

**AGH - UNIVERSITY OF SCIENCE AND TECHNOLOGY
POLAND**

**FACULTY OF ELECTRICAL ENGINEERING, AUTOMATICS,
COMPUTER SCIENCE AND ELECTRONICS**



Ali Abdrhman M. Ukasha, M.Sc.

Ph.D. Thesis

**METHODS OF CONTOUR
COMPRESSION IN TIME
AND SPECTRAL DOMAINS**

**Supervisor:
Prof. dr hab. eng. Andrzej Dziech**

**Krakow
© 2006**

ACKNOWLEDGEMENTS

I am very much grateful to my supervisor Prof. A. Dziech for his valuable assistance, guidance and advice throughout the course of this research. It is worth mentioning that this work could not have been realized without the support he provided, including the investigation of useful suggestions, made by him.

My deep appreciation and thanks are to my parents for their constant support and encouragement throughout my study.

The encouragement, understanding and patience of my wife and children during the course of this research are gratefully acknowledged.

I would like to express my thanks to Dr. I. Pardyka for many useful discussions, and Dr. R. Baran and Dr. P. Slusarczyk for their helping during the all thesis stages.

I am also using this medium to say thank you to W. Sornat and M. Lutwin for their helps and guidance during my staying period in Kielce.

I would also like to thank for all the help and advice receive from my colleagues and friends at the University of Computer Engineering and Telecommunications in Kielce and Academic Świętokrzyska in Kielce.

Finally, I wish also to express my gratitude to the Libyan Secretary of Higher Education for providing the scholarship to do this work.

ABSTRACT

A process that has reduced the size of image files such as eliminating information which is difficult to see by the human eye is called image compression. Reducing the quantity of information (number of bits) that is required to store or transmit a digital image is referred to as the image compression. In this dissertation, the methods of contour extraction, approximation, and compression in time and spectral domains are presented, investigated and compared. Three new algorithms in time domain which are fast and have good reconstruction are also proposed. These algorithms are “contour approximation using centroid method”, “contour compression using segment distances ratio method”, and “triangle family methods of contour compression”. These algorithms are compared with the contour compression in spectral domain. These transforms are periodic Haar piecewise-linear (PHL), Walsh, discrete cosine transform (DCT) and Haar. The comparison show that the fastest transform is the PHL transform which needs fewer operations.

Three different algorithms of contour extraction and image compression using low-pass filter (LPF) and high-pass filter (HPF) are presented and compared with Sobel and Canny detectors in this dissertation. These algorithms are “contour extraction and image compression in spectral domain using HPF”. The proposed algorithms used the transforms which are mentioned above. Effectiveness of the contour extraction for different classes of images is evaluated. The results show that these algorithms have a small number of operations and a good quality of contour extraction and image reconstruction.

A proposed algorithm for combining two real grey-scale images using contour extraction in transform domain (high pass filter) and correlation function is presented.

In the last part two proposed algorithms are considered in order to reconstruct the original image using PHL transform. The first algorithm is “Image reconstruction by contour extraction using HPF”. The second algorithm is “Image reconstruction by contour extraction using bit-planes decomposition”.

TABLE OF CONTENTS

	Page
<i>ACKNOWLEDGEMENTTS</i>	ii
<i>ABSTRACT</i>	iii
<i>LIST OF FIGURES</i>	vii
<i>LIST OF TABLES</i>	xvii
<i>LIST OF LISTINGS</i>	xviii
<i>INTRODUCTION</i>	xix
 1. CONTOUR EXTRACTION	 1
1.1 Overview	1
1.2 Edge Detection	1
1.3 Grey-level metrics	2
1.4 Contour extraction using different windows	9
1.4.1 Object contour following ‘OCF’ method	11
1.4.2 Multiple step contour extraction ‘MSCE’ method	12
1.4.3 Object-oriented contour extraction ‘OCE’ method (4x4 windows)	13
1.4.4 Single step parallel contour extraction ‘SSPCE’ method (3x3 windows)	15
1.4.5 Contour extraction based on 2x2 windows	17
1.5 Contour descriptors	18
1.6 Comparison between the contour extraction algorithms	19
 2. METHODS OF CONTOUR COMPRESSION IN TIME DOMAIN	 25
2.1 Overview	25
2.2 Contour compression methods	25

2.2.1 Ramer method	26
2.2.2 Tangent method	27
2.2.3 Algorithm for contour approximation using centroid method	30
2.2.3.1 Description of the algorithm	30
2.2.3.2 Results of the experiments	34
2.2.4 Algorithm for contour compression using segment distances ratio method	37
2.2.4.1 Description of the algorithm	38
2.2.4.2 Applied measures	40
2.2.4.3 Results of the experiments	41
2.3 Comparison between the contour approximation algorithms	46

