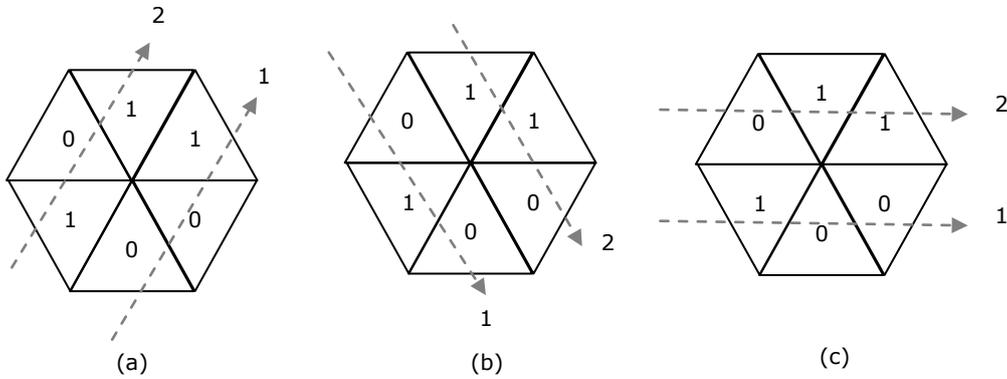


Figure 3.10. (a) First projection at angle $\pi/6$ degrees with the x -axis; (b) Second projection at angle $\pi/2$ degrees with the x -axis; (c) Third projection at angle $5\pi/6$ degrees with the x -axis.

These projections are the sum of the pixel values encountered by the beam along the path. After obtaining these 3 projections, the problem that has to be solved is how to determine the original hexagon-shaped image using the projections measured along the considered projection angles Figure 3.11.

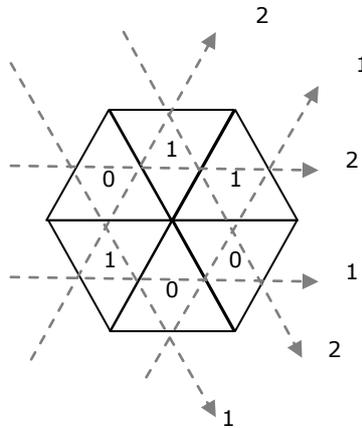


Figure 3.11. Three projections of a hexagonal image represented on the triangular grid

The reconstruction problem of a hexagon-shaped image from its projections is one that can have more than one solution for the same entering data. This means that for a specific set of projections corresponds more than one image.

Taking into consideration the existing methods for the square grid, there are many possible methods that can be used to solve the image reconstruction problem. These methods are not studied and used, yet, for the triangular grid case. Images can be reconstructed from projections using reconstruction algorithms like algebraic reconstruction, Fourier central Slice theorem, filtered backprojection, network flow algorithm, optimization algorithms or other algorithms.

3.8. CONCLUSIONS

In image processing and computer graphics, triangular and hexagonal grids are used more and more fervently [Nag03b][Luk12][Fay12][Mid05].

The aim of this chapter is to provide an overview of the triangular grid, its representation and some properties. The three coordinate representation of the triangular grid and also some topological aspects were presented. Because the structure of the grid contains two types of triangles - even-triangles and odd-triangles-, a three coordinate system is used to address the points in a triangular grid. Also some important aspects concerning lanes orthogonal to coordinate axes and diamond-chains parallel to coordinate axes are introduced and explained.

Traditionally, hardware for image capturing and displaying images has been based on the square grid. Unfortunately, for the moment there is no hardware equipment available for capturing and displaying triangular grid based images and, from our knowledge, there are no databases with images on the triangular grid that can be used to test image processing algorithms.

In the next chapters a deeper mathematical analysis of the representation of projections and of the generation of a solution using binary tomography approaches based on triangular grid, are proposed.

4. RADON TRANSFORM ON TRIANGULAR GRID BASIS

4.1. Introduction	44
4.2. The Radon transform	44
4.3. Conclusions	48

The Radon transform represents the projection data obtained after a tomography scan considered for an unknown object. The inverse of the Radon transform can be used to reconstruct the original object from the measured data. This represents the mathematical support for tomography reconstruction, also known as image reconstruction. In this chapter the Radon transform mathematical formulation for the triangular grid is derived by the author.

4.1. INTRODUCTION

In tomography, the first phase is the generation of the set of projections, which mathematically is accomplished using the Radon transform. In this chapter the mathematical framework concerning the projections which are generated applying the Radon transform on non-orthogonal basis, respectively with triangular grid basis, is described.

The Radon transform is used for object detection from 1965 [Baz65]. Then in 1969, Rosenfeld describes this technique in his work [Ros69]. But neither of the authors identifies this technique as the Radon transform. As mentioned in section 2.2, the Radon transform, named after the Austrian mathematician Johann Radon, is an integral transform consisting of the integral of a function over straight lines. In his work, Johann Radon showed how a function can be described in terms of its projections [Rad17]. The Radon transform is the mapping of the function on the projections. The inverse Radon transform is used to reconstruct images from their projections [Nat01][Nat011].

The proposed problem is to reconstruct an image or object using the triangular grid. In order to mathematically describe the projections on the triangular grid, the Radon transform is revised and modeled for the triangular grid.

4.2. THE RADON TRANSFORM

The basis vector that underlies the triangular grid is $[1 \ 0]^T, \left[\frac{1}{2} \ -\frac{\sqrt{3}}{2}\right]^T$,

which means that there is one vector parallel to the x axis and a second one rotated with 120° clockwise from the first one considered. In order to address a triangular grid, three axes of symmetry of the triangle are used instead of two. The third one is a linear combination of the other two axes. The corresponding coordinate scheme

uses a triplet of coordinates $(i, j, k) \in {}^*\mathbb{R}^3$ corresponding to the distance from the lines $x = 0$, $y = 0$, and $z = 0$ respectively, and they obey the following rule: $i+j+k=0$ or $i+j+k=1$.

A 2D object in the xyz triangular grid reference frame is considered and a beam L going through the object at angle θ , as shown in Figure 4.1. The equation of the beam L , is:

$$L(r, \theta) = \left\{ (x, y, z) \in {}^*\mathbb{R}^3 : \frac{1}{\sin 60^\circ} \left((x - z) \sin \theta + (y - z) \sin(60^\circ - \theta) \right) = r \right\} \quad (4.1)$$

where r is the distance of L from the origin and θ is the angle the normal vector L makes with the x axis.

Any point (x, y, z) on the line L , with specific r and θ , satisfies the equation $\frac{1}{\sin 60^\circ} \left((x - z) \sin \theta + (y - z) \sin(60^\circ - \theta) \right) = r$ where $x+y+z=0$ or $x+y+z=1$.

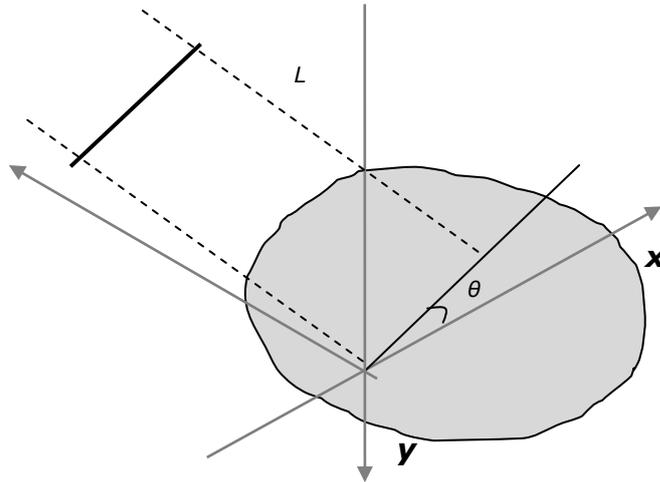


Figure 4.1. Parallel beam, L , going through an object at an angle θ .

If a function $f(x, y, z)$ represents an unknown object, then its integral over the line $L(r, \theta)$ is:

$$I = \int_L f(x, y, z) ds \quad (4.2)$$

with $(x, y, z) \in {}^*\mathbb{R}^3$, $x+y+z=0$ or $x+y+z=1$, where ds is the differential element of the line L .

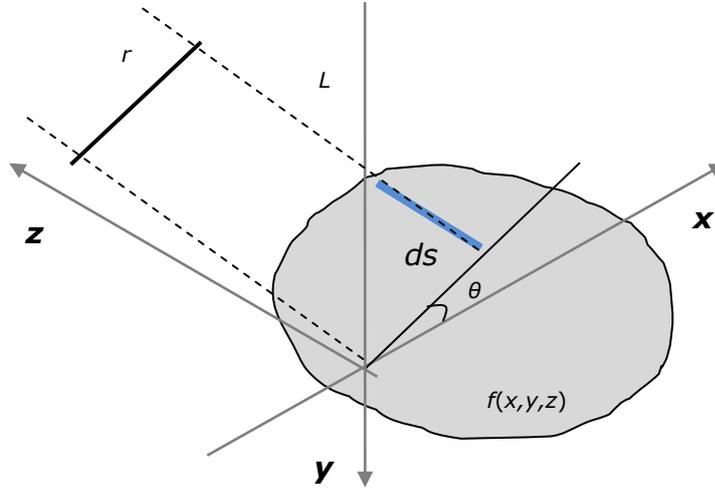


Figure 4.2. The differential element of the line L

In order to write the equation (4.2) in terms of x , y and z , the Dirac delta function is used, in this way the above equation can be rewritten as:

$$I = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y, z) \delta \left(\frac{1}{\sin 60^\circ} \left((x - z) \sin \theta + (y - z) \sin (60^\circ - \theta) \right) - r \right) dx dy \quad (4.3)$$

with $(x, y, z) \in \mathbb{R}^3$, $x + y + z = 0$ or $x + y + z = 1$.

In mathematics, the Dirac delta function [Dir27] or δ function is a generalized function that can be defined as the limit of a class of delta sequences. Function (4.3) is known as the Radon transform (or sinogram) of the 2D object $f(x, y, z)$ over the line $L(r, \theta)$. In mathematics, the projection-slice theorem states that having an infinite number of one-dimensional projections of an object taken at an infinite number of angles, the original object $f(x, y, z)$ can be perfectly reconstructed. So to get $f(x, y, z)$ back, from the above equation means finding the inverse Radon transform. The result of the formula (4.3) is a numerical value, representing the line integral value. Figure 4.3 shows a projection line in which the results of these line integrals are accumulated.

Considering many parallel lines and performing the same line integral of $f(x, y, z)$ over each line, many radon transform integrals are obtained as shown in Figure 4.4. Since all the considered lines are parallel to line L , all of them have the same angle θ and different r 's.

The projection shown in Figure 4.4 is a discrete function which depends on r and θ , called the total attenuation, and is denoted by:

$$p(r, \theta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y, z) \delta \left(\frac{1}{\sin 60^\circ} \left((x - z) \sin \theta + (y - z) \sin (60^\circ - \theta) \right) - r \right) dx dy \quad (4.4)$$

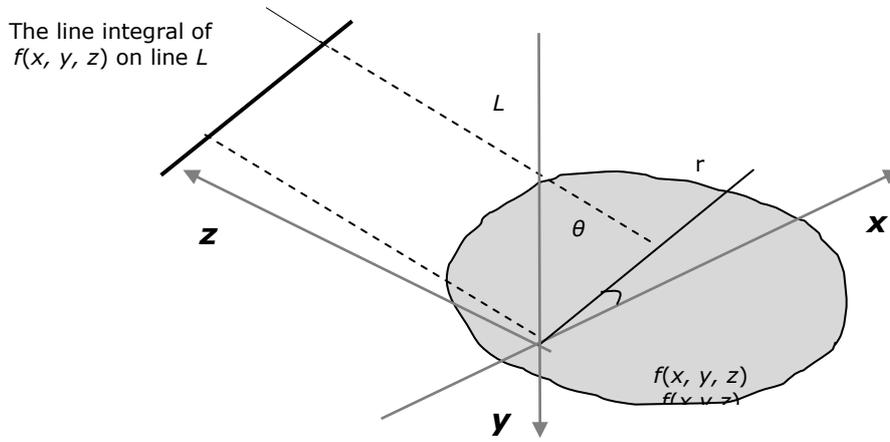


Figure 4.3. A projection line - the line integral of $f(x, y, z)$ on line L .

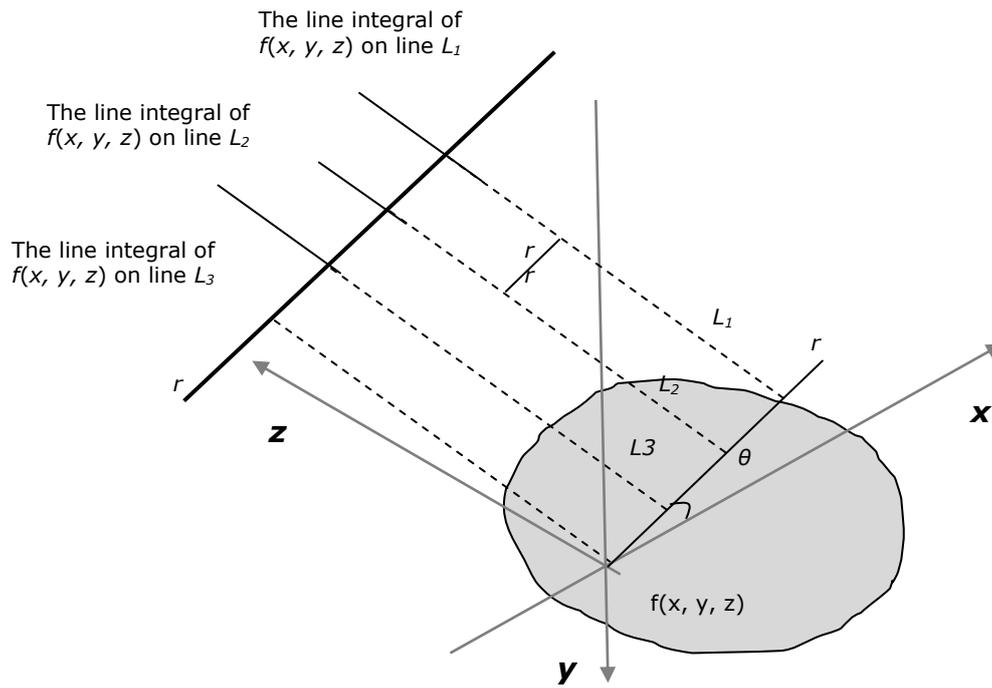


Figure 4.4. Radon transform integrals

The Radon transform $p(r, \theta)$ at angle θ is defined as the integral of function $f(x, y, z)$ with $(x, y, z) \in \mathbb{R}^3$, $z = -x - y$ or $z = 1 - x - y$ for all x, y and z which satisfy the equation $\frac{1}{\sin 60^\circ} \left((x - z) \sin \theta + (y - z) \sin(60^\circ - \theta) \right) - r = 0$. The coordinates (r, s) are related to $(x - z, y - z)$ by a rotation matrix:

$$\begin{bmatrix} r \\ s \end{bmatrix} = \frac{1}{\sin 60^\circ} \begin{bmatrix} \sin \theta & \sin(60^\circ - \theta) \\ \sin(60^\circ + \theta) & -\sin \theta \end{bmatrix} \begin{bmatrix} x - z \\ y - z \end{bmatrix} \quad (4.5)$$

$$\begin{bmatrix} x - z \\ y - z \end{bmatrix} = \frac{1}{\sin 60^\circ} \begin{bmatrix} \sin(60^\circ - \theta) & \sin \theta \\ -\sin \theta & \sin(60^\circ + \theta) \end{bmatrix} \begin{bmatrix} r \\ s \end{bmatrix} \quad (4.6)$$

Since $(x - z, y - z)$ and (r, s) are related by a rotation transformation, shown in equation (4.5) and (4.6) (refer to Appendix B for detailed derivation), equation (4.4) can be written as:

$$p(r, \theta) = \int_{-\infty}^{\infty} f \left(\frac{1}{\sin 60^\circ} \left(r \sin(60^\circ - \theta) + s \sin \theta, -r \sin \theta + s \sin(60^\circ + \theta) \right) \right) ds \quad (4.7)$$

Sampling this projection function produces the discrete sinogram $p(r_i, \theta_k)$.

4.3. CONCLUSIONS

The aim of this chapter is to derive the mathematical formulation of the Radon transform for the triangular grid. This transformation, that represents the equation of the projection, is used in the second phase of tomography, in the reconstruction of the hexagon-shaped images represented on the triangular grid.

In the following chapter it is considered that the obtained projections are discrete functions and each projection is the flux of the beam on the object. Considering as an input the projections obtained using the Radon transform, in the next chapter a mathematical analysis of the solution using the minimum cost maximum flow algorithm for solving the reconstruction problem on triangular grid from three projections is proposed.

5. MODELING THE RECONSTRUCTION PROBLEM ON TRIANGULAR GRID FROM THREE PROJECTIONS AS MINIMUM COST MAXIMUM FLOW PROBLEM

5.1. Introduction	49
5.2. Preliminaries.....	50
5.3. Three projection reconstruction.....	52
5.3.1. Parallel-parallel to coordinate axes projection	55
5.3.2. Orthogonal-orthogonal to coordinate axes projection	58
5.3.3. Parallel-orthogonal to coordinate axes projection.....	62
5.4. Conclusions	64

The aim of this chapter is to formulate the problem of reconstruction of binary hexagon-shaped images represented on the triangular grid from two projections and using a third projection as a restriction, by reducing the problem to the minimum cost maximum flow problem.

5.1. INTRODUCTION

The problem of reconstruction of binary images from few projections was studied by many authors. Most of the results use images that are defined on square grids and square lattices. The problem of binary image reconstruction can be modeled efficiently as a network flow problem. The first to describe the two-projection reconstruction problem in the context of flows in network was Gale, in 1957 [Gal57]. In his approach, he proposed a network flow algorithm for reconstructing matrices from their row and column sums. In 1956 Ford and Fulkerson [For56], in their seminar paper presented an algorithm for computing a maximum flow in a network, which can be used to solve the two-projection reconstruction problem. Several mathematical properties of the reconstruction problem on the square grid were derived using the network flow approach, by Anstee [Ans83]. The generalization of Gale's results to the case of square lattices from any pair of directions was presented in [Bat06b].

In this chapter we develop a minimum cost maximum flow approach for the triangular grid case starting from the generalization of the basic two-projection problem from the square grid. Since a triangular grid can be associated with a three direction coordinate system, the proposed approach for the minimum cost maximum flow algorithm tries to incorporate three projections. In this way the method uses two basic projections and a third projection as a restriction. Moreover, since the intersection of a direction with a lane can result in two points, as shown in section 3.5, in the next sections some special cases for modeling the reconstruction problem on triangular grid as minimum cost maximum flow problem are considered and described.

5.2. PRELIMINARIES

The input of the reconstruction is the discrete function

$$p(r, \theta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y, z) \delta\left(\frac{1}{\sin 60^\circ}((x-z)\sin\theta + (y-z)\sin(60^\circ - \theta)) - r\right) dx dy, \text{ with}$$

$(x, y, z) \in {}^*\mathbb{R}^3$, which means that $x+y+z=0$ or $x+y+z=1$, where θ is the angle that each projection is generated at. This angle belongs to the interval $[0, \pi)$.

Considering that the number of the generated projections is d , then $\Delta\theta = \frac{\pi}{d}$. Using

these notations, $p(r, \theta_k)$ represents the projection at angle θ_k which is equal to $k \frac{\pi}{d}$.

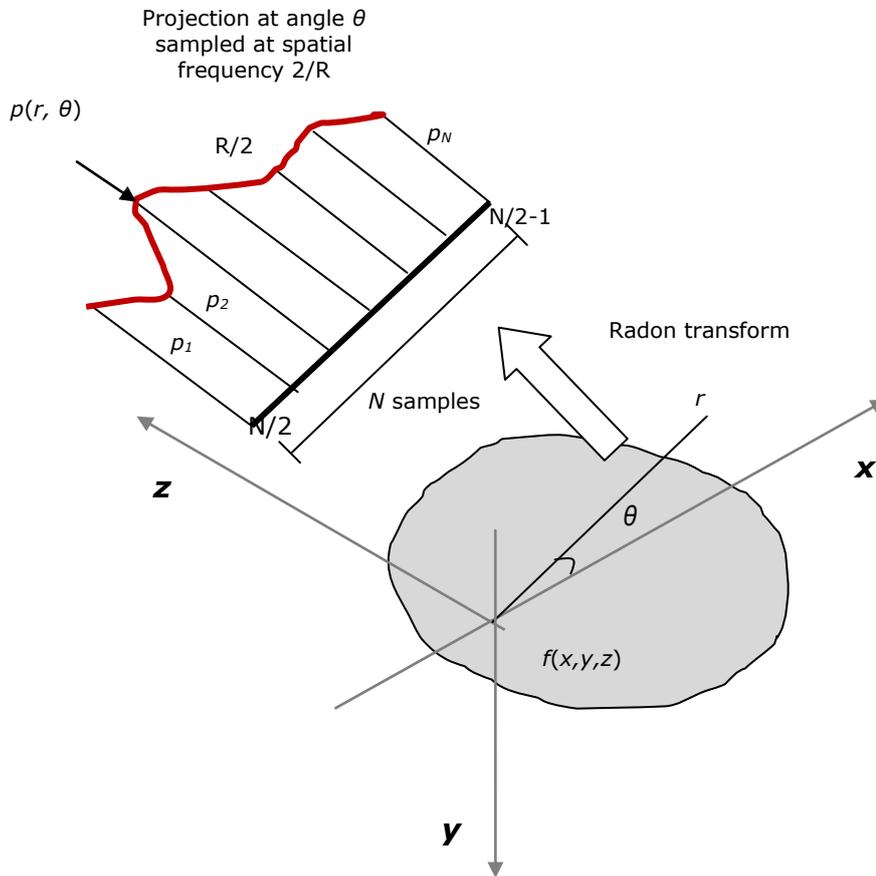


Figure 5.1. Radon transform representation

Let R be the smallest distance between 2 adjacent repeating cycles of intensities, and $B = \frac{1}{R}$ be the largest spatial frequency present in the projection data. The Nyquist sampling theory states that the frequency at which the projection is sampled is no less than $2B$. Using the sampling theory, the sampling frequency is $f_s=2B$, so that $p(r, \theta_k)$ is sampled at an interval of width $R/2$. Let N be the number of samples obtained from the sampling frequency and $t = -\frac{N}{2}, \dots, 0, 1, \dots, \frac{N}{2} - 1$. The result of sampling $p(r, \theta_k)$ is the sequence of numbers $p\left(\frac{t}{2B}, \theta_k\right)$, having different frequencies for different angle (see Figure 5.1).

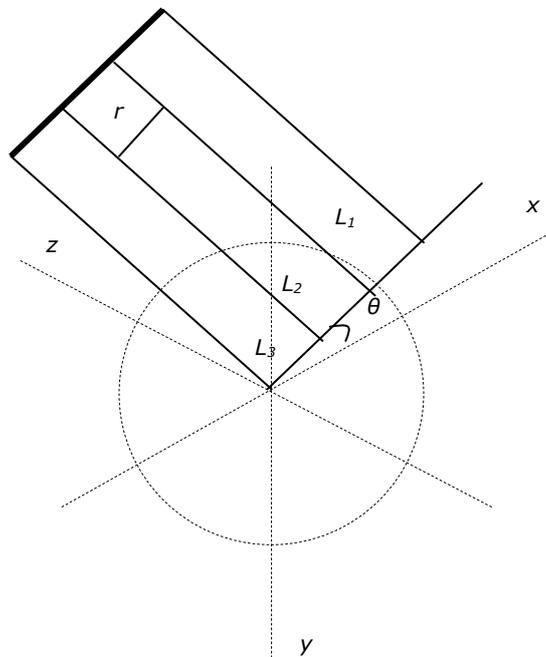


Figure 5.2. Parallel beam geometry which contains the projection lines.

The equation of a line which depends on r and θ is:

$$L(r, \theta) = \left\{ (x, y, z) \in \mathbb{R}^3 : \frac{1}{\sin 60^\circ} \left((x - z) \sin \theta + (y - z) \sin(60^\circ - \theta) \right) = r \right\} \quad (5.1)$$

This means that any point (x, y, z) on the line L satisfies the equation $\frac{1}{\sin 60^\circ} \left((x - z) \sin \theta + (y - z) \sin(60^\circ - \theta) \right) = r$.

Figure 5.2 shows the parallel beam geometry which contains the projection lines for a projection angle θ .

In this chapter, the problem of reconstruction of a hexagon-shaped image from its projections measured along three projection angles, θ_1 , θ_2 and θ_3 , is considered.

The unknown image is considered as an hexagon-shaped image $A = \{a_{i,j,k}\}$, of dimension $m \times m \times m$, where $i, k \in [-m+1, m]$, $j \in [-m, m-1]$ and $(i, j, k) \in {}^*\mathbb{R}^3$ with $i+j+k=0$ or $i+j+k=1$. Let \mathcal{L}_k , $k=1,2,3$ be the set of projection lines for projection angle θ_k and the finite set $L_k = \{L \in \mathcal{L}_k \mid L \cap A \neq \emptyset\}$. The elements of a line L_k are $L_{k,i}$, $i=1, \dots, |L_k|$. The image area represents the set of all points in ${}^*\mathbb{R}^3$ which are between $L_{k,1}$ and $L_{k,N}$, $N = |L_k|$.

The projections $P_k: L_k \rightarrow \mathbb{N}$ of image A for angle θ_k are defined as:

$$P_k(L) = |A \cap L| = \sum_{a \in L} f(a) \quad (5.2)$$

where f is the characteristic function of A .

5.3. THREE PROJECTION RECONSTRUCTION

The binary reconstruction problem from three projections on the triangular grid can be stated as follows:

Problem 5.1: Let θ_1 , θ_2 and θ_3 be three different projection angles and p_1 , p_2 and p_3 the measured projection data for the angles θ_1 , θ_2 and θ_3 . Construct a hexagon-shaped image A such that $P_k = p_k$, $k=1,2,3$.

The function $S: [0, 1]^{m \times m \times m} \rightarrow \mathbb{R}$ denoted by $S(A) = \sum_{i,j,k} a_{i,j,k}$ with $i, k \in$

$[-m+1, m]$, $j \in [-m, m-1]$ where $(i, j, k) \in {}^*\mathbb{R}^3$ is the total projection of the hexagon-shaped image.

When the measured projections are obtained from physical experiments, usually they contain errors. That is why, using three projections for the representation of the hexagon-shaped image it is possible to have zero images that correspond to the measured projections. In this case a solution, that has the projections as close as possible to the measured projections, for the reconstruction problem, is needed.

The reconstruction problem from few projections, in particular for our hexagon-shaped image from three projections is usually undetermined. Because it is important for the reconstruction to closely resemble the original image, prior knowledge has to be included in the reconstruction algorithm. The prior knowledge about the unknown image is incorporated by using a weight for each triangular pixel. By using the pixel weight, the reconstruction problem can be stated as:

Problem 5.2: Let θ_1, θ_2 and θ_3 be three different projection angles and p_1, p_2 and p_3 the measured projection data for the angles θ_1, θ_2 and θ_3 . Let $W:A \rightarrow \mathbb{R}$ be a given function. Construct a binary hexagon-shaped image A such that $P_k=p_k, k=1,2,3$ and the total weight $\sum_{a \in A} W(a)$ is maximal.

The basic idea used for the reconstruction of binary images represented on the square grid, from two projections as a minimum cost maximum flow algorithm was described by Gale in 1975 [Gal57]. In his approach, he proposed a network flow algorithm for reconstructing matrices from their row and column sums.

Unfortunately there is no generalization of the minimum cost maximum flow approach considering the case of three projections.. For the above reconstruction problem stated for the case of a hexagon-shaped image represented on the triangular grid, our new approach is based on the idea to reduce it to the reconstruction problem from two projections with restrictions imposed by the third projection. In this way, the obtained reconstruction has similar projections with two of the measured projections and also has some similar values with the third measured projection.

With the quadruplet $(A, \theta_1, \theta_2, \theta_3)$ a directed graph $G = (V, E)$ is associated, where V is the set of vertices and E represents the set of edges (see Figure 5.3). The set of vertices V contains a source vertex s , a sink vertex t , one vertex for each line of projection angle θ_1 and one vertex for each line of projection angle θ_2 . The vertex with label $v_{k,i}$ corresponds to $L_{k,i}$. Every pair of vertices $(v_{1,i}, v_{2,j})$, for which the vertices intersect each other in the image domain, is connected by an edge. This set of edges is denoted by E_p . Besides this set there are two other set of edges $E_s = \{(s, v_{1,i}) : i=1, \dots, |L_1|\}$ and $E_t = \{(v_{2,i}, t) : i=1, \dots, |L_2|\}$.

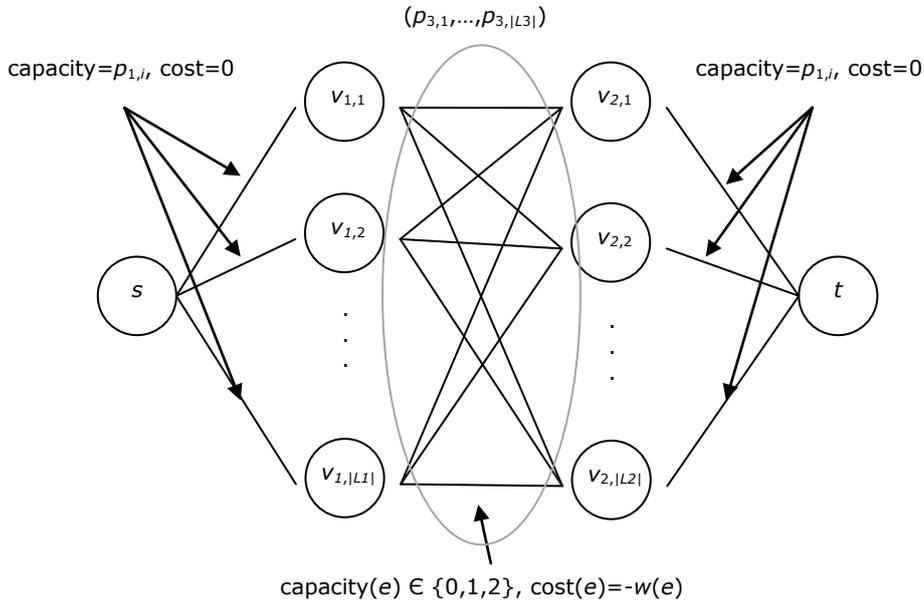


Figure 5.3. General representation of a graph G for the minimum cost maximum flow problem using three projection angles.

Each edge $(u,w) \in E$, $u, w \in V$, of G has associated a capacity $c(u,w) \geq 0$, a flow $h(u,w)$ and a cost $b(u,w)$. The third line of projection angle θ_3 is used in the computation of the capacity associated to each edge (u,w) . The value of the cost of each edge in E_p depends on the weight function. The cost of sending the flow through the network is $h(u,w) \cdot b(u,w)$. The proposed amount of flow to be sent from

$$s \text{ to } t \text{ is } \left(\sum_{k=1}^3 \sum_{i=1}^{|L_k|} p_{k,i} \right) / 3.$$

For using the associated graph G for solving the reconstruction problem, a capacity function $c : V \times V \rightarrow \mathbb{N}$ is defined for the vertices from the source and to the sink, as:

$$\begin{aligned} c(s, v_{1,i}) &= p_{1,i} \quad \text{for } i = 1, \dots, |L_1| \\ c(v_{2,j}, t) &= p_{2,j} \quad \text{for } j = 1, \dots, |L_2| \end{aligned} \quad (5.3)$$

The capacity of the edges between the layer of vertices $v_{1,i}$ and $v_{2,j}$ can have the values $\{0,1,2\}$, depending on the two projection lines orientation and on the values of the third projection measured for the projection angle θ_3 . The specific values of the capacity for some cases are detailed in the following sections.

A flow in a graph is a function $h : V \times V \rightarrow \mathbb{N}$ with the following properties:

Capacity constraints. For all u and $w \in V$, the flow along an edge (u,w) cannot exceed its capacity: $h(u,w) \leq c(u,w)$

Skew symmetry. For all u and $w \in V \setminus \{s,t\}$, the flow from u to w must be the opposite of the flow from w to u : $h(u,w) = -h(w,u)$

Flow conservation. For all $u \in V \setminus \{s,t\}$ the flow to a vertex is zero, except from the source and sink: $\sum_{w \in V} h(u,w) = 0$.

The flow conservation implies that for all vertex $u \in V \setminus \{s,t\}$

$$\sum_{v \in V} h(v,u) = \sum_{w \in V} h(u,w).$$

For solving the weighted version of the reconstruction problem, a cost function $b : V \times V \rightarrow \mathbb{N}$ is defined as:

$$b(u,w) = \begin{cases} 0 & \text{for } (u,w) \in E_s \cup E_t \\ -W(\Phi(u,w)) & \text{for } (u,w) \in E_p \end{cases} \quad (5.4)$$

where W is the weight function that associates to each pixel a weight value and Φ is a mapping which associates a pixel, or a set of two pixels, to each edge.

Considering the upper constraints, the requirement of the problem is to minimize the total cost of the flow:

$$\sum_{u,w \in V} b(u,w) \cdot h(u,w) \quad (5.5)$$

In the construction of the associated graph G , the features of an image represented on the triangular grid are taken into consideration. Using as support the triangular grid, the resulting bipartite graph is not complete, because not all the pairs of vertices are connected one to each other. The connection between two

vertices, $v_{1,i}$ and $v_{2,j}$ depends on the intersection of the vertices in the image domain. The Pseudocode sequence 5.1 describes the basic steps in the construction of the minimum cost maximum flow graph which is associated with the quadruplet $(A, \theta_1, \theta_2, \theta_3)$. In the graph, the cost function is computed using the values from the pixels weight map.

Pseudocode sequence 5.1. Algorithm for computing graph G .