3. TRIANGLE FAMILY METHODS OF CONTOUR COMPRESSION IN TIME DOMAIN 49

3.1 Overview	49
3.2 Applied measures	50
3.3 Algorithm for contour approximation using the triangle height over length ratio method	51
3.3.1 Description of the algorithm	51
3.3.2 Threshold determination.....	52
3.3.3 Results of the experiments	57
3.4 Algorithm for contour approximation using the triangle height method	61
3.4.1 Description of the algorithm	61
3.4.2 Results of the experiments	61
3.5 Algorithm for contour approximation using the triangle height square method	66
3.5.1 Description of the algorithm	66
3.5.2 Results of the experiments	66
3.6 Algorithm for contour approximation using the triangle area method	71

3.6.1 Description of the algorithm	71
3.6.2 Results of the experiments	72
3.7 Comparison between the triangle family algorithms	76
4. METHODS OF CONTOUR COMPRESSION IN TRANSFORM DOMAIN	80
4.1 Overview	80
4.2 Periodic Haar piecewise-linear (PHL) Transform	81
4.3 Walsh transform	86
4.4 Discrete cosine transform (DCT)	88
4.5 Haar transform	91
4.6 Comparison between the fast transforms and time domain methods	95
4.7 Algorithms for contour extraction and image compression in transform domain using High Pass Filter (HPF)	97
4.7.1 Description of the algorithms	98
4.7.2 Experimental results	100
4.8 Algorithm for combining two images using contour extraction in transform domain by High Pass Filter (HPF)	108
4.8.1 Description of the algorithm	108
4.8.2 Experimental results	109
4.9 Algorithms for reconstructing the image using contour extraction in transform domain by High Pass Filter (HPF) and bit-planes decomposition methods	116
4.9.1 Description of HPF algorithm	116
4.9.2 Description of bit-planes decomposition algorithm	116
4.9.3 Applied measures	118
4.9.4 Experimental results	118
CONCLUSIONS	124
REFERENCES	126

LIST OF FIGURES

Chapter 1		Page
Fig. 1.1	Gradient operators	5
Fig. 1.2	Roberts Cross convolution masks	6
Fig. 1.3	Sobel Cross convolution masks	7
Fig. 1.4	Laplacian operator convolution masks	7
Fig. 1.5a	Chain coding using the 8-Connectivity scheme	11
Fig. 1.5b	Chain coding using the 4-Connectivity scheme	11
Fig. 1.6	Object contour follower (OCF)	11
Fig. 1.7	Block diagram of the Multiple Step Contour Extraction (MSCE) method.....	12
Fig. 1.8	Object-oriented Contour Extraction (OCE) method	14
Fig. 1.9a	OCE procedure without correcting the first step	14
Fig. 1.9b	OCE procedure after correcting the first step	14
Fig. 1.10	The eight rules to extract closed contour using SSPCE method ...	16
Fig. 1.11	Pixel numbering for 2x2 windows	17
Fig. 1.12a	Binary image (Circle)	20
Fig. 1.12b	Binary image (Rectangle)	20
Fig. 1.12c	Binary image (E letter)	20
Fig. 1.13	Extracted contours using the three different algorithms	21
Fig 1.14a	Number of operations versus number of edges using the all algorithms of the Circle	22
Fig 1.14b	Number of operations versus number of edges using the all algorithms of the Rectangle	22
Fig 1.14c	Number of operations versus number of edges for the all	

	algorithms of the E letter	22
Fig. 1.15	Binary image for the Libya map	23
Fig. 1.16	Number of operations versus number of edges using the all algorithms of the Libya map	23
Chapter 2		
Fig. 2.1	Example for contour approximation using Ramer algorithm	27
Fig. 2.2	Fit criterion of the tangent algorithm	27
Fig. 2.3	Contour representation using (θ, l)	28
Fig. 2.4	Contour representation using (α, l)	29
Fig. 2.5a	Selection of output vertices at $I = 0$	31
Fig. 2.5b	Selection of output vertices at $I = 1$	31
Fig. 2.6	Flowchart of the algorithm	32
Fig. 2.7a	Test contour: Serpent	34
Fig. 2.7b	Test contour: Apple	34
Fig. 2.8	Selected results of Serpent contour test approximation	35
Fig. 2.9	Selected results of Apple contour test approximation	36
Fig. 2.10a	Comparison of the analyzed method with the Ramer algorithm: MSE versus CR	37
Fig. 2.10b	Comparison of the analyzed method with the Ramer algorithm: SNR versus CR	37
Fig. 2.11	Fit criterion of the algorithm for one segment	39
Fig. 2.12	Flowchart of the algorithm	40
Fig. 2.13a	Test contour: Italy	41
Fig. 2.13b	Test contour: Serpent	41
Fig. 2.14	Results of approximation for the Italy contour	42