Input: A compatible set of projections $p_k, k=1,2,3$

Step 1: Compute the vertices: 1 source vertex s , 1 sink vertex t , vertices $v_{1,i}$ and $v_{2,j}$, $i, j = 1, \dots, |L_k|$

Step 2: Compute the possible edges between the layers $v_{1,i}$ and $v_{2,j}$ based on the features of the hexagon-shaped image.

$$k_0 = -i - j + 2m - 1; k_1 = -i - j + 2m;$$

If $k_0 \in [-m+1, m]$ or $k_1 \in [-m+1, m]$ set the edge $(v_{1,i}, v_{2,j})$

Step 3: For each edge compute the capacity

Step 4: For each edge compute the cost

Step 5: Output graph G

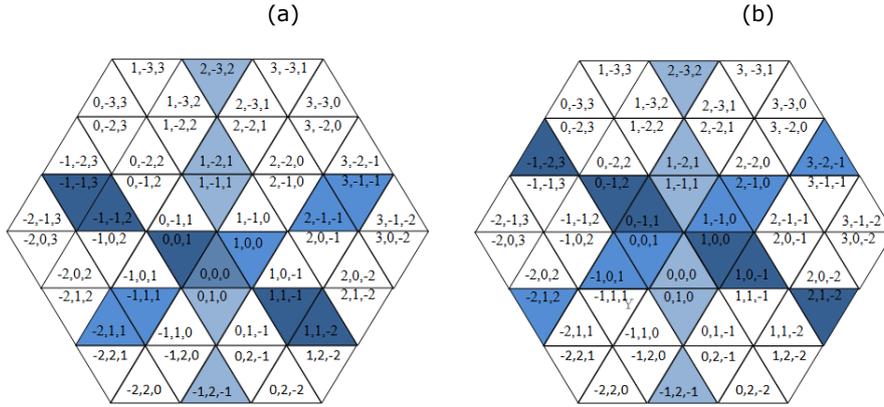
Considering the coordinate axes for a triangular grid, six natural directions are possible, three directions orthogonal and three directions parallel to the coordinate axes. These six directions correspond to the six projection angles $\theta \in \left\{0, \frac{\pi}{6}, \frac{\pi}{3}, \frac{\pi}{2}, \frac{2\pi}{3}, \frac{5\pi}{6}\right\}$. In the next sections we describe the properties of the associated graph, considering the possible combination of two main directions: two parallel directions (parallel-parallel), two orthogonal directions (orthogonal-orthogonal) and one orthogonal and one parallel direction to the coordinates axes projection, the third direction being chosen in a random way from the unconsidered projection angles.

5.3.1. PARALLEL-PARALLEL TO COORDINATE AXES PROJECTION

Considering the directions parallel to the coordinate axes it can be seen that they contain points for which the difference of the two coordinate values is the same. In this case the direction is called diamond-chain; for example, the light blue color diamond-chain on Figure 5.4 is formed by the set $\{(i, j, k) | i - k = 0\}$.

Using for the two considered directions the directions parallel to the coordinated axes, two cases are possible, one in which the intersection of the two diamond-chains is one pixel (see Figure 5.4.a), and one in which the diamond-chains do not intersect each other (see Figure 5.4.b). Mathematically, the three equations of the coordinate value are not independent of each other defining a subset of triplets, where the difference of two coordinates has a constant value. For example, considering a set $\{(u, u+l, u+q) | u, l, q \in \mathbb{Z}\}$ where the first coordinate value u is given, the other two coordinate values depend on the first one. However the sum of the values in these triplets can be divisible by 3, can have a remainder of 1 by dividing with 3, being 1 mod 3, or can have a remainder of 2 by dividing of 3,

being 2 mod 3. In the first case, when the sum of the values in these triplets is divisible by 3, the intersection of the two diamond-chains consists of an even pixel (having sum 0) (0,0,0) by choosing $u = 0$ as presented in Figure 5.4.a. In the second case, when the sum of the values in these triplets has a remainder of 1 by dividing with 3, being 1 mod 3, the intersection is an odd pixel (having sum 1); for example $(u, u + 2, u - 4)$ by fixing $u = 1$ gives the point (1, 3, -3). In the third case, when the sum of the values in the triplet has a remainder of 2 by dividing with 3, being 2 mod 3, there is no intersection pixel; for example Figure 5.4.b: $(u, u-1, u)$.



For the generation of the three projections parallel to the coordinate axes, the following three angles are considered $\theta_1 = 0$, $\theta_2 = \frac{\pi}{3}$ and $\theta_3 = \frac{2\pi}{3}$. In this case there are $N=4m-1$ projection lines for an angle θ , where m is the side length of the hexagon-shaped image (see Figure 5.5).

Considering a hexagon-shaped image $A=\{a_{i,j,k}\}$, $i, k \in [-m+1, m]$, $j \in [-m, m-1]$ where $(i, j, k) \in \mathbb{R}^3$, its projections $P_1, P_2, P_3: [0,1]^{N \times N \times N} \rightarrow \mathbb{R}^N$ for the angles θ_1, θ_2 and θ_3 are computed by summation of all entries in a x, y and z parallel to the axes direction of A . Figure 5.5 shows an example of a hexagon-shaped image with its projection values computed along the three diamond-chains.

Using the associated graph G for solving the reconstruction problem a capacity is associated to each edge from E_p as follows:

$$c(v_{1,j}, v_{2,j}) \in \{0, 1\} \quad \text{for } i, j = 1, \dots, N \quad (5.6)$$

The capacity values over the edges $(v_{1,i}, v_{2,j})$ depend on the values of the third projection measured for the projection angle θ_3 . So, the value of the capacity is zero if the value from the third projection is zero. In all the other cases, the edge has a capacity value of 1.

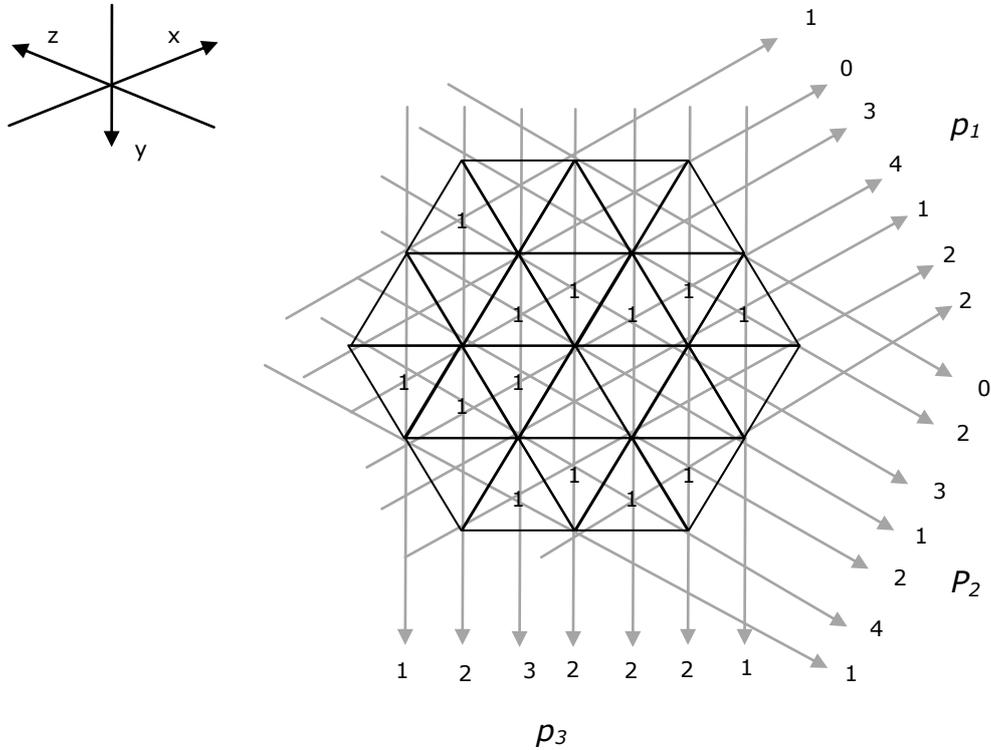


Figure 5.5. A $2 \times 2 \times 2$ hexagon-shaped image with its projections corresponding to the projection angles $\theta_1 = 0$, $\theta_2 = \frac{\pi}{3}$, $\theta_3 = \frac{2\pi}{3}$.

For a given flow the size of the flow is $|h| = \sum_{s, v_{1,i} \in V} h(s, v_{1,i})$. Using the flow conservation constraint the size of the flow is also $|h| = \sum_{v_{2,j}, t} h(v_{2,j}, t)$. In this case, in graph G all flow that leaves the source must pass through the edges $v_{1,i}$ and $v_{2,j}$, which implies that the size of the flow is $|h| = \sum_{v_{1,i}, v_{2,j} \in V} h(v_{1,i}, v_{2,j})$. Using this property, the flow in G has the values $h(v_{1,i}, v_{2,j}) \in \{0, 1\}$ for any edge $(v_{1,i}, v_{2,j}) \in E_p$, as the capacity is also $\{0, 1\}$.

The value of the cost for any edge $(v_{1,i}, v_{2,j}) \in E_p$ is given by:

$$b(v_{1,i}, v_{2,j}) = -W(L_{1,i} \cap L_{2,j}) \quad \text{for } i, j = 1, \dots, N \quad (5.7)$$

where $L_{1,i}$ and $L_{2,j}$ are the lines from the two diamond-chains associated to the projection angles θ_1 and θ_2 . The intersection of the two diamond-chains, if exists, is one pixel.

Figure 5.6 shows the construction of a graph associated to the hexagon-shaped image from Figure 5.5 using the measured projection at projection angles corresponding to the parallel directions to the coordinate axes. The lines between the two layers of vertices $v_{1,i}$ and $v_{2,j}$ represent flow 1.

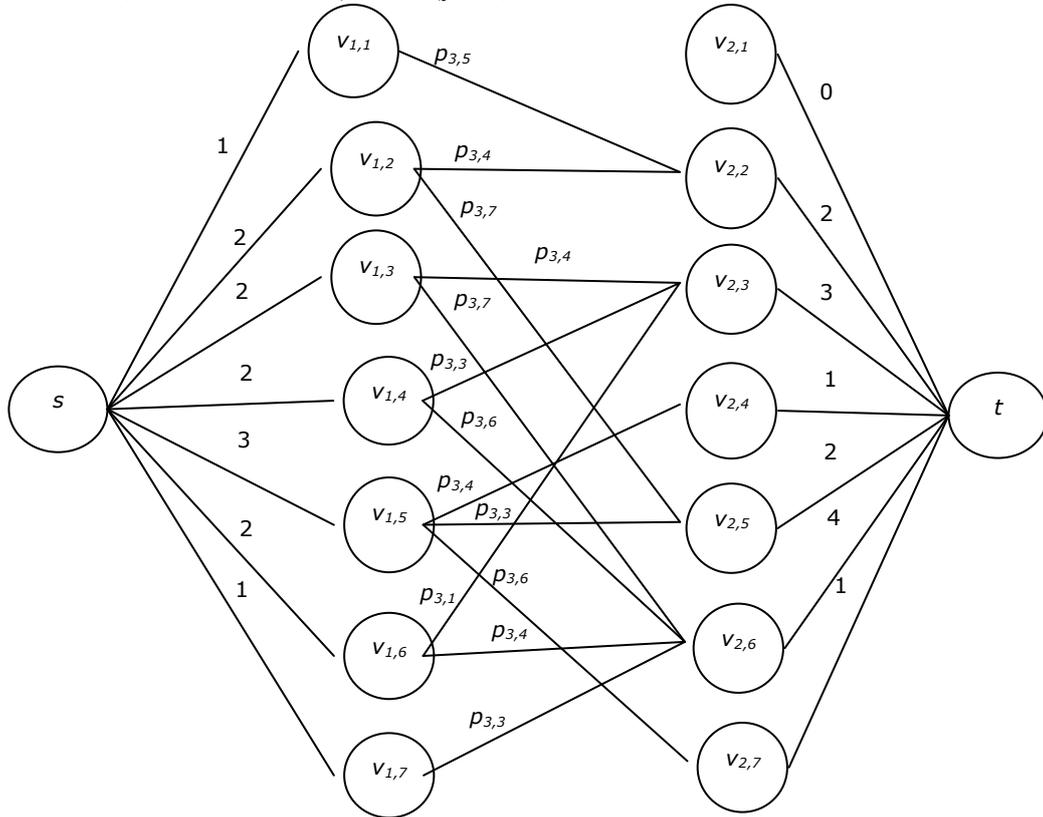


Figure 5.6. The corresponding graph associated to the hexagon-shaped image from Figure 5.5. The lines represent flow 1.

5.3.2. ORTHOGONAL-ORTHOGONAL TO COORDINATE AXES PROJECTION

Considering, for the chosen set of three projections the directions orthogonal to the coordinate axes all the points on such a direction have the same value for a corresponding coordinate. These points form a line which is called lane.

Two lanes that are not parallel have exactly two common triangles. So besides the two fixed coordinate values, they have the third coordinate value such that the sum of the three values become 0 and 1: i.e. one even pixel and one odd pixel, having two identical coordinates (the ones used for the network problem) and the third differs with one unit; e.g. considering directions x and y there are two points a_{i,j,k_0} and a_{i,j,k_1} where $|k_0 - k_1| = 1$ (see Figure 5.7).

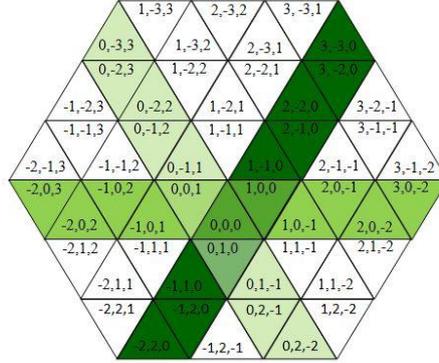


Figure 5.7. Examples for orthogonal projection to coordinate axes

In the case of orthogonal direction to the coordinate axes there are $N=2m$ projection lines for an angle θ , where m is the side length of the hexagon-shaped image. We developed an approach for solving the reconstruction problem for hexagon-shaped images on triangular grid using a maximum flow minimum cost graph representation model for three directions orthogonal to the coordinate axes in [Moi13a].

Considering a hexagon-shaped image $A=\{a_{i,j,k}\}$, $i, k \in [-m+1, m]$, $j \in [-m, m-1]$ where $(i, j, k) \in \mathbb{R}^3$, its projections for the $\theta_1 = \frac{\pi}{6}$, $\theta_2 = \frac{\pi}{2}$ and $\theta_3 = \frac{5\pi}{6}$, are computed by summation of all entries in a x , y and z orthogonal to the axes direction of A as presented in [Moi11]. In this way $P_1, P_2, P_3: [0,1]^{N \times N \times N} \rightarrow \mathbb{R}^N$, the reconstructed image projections, are defined as:

$$\left\{ \begin{array}{l} \sum_{\substack{j \in [-m-i, \dots, m-1] \\ k \in [-m+1-i, \dots, m]}} a_{i,j,k} = P_{1,i} \quad , \text{for } i = -m+1, \dots, 0 \\ \sum_{\substack{j \in [-m, \dots, m-i] \\ k \in [-m+1, \dots, m+1-i]}} a_{i,j,k} = P_{1,i} \quad , \text{for } i = 1, \dots, m \end{array} \right. \quad (5.8)$$

$$\left\{ \begin{array}{l} \sum_{\substack{i \in [-m-j, \dots, m] \\ k \in [-m-j, \dots, m]}} a_{i,j,k} = P_{2,j} \quad , \text{for } j = -m, \dots, -1 \\ \sum_{\substack{i \in [-m+1, \dots, m-j] \\ k \in [-m+1, \dots, m-j]}} a_{i,j,k} = P_{2,j} \quad , \text{for } j = 0, \dots, m-1 \end{array} \right. \quad (5.9)$$

$$\begin{cases} \sum_{\substack{i \in [-m+1-k, \dots, m] \\ j \in [-m-k, \dots, m-1]}} a_{i,j,k} = P_{3,k} & , \text{for } k = -m+1, \dots, 0 \\ \sum_{\substack{i \in [-m+1, \dots, m+1-k] \\ j \in [-m, \dots, m-k]}} a_{i,j,k} = P_{3,k} & , \text{for } k = 1, \dots, m \end{cases} \quad (5.10)$$

Assuming that the projection values are consistent, this implies that $\sum_{i \in [-m+1, m]} P_{1,i} = \sum_{j \in [-m, m+1]} P_{2,j} = \sum_{k \in [-m+1, m]} P_{3,k}$. Figure 5.8 shows an example of image with its corresponding orthogonal projections.

A representation of a graph $G = (V, E)$ and its properties for the projections orthogonal to the axes are proposed in [Moi13a]. Considering the graph $G = (V, E)$, the capacity value of each edge $(v_{1,i}, v_{2,j}) \in \{0, 1, 2\}$. The capacity of an edge $(v_{1,i}, v_{2,j})$ is 0, 1 or 2 because on the triangular grid the projection orthogonal to the coordinate axes for a specified direction contains both even and odd pixels (see Figure 5.7) and it depends on the third projection of projection angle θ_3 . Knowing that each possible edge corresponds either to one pixel or to two pixels for the hexagon-shape image A , the capacity of the edge is set to:

- 0 – if the intersection between the projection line at angle θ_1 and the projection line at angle θ_2 correspond two pixel, and the two values for the two possible measured projection for third projection angle θ_3 are both zero;
- 1 – if the edge corresponds only to one pixel, which implies two possible cases:
 - (1) the first one is when the intersection between the projection line at angle θ_1 and the projection line at angle θ_2 is one pixel and the value for the measured projection for third projection angle θ_3 is not zero (only one triangle shape is element of the image for a given pair of two coordinates, see Figure 5.7, e.g. pixel (3,0,-2));
 - (2) the second case is when the intersection between the projection line at angle θ_1 and the projection line at angle θ_2 are two pixel, but one of the two values for the two possible measured projection for third projection angle θ_3 is zero (both triangle shapes are elements of the image for a given pair of two coordinates and the projection corresponding for one of the third coordinate is zero);
- 2 – if the intersection between the projection line at angle θ_1 and the projection line at angle θ_2 correspond two pixel, and the two values for the two possible measured projection for third projection angle θ_3 are not zero (see Figure 5.7);

The flow and the cost functions are defined for a hexagon-shaped image, considering as the two main directions the directions orthogonal to x and y coordinate axes, and the third direction the one orthogonal to z coordinate axis. For the other two cases, the functions can be defined in a similar way.

For a hexagon-shaped image A , in the case of orthogonal directions to the coordinate axes projections, the corresponding flow is defined by:

$$h(v_{1,i}, v_{2,j}) = a_{i-m, j-m+1, k_0} + a_{i-m, j-m+1, k_1} \quad (5.11)$$

where $i, j=1, \dots, N$ and $k_0=-i-j+2m-1, k_1=-i-j+2m$ and $i-m, k_0, k_1 \in [-m+1, m], j-m+1 \in [-m, m+1]$.

Figure 5.8 shows an example of a $2 \times 2 \times 2$ regular hexagon-shaped image with its projections corresponding to the projection angles: $\theta_1 = \frac{\pi}{6}$, $\theta_2 = \frac{\pi}{2}$, $\theta_3 = \frac{5\pi}{6}$, corresponding to the three lanes.

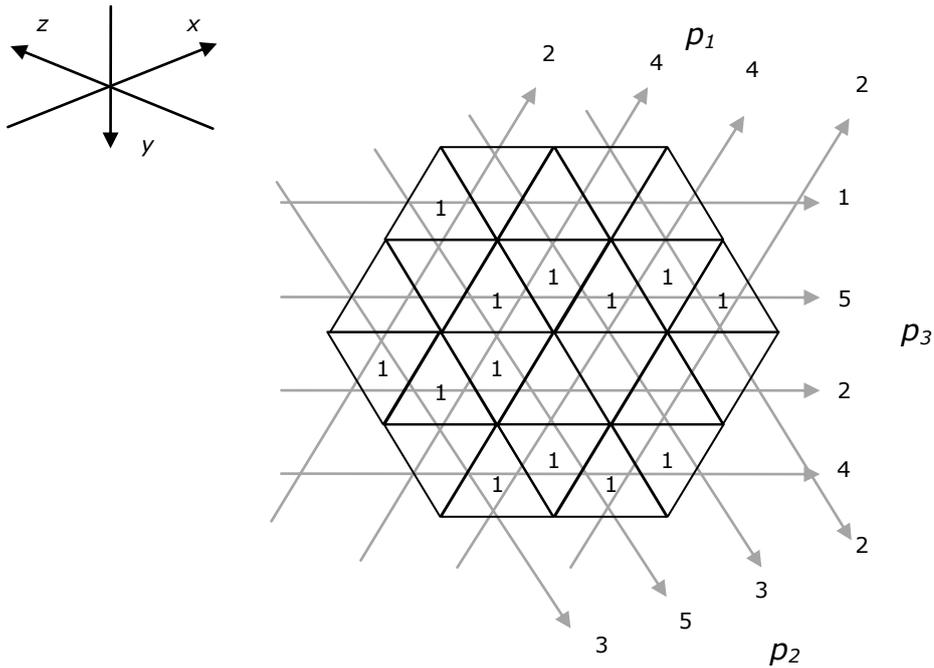


Figure 5.8. A $2 \times 2 \times 2$ hexagon-shaped image with its projections corresponding to the projection angles: $\theta_1 = \frac{\pi}{6}$, $\theta_2 = \frac{\pi}{2}$, $\theta_3 = \frac{5\pi}{6}$.

To each edge $(u, v) \in E_p$ a cost function is associated. This function determines the cost of sending a certain amount of flow through the edge. The cost depending on the weight map W and is defined as:

$$b(v_{1,i}, v_{2,j}) = -(w_{i-m, j-m+1, k_0} + w_{i-m, j-m+1, k_1}) \quad (5.12)$$

where $i, j=1, \dots, N$ and $k_0=-i-j+2m-1, k_1=-i-j+2m$ and $i-m, k_0, k_1 \in [-m+1, m], j-m+1 \in [-m, m+1]$.

The cost is computed using the weight function taking into consideration the fact that for the intersection of the projection line at angle θ_1 and the projection line at angle θ_2 consists of two pixels that lay on two different projection lines associated to the third projection angle θ_3 .

Figure 5.9 shows the associated graph for a hexagon-shaped image using the measured projection at projection angle $\theta_1 = \frac{\pi}{2}$, $\theta_2 = \frac{5\pi}{6}$, $\theta_3 = \frac{\pi}{6}$ which correspond to the orthogonal direction to the coordinate axes projections. The black lines between the two layers of vertices $v_{1,i}$ and $v_{2,j}$ represent flow 1 and dashed lines between the two layers of vertices $v_{1,i}$ and $v_{2,j}$ represent flow 2.

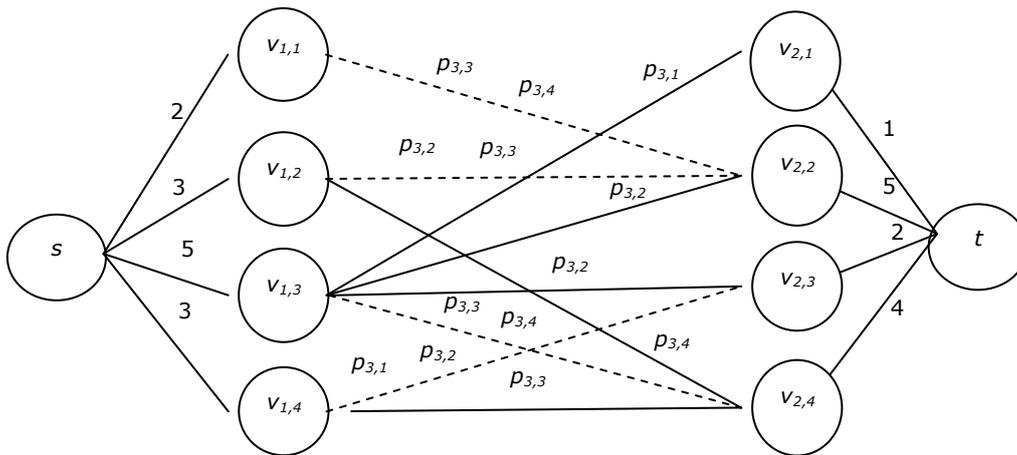


Figure 5.9. The corresponding graph associated to the hexagon-shaped image from Figure 5.8. The black lines represent flow 1 and dashed lines represent flow 2.

5.3.3. PARALLEL-ORTHOGONAL TO COORDINATE AXES PROJECTION

In the case of considering the main two projection directions as one orthogonal and one parallel to the coordinate axes, two possible cases exist. Here a lane and a diamond-chain intersect each other. In this case, for the chosen pair of angles there are $N_1=2m$ projection lines for an angle θ_1 and $N_2=4m-1$ projection lines for an angle θ_2 , where m is the side length of the hexagon-shaped image.

The first possible case is when the lane is orthogonal and the diamond chain is parallel to the same coordinate axis (see Figure 5.10.a). In this case there is one intersection point. By fixing a coordinate value, and the difference of the other two values, there is only one way to obtain points with coordinate triplets with 0 or 1 sum, there is no way to obtain both of them.

For the generation of the associated graph G , one of the following pair of angles $\theta_1 = 0, \theta_2 = \frac{\pi}{2}$ or $\theta_1 = \frac{\pi}{6}, \theta_2 = \frac{2\pi}{3}$ or $\theta_1 = \frac{\pi}{3}, \theta_2 = \frac{5\pi}{6}$ is considered as a first step, meanwhile the third angle θ_3 is randomly generated from the remaining angles. The associated capacity, flow and cost values respect the properties presented in section 5.3.1.

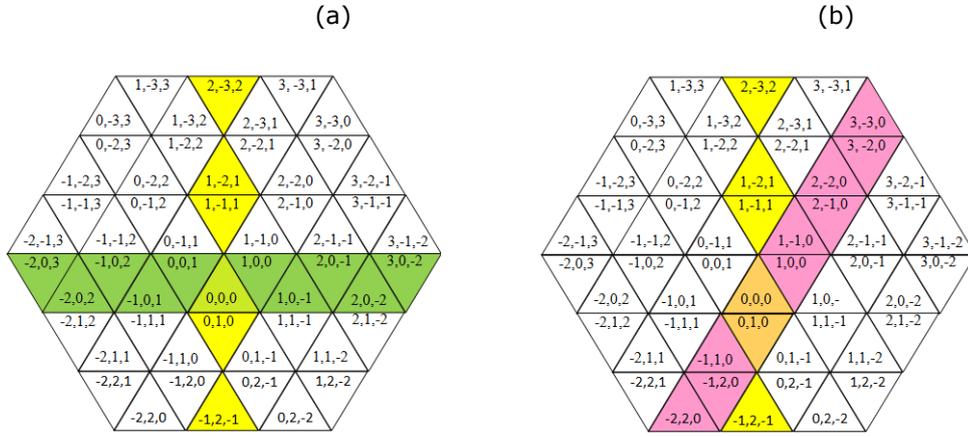


Figure 5.10: Two direction combinations (a) parallel-orthogonal same axes direction;
(b) parallel-orthogonal different axes direction;

The second case is when the lane is orthogonal to another axis than the one parallel with the diamond-chain (see Figure 5.10.b), when there are two intersection points, one even and one odd point, that differs by the parallel direction axes coordinate with one unit. Here there is a fixed coordinate, and the difference of that and another coordinate value is fixed: therefore having two fixed coordinate values, the third coordinate has two possible values so that their sum to be 0 or 1. For example, if the case of y parallel direction and z orthogonal direction is considered (see Figure 5.10.b.) there are two intersection points $a_{i,j,k}$ and $a_{i,j+1,k}$. Considering an hexagon-shape image, the associated graph G , is generated using one of the pairs of angles $\theta_1 = 0$, $\theta_2 = \frac{2\pi}{3}$ or $\theta_2 = \frac{5\pi}{6}$, or $\theta_1 = \frac{\pi}{6}$, $\theta_2 = \frac{\pi}{2}$ or $\theta_2 = \frac{5\pi}{6}$, or $\theta_1 = \frac{\pi}{3}$, $\theta_2 = \frac{\pi}{2}$ or $\theta_2 = \frac{2\pi}{3}$ meanwhile the third angle θ_3 is randomly generated from the remaining angles.

The associated capacity for the edges $(v_{1,i}, v_{2,j})$ has the values $\{0, 1, 2\}$. For a detailed description see section 5.3.2.

For a hexagon-shape image A , the flow in a graph corresponds to:

$$h(v_{1,i}, v_{2,j}) = A(L_{1,i} \cap L_{2,j}) \quad (5.13)$$

where $L_{1,i}$ is the lane associated to the projection angle θ_1 and $L_{2,j}$ is the diamond-chain associated to the projection angle θ_2 . The intersection of the lane and diamond-chain is contains two pixels from the image A .

The value for the cost in a graph is given by the weight function W values for the two pixels obtained through the intersection of a lane L_1 and diamond-chain L_2 :

$$b(v_{1,i}, v_{2,j}) = -W(L_{1,i} \cap L_{2,j}) \quad (5.14).$$

5.4. CONCLUSIONS

In this chapter we modeled the reconstruction problem of hexagon-shape images on triangular grid using three projections using the problem of finding a maximum flow with minimum cost in a graph. For the case of triangular grid, the proposed minimum cost maximum flow approach, starts from the algorithm using two projections and incorporates a restriction generated from a third projection. Considering the natural directions of the triangular grid, six projection angles can be considered for the orthogonal and parallel to the coordinate axes directions. Using these six projection angles three main cases are possible in the generation of the associated graph. In the previous sections these three cases are described.

Based on the proposed minimum cost maximum flow algorithm for the triangular grid, in the next chapters we propose two algorithms for solving the binary tomography reconstruction problem on triangular grid from the measured projections along few projection angles. Usually, the reconstruction problem from few projections is undetermined; the number of solutions can be very large. In order to solve this problem, we develop a memetic algorithm and an iterative approach for finding the reconstruction which minimizes an evaluation function. The evaluation function measures the quality of the reconstruction. The memetic algorithm uses the minimum cost maximum flow algorithm in the generation of the initial population. In the case of the iterative algorithm, in every iteration the minimum cost maximum flow algorithm is computed for a different triplet of projection angles.

6. MEMETIC ALGORITHM FOR IMAGE RECONSTRUCTION IN DISCRETE TOMOGRAPHY ON THE TRIANGULAR GRID FROM SIX PROJECTIONS

6.1. Introduction	65
6.2. Preliminaries.....	66
6.3. Algorithm description.....	67
6.3.1. Initial population	69
6.3.2. Fitness function.....	69
6.3.3. Crossover operator	69
6.3.4. Guided mutation operator.....	71
6.3.5. Compactness operator	72
6.3.6. Minimal hill climb operator.....	73
6.3.6.1. Rhombus switching component.....	73
6.3.6.2. Hourglass switching component.....	74
6.3.6.3. Rotation switching component	76
6.4. Experimental results.....	78
6.4.1. Hv-convex polyominoes	79
6.4.2. Generic regular hexagon-shaped images	83
6.4.3. Non-convex regular hexagon-shaped images	87
6.5. Performance analysis	97
6.6. Conclusions	101

One of the main problems in binary tomography on triangular grid is the reconstruction of a hexagon-shape image from acquired projection along few different projection angles. In this chapter we develop an original memetic algorithm, for reconstruction of binary hexagon-shape images from only six projections. The quality of the reconstruction is measured by the evaluation function. The proposed algorithm finds the reconstruction that minimizes an evaluation function. The algorithm generates an initial population using the minimum cost maximum flow algorithm. The reconstructed images evolve towards an optimal solution or close to the optimal solution, by using new crossover operators and guided mutation operators. The quality of the images is improved by using a new local search operators based on switching components and compactness. The benefits of the proposed algorithm were tested regarding the correctness, robustness and time of the reconstruction by generating hexagon-shaped images from public available datasets of images.

6.1. INTRODUCTION

Binary tomography is one special case of discrete tomography, which aim is to reconstruct a binary image using projections from few different angles. These

problems are usually hard problems, meaning that the solution of the problem is usually undetermined; there are several solutions for the same problem. Also, if more than three projections are considered, the reconstruction problem is NP-hard. In order to handle this problem, optimization algorithms, such as evolutionary algorithms can be used. Evolutionary algorithms are generic population-based metaheuristic optimization algorithms. In particular, in this chapter a memetic algorithm is proposed by the author, for resolving the binary tomography reconstruction problem on triangular grid for hexagon-shaped images. The proposed algorithm is somewhat analogous to the method given by Di Gesu in [DiG08]. Memetic algorithm represents a method bases on the concept of having a population of possible solutions that evolve towards a possible solution using genetic operators, crossover, mutation selections and local-search operators.

In the following sections, we define our mutation and crossover operators for the case of hexagon-shaped images. Also, we propose the so-called switching components and a compactness operator, which are used in the local search operator for improving the quality of the solutions. A switching component consists of a set of points such that inverting their values the projection data do not change. A compactness operator eliminates some of the isolated points of an image. It also changes some of the projection values. In the case of the triangular grid data from six projection angles for the reconstruction of images are used. The used directions are the natural directions of the triangular grid, based on the symmetry of the grid and thus, they can easily be described by the symmetric coordinate frame. Three natural directions are orthogonal to the coordinate axes, using projections that gather the information from pixels sharing a coordinate value (projections by lanes) and the other three directions are parallel to the symmetry axes (projections by diamond-chains). We developed specific operators using the orthogonal directions to the coordinate axes in [Moi11][Moi13b]. Also, for the case of parallel to the coordinate axes directions we developed specific operators and presented them in [Nag14]. Using the developed operators for the orthogonal and parallel to the coordinate axes, in this chapter we propose a memetic algorithm that uses all the six natural projection directions of the triangular grid.

6.2. PRELIMINARIES

This section introduces some notations and defines aspects regarding the discrete tomography problem for hexagon-shaped binary images represented on triangular grid. The introduction of the discrete tomography problem on the triangular grid is based on the well known theoretical results [Her99][Her07].

A hexagon-shaped binary image represented on the triangular grid is considered to be a set of size $m \times m \times m$, as presented in Section 3.3. The number of pixels in such a hexagonal set is $6m^2$. The pixels of an image have value 0, if the pixel is black which means that it belongs to the background and value 1, if the pixel is white it means that it belongs to the foreground. As the triangular grid is used for the representation of the images, there are two types of pixels even (\blacktriangle) or odd (\blacktriangledown). The two orientations of the pixels are also called parities of the triangles [Nag07]. Since the grid is a 2D one, the coordinate values are not independent of each other, but each triangle of the shape \blacktriangle is addressed with a triplet having sum 0, and each triangle of the shape \blacktriangledown is addressed with a triplet with sum 1. In the triangular grid, the directions orthogonal to the coordinate axes are called lanes and

the directions parallel to the coordinate axes are called diamond-chains. A lane is a set of points that share a coordinate value.

Given an angle θ and a direction r , the projection at the given angle is

$$p(r, \theta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y, z) \delta\left(\frac{1}{\sin 60^\circ}((x - z) \sin \theta + (y - z) \sin(60^\circ - \theta)) - r\right) dx dy$$

where $(x, y, z) \in \mathbb{R}^3$, which means that $x+y+z=0$ or $x+y+z=1$.

A projection of the triangular grid in a direction is a function that gives the number of points on each line which is parallel to that direction. Considering L_k , $k=1, \dots, d$ as the number of samples obtained from a sampling frequency, the vectors p_1, \dots, p_d of nonnegative values measure the projection at angles $\theta_1, \dots, \theta_d$ where d is the number of generated projections.