Fig. 2.15	Results of approximation for the Serpent contour	43
Fig. 2.16	Comparison of the analyzed method with the Ramer algorithm for NO versus CR of the tested contours	44
Fig. 2.17	Comparison of the analyzed method with the Ramer algorithm for NO versus MSE of the tested contours	44
Fig. 2.18a	Comparison of the analyzed method with the Ramer algorithm for NO versus SNR: Italy contour	45
Fig. 2.18b	Comparison of the analyzed method with the Ramer algorithm for NO versus SNR: Serpent contour	45
Fig. 2.19	Comparison of the analyzed method with the Ramer algorithm for SNR versus CR of the selected contours	46
Fig. 2.20a	Original contour	47
Fig. 2.20b	Approximated contour using the first algorithm	47
Fig. 2.20c	Approximated contour using the second algorithm	47
Fig. 2.20d	Approximated contour using the third algorithm	47
Fig. 2.21a	Plot comparing algorithms (NO versus CR)	48
Fig. 2.21b	Plot comparing algorithms (SNR versus CR)	48
Chapter 3		
Fig. 3.1a	Test contour: Italy	50
Fig. 3.1b	Test contour: Rose	50
Fig. 3.2	Illustration of the basic triangle for the analyzed algorithm	51
Fig. 3.3	Flowchart of the analyzed algorithm	52
Fig. 3.4	CR versus N for different thresholds of Rose contour	53
Fig. 3.5	SNR versus N for different thresholds of Rose contour	53
Fig. 3.6	Approximated contour using triangle height over length method for the Rose contour	54

Fig. 3.7	Approximated contour using Ramer method for the Rose contour	54
Fig. 3.8a	Test contour for Arabic name	55
Fig. 3.8b	Test contour for E letter	55
Fig. 3.9	SNR versus N for different thresholds of Italy, Arabic, and E letter contours	56
Fig. 3.10	Results of approximation for the Italy contour	57
Fig. 3.11	Results of approximation for the rose contour	58
Fig. 3.12a	Comparison of the analyzed method with the Ramer method for the contour of Italy	59
Fig. 3.12b	Comparison of the analyzed method with the Ramer method for the contour of Rose	59
Fig. 3.13a	Approximated Rose contour using Ramer method	59
Fig. 3.13b	Approximated Rose contour using analyzed method	59
Fig. 3.14	Comparison of the analyzed method with the Ramer and tangent methods of MSE versus CR for the Italy contour	60
Fig. 3.15a	Comparison of the analyzed method with the Ramer and tangent methods of SNR versus CR for the contour of Italy	60
Fig. 3.15b	Comparison of the proposed method with the Ramer and tangent methods of SNR versus CR for the contour of Rose	60
Fig. 3.16	Results of approximation for the Italy contour	62
Fig. 3.17	Results of approximation for the rose contour	63
Fig. 3.18	Comparison of the analyzed method with the Ramer method for the contours of Italy and Rose	64
Fig. 3.19a	Approximated Rose contour using Ramer method	64
Fig. 3.19b	Approximated Rose contour using analyzed method	64
Fig. 3.20	Comparison of the analyzed method with the Ramer and tangent methods for MSE versus CR of the Italy contour	65

Fig. 3.21a	Comparison of the analyzed method with the Ramer and tangent methods of SNR versus CR for the contour of Italy	65
Fig. 3.21b	Comparison of the analyzed method with the Ramer and tangent methods of SNR versus CR for the contour of Rose	65
Fig. 3.22	Results of approximation for the Italy contour	67
Fig. 3.23	Results of approximation for the rose contour	67
Fig. 3.24a	Comparison of the analyzed method with the Ramer method for the contour of Italy	69
Fig. 3.24b	Comparison of the analyzed method with the Ramer method for the contour of Rose	69
Fig. 3.25a	Approximated Rose contour using Ramer method	68
Fig. 3.25b	Approximated Rose contour using analyzed method	68
Fig. 3.26	Comparison of the proposed method with the Ramer and tangent methods of SNR versus CR for the contours of Italy and Rose	70
Fig. 3.27	Comparison of the analyzed method with the Ramer and tangent methods for MSE versus CR of the Italy contour	71
Fig. 3.28	Results of approximation for the Italy contour	72
Fig. 3.29	Results of approximation for the rose contour	73
Fig. 3.30a	Comparison of the analyzed method with the Ramer method for the contour of Italy	74
Fig. 3.30b	Comparison of the analyzed method with the Ramer method for the contour of Rose	74
Fig. 3.31a	Approximated Rose contour using Ramer method	75
Fig. 3.31b	Approximated Rose contour using analyzed method	75
Fig. 3.32	Comparison of the analyzed method with the Ramer and tangent methods for MSE versus CR of the Italy contour	75
Fig. 3.33a	Comparison of the analyzed method with the Ramer and tangent methods for SNR versus CR of the Italy contour	76