An example of image with its projections taken along the directions orthogonal to the coordinate axes is presented in Figure 6.1.

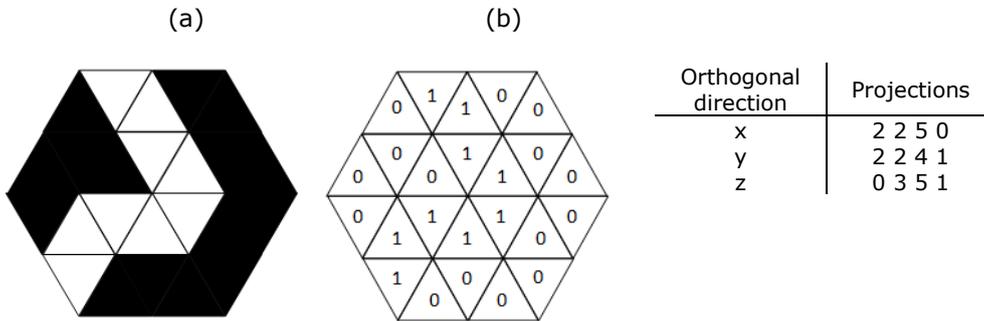


Figure 6.1. A hexagon-shaped binary image (a) and its representation (b), with its projections measured along the orthogonal to the axes directions.

6.3. ALGORITHM DESCRIPTION

In the next sections the reconstruction problem considering the case of triangular grid representation of images, is formulated as a memetic algorithm. The proposed algorithm tries to reconstruct hexagon-shaped binary images that satisfy six projections, by creating an initial population generated using the minimum cost maximum flow algorithm. The reconstructed solution is obtained by using memetic algorithm operators designed for the case of hexagon-shaped binary images. In the following sections it is supposed that the reader is familiar with the concepts of a memetic algorithm. A description of it can be found in [Mos89] [Mos10].

Pseudocode sequence 6.1. Basic steps of the memetic algorithm. Pseudocode sequence 6.1 shows the basic steps of the method proposed by the author in this thesis.

Pseudocode sequence 6.1. Basic steps of the memetic algorithm.

Input: Projections measured along the considered projections angles

Steps:

Create initial population using the minimum cost maximum flow algorithm for triangular grid

While (solution is not found or number of generations is not met)

 While (number of children are not created)

 Generate children by lane and diamond-chain directions crossovers

 End While

Select new population from the union of the current population and the generated children

 For (all images in the population)

 Apply a random guided mutation operator

 If (fitness improved) then

 Replace image with its mutated version

 End If

 End For

 For (all images in the population)

 Apply compactness operator

 If (fitness improved) then

 Replace image with its compacted version

 End If

 End For

 For (all images in the population)

 Apply minimal hill climb operator based on rhombus switching

 Apply minimal hill climb operator based on hourglass switching

 Apply minimal hill climb operator based on rotation switching

 End For

End While

Output: Reconstructed image

As an input of the memetic algorithm the measured projections along the projections angles are considered. The first step is the initialization of the initial population of individuals by using the minimum cost maximum flow algorithm (see Section 6.3.1). The stopping criteria of the main loop depend on the best found solution and on the number of generations. In the computation of the next generation, the crossover operator is applied using a random number of parents. The crossover operator is presented in Section 6.3.3. Subsequently, the guided mutation operator, described in Section 6.3.4, is applied. After applying the guided mutation, the compactness operator, described in Section 6.3.5, is used to improve the obtained individuals. To further improve the quality of the individuals a minimal hill climb operator based on the switching operators, proposed by the author in Section 6.3.6, are used. The order of the operators in the proposed memetic algorithm was experimentally established.

The next sections describe in detail the basic steps of the proposed algorithm. The fitness function and the new operators are defined. Also the generation of the initial population is presented.

6.3.1. INITIAL POPULATION

As we have discussed in Chapter 5, a hexagon-shaped image A , to which three direction projections are associated, can be represented in a convenient way by using a graph G associated to a minimum cost maximum flow problem. This graph has a source s , a sink t and two layer of vertices between s and t , one layer of vertices $v_{1,i}$ for the first direction projection and one layer of vertices $v_{2,j}$ for the second direction projection. To each edge of G a capacity, a flow and a cost are associated. The value of the capacity depends on the values of the measured projections for the third direction. The obtained maximal flow corresponds to an image that satisfies the projections along the two main directions and some of the projections from the third direction.

To compute the maximal flow of the graph the minimum cost maximum flow algorithm for the case of triangular grid can be used as shown in Chapter 5. The diversity in the initial population is guaranteed by the use of a binary randomly generated weight map which values are used in the computation of the cost.

6.3.2. FITNESS FUNCTION

The computation of the fitness function does not use any prior information about the image to reconstruct, only its projections are used; the following formulation [DiG08] of the fitness function on all considered directions d is used:

$$\mathfrak{F}(A) = \sum_{k=1}^d I_1(P_k, p_k) \quad (6.1)$$

$$I_1(P_k, p_k) = \sum_{i=1}^{|L_k|} |p_{k,i} - P_{k,i}| \quad (6.2)$$

where p_k is the measured projection along the projection angle θ_k and $p_k = (p_{k,1}, \dots, p_{k,|L_k|})$ is the projection of the reconstructed image A calculated along the projection angle θ_k . The value I_1 is the distance of two same-dimensional vectors and is defined as the (norm of the) difference of the projections P_k of the reconstructed image and the measured projections p_k along the projection angle θ_k . The goal of the algorithm is to minimize the function defined in equation (6.1). Actually, an error function is used to measure the fitness, there for the aim is to achieve images with 0 (or with very small) "fitness" value.

6.3.3. Crossover OPERATOR

The crossover operator is applied on all the six natural directions to the axes. The crossover can be applied to the orthogonal and parallel direction to any of

the axes x , y and z . In the following paragraph the crossover applied on the direction parallel to y axis is defined. The crossover operators for the other directions can be defined in a similar way.

Given two parents A_1, A_2 , their $y_{||}$ offspring C_1, C_2 are obtained by swapping the diamond-chains parallel to axis $y, A_1^{(j)}$ and $A_2^{(j)}$, where j is the position of 1's in the random binary mask $M=(M_1, \dots, M_N), N=4m-1$ (see Figure 6.2) as defined in the formulas from equation (6.3).

$$C_1^{(j)} = \begin{cases} A_1^{(j)} & \text{if } M_j = 0 \\ A_2^{(j)} & \text{if } M_j = 1 \end{cases} \quad \text{and} \quad C_2^{(j)} = \begin{cases} A_2^{(j)} & \text{if } M_j = 0 \\ A_1^{(j)} & \text{if } M_j = 1 \end{cases} \quad (6.3)$$

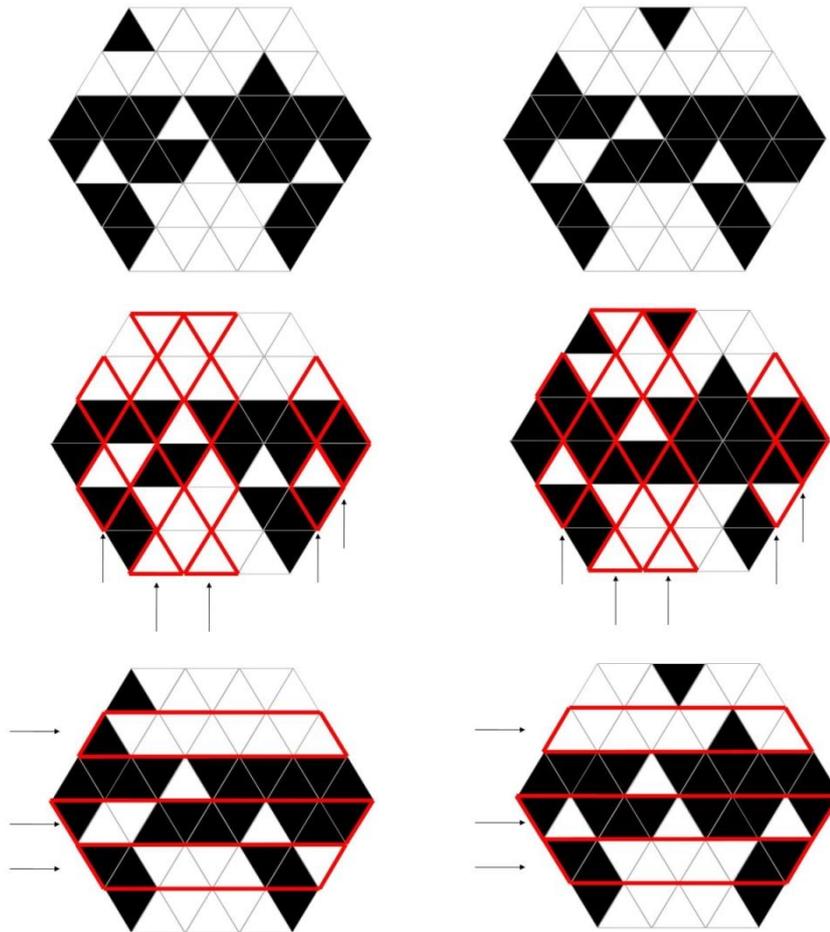


Figure 6.2. Two parents (first row) and their $y_{||}$ and y_{\perp} offsprings (second, third row).

Figure 6.2 shows an example for the crossover operator using directions parallel to y coordinate axis. In the first row two parents are represented. On the second row their $y_{||}$ offsprings for the two parents, obtained through the mask $M = (0; 1; 0; 1; 0; 1; 0; 0; 0; 1; 1)$ are represented. The third row represents the two y_{\perp} offsprings, obtained through the mask $M = (0; 1; 0; 1; 1; 0)$. The arrows, on the second and third rows, indicate the 1's in the binary mask M .

The main steps of the crossover procedure are presented in Pseudocode sequence 6.2.

The crossover operator mix the projection values of the parents in the chosen direction and usually changes the projection values in other directions.

Pseudocode sequence 6.2. Main steps for the crossover operator

Select two parents A_1, A_2
 Generate a random binary mask $M=(M_1, \dots, M_N), N=4m-1$
 Compute C_1 and C_2

6.3.4. GUIDED MUTATION OPERATOR

Guided mutation is applied on all three orthogonal directions. Given a hexagon-shaped binary image A its mutated version is obtained by changing the color of two neighbor black and white pixels. The pixels have the property that two coordinates have the same value and the third one differs by 1 unit and their color is different [Moi11]. The operator locates such pair of black and white pixels and chooses randomly the pairs of pixels inverting their color.

Figure 6.3 shows the mutated image, on the left-hand-image, which is obtained by inverting the color of two neighboring black and white pixels, the pixel pairs marked by the stars in the right-hand-side.

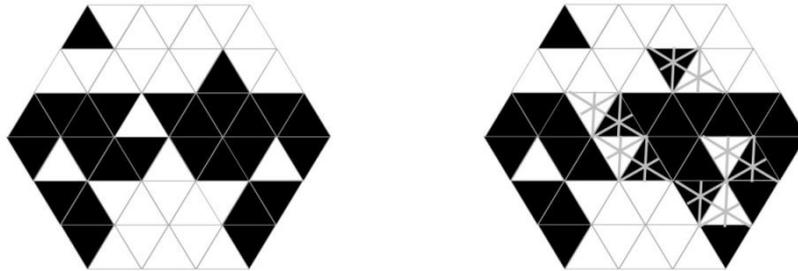


Figure 6.3. The mutated image (left) and the parent image (right).

The main steps for the guided mutation operator are presented in Pseudocode sequence 6.3.

Pseudocode sequence 6.3. Main steps for the guided mutation operator

```

Initialize the list of pairs of black and white pixels, that have two coordinates with
the same value and the third one differs by 1 unit
While (there are elements in the list )
    Generate a random number smaller than the number of elements from the list
    If (applying the random pair get better fitness value) then
        Apply the randomly selected pair
        Delete the pair and other pairs that contain elements of the selected pair
        from the list
    End If
End While

```

6.3.5. COMPACTNESS OPERATOR

Until now the operators described are suitable to reconstruct images with more components and holes, without using any prior information. The previously presented operators may introduce isolated pixels which usually worsen the solution. A pixel is called isolated if it is surrounded by 12 elements with opposite value.

Let A be a hexagon-shaped binary image. The compactness operation on A is an operation that eliminates some of the isolated points (independently of their original values).

Let $\sigma = \min(\alpha, \beta)$ where α is the number of isolated white pixels and β is the number of isolated black pixels. The compactness operator randomly locates σ isolated white pixels and σ isolated black pixels and sets them to the same value of their neighbors. Using the *min* function the availability of pairs of isolated black and white pixels is guaranteed.

Figure 6.4 shows two isolated pixels with their neighborhoods, on the left-hand-side. The image on the right-hand-side of the figure is obtained by setting the isolated pixels (on the left) to the same value of their neighbors.

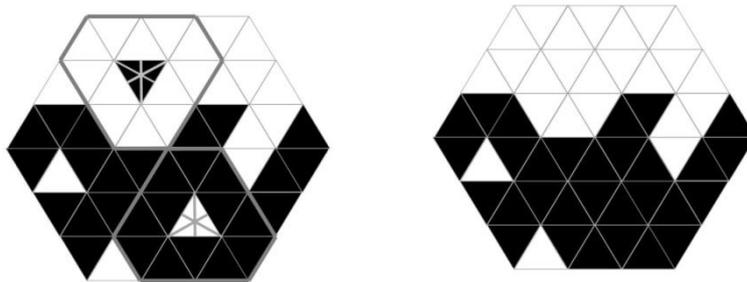


Figure 6.4. Two isolated pixels (a white and a black) and their neighborhoods. Image after applying compactness operator (right).

Pseudocode sequence 6.4 presents the main steps for the compactness operator.

Pseudocode sequence 6.4. Main steps for the compactness operator

Select a parent A
 Identify α the number of white isolated points
 Identify β the number of white isolated points and black
 Compute $\sigma = \min(\alpha, \beta)$
 Randomly locate σ isolated white pixels and σ isolated black pixels
 Sets the σ isolated pixels to the same value of their neighbors

6.3.6. MINIMAL HILL CLIMB OPERATOR

The minimal hill climb operators apply sequences of small modifications which increase the quality of the image. The proposed operators are based on the switching components which we define in the following sections.

The switching components maintain the three orthogonal projections of the image while swapping the values of pixels. Generally the operators change the values of the parallel direction projections.

In discrete tomography the concepts of switching components are important for finding reasonable solutions as a hexagon-shaped image can be transformed into another hexagon-shaped image that satisfies the same set of three projections by a finite sequence of switching components. For the triangular grid we define three type of switching components. Switching components have the properties that, by interchanging the values 0 and 1 of these pixels, the projection values do not change. We introduced the presented switching components for hexagon-shaped images in [Moi11]. In the following paragraphs, we propose a mathematical formulation of the switching components. After that, we describe the minimal hill climb operators based on the proposed switching components.

6.3.6.1. Rhombus switching component

Let A be a hexagon-shaped binary image. The rhombus switching component on A can be applied for the three directions orthogonal to the axes.

Considering the direction orthogonal to coordinate axis y , the rhombus switching component swaps $a_{i,j,k}$ with $a_{i,j,h}$ and $a_{i,j+1,k}$ with $a_{i,j+1,h}$, where the following conditions must be satisfied:

$$\begin{cases} a_{i,j,k} = a_{i,j+1,h} = 0 \\ a_{i,j,h} = a_{i,j+1,k} = 1 \end{cases} \quad \text{or} \quad \begin{cases} a_{i,j,k} = a_{i,j+1,h} = 1 \\ a_{i,j,h} = a_{i,j+1,k} = 0 \end{cases} \quad (6.4)$$

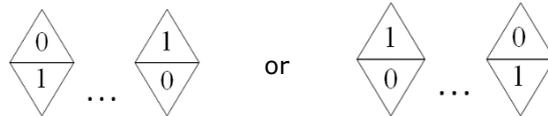
Considering the direction orthogonal to coordinate axis x , it swaps $a_{i,j,k}$ with $a_{i,j,h}$ and $a_{i-1,j,k}$ with $a_{i-1,j,h}$, where the next condition holds for these pixels of the image

$$\begin{cases} a_{i,j,k} = a_{i-1,j,h} = 0 \\ a_{i,j,h} = a_{i-1,j,k} = 1 \end{cases} \quad \text{or} \quad \begin{cases} a_{i,j,k} = a_{i-1,j,h} = 1 \\ a_{i,j,h} = a_{i-1,j,k} = 0 \end{cases} \quad (6.5).$$

Finally, considering the direction orthogonal to coordinate axis z , it swaps $a_{i,j,k}$ with $a_{i,j,h}$ and $a_{i,j,k-1}$ with $a_{i,h,k-1}$, where the next condition holds for these pixels of the image

$$\begin{cases} a_{i,j,k} = a_{i,h,k-1} = 0 \\ a_{i,h,k} = a_{i,j,k-1} = 1 \end{cases} \text{ or } \begin{cases} a_{i,j,k} = a_{i,h,k-1} = 1 \\ a_{i,h,k} = a_{i,j,k-1} = 0 \end{cases} \quad (6.6).$$

Actually, these rhombuses can appear in the images, for the direction orthogonal to coordinate axis y , as follows:



with the restriction that on the first row there are even parity points meanwhile on the second row we have odd parity points. The elements of these switching components form two rhombuses. Notice that the two neighbor pixels of a rhombus share two of their coordinate values and, consequently, by interchanging their pixel value (from 0 to 1 and vice-versa) data from two of the projection directions do not change. Only projection data of the direction orthogonal to the coordinate axis in which the coordinate values of the pixels differ is changed. In this way by having another rhombus that gives the opposite change a switching component is obtained.

Figure 6.5 shows an example for the rhombus switching component for direction orthogonal to coordinate axis y . The right-hand-image is obtained by a rhombus switching, the values of the pixels marked by the stars, in the left-hand-image, are inverted. For the other two directions the rhombus switching component works in the same way, actually one can obtain them by rotating the locations of the pixels of the previously shown components by $\frac{2\pi}{3}$ in both directions, respectively.

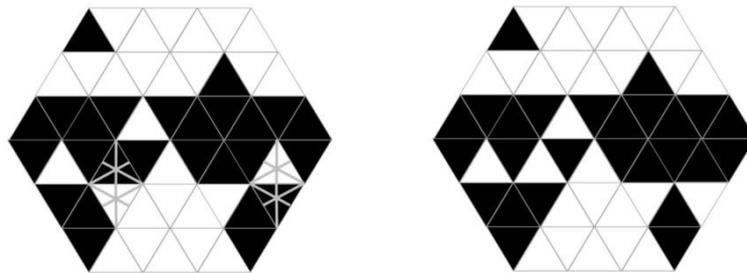


Figure 6.5. Rhombus switching operation for direction orthogonal to coordinate axis y .

6.3.6.2. Hourglass switching component

The hourglass switching operator on a hexagon-shaped binary image A can be applied for a subhexagon by swapping $a_{i,j,k}$ with $a_{i+1,j,k}$ or $a_{i+1,j,k-1}$ and $a_{i+1,j+1,k-1}$ with $a_{i,j+1,k}$ or $a_{i,j+1,k-1}$. The other two switching possibilities in the subhexagon consist of choosing two other pixels as the hourglass that has to be switched

(rotated) $a_{i+1,j,k}$ with $a_{i,j,k}$ or $a_{i+1,k,k-1}$ and $a_{i,j+1,k-1}$ with $a_{i,j+1,k}$ or $a_{i+1,j+1,k-1}$. The other switching possibility is to swap $a_{i+1,j,k-1}$ with $a_{i,j,k}$ or $a_{i+1,j,k}$ and $a_{i,j+1,k}$ with $a_{i,j+1,k-1}$ or $a_{i+1,j+1,k-1}$. The pixels from the subhexagon considered for the hourglass switching must satisfy one of the following constrains:

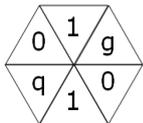
$$\begin{cases} a_{i,j,k} = a_{i+1,j+1,k-1} = v \\ a_{i+1,j,k} = a_{i,j+1,k-1} = w \end{cases} \text{ or } \begin{cases} a_{i,j,k} = a_{i+1,j+1,k-1} = w \\ a_{i+1,j,k-1} = a_{i,j+1,k} = v \end{cases}$$

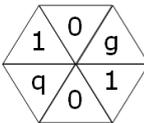
$$\begin{cases} a_{i+1,j,k} = a_{i,j+1,k-1} = v \\ a_{i,j,k} = a_{i+1,j+1,k-1} = w \end{cases} \text{ or } \begin{cases} a_{i+1,j,k} = a_{i,j+1,k-1} = w \\ a_{i+1,j,k-1} = a_{i,j+1,k} = v \end{cases} \quad (6.7)$$

$$\begin{cases} a_{i+1,j,k-1} = a_{i,j+1,k} = v \\ a_{i,j,k} = a_{i+1,j+1,k-1} = w \end{cases} \text{ or } \begin{cases} a_{i+1,j,k-1} = a_{i,j+1,k} = w \\ a_{i+1,j,k} = a_{i,j+1,k-1} = v \end{cases}$$

where $v, w \in \{0,1\}$, $v \neq w$.

Actually, the hourglass switching components, for example, can appear in

the images as follows:  that can be switched to subhexagons of the

form  where $g, q \in \{0,1\}$. In the case $g = q$ the hourglass can be

rotated both in 'clockwise' and 'anticlockwise' directions.

The combination of the triangles in an hourglass switching component is: 3 even-triangles and 3 odd-triangles, in the following way: first line even-odd-even triangles and second line odd-even-odd triangles.

Figure 6.6 shows the subhexagons of the hourglass switching for the three combinations from equation (6.7), where $v, w \in \{0,1\}$, $v \neq w$ and $g, q \in \{0,1\}$.

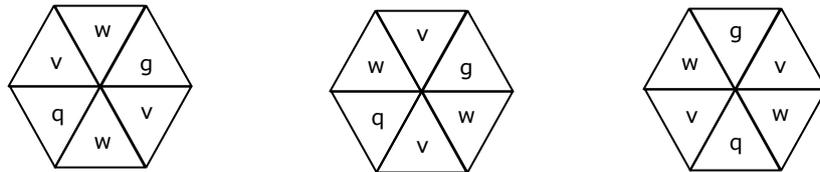


Figure 6.6. Subhexagons of the hourglass switching for the three combinations from equation (6.7).

Figure 6.7 shows an example for the hourglass switching component using the first combination from equation (6.7). The right-hand-image is obtained by using hourglass switching for the initial white pixels with their neighbors marked by the stars, in the left-hand-image.

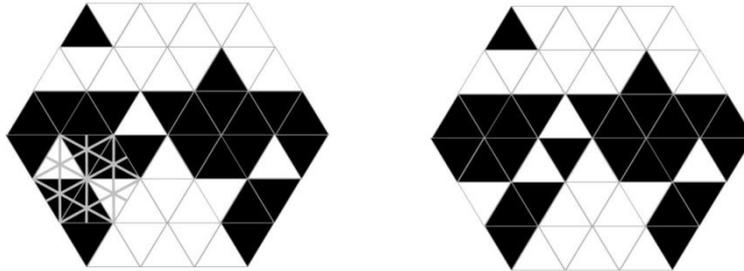


Figure 6.7. Example for the hourglass switching operation.

6.3.6.3. Rotation switching component

The rotation switching component on a hexagon-shaped binary image A can be applied for a subhexagon with size $r \leq m$ and it swaps $a_{i,j,k}$ with $a_{i,j+2r-1,k-2r+1}$, $a_{i+r,j+r,k-2r+1}$ with $a_{i-r+1,j+r,k}$ and $a_{i-r+1,j+2r-1,k-r+1}$ with $a_{i+r,j,k-r+1}$, where the pixels have to satisfy the following constraint:

$$\begin{cases} a_{i,j,k} = a_{i+r,j+r,k-2r+1} = a_{i-r+1,j+2r-1,k-r+1} = 0 \\ a_{i,j+2r-1,k-2r+1} = a_{i-r+1,j+r,k} = a_{i+r,j,k-r+1} = 1 \end{cases}$$

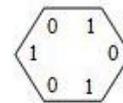
or (6.8)

$$\begin{cases} a_{i,j,k} = a_{i+r,j+r,k-2r+1} = a_{i-r+1,j+2r-1,k-r+1} = 1 \\ a_{i,j+2r-1,k-2r+1} = a_{i-r+1,j+r,k} = a_{i+r,j,k-r+1} = 0 \end{cases}$$

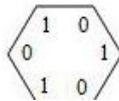
and each element of a pair $(a_{i,j,k}, a_{i,j+2r-1,k-2r+1})$, $(a_{i+r,j+r,k-2r+1}, a_{i-r+1,j+r,k})$, $(a_{i-r+1,j+2r-1,k-r+1}, a_{i+r,j,k-r+1})$ has the same parity. The pixels of such a pair are located at the opposite corners of the hexagon.

Actually, the rotation switching components can appear in the images as a

triplet of 0's and a triplet of 1's in the corner of subhexagons, as follows:



that changes into



, where a corner of the hexagon is formed by two

neighbor pixels and only one of them is used in the given switching component, as it is shown in Figure 6.8.

Figure 6.8 illustrates possible positions of the initial triplets in the rotation switching component for a hexagon of size 6 ($r = 3$). As one can see at any corner of the hexagon any of the two neighbor triangles can be used depending also on the other corners, so that in every direction (horizontal and directions / and \) exactly those lanes have a pixel with value 1 that have a pixel with value 0. Thus at each case, by switching the values of the pixels marked by 0 and 1 to their opposite value, the projections by lanes do not change.

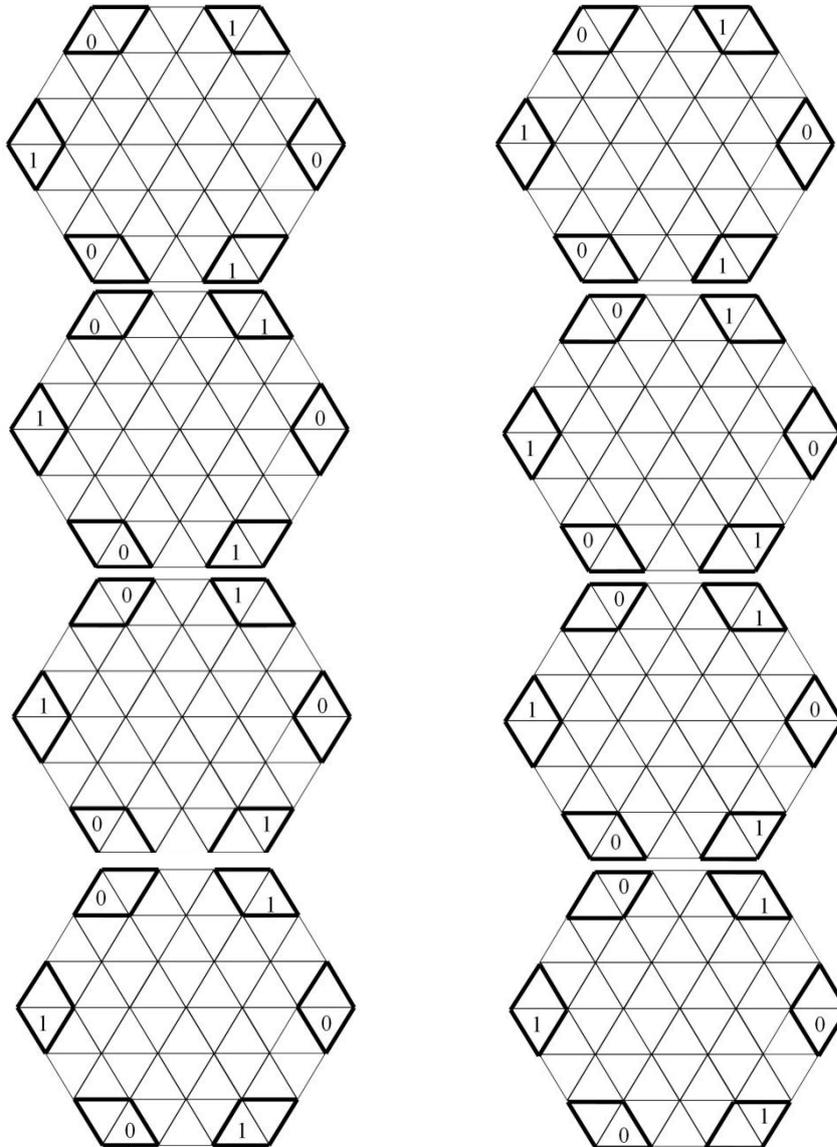


Figure 6.8. Possible positions of the initial triplets in the rotation switching component for a hexagon of size 6 ($r = 3$).

Considering a binary subhexagon image, the black and white pixels from the rotation switching component can be situated in $3! + 2 = 8$ ways at the corners of one subhexagon based on their parities, with $\text{size} \geq 2$ (see Figure 6.8). The possible combinations of the triangles in a rotation switching component are: 6 even-triangles, 6 odd-triangles, 3 possibilities in which there are 4 even-triangles and 2 odd-triangles and 3 possibilities in which there are 2 even-triangles and 4 even-triangles.

Based on the three previously proposed switching components, in the following paragraph we describe our minimal hill climb operators, considering a generalized case.

Because, using three directions of projections, a hexagon-shaped image can be transformed into another image with the same projection values using switching components, the minimal hill climb operators are used in the proposed memetic algorithm for local search. To each individual from the current population a list with each type of switching components is associated. A list contains the switching components that improve the quality of the reconstructed image. After applying randomly one switching component from the list, this is removed from the associated list, taking care to remove also the other switching components that contain an element from the considered switching component. Pseudocode sequence 6.5 shows the main steps for the generalized minimal hill climb operator, based on any of the three switching components. Usually the diamond-chain projections change their values.

Pseudocode sequence 6.5. Main steps of a generalized minimal hill climb operator

```
Initialize the lists with the switching component
Randomly select one switching component from the list
If (applying the switching component fitness improved) then
    Apply the selected switching component
    Remove the switching component from the list
End If
```

6.4. EXPERIMENTAL RESULTS

Regarding the proposed memetic algorithm we tested its benefits in terms of correctness, robustness and runtime. For the experimental session different databases of hexagon-shaped images were used. From the robustness point of view we will show that the presented approach produces stable results even if we use only six projection angles. The reconstructed solutions are obtained in a relatively low time even in the case of images with a big number of pixels.

For our tests we used regular hexagon-shaped images of dimension $m \times m \times m$. From our knowledge, there is not any database with hexagon-shape images represented on the triangular grid to test image processing algorithms. For this reason we used resampling methods to create our test images. The test images represented on the square grid were taken from the publicly available databases [Pat][MPE]. Images from these databases are resampled to images on the triangular grid. After resampling the image on the triangular grid, a number of pixels are suppressed, in order to obtain a hexagon-shaped image having triangular pixels. The dimension of the resampled images is $m \times m \times m$. The formed data-bases consist of hv-convex polyominoes, generic regular hexagon-shaped images and of non-convex/non-connected hexagon-shaped images resampled at different dimensions.

The algorithm was implemented in C#. Experiments were carried out on a system equipped with Intel(R)Core(TM) i5-2500 (3.30 GHz), 4 GB of RAM and operating system Windows 8. For the network flow problem which is used for the generation of the initial individuals in the population the MCF class [Ber04] is used.

To evaluate the reconstruction quality of our memetic algorithm the pixel difference (l_1 -distance) between the original image and the reconstructed image normalized by the number of the pixels of the image multiplied by 100 to obtain it in percentage (%), is computed. In the experiments various values of the parameter: number ng of generations and the size np of the population are used. For each session of runs we measured the average computation time in seconds. The computation time of our algorithm includes also the generation of the initial population.

In the next sections we consider the three classes of hexagon-shaped images. The first class contains 30 hv-convex polyominoes. The second one consists of 30 generic hexagon-shaped images. Images of both classes are generated from the database [Pat] and have a dimension of $26 \times 26 \times 26$ which means that have 4056 pixels. The dimension of such a hexagon-shaped image is very close to the dimension of a 64×64 square image in terms of the used pixels. The third considered class contains images from [MPE] database, resampled at dimensions of $4 \times 4 \times 4$, $8 \times 8 \times 8$, $13 \times 13 \times 13$ and $26 \times 26 \times 26$.

In order to validate our method, different experimental sessions were made. In the experimental session, different combinations of population size $\{50, 75, 100\}$ and number of generations $\{50, 100, 150\}$ is used. The algorithm runs as long as the number of generations is not met or a solution is found. In each generation a random number of pairs of individuals are selected for the crossover operator. Because pixel errors result in errors in each of the projections, when the number of projections is large, one pixel error in the reconstruction results in a large distance from the measured projections. Note that, for instance, replacing a pixel of the solution with one of its 1-neighbor pixels using projections by lanes and diamond-chains the projection values are changing in one direction on two consecutive lanes and in two diamond-chain directions on two-two consecutive diamond-chains. In this way, this small pixel error can cause 6 error points in fitness using six projection angles.

6.4.1. HV-CONVEX POLYOMINOES

Firstly we consider the data set of 30 hv-convex polyominoes having the dimension of $m=26$. Example of images from this class can be seen in Figure 6.10.

In order to lower the effect of randomness, the average of the error ε and the average of the fitness function f on all test cases, are computed. Also the number of solutions, $n\varepsilon$, with the pixel error less than the average pixel error ε is computed. For example, 74% of the hv-convex solutions obtained using a population of 100 and generation 150 has an error less than the average error of 0,0122%.

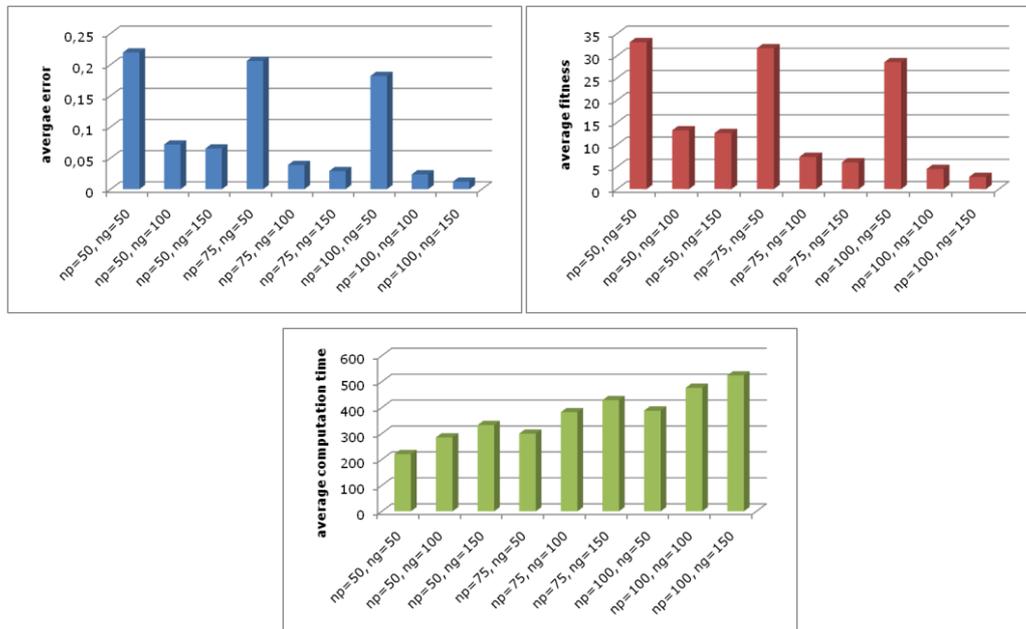
Table 6.1 shows the results for the considered hv-convex polyominoes with dimension $26 \times 26 \times 26$, using different combinations between the number of populations and generations.

Table 6.1. HV-convex polyominoes with dimension $26 \times 26 \times 26$.

Parameters		Quality of reconstruction			Computation time
np	ng	f	$\varepsilon(\%)$	$n\varepsilon$	(seconds)
50	50	33	0,2195	71%	209
50	100	13	0,0717	68%	285
50	150	13	0,0653	69%	333
75	50	32	0,2060	72%	299
75	100	7	0,0389	76%	382
75	150	6	0,0291	76%	428
100	50	29	0,1817	73%	388
100	100	5	0,0238	76%	476
100	150	3	0,0122	74%	524

We studied the effect of our algorithm considering different number of populations and generations. We found that for some cases we obtained even 76% of the solutions with an error less than the average error. The best obtained average error is of 0,0122%, with an average fitness value of 3 and the average computation time of 524 seconds. This means that almost all the 30 images from our database were perfectly reconstructed using a population of 100 and generation of 150.

Figure 6.9 shows the average results of the reconstruction error, fitness and computation time values for the 30 hv-convex polyominoes from our database. Average reconstruction error, fitness and computation time values are shown by color, blue, red and green, respectively.

Figure 6.9. Average results for hv-convex polyominoes of dimension $26 \times 26 \times 26$.

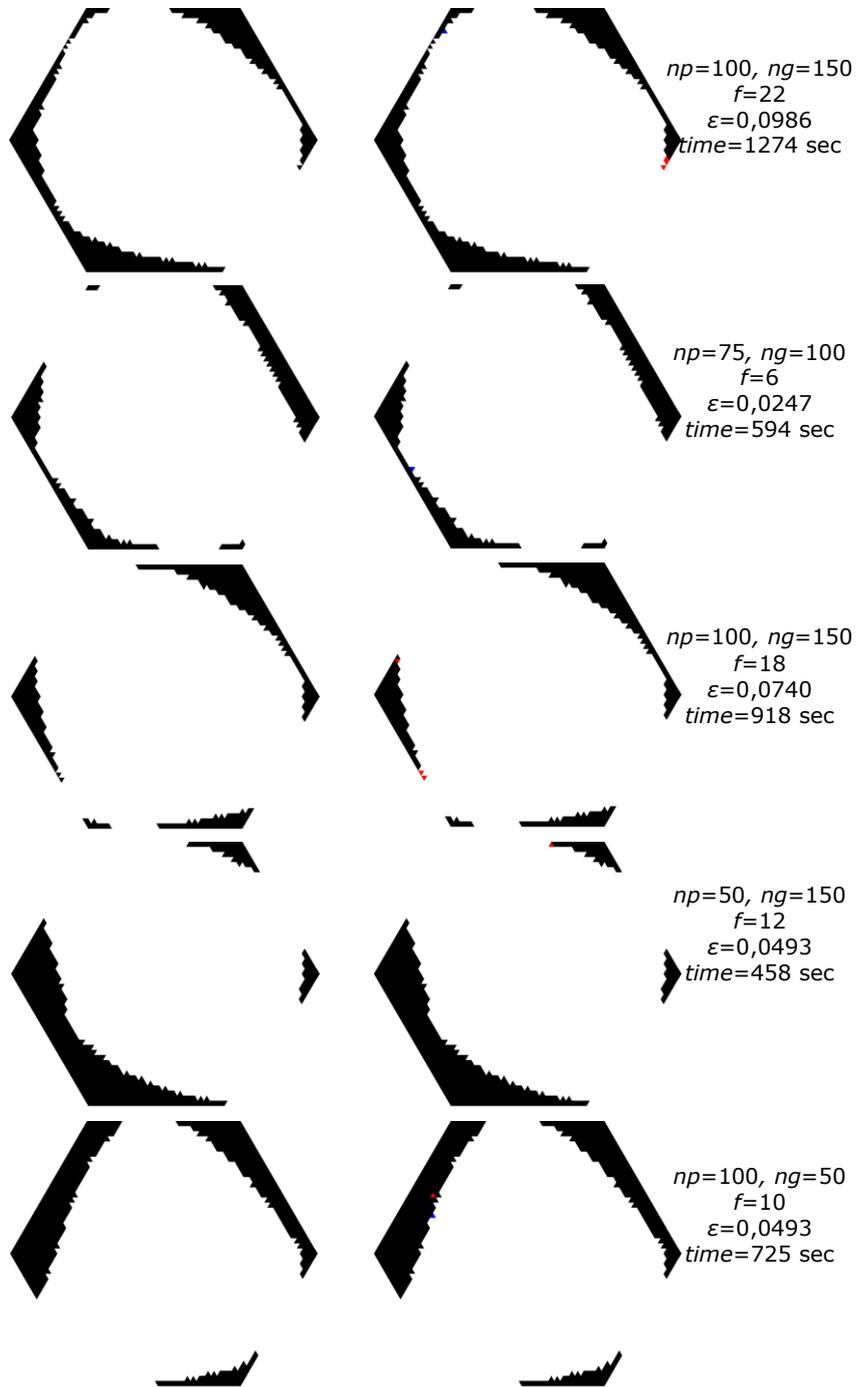


Figure 6.10. Five 26x26x26 hv-convex polyomino (left) and their reconstruction (right).

Figure 6.10 shows, on the left, five examples of hv-convex polyominoes with dimension $26 \times 26 \times 26$, taken from our test database. On the right hand side the reconstructions for the five considered examples of images are shown having different error and fitness values. Errors are shown by color: missing pixels are blue, additional pixels of the reconstructed image are red. Our test results show that using a population of 100 and generation of 150 for almost all the cases we obtained at least a perfect reconstruction in our 10 runs. But in case of other population and generation numbers like population of 50, 75 and 100 with generation of 100 and 150 we obtained in almost all the cases one perfect reconstruction.

Figure 6.11 - Figure 6.13 show the complete test results for the hv-convex polyominoes. The small blue circles indicate the fitness, reconstruction error and computation time less than the average values, for the test images. The small red circles indicate the fitness, reconstruction error and computation time greater than the average values.

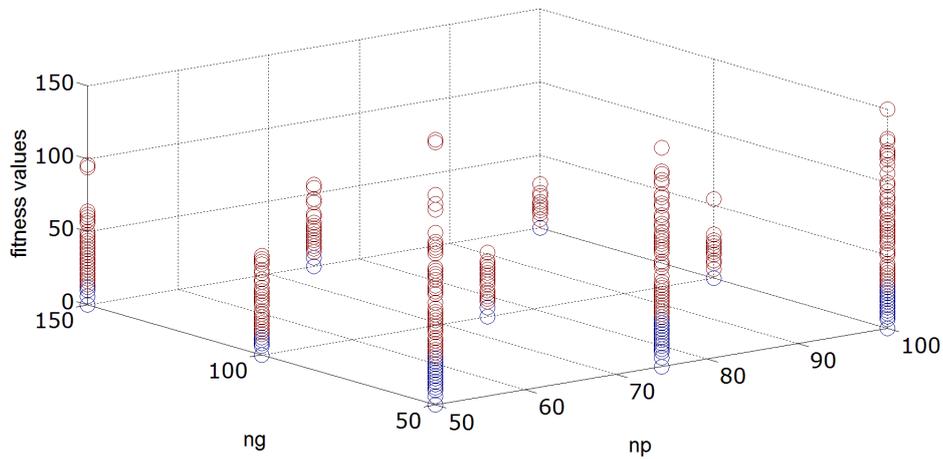


Figure 6.11. Fitness results for hv-convex polyominoes of dimension $26 \times 26 \times 26$.

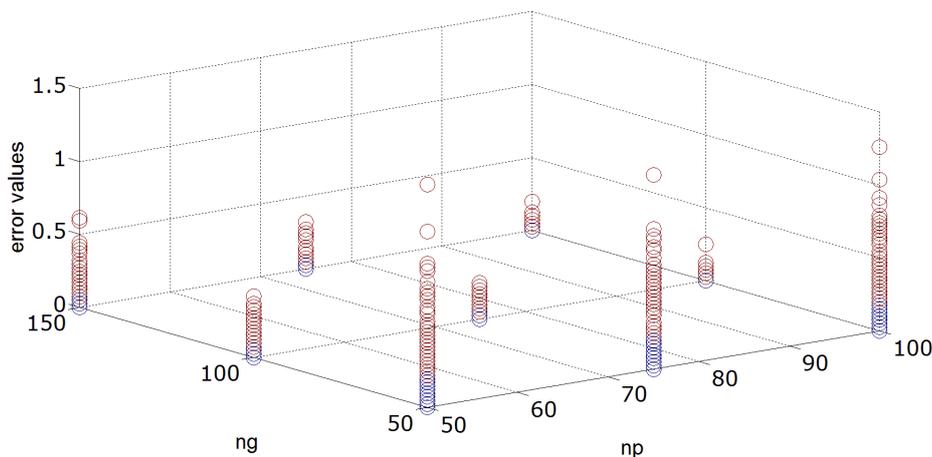


Figure 6.12. Error results for hv-convex polyominoes of dimension $26 \times 26 \times 26$.

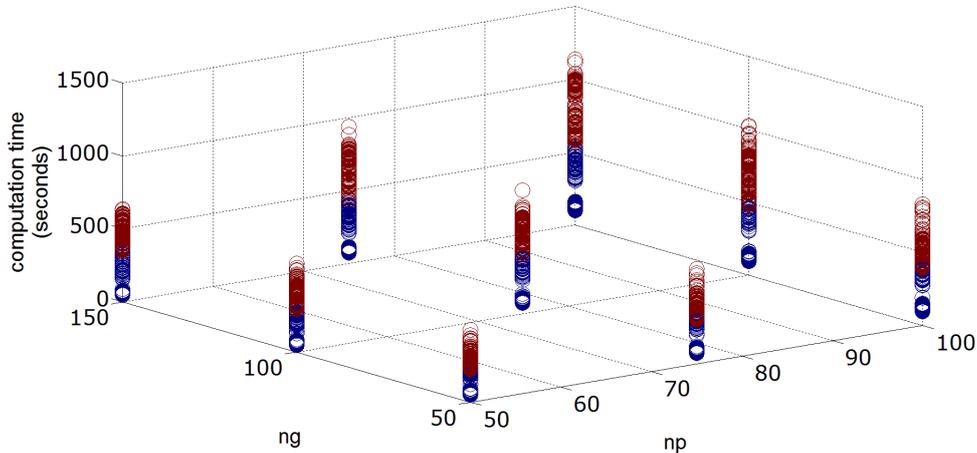


Figure 6.13. Computation time results for hv-convex polyominoes of dimension $26 \times 26 \times 26$.

In conclusions even choosing a reasonable number for the population and for generations, the proposed memetic algorithm leads us to robust solutions that are very close to the original images.

6.4.2. GENERIC REGULAR HEXAGON-SHAPED IMAGES

The second class of images is the one of 30 generic images that does not present any particular topological or geometrical properties resampled from [Pat] database, with dimension of $26 \times 26 \times 26$.

Table 6.2 shows the results for the considered set of images. In order to lower the effect of randomness, the average of the error ε and the average of the fitness function f on all test cases, are computed. Also the number of solutions, $n\varepsilon$, with the pixel error less than the average pixel error ε is computed. Experimental results show that for the considered data set we obtained more than 53% solutions that have the reconstruction error less than the average error $\varepsilon=6,0360\%$, for a population number of 100 and generation number 150.

Table 6.2. Generic hexagon-shaped images with dimension $26 \times 26 \times 26$.

Parameters		Quality of reconstruction			Computation time (seconds)
np	ng	f	ε	$n\varepsilon$	
50	50	363	8,9528	49%	1055
50	100	311	7,6783	51%	1413
50	150	283	6,9889	55%	1614
75	50	355	8,7528	49%	1504
75	100	293	7,2299	52%	1965
75	150	360	6,4121	55%	2194
100	50	341	8,4153	50%	1951
100	100	278	6,8497	52%	2501
100	150	245	6,0360	53%	2835

Figure 6.14 shows the average results of the error, fitness and computation time values for the 30 generic regular hexagon-shaped images from our database. Average error, fitness and computation time values are shown by color, blue, red and green, respectively.

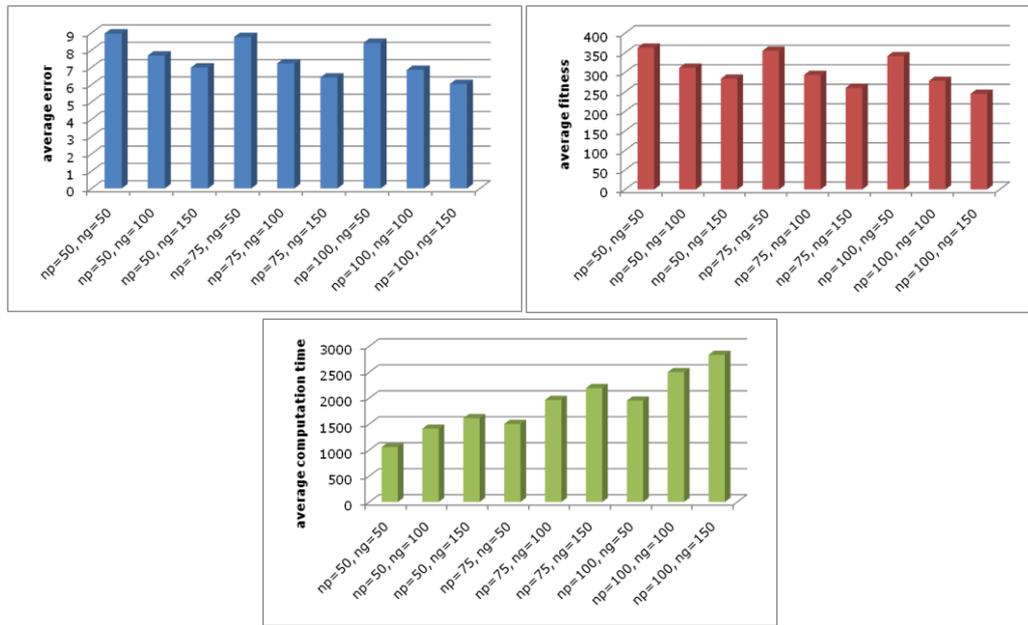


Figure 6.14. Average results for 30 generic hexagon-shaped images of dimension $26 \times 26 \times 26$.

Figure 6.15 shows, on the left, five examples of generic hexagon shaped images with dimension $26 \times 26 \times 26$, taken from our test database. On the right hand side the reconstructions for the five images are shown having reconstruction error and fitness values equal to the one shown. Errors are shown by color: missing pixels are blue, additional pixels of the reconstructed image are red. Our results show that the proposed memetic algorithm manages to reconstruct in a robust manner objects in such a way that the shape of objects can be distinguished even in more complex images.

Figure 6.16 shows the fitness results for the 30 generic hexagon-shaped images case. Figure 6.17 shows the reconstruction error results for the 30 generic hexagon-shaped images. The average computation time for each case is shown by Figure 6.18. The small blue circles indicate the fitness, reconstruction error and computation time less than the average values, for the test images. The small red circles indicate the fitness, reconstruction error and computation time greater than the average values.

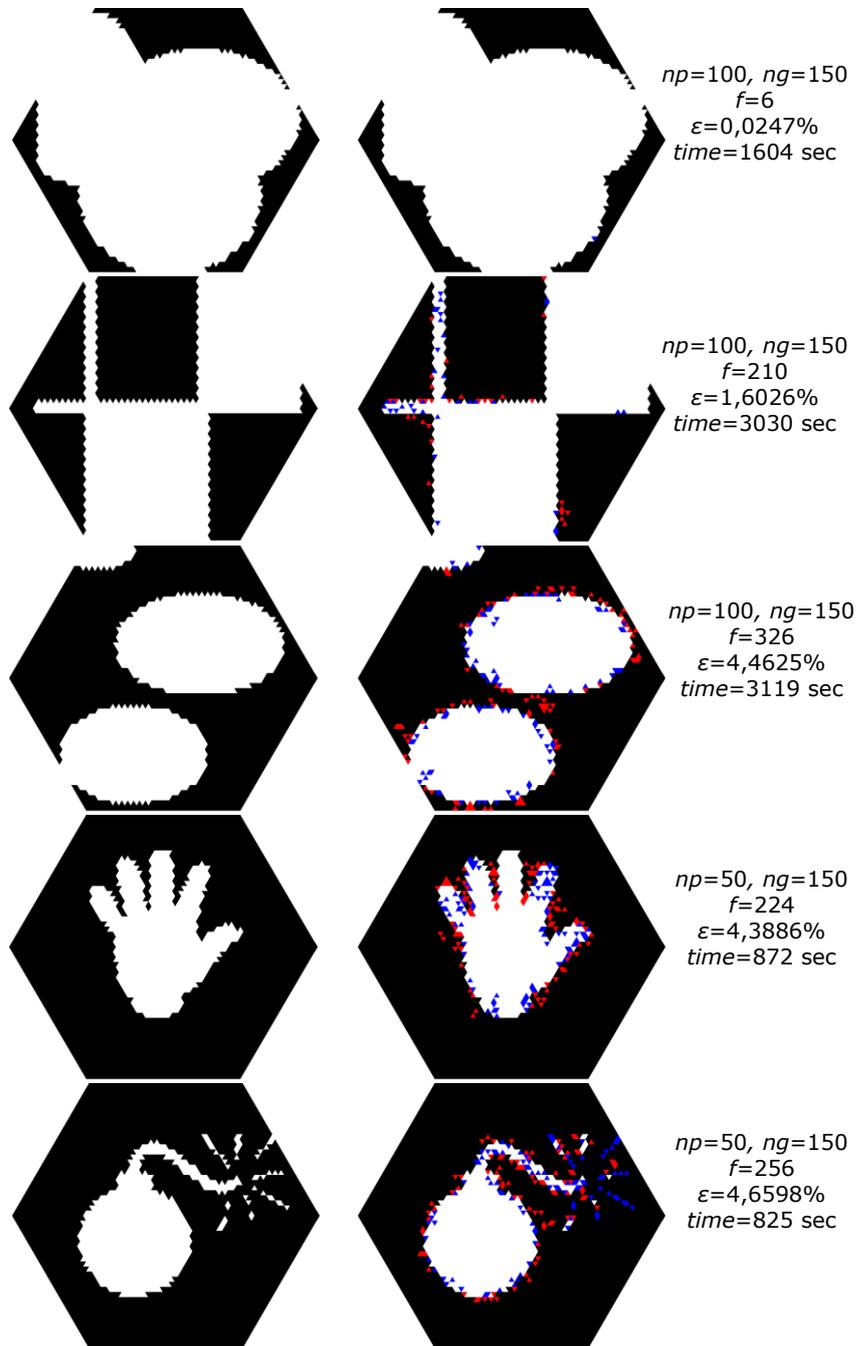


Figure 6.15. Five 26x26x26 generic hexagon-shaped images (left) and their reconstruction (right).

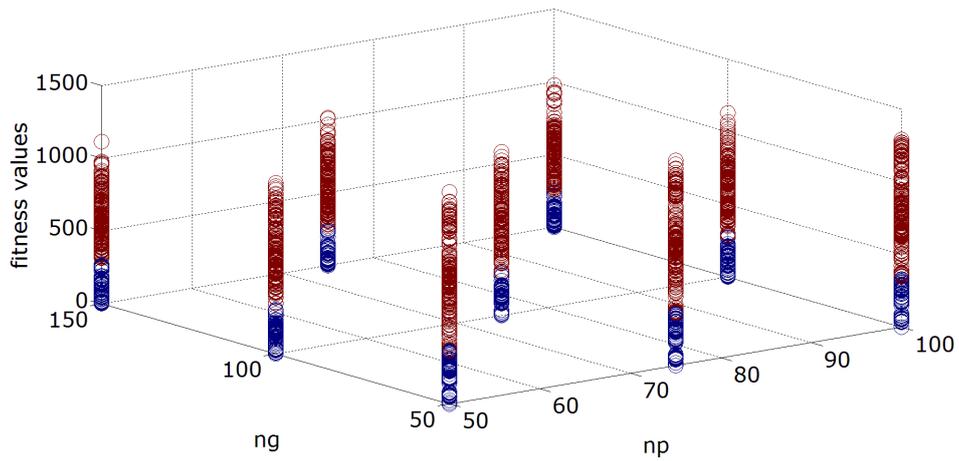


Figure 6.16. Fitness results for generic hexagon-shaped images of dimension $26 \times 26 \times 26$.

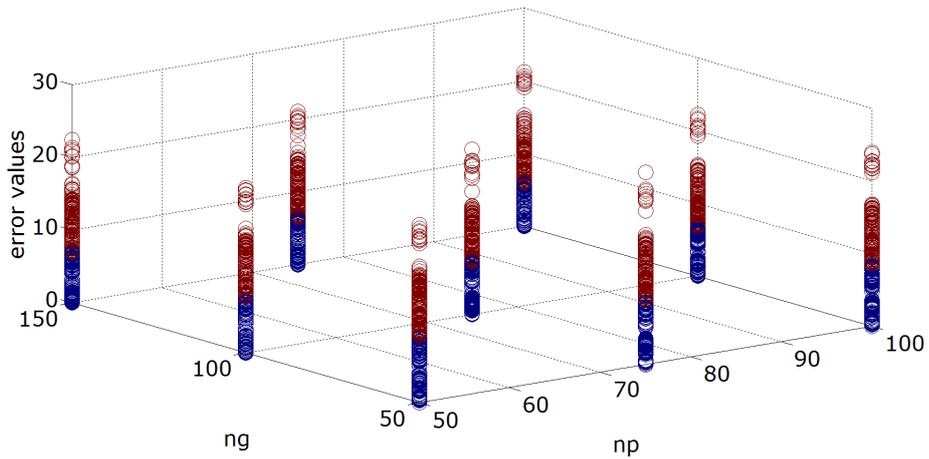


Figure 6.17. Reconstruction error results for generic hexagon-shaped images of dimension $26 \times 26 \times 26$.

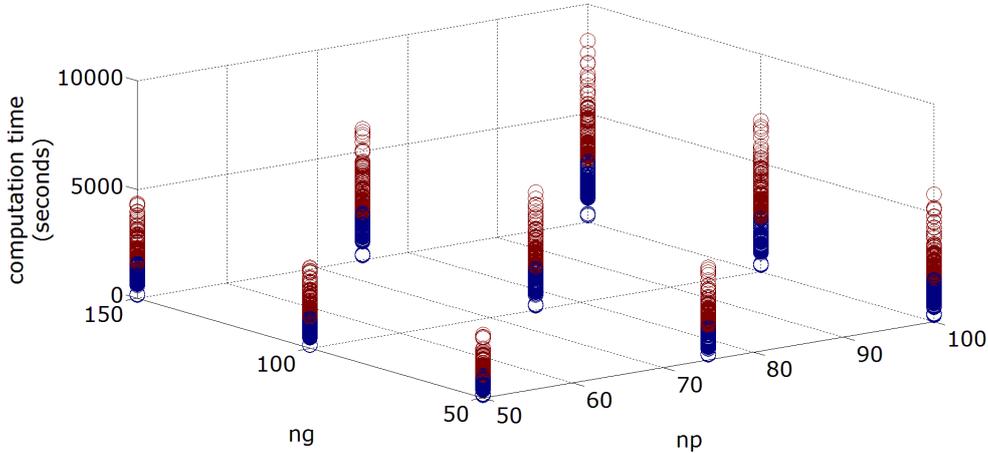


Figure 6.18. Computation time results for generic hexagon-shaped images of dimension $26 \times 26 \times 26$.

Experiments show that our memetic algorithm is effective even in the cases in which we consider non-convex or non-connected hexagon shaped images which do not satisfy any topological or geometrical properties. Our algorithm is efficient mainly in terms of computation time. The experiments show that for images that contain compact objects, for example the first and fourth image from Figure 6.15, our algorithm performs a very good reconstruction in a relatively short computation time. For images that contain more complex non-convex or non-connected objects our algorithm performs a good reconstruction, in terms that the shape of the object or objects contained in the image can be clearly distinguished.

6.4.3. NON-CONVEX REGULAR HEXAGON-SHAPED IMAGES

In order to validate our memetic algorithm we consider a class of 10 non-convex regular hexagon-shaped images resampled at dimension of $m = \{4, 8, 13, 26\}$. The number of pixels in the test images varies from 96 to 4056, respectively.

Figure 6.19 shows five non-convex images of dimension $26 \times 26 \times 26$ generated from database [MPE]. Even if, there are cases for the shown images when the reconstruction is perfect, there is no reconstruction error, in Figure 6.19 we present reconstructed images that have small reconstruction errors and fitness values. On the right hand side the reconstructions for the five images are shown with their reconstruction error and fitness values for the specified population and generation number. Errors are shown by color: missing pixels are blue, additional pixels of the reconstructed image are red.

In order to lower the effect of randomness, the average of the error ε and the average of the fitness function f on all test cases, are computed. Also the number of solutions, $n\varepsilon$, with the pixel error less than the average pixel error ε is computed.

Table 6.3 shows the average reconstruction results for the 10 considered non-convex hexagon shaped images having a size of $4 \times 4 \times 4$, which means 96 pixels. For this size of images our results show that for all the cases we obtained perfect reconstruction only in some seconds.

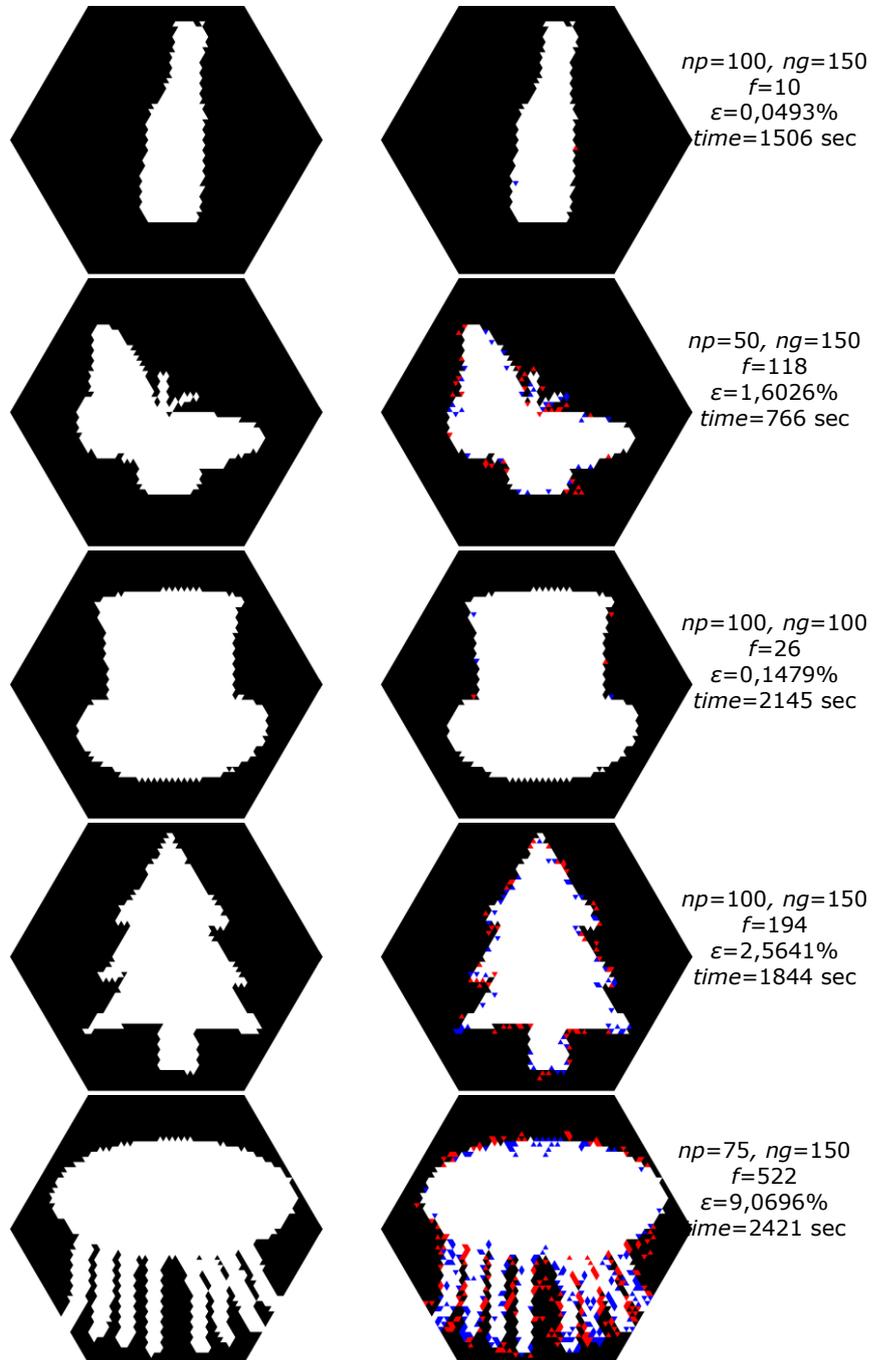


Figure 6.19. Five examples of 26x26x26 non-convex regular hexagon-shaped images (left) and their reconstruction (right).

Table 6.3. Non-convex regular hexagon-shaped images with dimension $4 \times 4 \times 4$.

Parameters		Quality of reconstruction			Computation time (seconds)
np	ng	f	ϵ	$n\epsilon$	
50	50	0	0,0000	100%	1
50	100	0	0,0000	100%	2
50	150	0	0,0000	100%	2
75	50	0	0,0000	100%	4
75	100	0	0,0000	100%	4
75	150	0	0,0000	100%	4
100	50	0	0,0000	100%	5
100	100	0	0,0000	100%	5
100	150	0	0,0000	100%	5

Figure 6.20 shows the average results of the reconstruction error, fitness and computation time values for the 10 non-convex regular hexagon-shaped images from our database. Average reconstruction error are shown in the blue color chart. The average fitness errors are shown in the chart of red color and the average computation time values are illustrated in the chart of green color.

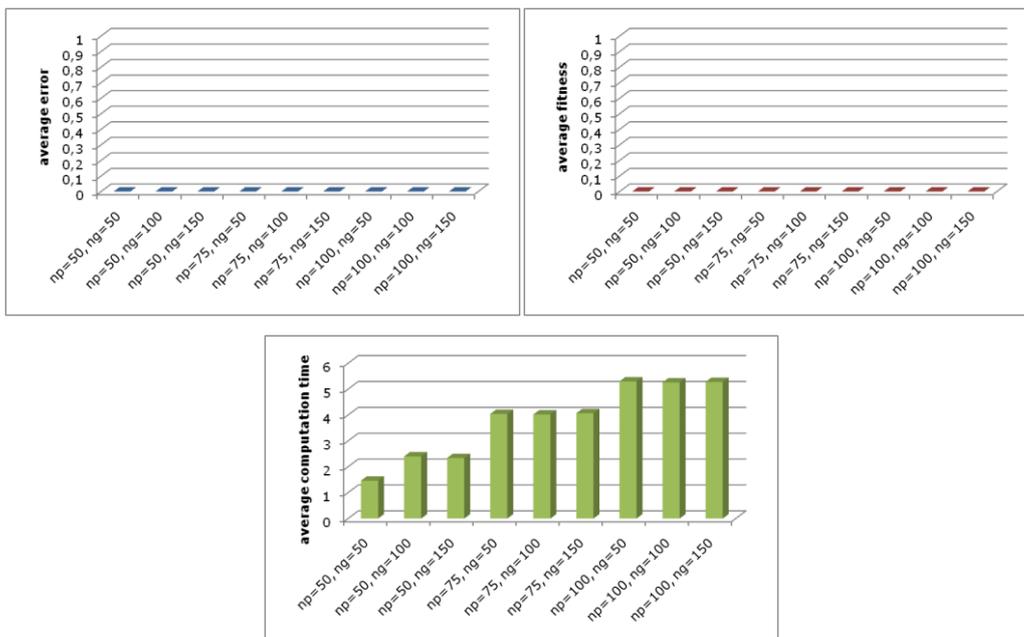
Figure 6.20. Average results for 10 non-convex regular hexagon-shaped images of dimension $4 \times 4 \times 4$

Table 6.4 shows the reconstruction results for the 10 on-convex hexagon-shaped images having a size of $8 \times 8 \times 8$, which means 384 pixels. Experimental results show that for the considered data set we obtained more than 82% solutions that have the reconstruction error less than the average error $\epsilon=1,5000\%$ and

fitness value $f=6$, for a population number of 100 and generation number 150, with a computation time of some seconds.

Table 6.4. Non-convex regular hexagon-shaped images with dimension $8 \times 8 \times 8$.

Parameters		Quality of reconstruction			Computation time (seconds)
np	ng	f	ϵ	$n\epsilon$	
50	50	10	2,4922	78%	8
50	100	9	2,2526	79%	14
50	150	8	1,9896	78%	16
75	50	10	2,4818	77%	18
75	100	7	1,8620	80%	20
75	150	6	1,5938	82%	21
100	50	10	2,5026	79%	23
100	100	8	1,9635	80%	25
100	150	6	1,5000	82%	28

Figure 6.21 shows the average results of the reconstruction error, fitness and computation time values for the 10 non-convex regular hexagon-shaped images with a size $8 \times 8 \times 8$. Average reconstruction error, fitness and computation time values are shown in the blue, red and green color chart, respectively.