Fig. 3.33b	Comparison of the analyzed method with the Ramer and tangent methods for SNR versus CR of the Rose contour	76
Fig. 3.34a	MSE versus CR for Italy contour	77
Fig. 3.34b	MSE versus CR for Rose contour	77
Fig. 3.35a	SNR versus CR for Italy contour	78
Fig. 3.35b	SNR versus CR for Rose contour	78
Fig. 3.36a	NO versus CR for Italy contour	79
Fig. 3.36b	NO versus CR for Rose contour	79
Chapter 4		
Fig. 4.1	Block diagram for contour extraction and compression in spectral domain	81
Fig. 4.2	Compressed contour using PHL transform	85
Fig. 4.3a	Sequency order	86
Fig. 4.3b	Hadamard order	86
Fig. 4.3c	Paley order	86
Fig. 4.4	Hadamard and Walsh ordered	87
Fig. 4.5	Compressed contour using Walsh transform	88
Fig. 4.6	The basis functions for the DCT transform	89
Fig. 4.7	Contour compressed using DCT transform	91
Fig. 4.8	Set of Haar functions	93
Fig. 4.9	Compressed contour using Haar transform	94
Fig. 4.10	MSE versus CR for the Italy contour	96
Fig. 4.11	SNR versus CR for the Italy contour	96
Fig. 4.12	NO versus CR for the Italy contour	96

Fig. 4.13a	HPF zonal method for the spectral image (256x256) using the first analyzed algorithm	99
Fig. 4.13b	HPF zonal method for the spectral image (256x256) using the second analyzed algorithm	99
Fig. 4.13c	HPF zonal method for the spectral image (256x256) using the third analyzed algorithm	99
Fig. 4.14	Block diagram of the analyzed algorithms	100
Fig. 4.15a	Test image: Tools (256x256)	100
Fig. 4.15b	Test image: Shoe (128x128)	100
Fig. 4.15c	Test image: Baby (128x128)	100
Fig. 4.15d	Test image: Montage (256x256)	100
Fig. 4.16a	Reconstruction of Tools image using I algorithm for the lower right block (128x128)	101
Fig. 4.16b	Reconstruction of Tools image using I algorithm for the lower right block (192x192)	101
Fig. 4.17a	Reconstruction of Baby image using I algorithm for the lower right block (64x64)	101
Fig. 4.17b	Reconstruction of Baby image using I algorithm for the lower right block (96x96)	101
Fig. 4.18a	Reconstruction of Tools image using II algorithm (128x128) block represented by the upper left block (192x192)	102
Fig. 4.18b	Reconstruction of Tools image using II algorithm (96x96) block represented by the upper left block (128x128)	102
Fig. 4.19a	Reconstruction of Baby image using II algorithm (64x64) block represented by the upper left block (96x96)	102
Fig. 4.19b	Reconstruction of Baby image using II algorithm (48x48) block represented by the upper left block (64x64)	102
Fig. 4.20a	Reconstruction of Tools image using III algorithm by rejecting the upper left block (128x128)	103

Fig. 4.20b	Reconstruction of Tools image using III algorithm by rejecting the upper left block (64x64)	103
Fig. 4.21a	Reconstruction of Baby image using III algorithm by rejecting the upper left block (64x64)	103
Fig. 4.21b	Reconstruction of Baby image using III algorithm by rejecting the upper left block (32x32)	103
Fig. 4.22	Contour extraction using III algorithm for the Tools image	104
Fig. 4.23	Contour extraction using III algorithm for the Shoe image	104
Fig. 4.24	Contour extraction using III algorithm for the Baby image	105
Fig. 4.25	Contour extraction using III algorithm for the Montage image ...	105
Fig. 4.26a	Using III algorithm and PHL transform: Image compression using LPF	107
Fig. 4.26b	Using III algorithm and PHL transform: Image reconstruction using HPF	107
Fig. 4.26c	Using III algorithm and PHL transform: Image reconstruction using LPF and HPF filters	107
Fig. 4.26d	Contour extraction from (b) image	107
Fig. 4.27	Block diagram of the analyzed algorithm	109
Fig. 4.28a	Real test images (256x256)	110
Fig. 4.28b	Extracted contours from both images	110
Fig. 4.28c	Image combination before and after the additional procedure	110
Fig. 4.29a	Real test images (256x256)	111
Fig. 4.29b	Extracted contours from both images	111
Fig. 4.29c	Image combination before and after the additional procedure	111
Fig. 4.30a	Real test images (256x256)	112

Fig. 4.30b	Extracted contours from both images	112
Fig. 4.30c	Image combination before and after the additional procedure	112
Fig. 4.31a	Real test images (256x256)	113
Fig. 4.31b	Extracted contours from both images	113
Fig. 4.31c	Image combination before and after the additional procedure	113
Fig. 4.32a	Real test images (256x256)	114
Fig. 4.32b	Extracted contours from both images	114
Fig. 4.32c	Image combination before and after the additional procedure	114
Fig. 4.33	Results using the analyzed algorithm	115
Fig. 4.34	Block diagram of the HPF algorithm	117
Fig. 4.35	Block diagram of the bit-planes decomposition algorithm	117
Fig. 4.36	Test images: (a) Rice (256x256) (b) Man face (128x128)	118
Fig. 4.37a	Reconstruction for Rice image using the HPF algorithm by LPF	119
Fig. 4.37b	Reconstruction for Rice image using the HPF algorithm by HPF	119
Fig. 4.37c	Reconstruction for Rice image using the HPF algorithm by LPF and HPF	119
Fig. 4.37d	Binary Rice image using the threshold for (c)	119
Fig. 4.38a	Image reconstruction for Rice image using the HPF algorithm ...	120
Fig. 4.38b	Image reconstruction for Rice image using bit-planes algorithm (number eight)	120
Fig. 4.39a	Reconstruction of Man face image using the HPF algorithm	120
Fig. 4.39b	Reconstruction of Man face image using the bit-planes	

	algorithm (number eight)	120
Fig. 4.40a	Woman face image	121
Fig. 4.40b	Reconstruction of Woman face image using the HPF algorithm	121
Fig. 4.40c	Reconstruction of Woman face image using the bit-planes algorithm (number eight)	121
Fig. 4.41a	Test image of Rice image using the bit-planes algorithm (bit-plane number seven)	122
Fig. 4.41b	Test image of Rice image using the bit-planes algorithm (the most significant bit (number eight))	122
Fig. 4.41 (c and d)	Test image of Rice image using the bit-planes algorithm (the complement for (a and b) respectively)	122
Fig. 4.42a	Test image of Man face image using the bit-planes algorithm (bit-plane number seven)	123
Fig. 4.42b	Test image of Man face image using the bit-planes algorithm (the most significant bit (number eight))	123
Fig. 4.42 (c and d)	Test image of Man face image using the bit-planes algorithm (the complement for (a and b) respectively)	123

LIST OF TABLES

	Page
Table 1.1 Comparison between the algorithms for the Circle image	20
Table 1.2 Comparison between the algorithms for the Rectangle image	20
Table 1.3 Comparison between the algorithms for the E letter image	21
Table 1.4 Comparison between the algorithms for the all test images	24
Table 2.1 Results after approximation for the Serpent contour	35
Table 2.2 Results after approximation for the Apple contour	36
Table 2.3 Results of approximation for the Italy contour	41
Table 2.4 Results of approximation for the Serpent contour	42
Table 2.5 Approximated contour using the three algorithms for the Libya map contour	47
Table 3.1 Approximated contour using triangle height over length method for the Rose contour	54
Table 3.2 Approximated contour using Ramer method for the Rose contour	55
Table 3.3 Results of approximation for the Italy contour	57
Table 3.4 Results of approximation for the Rose contour	58
Table 3.5 Results of approximated Rose contour using Ramer and analyzed methods	59
Table 3.6 Results of approximation for the Italy contour	61
Table 3.7 Results of approximation for the Rose contour	62
Table 3.8 Results of approximated Rose contour using Ramer and analyzed methods	65
Table 3.9 Results of approximation for the Italy contour	66
Table 3.10 Results of approximation for the Rose contour	68

Table 3.11	Results of approximated Rose contour using Ramer and analyzed methods	69
Table 3.12	Results of approximation for the Italy contour	72
Table 3.13	Results of approximation for the Rose contour	73
Table 3.14	Results of aapproximated Rose contour using Ramer and analyzed methods	75
Table 4.1	Results of approximation for the Italy contour using PHL transform	85
Table 4.2	Results of approximation for the Italy contour using Walsh transform	88
Table 4.3	Results of approximation for Italy contour using DCT transform	90
Table 4.4	Results of approximation for the Italy contour using Haar transform	95
Table 4.5	Results of reconstruction for Montage image	107
Table 4.6	Comparison using the third algorithm	107
Table 4.7	Results for the Rice image of Fig. 4.38	119
Table 4.8	Results for the Man face image	120
Table 4.9	Results for the Women face image	121

LIST OF LISTINGS

		Page
Listing 1.1	Implementation of the eight rules for contour extraction (4x4 windows)	13
Listing 1.2	Implementation of the eight rules for contour extraction (3x3 windows)	17
Listing 1.3	Implementation of the eight rules for contour extraction (2x2 windows)	18

INTRODUCTION

The main purpose of this thesis is to investigate some methods for contour approximation and compression and compare the analyzed methods with the well known methods.