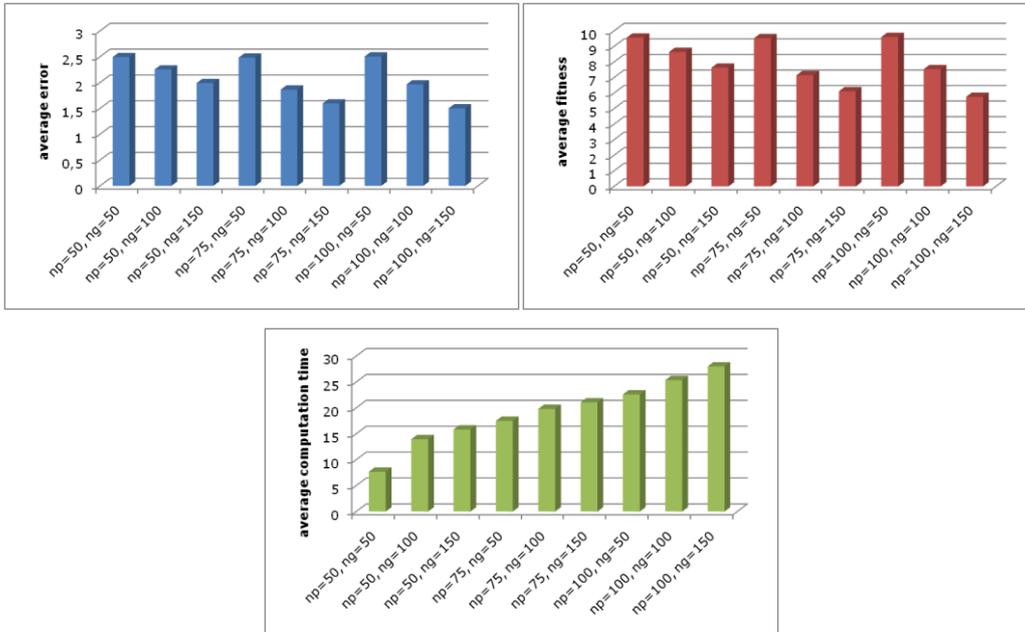


Figure 6.21. Average results for 10 non-convex regular hexagon-shaped images of dimension $8 \times 8 \times 8$

Table 6.5 shows the reconstruction results for the 10 on-convex hexagon-shaped images having a size of $13 \times 13 \times 13$, which means 1014 pixels. Experimental results show that for the considered data set we obtained more than 80% solutions

that have the reconstruction error less than the average error $\varepsilon=2,0148\%$ and fitness value $f=20$, for a population number of 100 and generation number 150, with a computation time of 136 seconds. For the other cases we obtained more than 77% solutions that have the reconstruction error less than the average error computed for the considered population and generation number.

Table 6.5. Non-convex regular hexagon-shaped images with dimension $13 \times 13 \times 13$.

Parameters		Quality of reconstruction			Computation time
np	ng	f	ε	$n\varepsilon$	(seconds)
50	50	33	3,2278	77	40
50	100	24	2,3314	79	67
50	150	23	2,2258	78	79
75	50	33	3,2288	78	74
75	100	22	2,2041	78	92
75	150	20	1,9280	80	106
100	50	33	3,2643	80	94
100	100	22	2,2022	80	120
100	150	20	2,0148	80	136

Figure 6.22 shows the average results of the reconstruction error, fitness and computation time values for the 10 non-convex regular hexagon-shaped images with a size $13 \times 13 \times 13$. Blue, red and green color charts represent the average reconstruction error, fitness and computation time values.

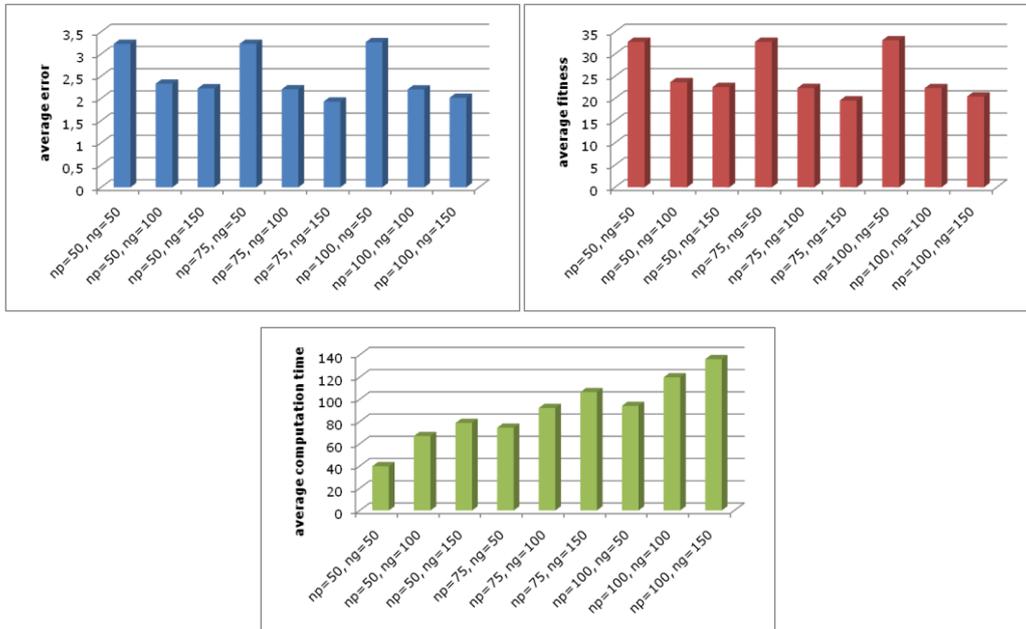


Figure 6.22. Average results for 10 non-convex regular hexagon-shaped images of dimension $13 \times 13 \times 13$

Table 6.6 shows the reconstruction results for the 10 on-convex hexagon-shaped images having a size of $26 \times 26 \times 26$, which means 4056 pixels. Experimental results show that for the considered data set we obtained more than 71% solutions that have the reconstruction error less than the average error $\epsilon = 3,2040\%$ and fitness value $f = 130$, for a population number of 100 and generation number 150, with a computation time of some minutes. More than 68% solutions that have the reconstruction error less than the average error computed for the considered population and generation number was obtained for the other cases too.

Table 6.6. Non-convex regular hexagon-shaped images with dimension $26 \times 26 \times 26$.

Parameters		Quality of reconstruction			Runtime (seconds)
np	ng	f	ϵ	$n\epsilon$	
50	50	186	4,5811	68	807
50	100	150	3,7029	69	1018
50	150	140	3,4440	71	1165
75	50	182	4,4931	71	1125
75	100	144	3,5621	70	1404
75	150	130	3,2061	70	1599
100	50	179	4,4019	68	1462
100	100	142	3,5096	70	1829
100	150	130	3,2040	71	2035

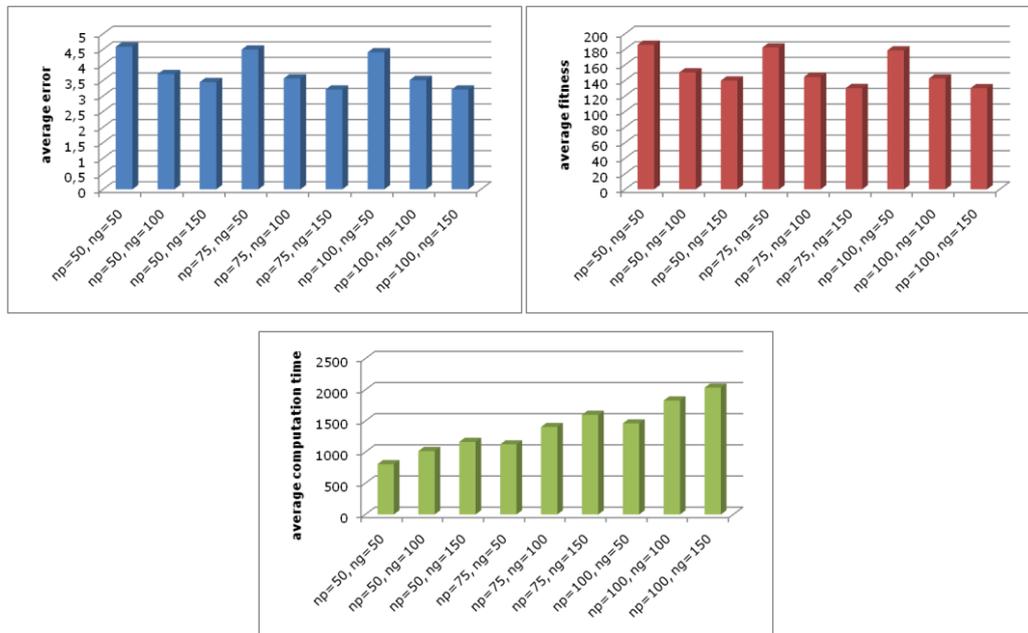


Figure 6.23. Average results for 10 non-convex regular hexagon-shaped images of dimension $26 \times 26 \times 26$

Figure 6.23 shows the average results of the reconstruction error, fitness and computation time values for the 10 non-convex regular hexagon-shaped images with a size $26 \times 26 \times 26$. The blue color chart illustrates the average reconstruction error values for the considered combinations of population and generation numbers. The red color chart illustrates the average reconstruction fitness values for the considered combinations of population and generation numbers. The green color chart shows the average computation time values for the considered combinations of population and generation numbers.

Figure 6.24 shows the complete test results for the 10 non-convex regular hexagon-shaped test images of dimension $4 \times 4 \times 4$. The small blue circles indicate the fitness, reconstruction error and computation time less than the average values for the considered test images. The small red circles indicate the fitness, reconstruction error and computation time greater than the average values.

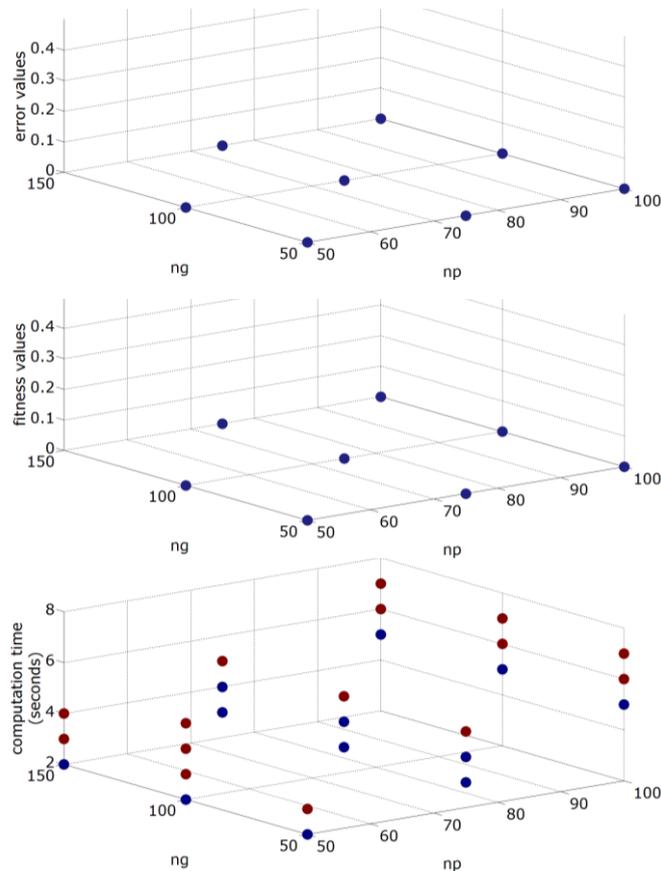


Figure 6.24. Results for non-convex regular hexagon-shaped images of dimension $4 \times 4 \times 4$.

Figure 6.25 shows the complete test results for the 10 non-convex regular hexagon-shaped test images of dimension $8 \times 8 \times 8$. The small blue circles indicate the fitness, reconstruction error and computation time less than the average values

for the considered test images. The small red circles indicate the fitness, reconstruction error and computation time greater than the average values.

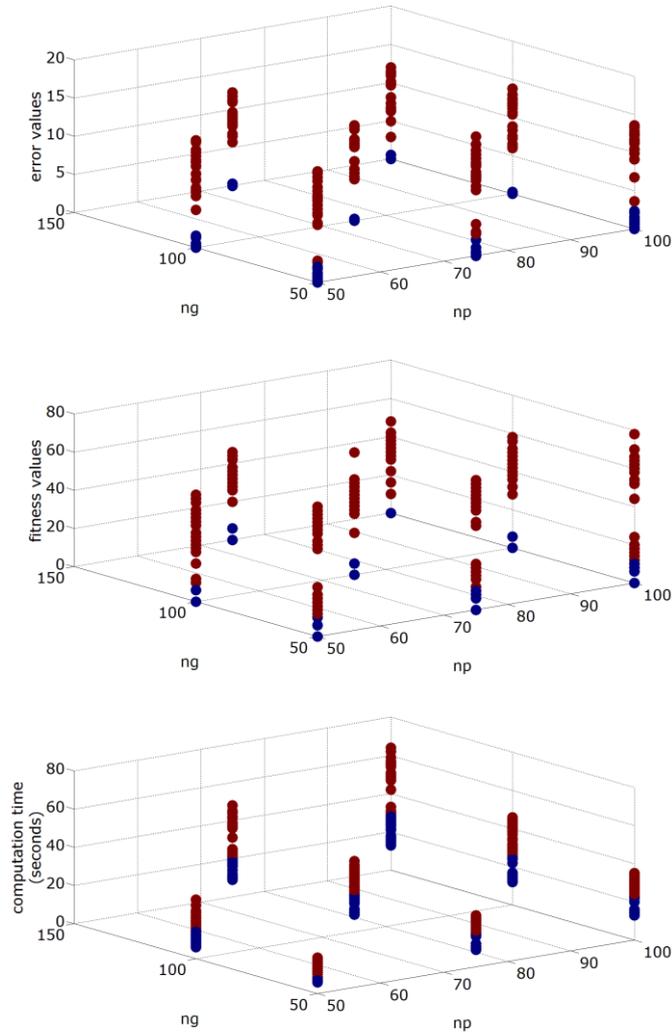


Figure 6.25. Results for non-convex regular hexagon-shaped images of dimension $8 \times 8 \times 8$

Figure 6.26 shows the complete test results for the 10 non-convex regular hexagon-shaped test images of dimension $13 \times 13 \times 13$. The small blue circles indicate the fitness, reconstruction error and computation time less than the average values for the considered test images. The small red circles indicate the fitness, reconstruction error and computation time greater than the average values.

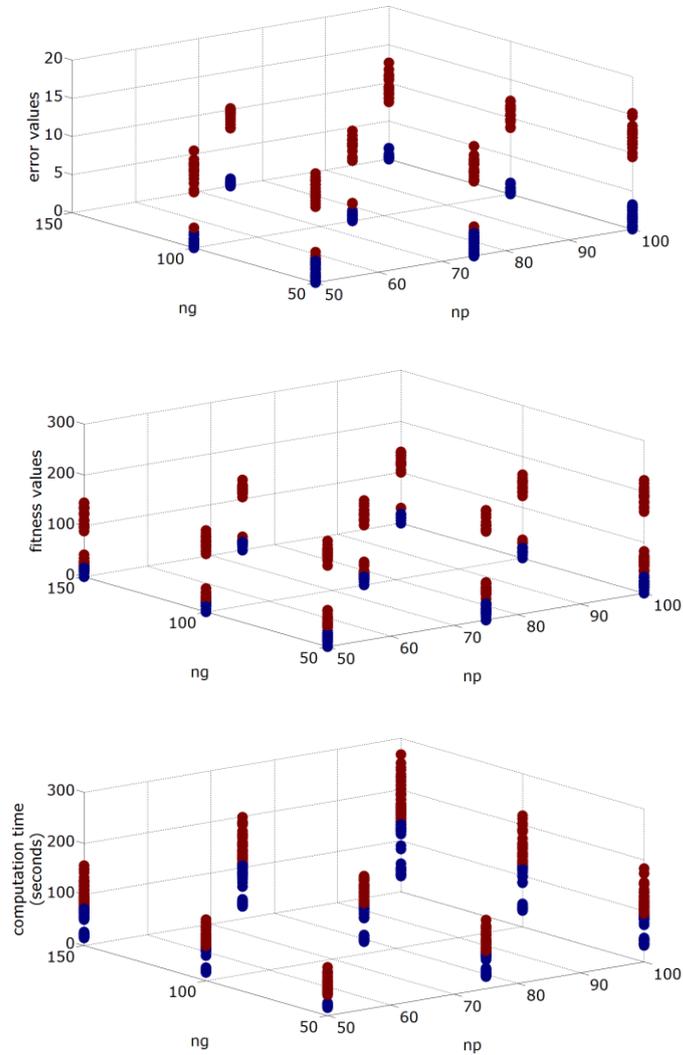


Figure 6.26. Results for non-convex regular hexagon-shaped images of dimension $13 \times 13 \times 13$.

Figure 6.27 shows the complete test results for the 10 non-convex regular hexagon-shaped test images of dimension $26 \times 26 \times 26$. The small blue circles indicate the fitness, reconstruction error and computation time less than the average values for the considered test images. The small red circles indicate the fitness, reconstruction error and computation time greater than the average values.

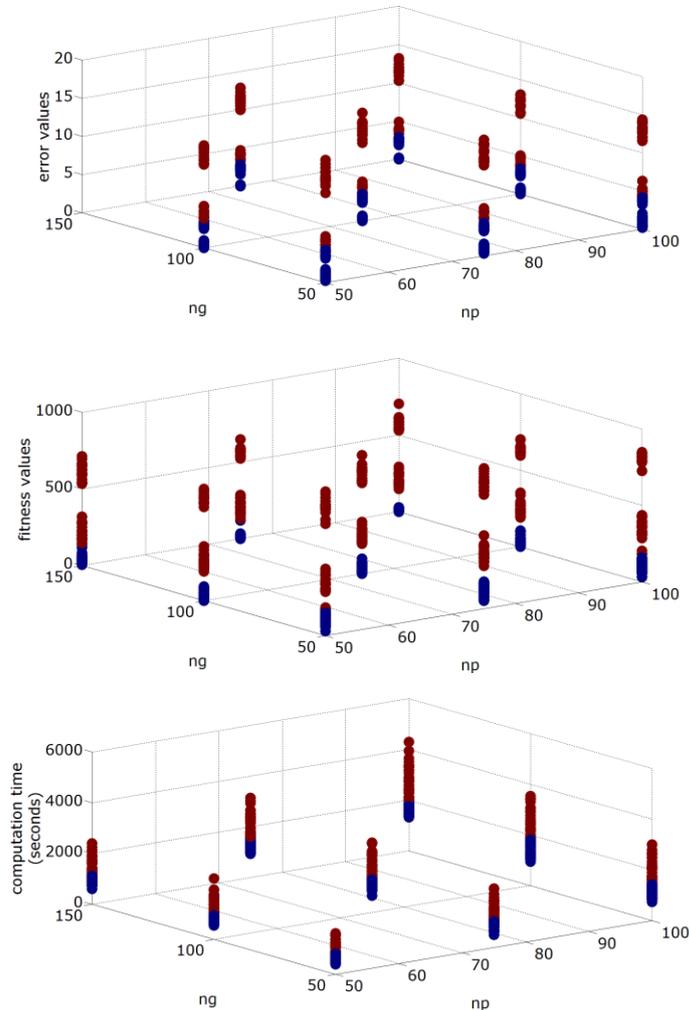


Figure 6.27. Results for non-convex regular hexagon-shaped images of dimension $26 \times 26 \times 26$.

For the non-convex regular shaped images, experimental results show that the proposed memetic algorithm is able to reconstruct in a robust manner images, in such a way that the shape of objects can be distinguished in each case. The computation time for such images is lower than an hour.

We tested our memetic algorithm on three test classes that contain sets of binary hexagon-shaped images. Experimental results show that the proposed memetic algorithm is robust enough, although hexagon-shaped images that satisfy prior knowledge, like the hv-convex polyomino constraint returned better results.

In the case of considering different dimension hexagon-shaped images, experimental results show that images that satisfy topological or geometrical constraints returned better results. It should be noted that the generic hexagon-

shaped images are pretty difficult to reconstruct since they do not satisfy any particular implicit constraint and are pretty different from each other.

In the case of classes of hexagon-shaped images that we considered, the introduced guided mutation and minimized hill climb operators significantly decreases the average fitness and reconstruction error. This is clearly shown in all the experimental results that by using a significantly high number of generations the average values of the fitness and reconstruction error significantly improved.

Experimental results also show that computation time needed to reconstruct a considered hexagon-shaped image increases in a quasi linear manner, because it depends on the number of population and generations. Also the gap of the algorithm, regarding the run time, is the computation of all the switching components implied in the hill climb operators, which depends also on the dimension of the image and the population number.

6.5. PERFORMANCE ANALYSIS

In this section we briefly describe and compare our results to the results of the reconstruction based on simulated annealing (SA) introduced for the triangular grid case [Luk12][Nag13a]. Simulated annealing is a stochastic optimization algorithm based on the simulation of physical process of slow cooling of the material in a heat bath [Kir83].

The considered problem to be solved is the binary tomography reconstruction problem where the imaging process is represented by the linear system of equations:

$$Ax = b, \quad A \in \mathbb{R}^{m \times n}, \quad x \in \{0, 1\}^n, \quad b \in \mathbb{R}^m \quad (6.9)$$

where A is the projection matrix, b contains the detected m projection values and x represents the unknown-image.

For the simulated annealing case the considered binary tomography image reconstruction problem is reformulated into an energy-minimization problem given by [Luk12]:

$$\min_{x \in \{0, 1\}^n} E(x) \quad (6.10)$$

where the energy function is:

$$E(x) = \frac{1}{2} \left(\|Ax - b\|^2 + \lambda \sum_i \sum_{j \in Y(i)} (x_i - x_j)^2 \right) \quad (6.11)$$

where $Y(i)$ is the set of indices of 3-neighborhood pixels of x_i ; if x_i is even pixel then $Y(i)$ contains indices of neighborhood pixels in directions parallel to x and y coordinate axes, else, when x_i is odd pixel, it contains indices of neighborhood pixels in direction parallel to z axis. The parameter $\lambda > 0$ is the balancing parameter between data fitting and smoothing terms. The first term is called data fitting and the second one smooth regularization [Luk12].

The energy function is the fitness function considered for the minimization task in the Simulated Annealing (SA) algorithm. Pseudocode sequence 6.6 shows the steps of the simulated annealing algorithm.

Pseudocode sequence 6.6. Simulated Annealing algorithm [Luk12].

Parameters supplied by the user:

$T_{start} > 0$ {start temperature},

$T_{min} > 0$ {minimum temperature},

$T_{factor} \in (0, 1)$ {multiplicative factor for reducing the temperature},

$NoChgLimit \in \mathbb{N}$ {number of required successively reduced temperature levels without accepted change attempts}.

Initial settings:

$x = [0, 0, \dots, 0]^T$, $T = T_{start}$,

$NoChg = 0$, $E_{current} = E(x)$.

while ($T \geq T_{min}$) \wedge ($NoChg \leq NoChgLimit$)

 for $i = 1$ to $sizeof(x)$,

 choose a random position j in the vector x ;

$x' = x$; $x'_j = 1 - x_j$;

$E_{attempt} = E(x')$;

$\Delta E = E_{attempt} - E_{current}$;

$z = rand(U(0, 1))$;

 if ($\Delta E < 0$) \vee ($\text{Exp}(-\Delta E/T) > z$), then

$x = x'$; {accept change}

$E_{current} = E_{attempt}$;

$NoChg = 0$;

 end if

 end for

$T = T * T_{factor}$;

$NoChg = NoChg + 1$;

end while.

For the experimental part we consider two test images of size $26 \times 26 \times 26$, having 4056 pixels. Images used for the test part are presented in Figure 6.28.

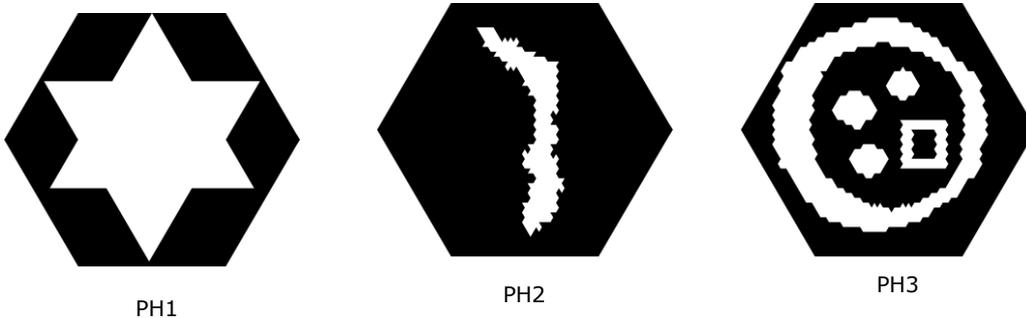


Figure 6.28. Images used in tests.

For the simulated annealing algorithm the following parameters were used $T_{start}=4$, $T_{min}=0.001$, $T_{factor}=0.97$, $NoChgLimit=10$ and $\lambda=5$.

That algorithm was written in MathLab. Smooth regularization term also was used, and the pixel error using six projections was usually below 2 percentages for any tested image. Figure 6.29 shows the reconstruction results for the two test images from Figure 6.28. The values under the reconstructed results show the number of reconstructed pixel error. Also the pixel errors are shown.

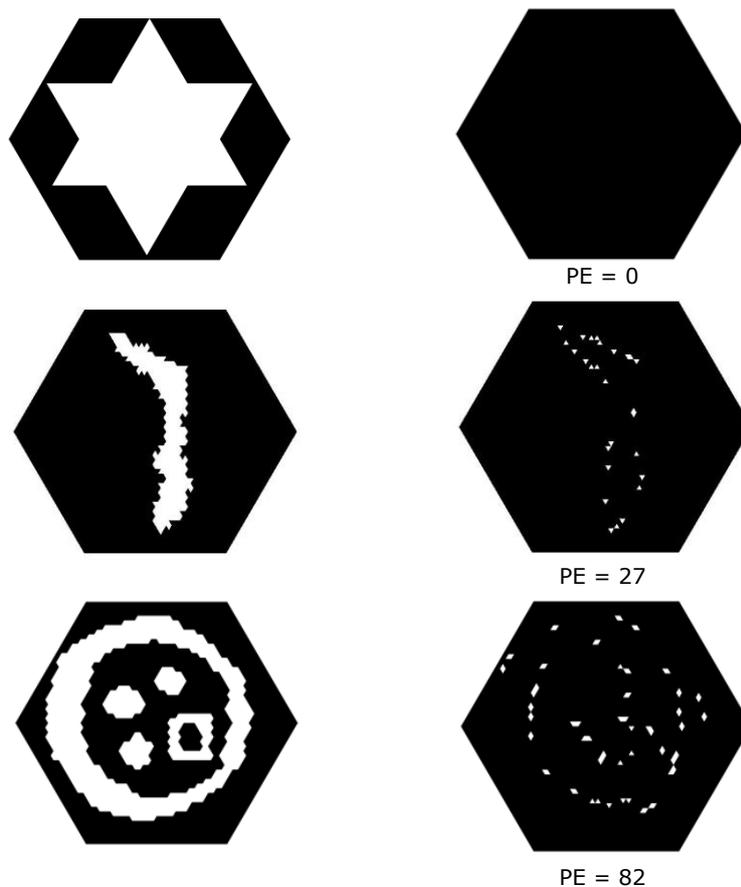


Figure 6.29. Reconstruction results for the three test images using simulated annealing.

Results show that using six projections the values are very close to 100%. These very small errors for the reconstructed images were due to the smooth regularization term that helped a lot to obtain solution with connected parts (without random noise). However the algorithm was used to do numerous number of iterations (usually it did more than a half million iterations) and thus its runtime was at least 15 hours, but usually was about or more than a day to obtain these results.

For the memetic algorithm the following parameters were used $np=\{50, 75, 100\}$ and $ng=\{50, 100, 150\}$. The tests were run 10 times for each test image. Figure 6.30 shows the best reconstruction results obtained for the test images from

Figure 6.28. The values under the reconstructed results show the number of reconstructed pixel error value. The memetic algorithm was written in C#.

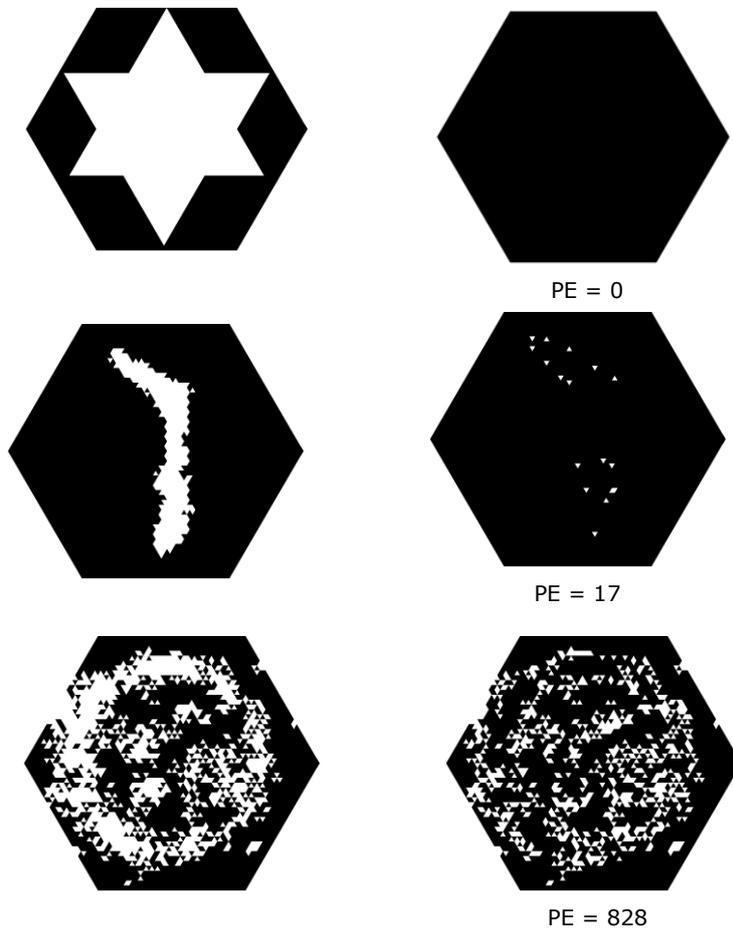


Figure 6.30. Reconstruction results for the three test images using memetic algorithm.

Table 6.7 shows the reconstruction results illustrated in Figure 6.29 and Figure 6.30 using the simulated annealing and the memetic algorithm, respectively. The error ϵ and the percentage, in which the reconstructions correspond to the original image, are computed. Also the pixel error PE is shown.

Table 6.7. Reconstruction results for the test images

Image	Simulated annealing			Memetic algorithm		
	PE	ϵ	$p\epsilon$	PE	ϵ	$p\epsilon$
PH1	0	0,0000	100%	0	0,0000	100%
PH2	27	0,6657	99,33%	17	0,4191	99,58%
PH3	82	2,0216	97,97%	828	20,4142	79,60%

Figure 6.31 shows the percentage, in which the reconstructions correspond to the original image using the simulated annealing and the memetic algorithm. The

blue columns correspond to the percentage obtained using the simulated annealing algorithm, for each image. The red columns, correspond to the percentage obtained using the memetic algorithm for each of the three images.

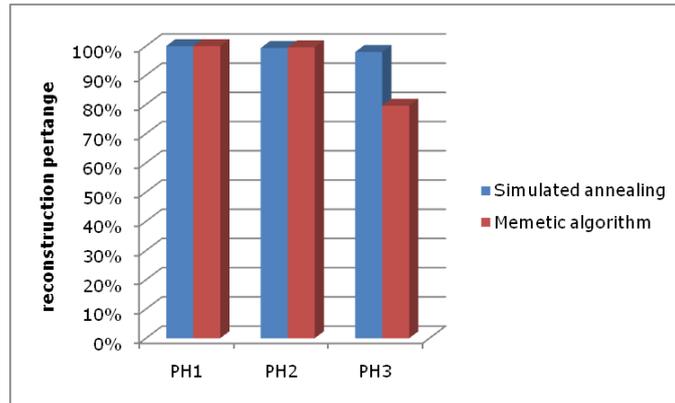


Figure 6.31. Reconstruction results for the three test images.

The considered fitness function for the memetic algorithm computes the difference between the measured projections and the projections of the reconstructed image. This function does not take into consideration any prior knowledge about the images, so the smoothness of images is not considered. This leads to not so good reconstruction in the case of images like PH3 from Figure 6.28. Our experiments show that the more complex test image PH3, have unacceptable bad reconstructions using the memetic algorithm, see Figure 6.29.

However for the simple test images that satisfy some topological or geometrical properties, like PH1 and PH2, the reconstruction results using the memetic algorithm are comparable or even better then reconstruction results obtained using the simulated annealing algorithm. Also in terms of computation time the memetic algorithm gave better results than the simulated annealing algorithm. For example for PH2 the results obtained with memetic algorithm, shown in Figure 6.30, have a computation time of 4137 seconds; meanwhile the results obtained with simulated annealing, see Figure 6.29, have a computation time of approximate 123h which means 442800 seconds.

Experimental results show that our memetic algorithm is much faster and its reconstruction results are comparable with the results obtained by simulated annealing in test cases which satisfy some topological or geometrical properties.

A comparison with other algorithms is often difficult, as each algorithm makes different assumptions on the class of images, the detector settings or other prior knowledge.

6.6. CONCLUSIONS

Discrete tomography for images on the triangular grid is a difficult task to solve and requires a lot of time (with respect to the size of the images) to reconstruct them without any a priori knowledge of the image. Actually we have used some assumptions about their compactness, but not in the fitness value. In

this chapter we have presented memetic algorithms on the triangular grid for hexagon-shaped binary images reconstruction with six projection angles. Experiments were performed using different type and different size hexagon-shaped images. Experimental results show that the proposed method is robust enough when images that satisfy topological and geometric constraints are used. The average pixel error is lower for hv-convex polyominoes than for generic images, because the additional constraint of convexity. It has to be noted that the considered generic images are pretty difficult to be reconstructed since they do not respect any constraints and are pretty different one from each other. In our experiments the guided mutation helped a lot to improve the quality of our reconstructed images. Results also show that computation time necessary to reconstruct a given hexagon-shaped image, scales in a quasi-linear way, because of the population and generation size. Regarding the runtime, the gap of our algorithm is the computation of all the switching components implied in the hill climb operators. Test results show that our memetic algorithm is much faster and its reconstruction results are comparable with the results obtained by simulated annealing for the test cases which satisfy some topological or geometrical properties.