The basic shape important for data compression is the compression ratio (CR), or it can be referred to as the ratio of the size of a compressed file to the original uncompressed file.

Data compression decreases the size of data that can be transmitted over a network link. The time required to transmit a data across the network can be shortened by reducing the size of the data. Image processing is a quickly progressing field with growing applications in different markets (professional and consumer).

Computer-aided tomography (CAT) is extensively used in life sciences. Its applications in the professional market and digital photography prove to be a significantly successful application in the consumer market.

There exist fundamental issues studied in this field that embrace problems related to visual perception, sampling and quantization of visual data, image restoration and enhancement, image segmentation, image recognition, and image transmission and storage (such as image compression, watermarking).

There exist many reasons gives interpretation why active contours provide suitable procedure to the process of front view region extraction. The most important one is that the front view is a well defined curve, i.e. is responsible for the characteristics of the curve approximation of active contours.

Applications of shape description and representation offer an important tool in the wide fields of pattern recognition and computer vision. Shape recognition can be represented in character recognition, contour matching for medical imaging 3-D

reconstruction and many other visual tasks. Contours and regions give two main approaches for the shape representation.

New algorithms of contour compression and approximation in both time and spectral domains are introduced after having been paid a lot of attention.

When working with a region or object, several representations are present and can simplify the treatment and measurements of the object.

The main target of this work is to introduce the developed set of tools that may be used to describe and compress any contour in time and spectral domains.

The separated contours give description for the whole image when the contour is defined with three streams of information:

1. Initiating point of the contour of the scan line sample image (i.e. the top left hand corner of the contour).
2. Height boundary value.
3. Contour boundary description.

Using these streams of information the contours can be separated after isolation by contourization. The header contains the horizontal and vertical resolutions of the entire image after preceding the streams of information mentioned above. Then using three stream encoders the three streams of contour information are encoded.

Each contour can be determined by expliciting its grey-level of x and y co-ordinates of the initiating point and also a sequence of the external limits of the contour.

Placing objects in a scene is the goal of the object detection. There are many important questions that relate to this goal. How the object is defined? What should be done to represent something as an object? Designing a system in the computer vision is the main achievement that would be capable of analyzing a scene and determining which objects fit the scene. Definitely the definition of the object - whether it is a pen, foot, or a mobile - depends on the designer's system. In order to find these objects the system must use some algorithms we will discuss about in the first chapter. They are assigned to either of two major classes known as object contour following (OCF) [38] and [39], and multiple

step contour extraction (MSCE) [1]. Another algorithm was introduced by Nabout and others [1] and [40] have developed an object-oriented contour extraction (OCE) algorithm. A new algorithm has imitated a new class of contour extraction algorithms, single step parallel contour extraction (SSPCE) [33] and [41]. Different windows of pixels are used by these algorithms in order to extract rules and suppress various noises. These algorithms are built up to look for specific objects, and in general, they have to be applied one at a time. For that reason, for how many different types of objects can be detected is influenced in the computational power of the system.

Measuring the difference in grey-levels between a processed pixel and its surrounding neighbours is referred to as grey-level metric. The larger difference in grey-levels is proportional to the larger metric value. The image is represented as a collection of traditional rectangular pixels by grey-level metrics which has been proposed in the recent years. The more explained information for some of these grey-level metrics such as Robert, Sobel, Prewitt and Canny which are defined in terms of rectangular pixels can be found in the following sources [2], [6], [7], [8], and [9].

Image of line (or curve segments) that may be connected or disconnected gives definition for the line drawing. The storage of line drawings can be used in many applications such as computer graphics, image processing, pattern recognition, and automated cartography in geographic information systems (ACGIS).

Freeman [30] and [32] proposed several methods called chain coding in order to represent contour data representations for better transmission and storage of such data. To do that, quantization and encoding the 2D figure is required. Superimposition on the line drawing is a uniform mesh. Curve points are used to approximate and quantize the line drawing. Different quantization scheme gives different representations by using mesh nodes which allow the chosen curve points are represented by points on the drawn line.

There is no noticeable development in the application of transform methods for the compression of contour data, as far as the contour data compression is concerned.

The contour approximation and compression that require the existing procedures concentrate on the coding schemes in the time domain. The well known algorithm is Ramer [31], who has presented repeated end points that suit the algorithm. Connecting a number of points by joining all the line end points gives the approximation of such data and that is the basic idea of this algorithm. There exists algorithm relevant to minimum length polygonal approximation of contours which is used as a fit criterion of the algorithm was introduced by Montanari [46]. There exist other algorithms also concerning the approximation and/or smoothing of curves such as, Sirjani [45], and tangent method [34], [36] and [37].

There exists a type of data compression in order to compress the data like audio signals or photographic images, which is refereed to as transform coding.

The conversion from spatial image pixel values to transform coefficient values can be done by using transform coding. There is no loss in the information because of the linear process, so the number of coefficients produced is equal to the number of pixels transformed.

One of the goals of the transformation is that most of the energy in the image will be contained in a few large transform coefficients. In general when these few coefficients contain most of the energy in most pictures, they may be further coded by lossless entropy coding. In lossy coding the smaller coefficients can be coarsely quantized or even deleted without doing any visible distortion for the reconstructed image.

Several types of transforms have been tested for picture coding, including for example Fourier [32], Karhonen-Loeve [32], Haar [56], periodic Haar piecewise-linear (PHL) [53], Walsh [49] and [50], discrete cosine transform (DCT) [54], and recently, wavelets [24]. There are three distinct points that are of interest in picture coding gives differ among the various transforms:

- 1) The density of concentration of energy in a minimum number of coefficients;
- 2) The effectiveness of region for each coefficient in the reproduced image;

3) The coarse quantization of the coefficients and its influence in the appearance and visibility of coding noise.

Coefficients in an image are represented as a two-dimensional array, in which each of them gives the brightness level in that point. In general, we cannot differentiate between coefficients which are more or less important. But after further detailed interpretation, we can. Most natural images have smooth color variations (low frequency) with some details represented as sharp edges between the smooth variations (high frequency). Base of an image is determined by the low frequency components while detailed image is giving by the high frequency components. Realistically, more importance are requested for the smooth variations than that the details.

In high pass filtering the main idea is to obtain the low frequency (slowly changing areas) of the image and to bring out the high frequency (fast changing details in the areas) of the image.

The main goals which are considered in this work are:

1. Comparison between the existed contour extraction methods using the number of operations.
2. Two proposal algorithms for contour approximation and compression in time domain and comparing it with well-known Ramer algorithm.
3. The new family known as “triangle family” containing four algorithms for contour approximation and compression in time domain and comparing it with well-known Ramer algorithm.
4. A proposed algorithm for contour extraction using high pass filter and also algorithm for combination real images using high pass filter and correlation function.
5. Two proposed algorithms to reconstruct image by contour extraction using high pass filter, double transforms and bit-planes decomposition.

According to the organization of the thesis we introduce the description of the layout of this work to fulfill the research objectives.

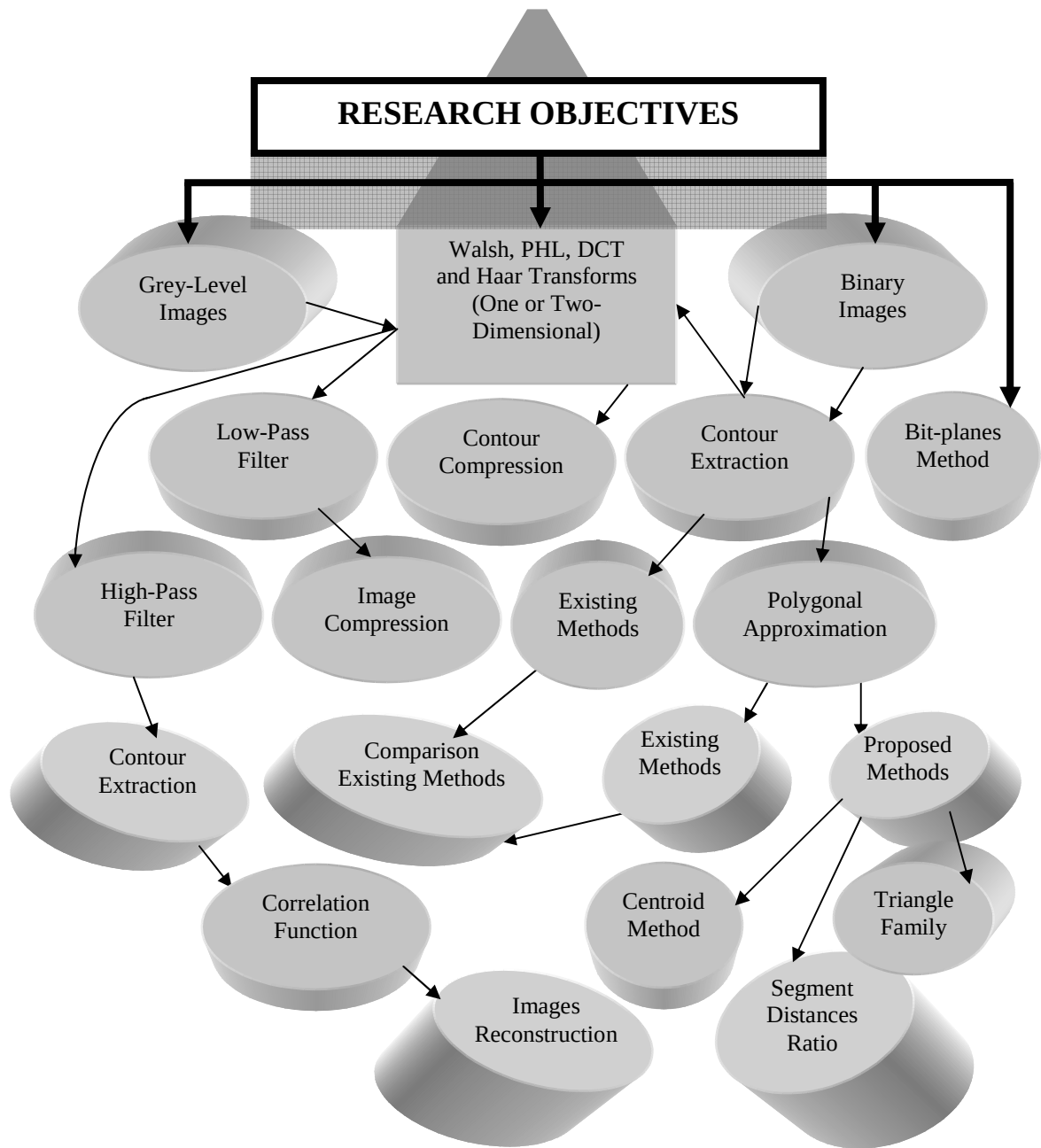
Chapter (1) presents a brief discussion of general background and basic concepts involved in contour data extraction. This chapter also presents a brief review of the two main classes of contour extraction methods. The first class is an object contour following (OCF) where as the second class is multiple step contour extraction (MSCE). Also general background for contour extraction using edge detection operators is given. At the end of this chapter the comparison between the contour extraction methods is presented.