7. ITERATIVE RECONSTRUCTION ALGORITHM BASED ON MINIMUM COST MAXIMUM FLOW ALGORITHM ON THE TRIANGULAR GRID

7.1. Introduction	103
7.2. Algorithm description.....	104
7.2.1. Projection angles selection.....	105
7.2.2. Pixel weight function on the triangular grid	105
7.2.3. Stopping criterion.....	106
7.3. Experimental results	107
7.3.1. HV-convex polyominoes	108
7.3.2. Generic regular hexagon-shaped images	112
7.3.3. Non-convex regular hexagon shaped images	115
7.4. Conclusions	125

Binary tomography is one special case of discrete tomography, where the function that represents the object or the image can take only two values: 0 or 1. So, the main aim of binary tomography is to reconstruct a binary image, where the object is represented in white and the background in black, using projections of the image from few different projection angles. In this chapter we propose an iterative reconstruction algorithm based on the minimum cost maximum flow algorithm on triangular grid. In each iteration a new triplet of projection angles are selected and the capacity of each edge of the associated graph is computed. Also the cost associated to the minimum cost maximum flow problem is computed using the reconstruction from the previous iteration. Then the minimum cost maximum flow problem, which incorporates the obtained data, is solved.

7.1. INTRODUCTION

In Chapter 5 a minimum cost maximum flow algorithm for resolving the reconstruction problem for binary hexagon-shaped images on triangular grid using three projection directions was presented. In the proposed algorithm two main projections with restrictions imposed by the third projection is used. Unfortunately there is no generalization of the minimum cost maximum flow algorithm for the case of more than two projections. In this chapter an iterative approach to solve the problem of hexagon-shaped images on triangular grid from more than three projections is presented. The proposed approach uses, in each iteration, the minimum cost maximum flow algorithm presented in chapter 5, each time for a different triplet of projection angles. The reconstruction problem that has to be solved is the following:

Problem 5.3: Let $\theta_1, \dots, \theta_d$ be three different projection angles and p_1, \dots, p_d the measured projection data for the angles $\theta_1, \dots, \theta_d$. Construct a hexagon-shaped image A such that $P_k = p_k$ for $k=1, \dots, d$.

7.2. ALGORITHM DESCRIPTION

The proposed algorithm computes the reconstruction from projections by solving a series of three projection reconstructions using the maximum flow minimum cost algorithm proposed in chapter 5. The three projections are measured along three different projection angles. The proposed algorithm finds an appropriate solution for the reconstruction problem. In each iteration, a triplet of three projection angles is selected, and then the reconstruction problem for these three projections is solved using a minimum cost maximum flow model that corresponds to those three projection angles. The obtained solution satisfies the projections in the first two directions exactly and some of the projections in the third direction also (especially those that have 0 value). This solution is used to determine the pixel weight for the network that corresponds to other triplet of directions. In this way prior knowledge can be incorporated in the reconstruction.

In the first step a start solution is computed on the triangular grid as a $m \times m \times m$ hexagon-shaped binary image. At start the algorithm has no reconstruction solution from a previous step. So there are no values to be used for the generation of the pixel weight from the first network. One of the solutions that can be considered in the computation is to use an algorithm based on Ryser's theory [Rys57][Rys60][Rys63] for reconstructing the original image as we proposed in [Nag13b]. The other solution is to use the same value, for example zero, for each pixel. In the proposed approach, we consider an initial image with pixel values zero to compute the start solution, which yields a black image.

In the next step the total number of white pixels is computed

$$T = \left(\sum_{k=1}^d \sum_{i=1}^{|L_k|} p_{ki} \right) / 3 \text{ where } d \text{ is the number of the considered projection angles, } p_k$$

are the measured projection data for the considered projection angles and $|L_k|$ is the number of samples over a sampling direction k . This value is used in the calculation of the total flow in each minimum cost maximum flow problem considered in each iteration of the algorithm.

Next, the algorithm enters in a loop. In each iteration of the loop, a new set of $(\theta_1, \theta_2, \theta_3)$ projection angles and the measured projections for these angles are selected. Section 7.2.1 describes in detail the way in which the angles are selected. Subsequently the pixel weight W is computed using the reconstructed image obtained in the previous iteration. Section 7.2.2 describes in detail the computation of the pixel weight. Using the projection associated to the selected projection angles, the pixel weight values and the total flow the minimum cost maximum flow problem can be solved, obtaining a new reconstruction of the image. The stopping condition depends on the value of the projections obtained for the reconstructed image and on the number of iterations. These stopping criteria are presented in 7.2.3.

A similar approach was presented in [Gri98] and extended in [Bat06b][Bat07a][Bat08] with a smoothness assumption in the context of reconstructing binary image represented on square grid from discrete and continuous X-rays. Our approach modifies the algorithm, for the case of binary

hexagon-shaped images represented on triangular grid. This is a new algorithm which uses the features and properties of the triangular grid, although it is very similar to the algorithm proposed by Batenburg.

In the following sections a detailed description of each of the proposed algorithm steps, from Pseudocode sequence 7.1, is given for the case of binary hexagon-shaped images represented on triangular grid.

Pseudocode sequence 7.1. Basic steps of the iterative reconstruction algorithm

Computing the start solution on the triangular grid

Computing the total number of white pixels

Generation = 1

While (condition is not met)

Generation = Generation + 1

Select a new set of projection angles

Compute the triangle pixel weight values using the reconstruction from the previous iteration

Compute the total number of white pixels

Generate a new solution using the minimum cost maximum flow algorithm for triangular grid

End While

7.2.1. PROJECTION ANGLES SELECTION

In the proposed algorithm the iterations are made using possibilities of three combinations. In order to select a different triplet $(\theta_1, \theta_2, \theta_3)$ for the projection angles in each iteration of the algorithm, the steps to follow are: first select two different directions, and then choose randomly the third direction from the ones not yet chosen. For choosing the two projection angles that are used to solve the minimum cost maximum flow problem, the pair of directions have to satisfy the constraint that the difference between the two angles is set to $\frac{2\pi}{3}$ in all the experimental results.

There can be other, may be better, direction selection steps used for the first two projection angles. In the case of using few projection angles, for e.g. $d=3,4,5,6$, all the possible combinations of directions can be used in order to avoid the usage of the same pair of projection angles, which could influence in a negative way the reconstruction.

7.2.2. PIXEL WEIGHT FUNCTION ON THE TRIANGULAR GRID

In order to incorporate preferences for smoothness of the image the pixel weight of each pixel in the image is computed. This pixel weight depends on the pixel value obtained in the previous reconstruction and also on the values of the neighborhood pixels.

In [Bat06b], the author introduces the concept of local weight function in the case of square grid image representation. For the computation of the pixel weights in our approach the definitions and functions introduced in

[Bat06b][Bat06a] are considered. Based on his approach, the construction of a weight function in the case of triangular grid image representation taking into consideration the twelve 3-neighbors of a pixel on the triangular grid is presented.

Let H_a be the neighborhood of a pixel $a_{i,j,k} \in A$, defined as $H_a = \{a_{x,y,z} \in A: I - r \leq x \leq i + r, j - r \leq y \leq j + r, k - r \leq z \leq k + r\}$ where r represents the neighborhood radius and $x, z \in [-m+1, m]$, $y \in [-m, m-1]$ and $x+y+z = 0$ or $x+y+z = 1$, and s_a be the number of pixels, from the neighborhood, that has the same color with $a_{i,j,k}$, $s_a = |\{ax,y,z \in H_a: A(ax,y,z) = A(a_{i,j,k})\}|$ [Bat06b]. In the case that the pixel $a_{i,j,k}$ is situated near the boundary, the neighborhood contains fewer pixels.

Considering a pixel $a_{i,j,k} \in A$ of a hexagon-shaped image the weight of the pixel is defined as:

$$W(a_{i,j,k}) = \left(A(a_{i,j,k}) - \frac{1}{2} \right) \cdot g\left(\frac{s_a}{|H_a|} \right) \cdot 10000 \quad (7.1)$$

where A is the reconstructed image from the previous iteration, and $g\left(\frac{s_a}{|H_a|} \right)$ is the weight function defined as:

$$g\left(\frac{s_a}{|H_a|} \right) = \begin{cases} 1 & , \frac{s_a}{|H_a|} \leq 0.65 \\ 4 \frac{s_a}{|H_a|} & , 0.65 < \frac{s_a}{|H_a|} < 1 \\ 9 & , \frac{s_a}{|H_a|} = 1 \end{cases} \quad (7.2)$$

The weight function depends on the pixel neighborhood H_a computed using the twelve 3-neighbors of the pixel $a_{i,j,k}$ and the number of pixels from the neighborhood that has the same color with $a_{i,j,k}$, s_a .

Considering the weight equal to 1 there is no smoothness incorporated in the reconstruction. The preference for maintaining the pixel value in the next reconstruction is very strong when all the neighbors of the pixel have the same color as the pixel itself. This preference is smaller as the neighborhood contains fewer pixels with the same color as the considered one.

In the experimental part, the neighborhood radius is set at $r = 5$ for the first 50 iterations, and after that at $r = 1$. Setting the neighborhood to a small value implies that the fine details are also reconstructed.

7.2.3. STOPPING CRITERION

In order to evaluate how similar the reconstructed image is with the original one, the distance between the measured projections and the projections of the reconstructed image is computed as the stopping condition of the algorithm. This measure is specified in the theory that is not a good measure because there can be two images with similar projections even if they are very different. But in the case of smooth images, considering more than three projections, this is not a problem. For

computing the distance between the projections of the reconstructed image and the measured projections as defined in equation (7.3), first the projections of the reconstructed image are computed for all the considered projection angles. Then the norm l_1 of the error with respect to the measured projections is computed as defined in equation (7.4):

$$\mathfrak{F}(A) = \sum_{k=1}^d l_1(P_k, p_k) \quad (7.3)$$

$$l_1(P_k, p_k) = \sum_{i=1}^{|L_k|} |P_{k,i} - p_{k,i}| \quad (7.4)$$

where p_k is the measured projection along the projection angle θ_k and $P_k=(P_{k,1}, \dots, P_{k,|L_k|})$ is the projection of the reconstructed image A calculated along the projection angle θ_k .

The algorithm stops when for the reconstructed image has exactly the measured projections. If there is no improvement for the distance between the reconstructed image projections and the measured projections in the last $N_i \in \mathbb{N}_{>0}$ iterations the algorithm stops. We used $N_i=200$ for our experiments. We determined the value of N_i experimentally. The algorithm always stops after a maximum number $M_i \in \mathbb{N}_{>0}$ of iterations. We used $M_i=2000$ for our experiments.

7.3. EXPERIMENTAL RESULTS

Regarding the proposed iterative method, we tested its benefits in terms of correctness, robustness and runtime. For the experimental session different databases of hexagon-shaped images were used. From the robustness point of view we will show that the presented approach produces stable results even if we use only six projection angles. The reconstructed solutions are obtained in a relatively good time even in the case of images with a big number of pixels.

The implementation of the algorithm is made in C#. For solving the minimum cost maximum flow problem we use the MCF solver [Ber04][Ber94]. This project is public available for noncommercial use. All the results from this sections were obtained using a system equipped with Intel(R) Core(TM) i5-2500 (3.30 GHz), 4 GB of RAM and the Windows 8.

Although, in real cases, only the projections of the input image will be available, in practice, artificial test images are used. In order to evaluate the algorithm the pixel difference between the original image and the reconstructed image is computed. The value is normalized by the number of the pixels of the image and multiplied by 100 to obtain the pixel error in percentage (%).

For the moment, no standard database of images represented on the triangular grid exists to compare different discrete tomography reconstruction algorithms. Anyway, to validate the algorithm two public available databases of binary images represented on the square grid [MPE][Pat] are used. Images from these databases are resampled to images on the triangular grid. After resampling the image on the triangular grid, a number of pixels are suppressed, in order to

obtain a hexagon-shaped image having triangular pixels. The dimension of the resampled images is $m \times m \times m$.

Considering different classes of images, different number of projections can be necessary to reconstruct correctly the images. There is no way to know in advance how many projections are necessary. That is why, we used a fixed number of projection angles, $\theta \in \left\{0, \frac{\pi}{6}, \frac{\pi}{3}, \frac{\pi}{2}, \frac{2\pi}{3}, \frac{2\pi}{6}\right\}$. Sets of three projection angles are used for the reconstructions.

For all the tests we include the average results for the runs were the test results are perfect and the average results of the reconstruction for which the test is successful. A reconstruction is considered perfect if the projections of the reconstructed image are the same with the measured projections. A reconstruction is considered successful if the distance between the reconstructed image projections and the measured projections is less than the average difference between the reconstructed image projections and the measured projection multiplied by d , where d is the number of projection angles. This implies that the reconstructed image approximates very well the measured projection values. We use the number of projection angles d , in the definition of successful reconstruction, because pixel errors result in errors in each of the projections. When the number of projections is large, one pixel error in the reconstruction results in a large distance from the measured projections.

7.3.1. HV-CONVEX POLYOMINOES

For our first test class, we considered 20 hv-convex polyominoes. We performed 100 runs using a varying number of projection angles. The hexagon-shaped images from this class have the size $26 \times 26 \times 26$, which means 4056 pixels.

Figure 7.1 shows 10 samples of hexagon-shaped images that contain hv-convex polyominoes that we generated. Results show that using 4 projection directions all our 20 hv-convex polyominoes were perfectly reconstructed in only some minutes. Using 3 projection directions is not enough for all the 20 hv-convex polyominoes to be perfectly reconstructed.

Table 7.1 shows the test results for the hexagon-shaped test images from Figure 7.1. The first column contains the considered test image. The second column contains the number of considered projection angles. The next two columns contain the average of perfect and successful reconstructions. Then, the next two columns contain the average projection error and the average pixel error, meanwhile the last two columns contain the number of average iterations and the number of average runtime measured in seconds.

Figure 7.2 shows the test results for the 10 hv-convex polyominoes from Figure 7.1. The blue column illustrate the results obtained using 3 projection angles, meanwhile the red columns indicate the reconstruction results using 4 projection angles.

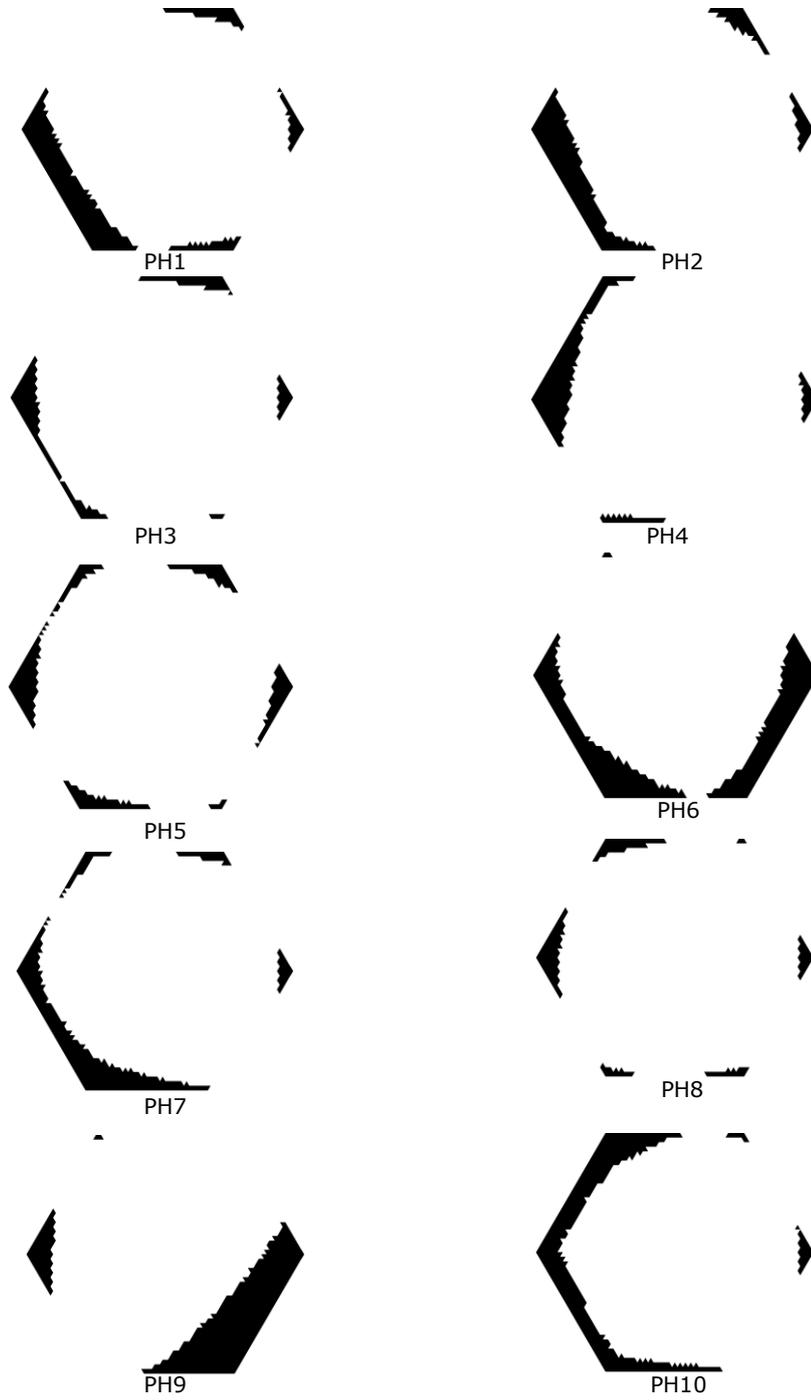


Figure 7.1. Hv-convex polyominoes with dimension of 26x26x26.

Table 7.1. Reconstruction results for the hv-convex polyominoes case.

Image	d	Perfect reconstruction	Successful reconstruction	Projection error	Pixel error	Iteration	Runtime (s)
PH1	3	18	100	20,52	7,08	385	615
	4	100	100	0,00	0,00	14	18
PH2	3	100	100	0,00	0,00	57	99
	4	100	100	0,00	0,00	8	10
PH3	3	100	100	0,00	0,00	1	2
	4	100	100	0,00	0,00	1	2
PH4	3	100	100	0,00	0,00	9	15
	4	100	100	0,00	0,00	1	2
PH5	3	4	100	21,9	7,30	339	552
	4	100	100	0,00	0,00	4	5
PH6	3	100	100	0,00	0,00	7	12
	4	100	100	0,00	0,00	4	5
PH7	3	2	100	28,32	10,46	359	584
	4	100	100	0,00	0,00	8	11
PH8	3	95	95	0,82	0,24	127	209
	4	100	100	0,00	0,00	9	12
PH9	3	100	100	0,00	0,00	44	76
	4	100	100	0,00	0,00	4	7
PH10	3	100	100	0,00	0,00	28	46
	4	100	100	0,00	0,00	2	3

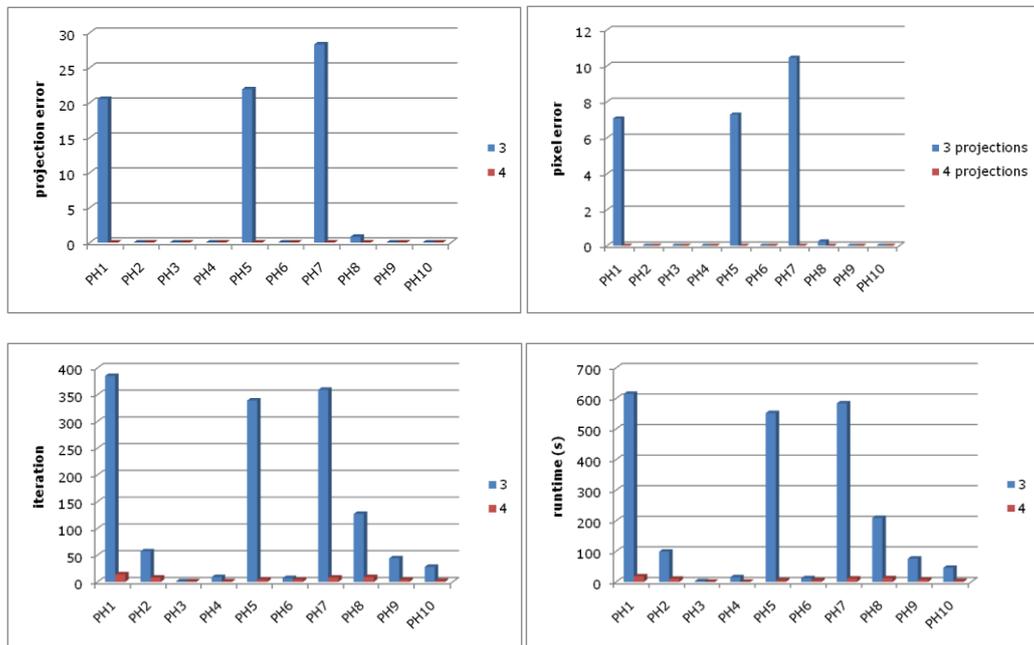


Figure 7.2. Results for the 10 hv-convex polyominoes.

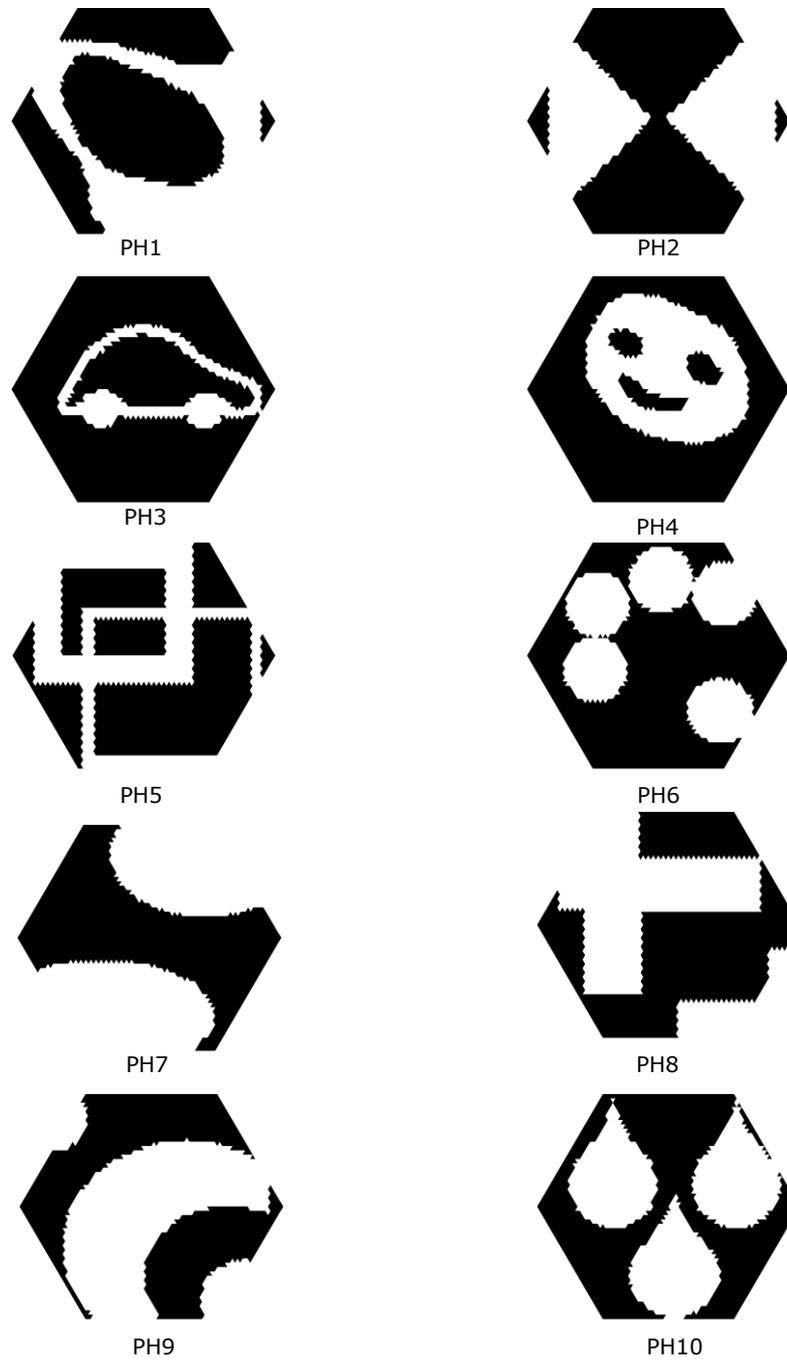


Figure 7.3. Generic regular hexagon-shaped images with dimension of 26x26x26.

The test results show that if using more than three projections, the algorithm every time converges to a hexagon-shaped image which perfectly satisfies

the measured projections. Only when we use three projections, in some cases for example for PH1, PH5, PH7 and PH8, the difference between reconstructed image and the original one was very small. Using four projection angles all the images were perfectly reconstructed.

7.3.2. GENERIC REGULAR HEXAGON-SHAPED IMAGES

The secondly considered test class is of 20 generic hexagon-shaped images. In this case, we resampled some generic images represented on the square grid. The test images were resampled from the public available databases [Pat]. The considered hexagon-shaped images for our tests have the size $m=26$.

Figure 7.3 shows 10 examples of images that we used in our tests. For each of our images we performed 100 runs.

Table 7.2 and Table 7.3 show the test results. The first column contains the considered image. The second and third column contain the image dimension and the number of projection angles used for the test. Column four, five, six and seven contain the average perfect and successful runs, and the average projection and pixel errors. The eighth and ninth columns contain the number of iterations and the runtime in seconds.

Table 7.2. Reconstruction results for PH1-PH5 generic regular hexagon-shaped images.

Image	d	Perfect reconstruction	Successful reconstruction	Projection error	Pixel error	Iteration	Runtime (s)
PH1	3	0	100	581	317	352	564
	4	0	100	162	85	409	451
	5	0	100	192	98	409	434
	6	0	100	159	76	406	415
PH2	3	100	100	0,0	0,0	1	2
	4	100	100	0,0	0,0	1	2
PH3	3	0	100	557	494	227	361
	4	0	100	437	335	340	369
	5	0	100	400	269	383	385
	6	0	100	220	127	414	422
PH4	3	0	100	1035	535	415	892
	4	0	100	188	103	573	619
	5	0	100	379	208	477	478
	6	0	100	181	93	406	414
PH5	3	0	100	81	23	402	595
	4	0	100	85	25	405	443
	5	0	100	192	90	406	409
	6	0	100	108	38	405	413

Figure 7.4 shows the test results for the 10 hv-convex polyominoes from Figure 7.3. The blue color column illustrate the results obtained using 3 projection angles, the red color columns indicate the reconstruction results using 4 projection angles, the green color column shows the reconstruction results obtained using 5 projection angles and the mauve color column illustrated the results for 6 projection angles. It can be seen that not all the time using more projection angles the reconstruction results decrease. This is because of the high number of iterations to which is associated a triplet of projection angles for which the first two projections are not orthogonal enough. Also, we saw from our test results that for some cases,

better reconstruction results can be obtained by using in each iteration triplets of projection angles in which the difference between the first two projection angles is orthogonal enough, as the case of 4 projections.

Table 7.3. Reconstruction results for PH6-PH10 generic regular hexagon-shaped images.

Image	d	Perfect reconstruction	Successful reconstruction	Projection error	Pixel error	Iteration	Runtime (s)
PH6	3	0	100	440	226	620	985
	4	0	100	218	124	468	517
	5	0	100	343	213	417	419
	6	0	100	212	110	425	433
PH7	3	0	100	55	23	390	615
	4	0	100	53	18	377	411
	5	0	100	115	50	397	399
PH8	3	100	100	0,0	0,0	100	168
	4	100	100	0,0	0,0	95	111
PH9	3	0	100	176	84	528	834
	4	0	100	130	56	406	442
	5	0	100	148	62	395	397
PH10	6	0	100	114	49	395	406
	3	0	100	958	478	435	437
	4	0	100	197	108	418	493
	5	0	100	212	107	401	379
	6	29	100	18	6	359	395

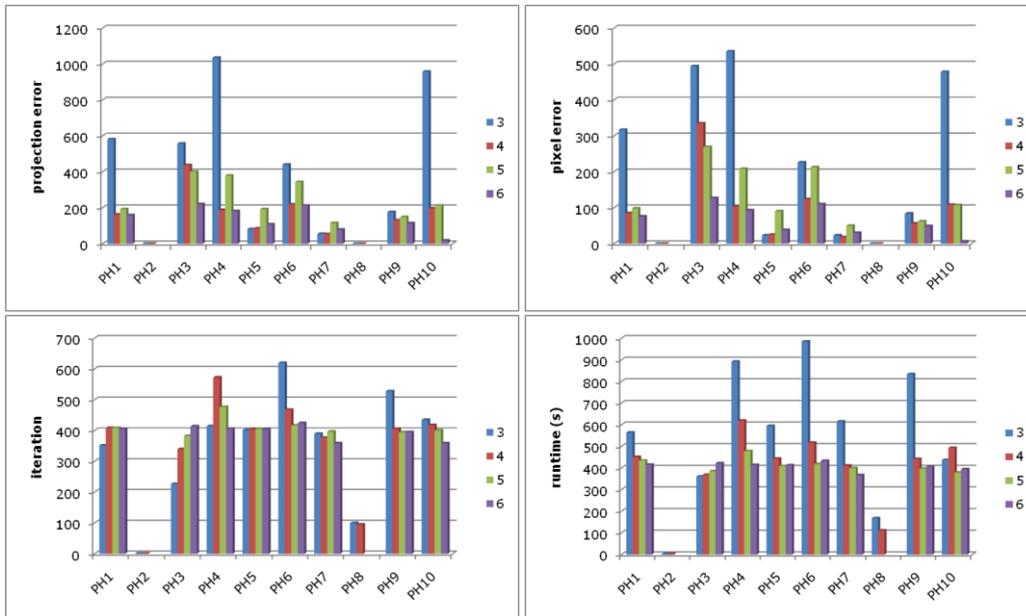


Figure 7.4. Reconstruction results for the 10 generic regular hexagon-shaped images.

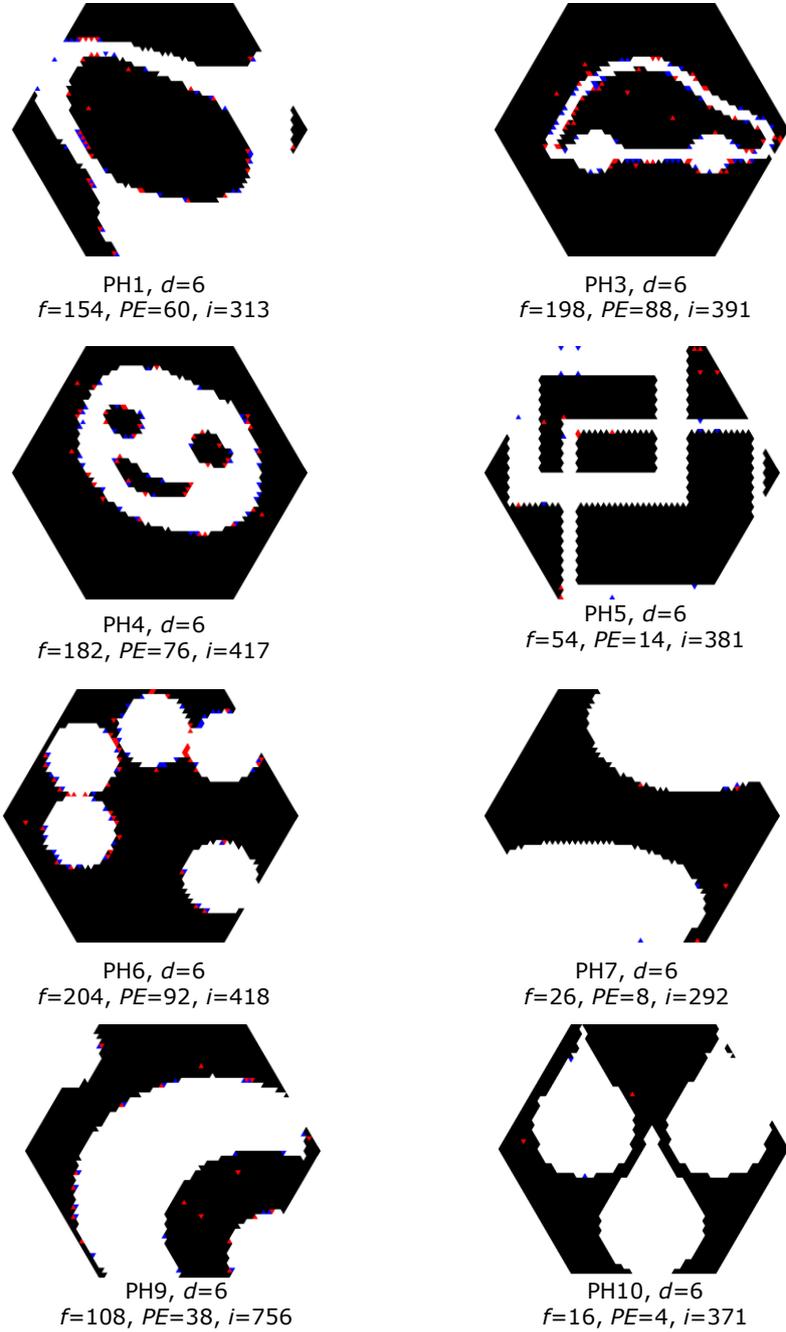


Figure 7.5. Generic hexagon-shaped images reconstruction examples.

Figure 7.5 shows reconstruction examples for images the test images. Errors are shown by color: missing pixels are blue, additional pixels of the reconstructed

image are red; f means fitness function value, PE means pixel error (sum of blue and red pixels) and i means number of iterations made for obtaining the solution.

The results show that using enough projection angles the binary hexagon-shaped images can be perfectly reconstructed, see for example image PH2, PH8 and in some cases PH10. Not for all images the six considered projection angles are enough. But even if the solutions are not perfect, a successful reconstruction was obtained in all the cases, the shape of the object from the images being clearly distinguished. In conclusions even choosing a reasonable number of projection angles, the proposed algorithm leads us to robust solutions that are very close to the original images.

7.3.3. NON-CONVEX REGULAR HEXAGON SHAPED IMAGES

For the third tests class we resampled, at different dimensions, five images from the [MPE] database. All the hexagon-shaped test images are resampled from square images of 100×100 at a dimensions of $m = \{4, 8, 13, 26, 50\}$. In our tests, for each of the considered test images we used a number of 100 runs.

Figure 7.6 show the considered non-convex regular hexagon-shaped images at a dimension of $50 \times 50 \times 50$.

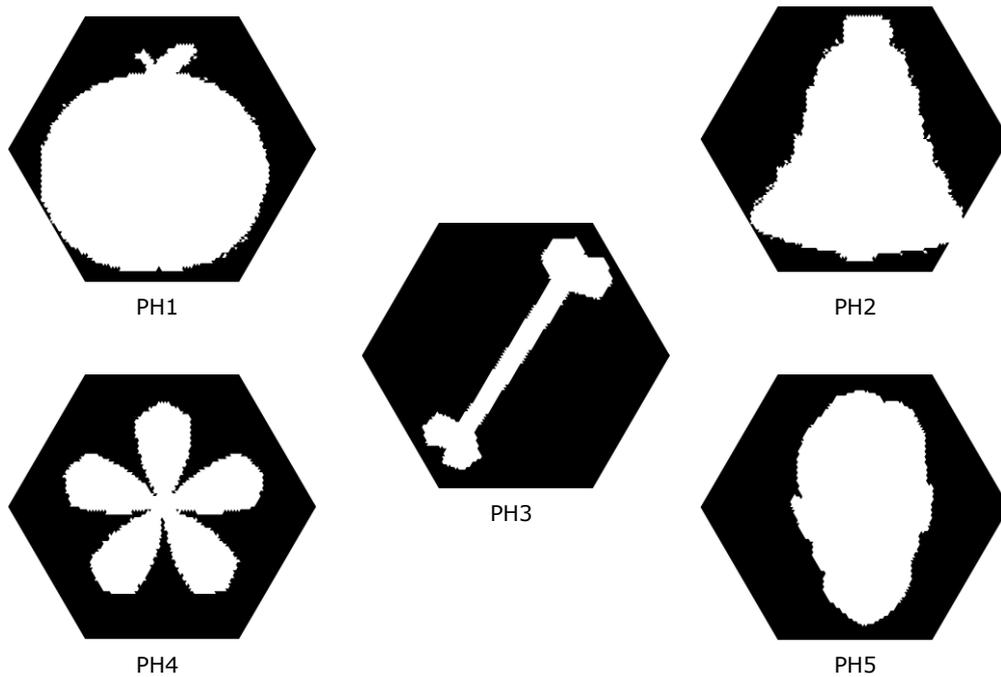


Figure 7.6. Non-convex hexagon-shaped test images of dimension $50 \times 50 \times 50$.

Table 7.4 shows the test results for the images from Figure 7.6. The first and second columns contain the considered test image and the number of projection angles. Then, the next four columns contain the average of perfect and successful reconstructions and the average projection and pixel error. Columns seven and eight

contain the number of average iterations and the number of average runtime measured in seconds.

Table 7.4. Reconstruction results for non-convex hexagon-shaped images with dimension $4 \times 4 \times 4$.

Image	d	Perfect reconstruction	Successful reconstruction	Projection error	Pixel error	Iteration	Runtime (s)
PH1	3	100	100	0,0	0,0	1	0
PH2	3	100	100	0,0	0,0	1	0
PH3	3	71	84	2	1	168	8
	4	100	100	0,0	0,0	36	2
PH4	3	100	100	0,0	0,0	16	0
PH5	3	100	100	0,0	0,0	3	0

Figure 7.7 shows the charts that contain the reconstruction results for the non-convex hexagon-shaped images at a dimension of $4 \times 4 \times 4$. The columns indicate the number of projections used for the reconstruction. Results show that the reconstruction values become lower as the number of projections increase.

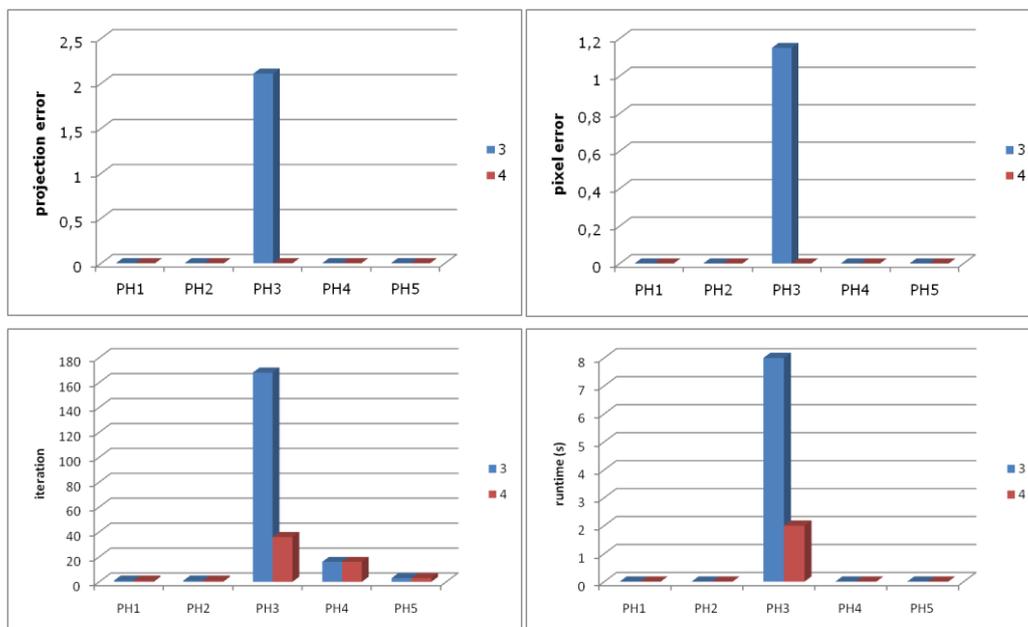


Figure 7.7. Reconstruction results for non-convex hexagon-shaped images with dimension $4 \times 4 \times 4$.

Table 7.5 shows the reconstruction results for the resampled images at a dimension of $8 \times 8 \times 8$. The first and second columns contain the considered test image and the number of projection angles. The fourth-eighth columns contain the average of perfect and successful reconstructions and the average projection and pixel error. The next two columns contain the number of average iterations and the number of average runtime measured in seconds.

Table 7.5. Reconstruction results for non-convex hexagon-shaped images with dimension $8 \times 8 \times 8$.

Image	d	Perfect reconstruction	Successful reconstruction	Projection error	Pixel error	Iteration	Runtime (s)
PH1	3	100	100	0,0	0,0	1	0
	4	100	100	0,0	0,0	1	0
	5	100	100	0,0	0,0	1	0
PH2	3	1	100	13	6	301	38
	4	100	100	0,0	0,0	46	7
	5	51	92	6	3	182	25
PH3	3	0	100	50	48	208	26
	4	0	100	42	27	328	50
	5	0	100	34	15	350	48
PH4	3	0	100	72	51	258	33
	4	0	100	57	41	291	44
	5	0	100	52	36	291	39
PH5	3	57	100	7	2	109	14
	4	100	100	0,0	0,0	18	3
	5	100	100	0,0	0,0	24	3

Figure 7.5 shows the charts that contain the reconstruction results associated to Table 7.5. The color of the columns indicates the number of projections used for the reconstruction 3, 4, 5 and 6, respectively. In the charts we can see that the projection and pixel errors become less, almost for all the cases, as the number of projections increases.

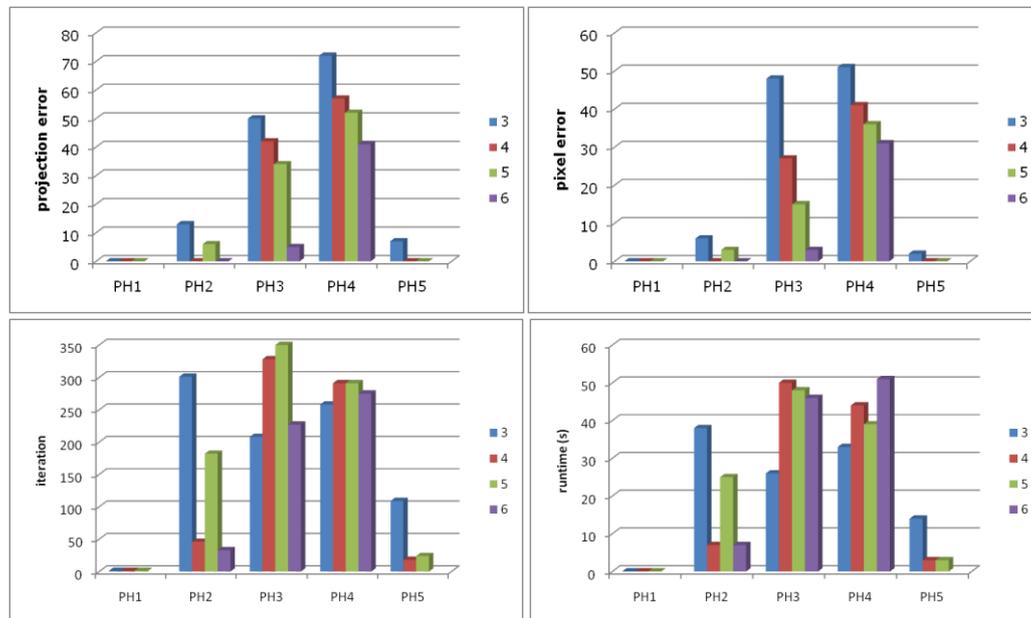
Figure 7.8. Reconstruction results for non-convex hexagon-shaped images with dimension $8 \times 8 \times 8$.

Table 7.6 shows the reconstruction results for the test images having a dimension of $13 \times 13 \times 13$. The first two columns contain the test images and the number of projection angles. The following two columns contain the average of perfect and successful reconstructions. The next columns contain the average reconstruction values for the projection and pixel error. The last two columns contain the number of average iterations and the number of average runtime measured in seconds. The reconstruction values are summarized in Figure 7.9, where the color of the columns corresponds to the number of projection angles, 3, 4, 5 and 6, respectively. Experimental results show that the reconstruction values depend a lot on the number of projections and especially on the order of the triplets of projection angles used in each iteration. Also the results show that for each type of image there is a lower value for the number of projection angles to compute an accurate solution. We remark that we have a case of image, the PH1, in which using only 3 projections, the algorithm obtained perfect reconstruction, meanwhile increasing the number of projection angles the reconstruction becomes worse. This is because of the considered order for the triplets of projection angles. Our test results show that the accurate of the reconstruction depends a lot on the order in which the triplets of projection angles are associated to each iteration. This is because the triplets for which the first two projection angles are not orthogonal enough worsen the reconstruction, in some cases. For the other considered images the reconstruction values linearly decrease with the increasing of the number of projection angles.

Table 7.6. Reconstruction results for non-convex hexagon-shaped images with dimension $13 \times 13 \times 13$.

Image	d	Perfect reconstruction	Successful reconstruction	Projection error	Pixel error	Iteration	Runtime (s)
PH1	3	98	98	0,3	0,1	68	19
	4	80	80	2	1	80	26
	5	1	100	19	7	299	82
	6	78	81	3	1	134	57
PH2	3	100	100	0,0	0,0	15	4
	4	100	100	0,0	0,0	15	5
	5	100	100	0,0	0,0	52	15
	6	100	100	0,0	0,0	43	21
PH3	3	0	100	105	63	375	99
	4	0	100	65	31	424	130
	5	0	100	51	21	368	100
	6	0	100	34	14	370	158
PH4	3	0	100	166	86	349	93
	4	0	100	116	66	373	114
	5	0	100	103	58	406	106
	6	0	100	89	49	381	153
PH5	3	100	100	0,0	0,0	4	1
	4	100	100	0,0	0,0	7	2
	5	100	100	0,0	0,0	10	3
	6	100	100	0,0	0,0	3	2

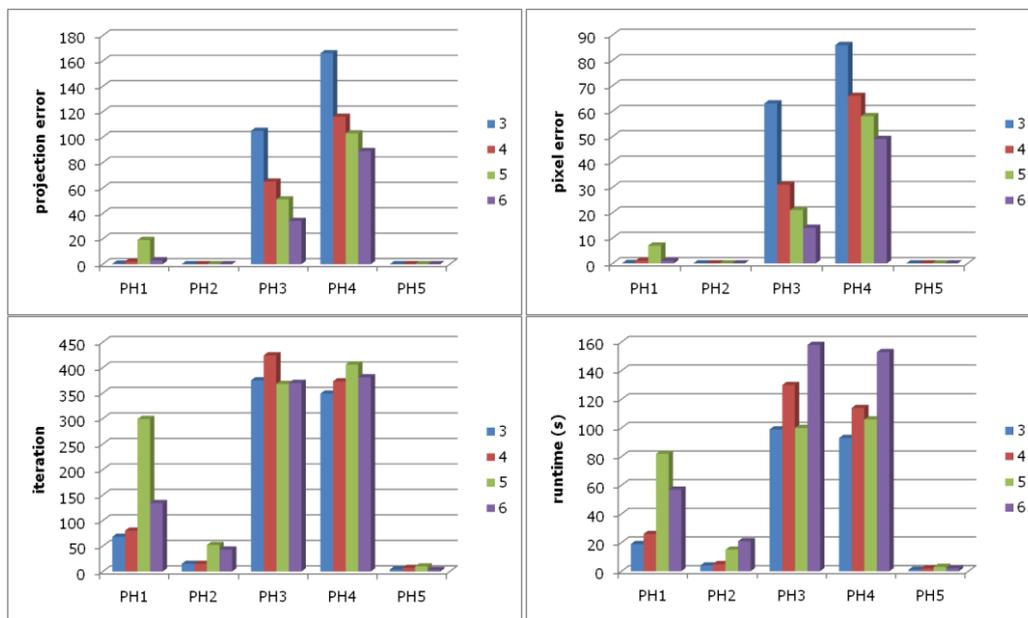


Figure 7.9. Reconstruction results for non-convex hexagon-shaped images with dimension $13 \times 13 \times 13$.

Figure 7.10 shows reconstruction examples for three test images of resolution $13 \times 13 \times 13$. Missing pixels are shown with blue, additional pixels of the reconstructed image are red. Information about the reconstructions are shown below the images: f means fitness function value, PE means pixel error (sum of blue and red pixels) and i means number of iterations made for obtaining the solution. Results show that the shape of the objects can be clearly distinguished.

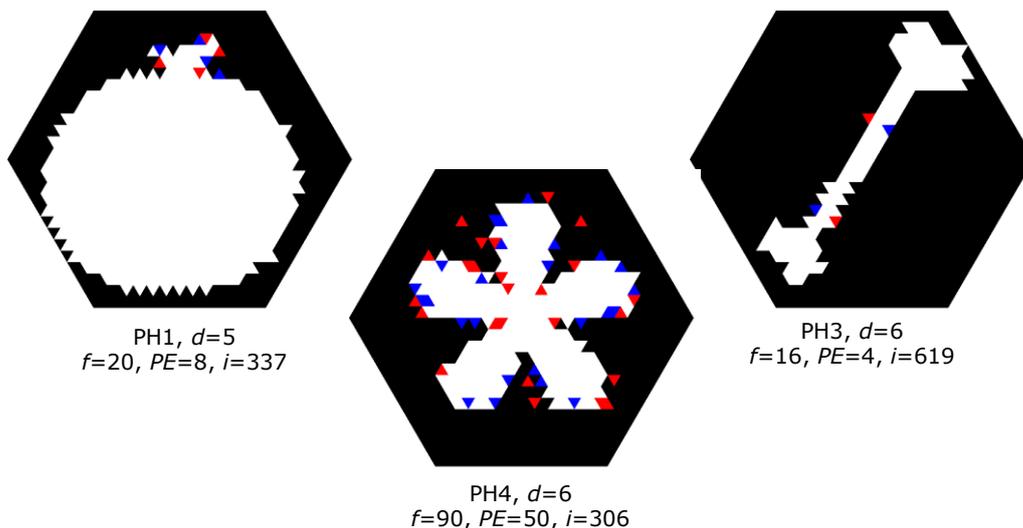


Figure 7.10. Reconstruction examples for non-convex hexagon-shaped images with dimension $13 \times 13 \times 13$.

Table 7.7 and Figure 7.11 summarize the reconstruction results for the test images having a dimension of $26 \times 26 \times 26$. The first two columns contain the test images and the number of projection angles. Column three and four contain the average of perfect and successful reconstructions. The next two columns contain the average reconstruction values for the projection and pixel error. The last two columns contain the number of average iterations and the number of average runtime measured in seconds. The colors of the columns from the charts of Figure 7.11 indicate the number of projection angles used in the reconstruction. Results show that using five projection angles the reconstruction values are the worst for almost all the cases. This is because of the number of triplets of projection angles, in which the difference the first two projections are not enough orthogonal one to each other, is high. Also using 4 projections instead of 6 can be advantageous, in some cases, because of the number of the triplets for which the first two projections are not enough orthogonal is lower than in 6 projection case. This leads for better results for test images like the one from PH1 and PH2.

Table 7.7. Reconstruction results for non-convex hexagon-shaped images with dimension $26 \times 26 \times 26$.

Image	d	Perfect reconstruction	Successful reconstruction	Projection error	Pixel error	Iteration	Runtime (s)
PH1	3	91	91	0,8	0,4	156	164
	4	95	95	0,4	0,2	138	159
	5	0	100	43	16	413	387
	6	8	100	14	5	356	380
PH2	3	0	100	27	10	351	356
	4	38	98	9	3	297	331
	5	0	100	79	30	374	353
	6	15	100	24	7	364	380
PH3	3	0	100	81	36	508	509
	4	0	100	87	39	443	487
	5	0	100	118	52	406	418
	6	0	100	89	41	401	416
PH4	3	0	100	808	427	441	436
	4	0	100	303	205	419	462
	5	0	100	292	170	435	442
	6	0	100	221	116	418	435
PH5	3	23	100	13	4	239	240
	4	84	84	2	1	139	159
	5	0	100	57	22	394	400
	6	84	88	3	1	195	208

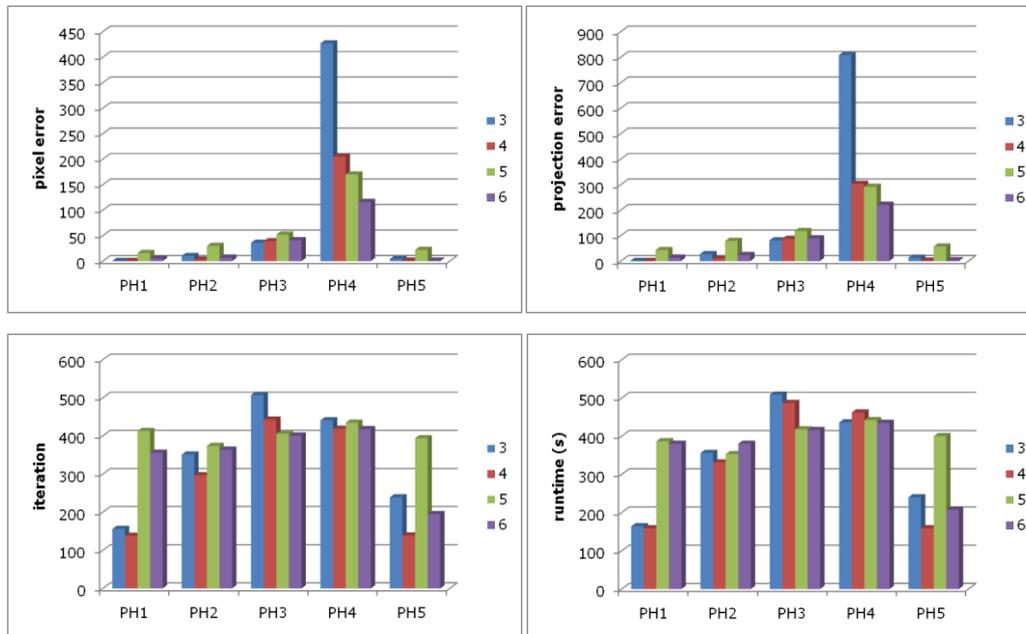


Figure 7.11. Reconstruction results for non-convex hexagon-shaped images with dimension $26 \times 26 \times 26$.

Figure 7.12 shows reconstruction examples for the test images resampled at dimension $26 \times 26 \times 26$. Missing pixels are shown with blue, additional pixels of the reconstructed image are red. Information about the reconstructions are shown below the images: f means fitness function value, PE means pixel error (sum of blue and red pixels) and i means number of iterations made for obtaining the solution. Reconstruction examples show that even if the projection and the pixel error are high the shape of the objects from the images can be clearly distinguished.

Table 7.8 shows the reconstruction results for the five test images resampled at dimension of $50 \times 50 \times 50$. The first and second columns contain the considered test image and the number of projection angles. The fourth-eighth columns contain the average of perfect and successful reconstructions and the average projection and pixel error. The next two columns contain the number of average iterations and the number of average runtime measured in seconds.

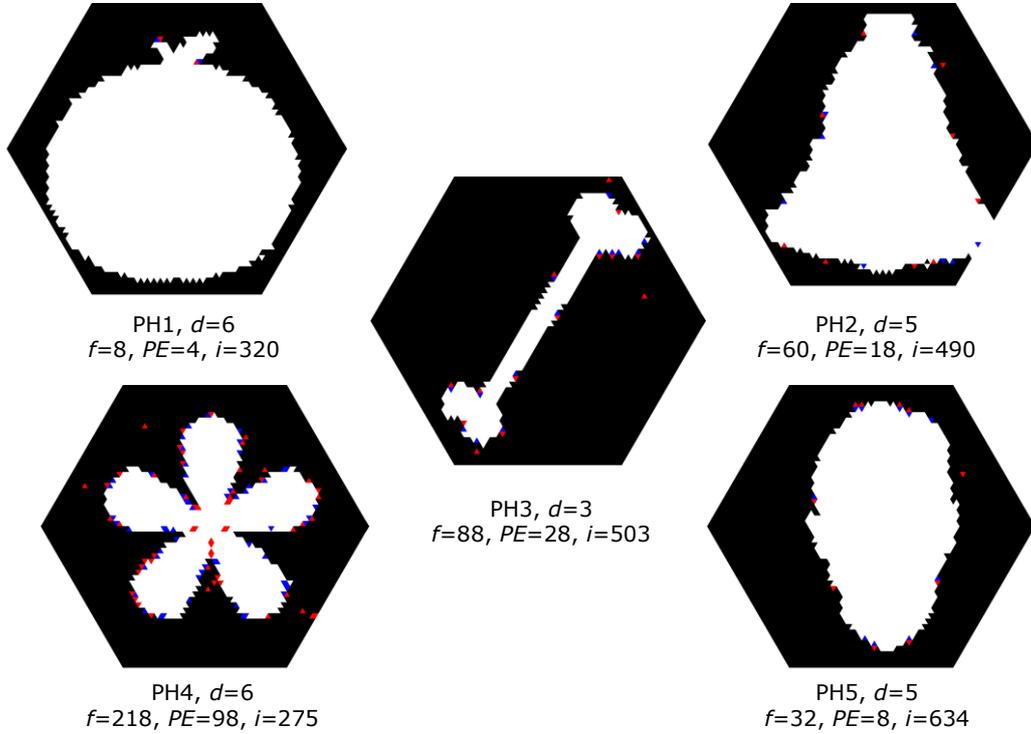


Figure 7.12. Reconstruction examples for non-convex hexagon-shaped images with dimension $26 \times 26 \times 26$.

Table 7.8. Reconstruction results for non-convex hexagon-shaped images with dimension $50 \times 50 \times 50$.

Image	d	Perfect reconstruction	Successful reconstruction	Projection error	Pixel error	Iteration	Runtime (s)
PH1	3	0	100	171	73	400	2112
	4	0	100	141	60	402	2015
	5	0	100	199	83	412	2216
	6	0	100	197	85	346	1857
PH2	3	0	100	217	94	401	2118
	4	0	100	202	87	412	2062
	5	0	100	345	164	382	2043
	6	0	100	267	111	416	2230
PH3	3	0	100	175	85	573	3015
	4	0	100	222	105	454	2261
	5	0	100	434	202	405	2164
	6	0	100	189	114	430	2202
PH4	3	0	100	2744	1453	820	4195
	4	0	100	983	763	661	3268
	5	0	100	899	649	349	1868
	6	0	100	505	299	416	2066
PH5	3	0	100	46	15	401	2006
	4	36	90	24	9	322	1611
	5	0	100	150	59	406	2168
	6	0	100	128	53	391	1950

Figure 7.13 shows the charts that contain the reconstruction results for the non-convex hexagon-shaped images resampled at dimension of $50 \times 50 \times 50$. The color of the columns indicates the number of projections used for the reconstruction, 3, 4, 5 and 6, respectively. Results show that, similarly to the case of dimension $26 \times 26 \times 26$, using five or six projections affects a lot the accurateness of the reconstructed images. This is because of the high number of iterations to which is associated a triplet of projection angles for which the first two projections are not orthogonal enough. This case happens for images like the ones from PH1, PH2 and PH3. Meanwhile results show that for other type of images the reconstruction results increase by using the same high number of iterations to which is associated a triplet of projection angles for which the first two projections are not orthogonal enough. See, for example the case of PH4.

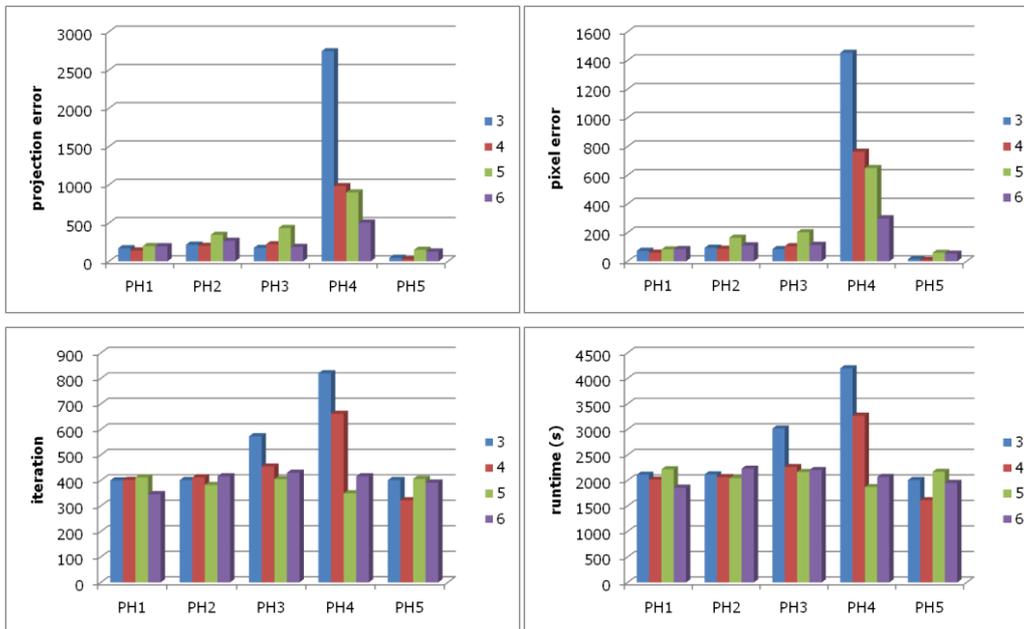


Figure 7.13. Reconstruction results for non-convex hexagon-shaped images with dimension $50 \times 50 \times 50$.

Figure 7.14 shows reconstruction examples for the test images resampled at dimension $50 \times 50 \times 50$. Missing pixels are shown with blue, additional pixels of the reconstructed image are red. Information about the reconstructions are shown below the images: f means fitness function value, PE means pixel error (sum of blue and red pixels) and i means number of iterations made for obtaining the solution. Reconstruction examples show that even if the projection and the pixel error are high the shape of the objects from the images can be clearly distinguished.

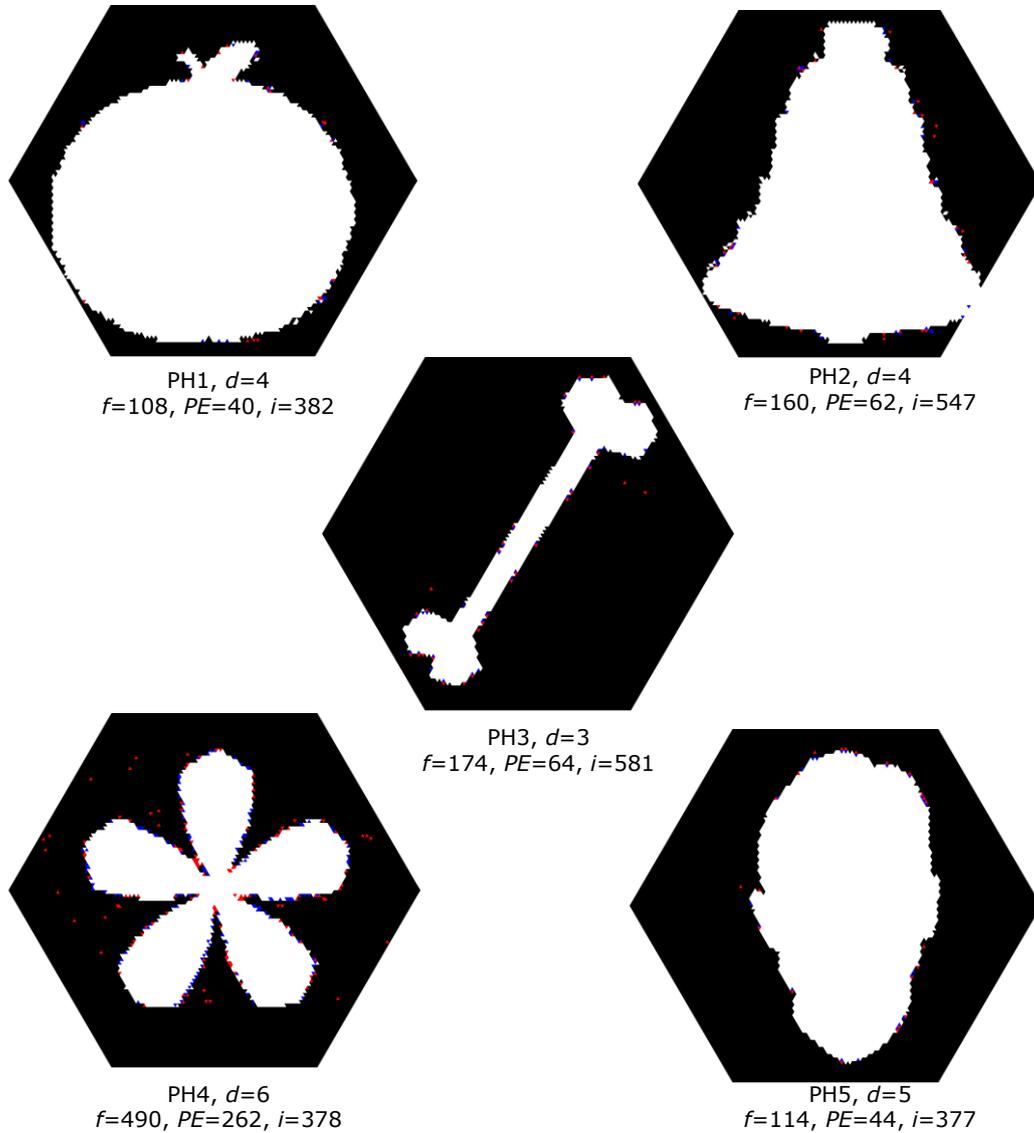


Figure 7.14. Reconstruction examples for non-convex hexagon-shaped images with dimension $50 \times 50 \times 50$.

Experimental results show that for small dimension images and for images that contain simple objects there are cases when the reconstruction is perfect, even if the image has a high dimension. For images that contain more complex objects the reconstructions are successful but neither of them is perfect, for high dimension objects. In order to obtain perfect reconstruction there are needed more projection angles. Also, in some cases we saw that better reconstruction results can be

obtained by using in each iteration triplets of projection angles in which the difference between the first two projection angles is orthogonal enough.

We tested our algorithm on three types of datasets. Test results show that our proposed algorithm computes accurate reconstructions also for images of higher dimensions, even if the reconstruction takes several hours. In some of the considered test images, having triplets of projection angles with difference between the first two projection angles less or equal to $2\pi/3$ is apparently not sufficient for reconstructing them. The first two projection angles have to be sufficiently orthogonal. By not using sufficiently orthogonal directions, the reconstructed image results in a flat reconstruction over the chosen directions, and this worsens our reconstruction. We obtained perfect reconstruction in only some seconds for the dataset of images that satisfy some constraints like the hv-polyominoes. For all the considered test cases our algorithm manage to successfully reconstruct the images, using only maximum six projection angles, in such a way that the object or objects contained in the image are clearly distinguished. In conclusions, choosing a reasonable number of projection angles, the proposed algorithm leads us to robust solutions that are very close to the original images.

Unfortunately, we could not make comparisons with alternative approaches for algorithms that reconstruct hexagon-shaped images using the triangular grid, because no alternative is yet available. Future work may include the development of alternative algorithms in order to be able to compare the feasibility and flexibility of the proposed algorithm for the case of triangular grid represented images.

7.4. CONCLUSIONS

In this chapter, we have presented an algorithm for reconstructing binary hexagon-shaped images from few projection angles, maximum six projection angles. This gives confidence on the feasibility of reconstruction hexagon-shaped images from few projection angles. The experimental results show that the algorithm can make a very good reconstruction in a relatively short time for different test images. Even if in our proposed method we use the smoothness assumption of an image, the algorithm performs well for reconstructing details also. Usually in practice images are not completely random and they do not satisfy mathematical constrains. The proposed method performs well by using the assumption that the images are relatively smooth, but they do not have any rigid structure assumption. In its actual form the algorithm assumes that the projection data it's perfect, there are no noisy projections.

8. CONCLUSIONS AND FUTURE WORK

8.1. Conclusions	126
8.2. Summary of contributions	127
8.3. Research perspectives	128

8.1. CONCLUSIONS

We have presented in this thesis our contribution to resolve the binary tomography reconstruction problem for digital images sampled on the triangular grid.

The field of tomography is still an active area of research, not being an easy task. The goal of tomography is to obtain image of object sections. This method is used in several fields in practice, when data about the inner structure of the object is needed without breaking it. It is applied in medicine, biology, material science, archeology, geophysics and in other field also.

Tomography techniques are based on acquiring projections of the image using multiple angles. These projection data is used by a tomography reconstruction software algorithm to obtain the section image.

In discrete tomography, the object, from mathematical point of view, corresponds to an attenuation function. For the attenuation function some integrals or sums over a subset are known. Binary tomography is a special case of discrete tomography where the function has only two values 0 and 1. The solution of the discrete topographic problem is usually undetermined, being several solutions for the same problem. Also discrete tomography problem is NP-hard when three or more than three projections are considered.

The first studied grid for discrete tomography was the square grid [Rys57][Rys63][Gal57][Wan98] and the cubic grid because of the Cartesian coordinate system use. However other regular and non-traditional grids in image processing can be used. In the plane, considering the possible tiling, the hexagonal and triangular grids can be used instead of the square grid. The advantage of using hexagonal grid is that it is very simple having only one neighborhood relation and it has better symmetric properties than the square grid has. The symmetry of the hexagonal grid is mirrored by its description by three coordinates with zero coordinate sums [Her95]. There is a relative wide literature on binary tomography using the hexagonal grid [Mat98][Mid05][Lai93]. The dual of the hexagonal grid is not hexagonal, but triangular, so the triangular grid (of areas), is topologically the same as the hexagonal grid of nodes. The triangular grid has a similar symmetry as the hexagonal grid, a $\frac{2\pi}{3}$ rotation moves the grid to itself, a three coordinate description being possible [Nag04][Nag07]. Also the triangular grid can represent any circular band limited signal in a more efficient way than conventional square grid.

In this thesis we have defined the digital image sampled on the triangular grid, and called it hexagon-shaped image. Also we have developed the mathematical background for the two phases of the binary tomography reconstruction problem, we have defined the Radon transform on triangular grid and we have proposed an image reconstruction algorithm from few different angles. Next section summaries our contributions.

8.2. SUMMARY OF CONTRIBUTIONS

The main purpose of this thesis is to solve the problem of the reconstruction of binary hexagon-shaped images represented on the triangular grid using projections of the image from few different angles.

In accordance with the established thesis objectives, we have introduced the fundamental bases of triangular grid and digital image representation using triangular grid. In this context, in chapter 3, we have presented the data structures, definitions and properties of the triangular grid. We also have defined the digital images on the triangular grid naming them hexagon-shaped images, because of their hexagon shape, fulfilling the first objective of the thesis. Some of the definitions presented in this chapter were introduced in [Moi13a][Nag13b][Nag13a][Nag14].

The binary tomography reconstruction problem presumes two main phases: the generation of the set of projections, and the combination of images to reconstruct an approximation of the original image.

Mathematically, the first phase, the generation of the set of projections of a hexagon-shaped image measured along different projection angles, is accomplished using the Radon transform. In chapter 4, we have developed the mathematical equations of the Radon transform on triangular grid bases achieving the second thesis objective. This transformation represents the equation of the measured projection, which is used in the second phase of tomography, in the reconstruction of the hexagon-shaped images.

The second phase of the considered problem, mathematically, consists of image reconstruction algorithms. For the image reconstruction process, the first step in our approach has been to model the reconstruction problem of hexagon-shaped images on triangular grid using three projections based on the problem of finding a maximum flow with minimum cost in a graph, fulfilling the third thesis objective. The proposed model, presented in chapter 3, takes into consideration the properties of a hexagon-shaped image, mainly the property that on the triangular grid a pixel is represented using three coordinates that correspond to the three coordinate axes of the grid and the domain in which these coordinates are defined in. In [Moi13a] and [Nag14] we presented the way in which the reconstruction problem of hexagon-shaped images, using orthogonal, respectively parallel to the triangular coordinate axes directions, can be modeled as the problem of finding maximum flow with minimum cost in a graph.

Using the model of the minimum cost maximum flow algorithm with restrictions imposed by the third projection, two image reconstruction algorithms were proposed.

The first developed method is a memetic algorithm, presented in chapter 6 which fulfills the fourth objective of the thesis. In this algorithm the initial population is generated using the minimum cost maximum flow algorithm for the case of hexagon-shaped images. We introduced new mutation and crossover operators for

hexagon-shaped images. Also a new compactness operator and a minimal hill climb operator, based on the switching components are defined. The compactness and the switching components are two important components of discrete tomography. We performed extension simulations on databases containing hv-convex polyominoes, generic hexagon-shaped images and non-convex regular hexagon-shaped images with various dimensions. Preliminary results show that the proposed method is robust enough, especially in the case in which the hexagon-shaped images satisfy prior knowledge, like the hv-polyominoes, or satisfy topological and geometrical constraints, like the generic images that contain objects without holes. The proposed operators for the case of hexagon-shaped images on triangular grid and some results using the orthogonal and respectively the parallel to the coordinate axes projection directions, are presented in [Moi11][Moi13b][Nag13a] and [Nag14].

The second proposed method for solving the image reconstruction problem from few projections is an iterative one and it is presented in chapter 7, fulfilling the last objective of the thesis. In each iteration, of the proposed algorithm, a new triplet of projection angles is selected. The reconstruction problem that depends on the selected triplet of projection angles and on the reconstruction from the previous iteration is solved. This reconstruction problem is equivalent with the problem of finding a flow with minimum cost in a graph. In our approach, we used the minimum cost maximum flow algorithm in order to solve the reconstruction problem. Our method is based on Batenburg approach for the case of the square grid, but some parts of the algorithm are adapted for the triangular grid case. The experimental results show that our algorithm produces good quality reconstruction results in a short time, even if we use high dimension hexagon-shaped images. Our results show that for images with more complex objects, it is not enough to use only six projections, but more different projection angles are needed.

8.3. RESEARCH PERSPECTIVES

In future we intend to generalize our methods to take into consideration specific models of the objects from the images, for example images that contain real test data from different domains. In this case, we want to extend the proposed method to 3D volumes of data based on the triangle representation of a pixel in 2D.

In real cases, the projection data can be perturbed by quantization and instrumental noises. In computerized tomography, usually these perturbations are disregarded due to the huge number of different projection angles which distribute the errors on the whole reconstructed image. In discrete tomography, this assumption is no longer valid because there are used a small number of projections in order to reconstruct the image. In this case, our method still needs to be verified.

Moreover, we intend to propose a parallel implementation for the proposed memetic algorithm for binary discrete tomography reconstruction on triangular grid, using few projection angles and without any further a priori information. Using parallel implementation would speed up our method due to the use of different physical CPUs.

Also we intend to implement other reconstruction algorithms for the case of hexagon-shaped images represented on the triangular grid, in order to make comparisons between different binary discrete tomography methods.

9. BIBLIOGRAPHY

- [Abb08] N. M. Abbasi, "The application of Fourier analysis in solving the Computed Tomography (CT) inverse problem," 2008.
- [Ahu93] R. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.
- [Ans83] R. P. Anstee, "The network flows approach for matrices with given row and column sums," *Discrete Mathematics*, vol. 44, pp. 125-138, 1983.
- [Bat05] J. K. Batenburg, "An evolutionary algorithm for discrete tomography," *Discrete Applied Mathematics*, vol. 151, pp. 36-54, 2005.
- [Bat06a] J. K. Batenburg, "Network flow algorithms for discrete tomography," Ph.D.Thesis, University of Leiden, The Netherlands, 2006.
- [Bat06b] J. K. Batenburg, "A network flow algorithm for binary image reconstruction from few projections," *Lecture Notes Comp. Sci.*, vol. 4245, pp. 86-97, 2006.
- [Bat07a] J. K. Batenburg, "A Network Flow Algorithm for Reconstructing Binary Images from Discrete X-rays," *Journal of Mathematical Imaging and Vision-JMIV*, vol. 27, no. 2, pp. 175-191, 2007.
- [Bat08] K. J. Batenburg, "A network flow algorithm for reconstructing binary images from continuous X-rays," *J. Math. Imag. Vis.*, vol. 30, pp. 231-248, 2008.
- [Baz65] M. J. Bazin and J. W. Benoit, "Off-line global approach to pattern recognition for bubble chamber pictures," in *IEEE Transactions on Nuclear Science*, 1965, p. 291-295.
- [Ber04] D. Bertsekas, A. Frangioni, and C. Gentile. (2004) RelaxIV, The MCFClass Project. [Online]. www.di.unipi.it/di/groups/optimize/Software/MCF.html
- [Ber94] D. Bertsekas and P. Tseng, "RELAX-IV: a faster version of the RELAX code for solving minimum cost flow problems," MIT, LIDS Technical Report LIDSP-P-2276, 1994.
- [Cel00] A. Cellerino, E. Novelli, and L. Galli-Resta, "Retinal ganglion cells with NADPH-diaphorase activity in the chick form a regular mosaic with a strong dorsoventral asymmetry that can be modelled by a minimal spacing rule," *Eur J Neurosci*, vol. 12, no. 2, pp. 613-620, 2000.
- [Cie11] R. Cierniak, "Reconstruction from Parallel-beam Projections," in *X-Ray Computed Tomography in Biomedical Engineering*. Springer London, 2011, pp. 83-125.
- [Daw76] R. Dawkins, *The Selfish Gene*. Oxford University Press, 1976.
- [Daw83] R. Dawkins, "Universal Darwinism," in *Evolution from molecules to man*, D. S. Bendall, Ed. Cambridge University Press, 1983.
- [Deu72] E. S. Deutsch, "Thinning algorithms on rectangular, hexagonal and triangular arrays," *Communications of the ACM*, vol. 15, no. 3, pp. 827-837, 1972.

- [DiG08] V. Di Gesu, G. Lo Bosco, F. Millonzi, and C. Valenti, "A memetical gorithm for binary image reconstruction," *Lecture Notesin Computer Science*, vol. 4958, pp. 384-395, 2008.
- [DiG10] V. Di Gesu, G. Lo Bosco, F. Millonzi, and C. Valenti, "A Memetic Approach to Discrete Tomography from Noisy Projections," *Pattern Recognition, Elsevier*, vol. 43, no. 9, pp. 3073-3082, 2010.
- [Dir27] P. Dirac, "The physical interpretation of the quantum dynamics," in *Proceedings of the Royal Society of London*, 1927, pp. 621-641.
- [Fay12] A. Fayas, H. Nisar, and A. Sultan, "Study on Hexagonal Grid in Image Processing," *IPCSIT*, vol. 45, pp. 282-288, 2012.
- [Fit96] A. P. Fitz and R. J. Green, "Fingerprint classification using a Hexagonal Fast Fourier Transform," *Pattern Recognition*, vol. 29, no. 10, p. 1587-1597, 1996.
- [Fon01] F. Fontan and D. Rocchesso, "Signal-theoretic characterization of waveguide mesh geometries for models of two-dimensional wave propagation in elastic media," in *IEEE Transactions on Speech and Audio Processing*, vol. 9 (2), 2001, pp. 152-161.
- [Fon98] F. Fontan and D. Rocchesso, "Signal-theoretic Characterization of Wavelength Mesh Geometries for Membrane Simulation," in *International Computer Music Conference Proceedings, MPublishing*, University of Michigan Library, 1998.
- [For56] L. R. Ford and D. R. Fulkerson, "Maximal flow through a network," in *Canad. J. Math*, vol. 8, 1956, pp. 399-404.
- [Gal57] D. Gale, "A theorem on flows in networks," *Pacific J. Math.*, vol. 7, no. 2, pp. 1073-1082, 1957.
- [Gar09] P. Garg, "A Comparison between Memetic algorithm and Genetic algorithm for the cryptanalysis of Simplified Data Encryption Standard algorithm," *International Journal of Network Security & Its Applications*, vol. 1, pp. 34-42, 2009.
- [Gol92] A. V. Goldberg, "An Efficient Implementation Of A Scaling Minimum-Cost Flow Algorithm," *Journal of Algorithms*, vol. 22, pp. 1-29, 1992.
- [Gon07] R. C. Gonzalez and R. E. Woods, *Digital image processing*, 3rd ed. Prentice Hall, 2007.
- [Gri00] P. Gritzmann, S. de Vries, and M. Wiegelmann, "Approximating binary images from discrete X-rays," *SIAM J. Optim*, vol. 11, pp. 522-546, 2000.
- [Gri98] P. Gritzmann, D. Prangenberg, S. de Vries, and M. Wiegelmann, "Success and failure of certain reconstruction and uniqueness algorithms in discrete tomography," *Int. J. Imaging Systems Technology*, vol. 9, pp. 101-109, 1998.
- [Gru90] B. Grunbaum and G. C. Shephard, *Tilings and Patterns*, 1st ed. W. H. Freeman and Company, 1990.
- [Haj01] L. Hajdu and R. Tijdeman, "An algorithm for discrete tomography," *Linear Algebra and its Applications*, vol. 339, pp. 147-169, 2001.
- [Har84] N. P. Hartman and S. L. Tanimoto, "A Hexagonal Pyramid data structure for Image Processing," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 14, p. 247-256, 1984.

- [Her07] G. T. Herman and A. Kuba, *Advances in Discrete Tomography and Its Applications*. Birkhäuser Boston, 2007.
- [Her94] I. Her and C. T. Yuan, "Resampling on a Pseudo-hexagonal Grid," *Graphical Models and Image Processing*, vol. 56, p. 336–347, 1994.
- [Her95] I. Her, "Geometric transformations on the hexagonal grid," *IEEE Transactions on Image Processing*, vol. 4, no. 9, pp. 1213–1222, 1995.
- [Her99] G. T. Herman and A. Kuba, *Discrete tomography: foundations, algorithms and applications*. Boston: Birkhauser, 1999.
- [Kak01] A. C. Kak, S. M. Kak, *Principles of computerized tomographic imaging*. Society for Industrial and Applied Mathematics, 2001.
- [Kir83] S. Kirkpatrick, C. Gelatt, and M. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [Kon87] T. Y. Kong and A. Rosenfeld, "Digital topology: Introduction and survey," *Computer vision, Graphics, and Image Processing*, vol. 48, pp. 357–393, 1987.
- [Kov04] V. Kovalevsky, "Algorithms in Digital Geometry Based on Cellular Topology," *Lecture Notes in Computer Science*, vol. 3322, pp. 366–393, 2004.
- [Kov08] V. Kovalevsky, *Geometry of Locally Finite Spaces (Computer Agreeable Topology and Algorithms for Computer Imagery)*. House Dr. Baerbel Kovalevski, 2008.
- [Kub99] A. Kuba and G. T. Herman, "Discrete tomography: a historical overview," in *Discrete tomography: foundations, algorithms and applications*. Boston: Birkhauser, 1999, pp. 3–34.
- [Lai93] A. F. Laine, S. Schuler, W. Huda, J. C. Honeyman, and B. Steinbach, "Hexagonal wavelet processing of digital mammography," *Proceedings of SPIE*, vol. 1898, p. 559–573, 1993.
- [Luk12] T. Lukić and B. Nagy, "Energy-Minimization Based Discrete Tomography Reconstruction Method for Images on Triangular Grid," *Lecture Notes in Computer Science*, vol. 7655, p. 274–284, 2012.
- [Mat98] S. Matej, G. T. Herman, and A. Vardi, "Binary tomography on the hexagonal grid using Gibbs Priors," *International Journal of Imaging Systems and Technology*, vol. 9, p. 126–131, 1998.
- [Mel91] R. A. Melter, "A Survey on Digital Metrics," *Contemporary Mathematics*, vol. 119, pp. 95–106, 1991.
- [Mid05] L. Middleton and J. Sivaswamy, *Hexagonal Image Processing, A Practical Approach*. Springer, 2005.
- [Moi11] E. Moisi and B. Nagy, "Discrete Tomography on the Triangular Grid: a Memetic Approach," in *7th International Symposium on Image and Signal Processing and Analysis*, Dubrovnik, 2011, pp. 579–584.
- [Moi13] E. V. Moisi, "Soft Computing Methods for Discrete Tomography on the Triangular Grid," "Politehnica" University of Timișoara, Timișoara, Ph.D. report, October 2013.

- [Moi13a] E. Moisi, B. Nagy, and V. Cretu, "Maximum flow minimum cost algorithm for reconstruction of images represented on the triangular grid," in *IEEE 8th International Symposium on Applied Computational Intelligence and Informatics*, Timisoara, 2013, pp. 35-40.
- [Moi13b] E. V. Moisi, I. V. Cretu, and B. Nagy, "Reconstruction of Binary Images Represented on Equilateral Triangular Grid Using Evolutionary Algorithms," in *Soft Computing Applications, Advances in Intelligent Systems and Computing*, vol. 195, 2013, pp. 561-571.
- [Moi14] E. V. Moisi, "Minimum Cost Maximum Flow Algorithm for Solving the Reconstruction Problem on Triangular Grid," "Politehnica" University of Timișoara, Timișoara, Ph.D. report, April 2014.
- [Mos10] P. Moscato and C. Cotta, "A Modern Introduction to Memetic Algorithms," in *Handbook of Metaheuristics*, M. a. P. J.-Y. Gendreau, Ed. Springer US, 2010, pp. 141-183.
- [Mos89] P. Moscato, "On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms," Caltech Concurrent Computation Program 826, 1989.
- [MPE] MPEG7 CE Shape-1 Part B. [Online]. http://www.imageprocessingplace.com/root_files_V3/image_databases.htm
- [Nag01] B. Nagy, "Finding Shortest Path with Neighborhood Sequences in Triangular Grids," in *2nd IEEE R8-EURASIP International Symposium on Image and Signal Processing and Analysis (ITI-ISPA 2001)*, Pula, Croatia, 2001, pp. 55-60.
- [Nag02] B. Nagy, "Metrics Based on Neighbourhood Sequences in triangular grid," *Pure Mathematics and Applications*, vol. 13, pp. 259-274, 2002.
- [Nag03a] B. Nagy, "Shortest Path in Triangular Grids with Neighbourhood Sequences," *Journal of Computing and Information Technology*, vol. 11, pp. 111-122, 2003.
- [Nag03b] B. Nagy, "A family of triangular grids in digital geometry," in *3rd International Symposium on Image and Signal Processing and Analysis*, Rome, 2003, pp. 101-106.
- [Nag04] B. Nagy, "Generalised triangular grids in digital geometry," *Acta Mathematica Academiae Paedagogicae Nyíregyháziensis*, vol. 20, p. 63-78, 2004.
- [Nag04b] B. Nagy, "Characterization of digital circles in triangular grid," *Pattern Recognition Letters*, vol. 25, no. 11, p. 1231-1242, 2004.
- [Nag06] A. Nagy, "Discrete Tomographic and PACS Image Processin System," University of Szeged Doctoral Thesis, 2006.
- [Nag07] B. Nagy, "Distances with Neighbourhood Sequences in Cubic and Triangular Grids," *Pattern Recognition Letters*, vol. 28, pp. 99-109, 2007.
- [Nag08a] B. Nagy and R. Strand, "A Connection between Z_n and Generalized Triangular Grids," *Lecture Notes in Computer Science*, vol. 5359, pp. 1157-1166, 2008.
- [Nag08b] B. Nagy and R. Strand, "Non-traditional grids embedded in Z_n ," *International Journal of Shape Modeling*, vol. 14, no. 2, pp. 209-228, 2008.

- [Nag12] B. Nagy, "Cellular Topology on the Triangular Grid," *Lecture Notes in Computer Science*, vol. 7655, pp. 143-153, 2012.
- [Nag13a] B. Nagy, T. Lukic, and E. Moisi, "Diszkrét tomográfia a háromszögrácson természet-motiválta algoritmusokkal (Soft computing methods for discrete tomography on the triangular grid)," in *KÉPAF*, Bakonybél, 2013, pp. 450-463.
- [Nag13b] B. Nagy, E. V. Moisi, and V. I. Cretu, "Discrete tomography on the triangular grid based on Ryser's results," in *8th International Symposium on Image and Signal Processing and Analysis (ISPA2013)*, 2013, pp. 794-799.
- [Nag14] B. Nagy and E. V. Moisi, "Binary tomography on the triangular grid with 3 alternative directions – a genetic approach," in *International Conference on Pattern Recognition*, Stockholms, 2014, accepted.
- [Nak94] J. Nakata and K. Ogawa, "Image reconstruction from projections using a genetic algorithm," in *Nuclear Science Symposium and Medical Imaging Conference*, vol. 4, 1994, pp. 1706-1709.
- [Nat01] F. Natterer, *The Mathematics of Computerized Tomography (Classics in Applied Mathematics, Book 32)*. Society for Industrial and Applied Mathematics, 2001.
- [Nat011] W. F. Natterer F., *Mathematical Methods in Image Reconstruction (Monographs on Mathematical Modeling and Computation)*. Society for Industrial and Applied Mathematics; 1st edition, 2001.
- [Neh12] N. Dixit, "Improving image quality of synthetically zoomed tomographic images," Master of Science (by Research) in Communication system and signal processing, International Institute of Information Technology, Hyderabad, India, 2012.
- [Pac05] O. Packer and D. R. Williams, "Light, the Retinal Image, and Photoreceptors," University of Rochester, The College, Brain & Cognitive Sciences course syllabus, 2005.
- [Pat] Pattern Recognition and Discrete Geometry. [Online]. <http://math.unipa.it/predige/imgs.zip>
- [Rad17] J. Radon, "Über die Bestimmung von Funktionen durch ihre Integralwerte längs gewisser Mannigfaltigkeiten," *Berichte Sächsische Akademie der Wissenschaften*, vol. 69, pp. 262-277, 1917.
- [Ros69] A. Rosenfeld, *Picture Processing by Computer*. Academic Press, 1969.
- [Ros86] A. Rosenfeld and J. L. Pfaltz, "Distance functions on digital pictures," *Pattern Recognition*, vol. 1, pp. 33-61, 1986.
- [Ros89] A. Rosenfeld and R. A. Melter, "Digital Geometry," *The mathematic intelligencer*, vol. 11, pp. 69-72, 1989.
- [Rys57] H. J. Ryser, "Combinatorial properties of matrices of zeros and ones," *Can J. Math.*, vol. 9, pp. 371-377, 1957.
- [Rys60] H. J. Ryser, "Matrices of zeros and ones," *Bulletin of American Mathematical Society*, vol. 66, no. 6, pp. 442-464, 1960.
- [Rys63] H. J. Ryser, *Combinatorial mathematics*. New York: Mathematical Assn of Amer, 1963.

- [Shi81] K. Shimizu, "Algorithm for Generating a Digital Circle on a Triangular grid," *Computer Graphics and Image Processing*, vol. 15, no. 4, pp. 401-402, 1981.
- [Slu82] C. H. Slump and J. J. Gerbrands, "A network flow approach to reconstruction of the left ventricle from two projections.," in *Computer Graphics and Image Processing*, vol. 18, 1982, pp. 18-36.
- [Tan71] K. Tanabe, "Projection method for solving a singular system of linear equations and its applications," *Numerische Mathematik*, vol. 17, no. 3, pp. 203-214, 1971.
- [Val08] C. Valenti, "A genetic algorithm for discrete tomography reconstruction," *Genetic Programming and Evolvable Machines*, vol. 9, no. 1, pp. 85-96, Mar. 2008.
- [Van02] D. Van De Ville, R. Van de Walle, W. Philips, and I. Lemahieu, "Image resampling between orthogonal and hexagonal lattices," in *IEEE International Conference on Image Processing*, 2002, p. 389-392.
- [Van03] D. Van De Ville, W. Philips, I. Lemahieu, and R. Van de Walle, "Suppression of sampling moire in color printing by spline-based least-squares prefiltering," *Pattern Recognition Letters*, vol. 24, p. 1787-1794, 2003.
- [Van04] D. Van De Ville, et al., "Hex-splines: A novel spline family for hexagonal lattices," *IEEE Transactions on Image Processing*, vol. 13, no. 6, p. 758-772, 2004.
- [Wan98] B. Wang and F. Zhang, "On the precise number of $(0;1)$ -matrices in $A(R;S)$," *Discrete Math*, vol. 187, pp. 211-220, 1998.
- [Wat89] A. B. Watson and J. A. J. Ahumada, "A hexagonal orthogonal-oriented pyramid as a model of image representation in the visual cortex," *IEEE Transactions on Biomedical Engineering*, vol. BME-36, p. 97-106, 1989.
- [Wut91] C. A. Wuthrich and P. Stucki, "An algorithmic comparison between square- and hexagonal-based grids," *Graphical Models and Image Processing*, vol. 53, pp. 324-339, 1991.
- [Xia05] H. Xiangjian and J. Wenjing, "Hexagonal Structure for Intelligent Vision," in *Information and Communication Technologies*, 2005, pp. 52-64.

APPENDIX A

In this appendix it is shown the formula used for converting coordinates of pixels on a triangular grid into corresponding pixel coordinates on a square grid. The Cartesian grid is a well known grid used in image processing. Triangular grid is, in many respects as the Cartesian grid, the pixels in the triangular grid are triangles instead of squares as in the case of the Cartesian grid. Figure A. 1 shows the triangular and Cartesian grid coordinate systems.

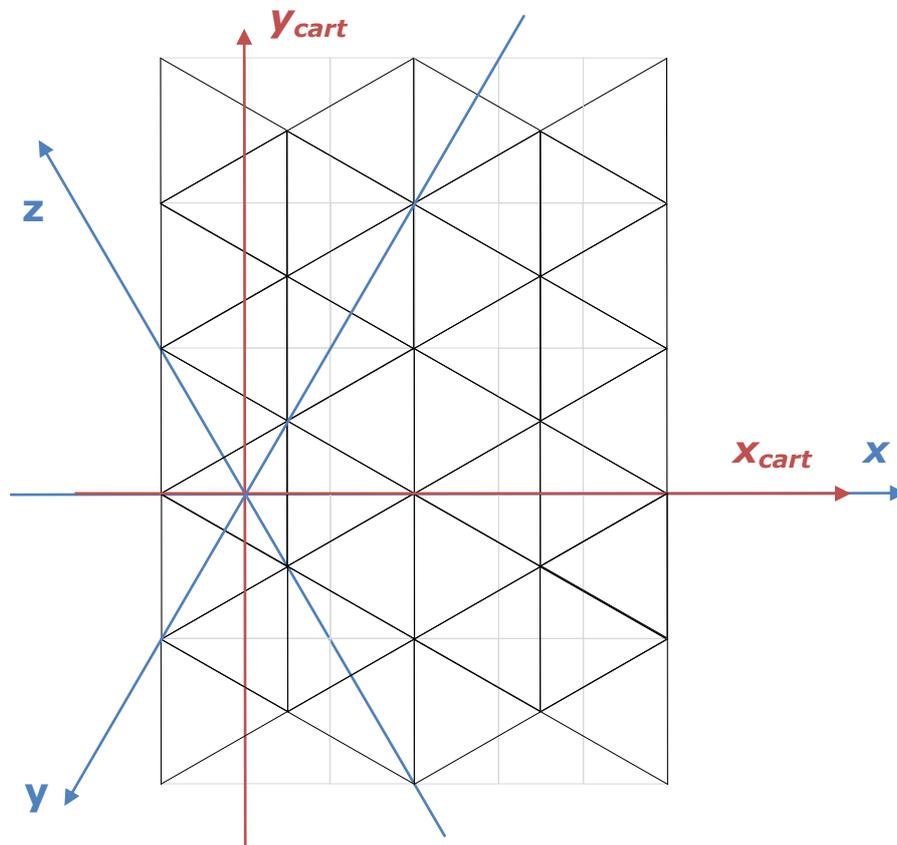


Figure A. 1. Triangular grid coordinate and Cartesian grid coordinate systems

Using the Pythagorean Theorem, the relationship between y -Cartesian and x -Cartesian grid line spacing is

$$y - \text{Cartesian space} = x - \text{Cartesian space} * \sqrt{3} \quad (\text{A.1})$$

If the x -Cartesian space is considered to be 1 unit, than the y -Cartesian space is $\sqrt{3}$ units. The length of a side of the equilateral triangle is $\sqrt{3}$. The length of the height of the equilateral triangle is $\frac{3}{2}$.

It can be noticed that any row of triangles parallel to x -coordinate have a constant $y - z$ coordinate, so y_{cart} depends only on $y - z$. The centers of the triangles in one row parallel to x -coordinate are half side-length above/below the centers in the next row, so:

$$y_{cart} = -\frac{\sqrt{3}}{2}(y - z) \quad (\text{A.2})$$

The triangles with a given x and z coordinates all have centers on a line perpendicular to x axis that is $\frac{1}{2}$ from the origin. The x axis has slope $-\frac{\sqrt{3}}{3}$, so the line perpendicular to it has slope $\sqrt{3}$. A point on the x axis and on the line has coordinates $\left(\frac{1}{2}(x - z), -\frac{\sqrt{3}}{2}(x - z)\right)$. An equation in x_{cart} and y_{cart} for the line containing the centers of the triangles with x and z coordinates is $y_{cart} + \frac{\sqrt{3}}{2}(x - z) = \sqrt{3}\left(x_{cart} - \frac{1}{2}(x - z)\right)$. Substituting for y_{cart} using equation (A.2) and solving for x_{cart} gives the formula:

$$x_{cart} = x - \frac{1}{2}(y + z) \quad (\text{A.3})$$

Using the coordinate transformation between the triangular coordinates and the Cartesian coordinates the radon transform, which is the mathematical basis of the projection operation, is presented.

APPENDIX B

This appendix provides the derivation for the rotation transformation on the triangular grid.

In Figure B. 1, OP , OQ , OR are axes of triangular coordinate and OP' , OQ' , OR' are axes of the triangular coordinate rotates by an angle of θ with respect to OP and OQ . Point E is represented as $(x, -y, z)$ and $(x', -y', z')$ in OP , OQ , OR and OP' , OQ' , OR' coordinate system, where $x+(-y)+z=0$ or $x+(-y)+z=1$ and $x'+(-y')+z'=0$ or $x'+(-y')+z'=1$.

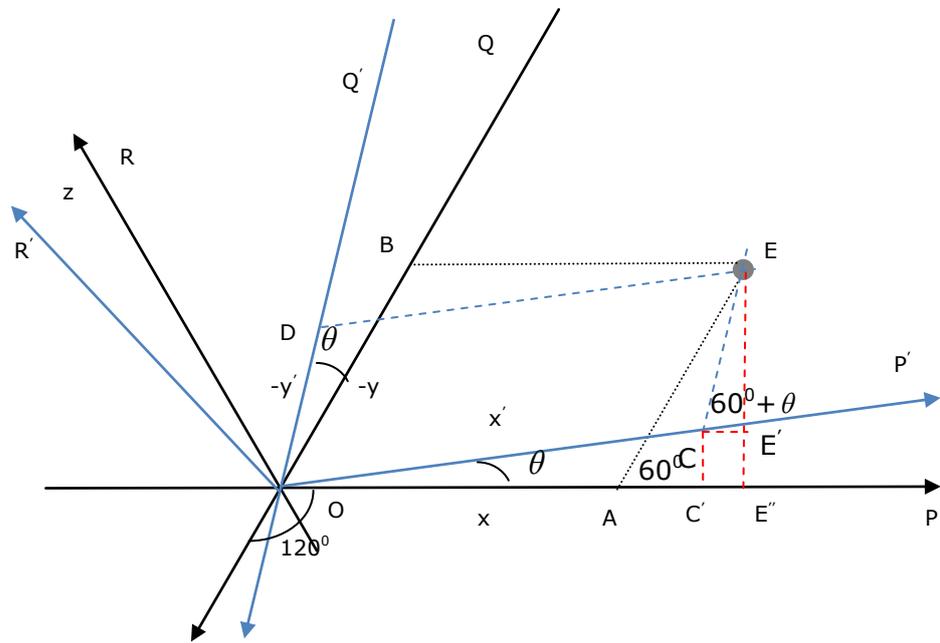


Figure B. 1. Triangular and rotated triangular coordinate

Substituting:

$$OA = x - z$$

$$OE'' = -(y - z) \cos 60^\circ$$

$$OC' = (x' - z') \cos \theta$$

$$CE' = -(y - z) \cos(60^\circ + \theta)$$

equation(B.1) becomes:

$$(x - z) - (y - z) \cos 60^\circ = (x' - z') \cos \theta - (y' - z') \cos(60^\circ + \theta) \quad (B.2)$$

For EE'' we can write:

$$EE'' = E'E'' + EE' = CC' + EE' \quad (B.3)$$

Substituting:

$$\begin{aligned} EE'' &= -(y-z) \sin 60^\circ \\ CC' &= (x'-z') \sin \theta \\ EE' &= -(y'-z') \sin(60^\circ + \theta) \end{aligned}$$

Equation(B.3) becomes:

$$-(y-z) \sin 60^\circ = (x'-z') \sin \theta - (y'-z') \sin(60^\circ + \theta) \quad (B.4)$$

Solving equations (B.2) and (B.4) for $x-z$ and $y-z$ we get:

$$\begin{bmatrix} x'-z' \\ y'-z' \end{bmatrix} = \frac{1}{\sin 60^\circ} \begin{bmatrix} \sin \theta & \sin(60^\circ - \theta) \\ \sin(60^\circ + \theta) & -\sin \theta \end{bmatrix} \begin{bmatrix} x-z \\ y-z \end{bmatrix} \quad (B.5)$$

$$\begin{bmatrix} x-z \\ y-z \end{bmatrix} = \frac{1}{\sin 60^\circ} \begin{bmatrix} \sin(60^\circ - \theta) & \sin \theta \\ -\sin \theta & \sin(60^\circ + \theta) \end{bmatrix} \begin{bmatrix} x'-z' \\ y'-z' \end{bmatrix} \quad (B.6)$$

LIST OF PUBLICATIONS

ISI PROCEEDINGS

- [Moi13a] *Elisa Valentina Moisi*, Vladimir Ioan Cretu, Benedek Nagy, "Maximum Flow Minimum Cost Algorithm for Reconstruction of Images Represented on the Triangular Grid", Proceedings of the 8-th IEEE International Symposium on Applied Computational Intelligence and Informatics, SACI 2013, Editor(s): Fodor, Janos; Ciocarlie, Horia, IEEE, Timisoara, Romania, 23-25 May. 2013, pp. 35-40
- [Moi13b] *Elisa Valentina Moisi*, Vladimir Ioan Cretu, Benedek Nagy, "Reconstruction of Binary Images Represented on Equilateral Triangular Grid Using Evolutionary Algorithms", "Soft Computing Applications", Proceedings of the 5th International Workshop Soft Computing Applications (SOFA 2012), Advances in Intelligent Systems and Computing series ASIC 195, Springer Heidelberg New York. Dordrecht London 2013, pp.561-572

IEEE PROCEEDINGS

- [Nag14] Benedek Nagy, *Elisa Valentina Moisi*: "Binary tomography on the triangular grid with 3 alternative directions – a genetic approach," in *International Conference on Pattern Recognition*, Stockholm, 2014, accepted
- [Nag13b] Benedek Nagy, *Elisa Valentina Moisi*, Vladimir Ioan Cretu: "Discrete Tomography on the Triangular Grid based on Ryser's Results", 8th International Symposium on Image and Signal Processing and Analysis (ISPA 2013), Trieste, Italy, pp. 794-799
- [Moi11] *Elisa Valentina Moisi*, Benedek Nagy, "Discrete Tomography on the Triangular Grid: a Memetic Approach", 7th International Symposium on Image and Signal Processing and Analysis (ISPA 2011), Dubrovnik, Croatia, pp. 579-584

SCIENTIFIC CONFERENCE VOLUMES

- [Nag13a] Benedek Nagy, Tibor Lukic, *Elisa Valentina Moisi*, "Soft computing methods for discrete tomography on the triangular grid" (Diszkrét tomográfia a háromszögrácson természet-motiválta algoritmusokkal), KÉPAF 2013, Bakonybél, pp. 450-463

TECHNICAL REPORTS

- [Moi13] *Elisa Valentina Moisi*, "Soft Computing Methods for Discrete Tomography on the Triangular Grid," "Politehnica" University of Timișoara, Timișoara, Ph.D. report, October 2013.
- [Moi14] *Elisa Valentina Moisi*, "Minimum Cost Maximum Flow Algorithm for Solving the Reconstruction Problem on Triangular Grid," "Politehnica" University of Timișoara, Timișoara, Ph.D. report, April 2014.