Chapter (2) provides a brief discussion of general background and most existing methods for the contour data approximation and compression in time domain. These methods are Ramer and tangent methods. Two algorithms of the same class are proposed. The two algorithms are “contour approximation using centroid method” and “contour compression using segment distances ratio”. A performance comparison between the two proposed algorithms and the Ramer and tangent algorithms is presented.

Chapter (3) introduces a proposed family for contour data approximation and compression in time domain. This family is known as “triangle family”. Four algorithms of this family are proposed. A performance comparison between these algorithms and the Ramer algorithm is presented.

Chapter (4) presents a brief review of orthogonal transforms commonly used for image compression. One and two-dimensional for contour and image respectively use periodic Haar piecewise-linear (PHL), Walsh, discrete cosine transform (DCT) and Haar transforms are discussed in detail. This chapter concludes with a comparison of the compression techniques for the all transforms with the algorithms in time domain (Ramer, segment distances ratio and triangle family). Three algorithms of contour extraction and image compression using low-pass filter (LPF) and high-pass filter (HPF) are proposed and compared with Sobel and Canny detectors in this chapter. These algorithms used periodic Haar piecewise-linear (PHL) transform, discrete cosine transform (DCT) and Haar transform. This chapter also presents an another algorithm for

real grey-level images construction by combining it's extracted contours which have been obtained using high-pass filters and by using one of the following transforms (PHL or DCT or Haar). At the end of this chapter introduces two algorithms that can reconstruct the image using the contours extraction by HPF and only the most significant bit.



Block diagram of the main objectives of the thesis

Chapter1

CONTOUR EXTRACTION

1.1 Overview

In this chapter we briefly summarize previous methods of contour extraction. There exist two main fields of research for the 2D-object contour extraction problem: The object contour following (OCF) or sequential methods and the multiple step contour extraction (MSCE) or parallel methods. Later, they develop an object-oriented contour extraction algorithm (OCE). Also there exists an algorithm which initiated a third class of contour extraction algorithms, single step parallel contour extraction (SSPCE).

Finally, the comparison between the OCE (4 x 4 windows), SSPCE (3 x 3 windows) and contour extraction based on (2 x 2 windows) is performed.

To present the results, the comparisons between these contour extraction methods are done using the number of arithmetic operations versus number of edges.

At the beginning of this chapter, we summarize previous edge detection, and object recognition.

1.2 Edge Detection

The most important study field of computer vision is the detection of object (2-dimensional) image as edge points (pixels) of a 3-dimensional physical object. The correctness and completeness of edges constitutes is an essential tool to extract object contours and object recognition [3]. For simplicities and facilitates of image analysis, the edge detection is very important [5].

Once a large change in image brightness has occurred, using the edge detection we can extract and localize points (pixels). The relationship between a pixel and its surrounding neighbours interprets the edge detection. The pixel is represented as an edge point if the regions around a pixel are not similar, otherwise, the pixel is not appropriate to be recorded an edge point.

1.3 Grey-level metrics

Detection, recognition and extraction of 2D-objects in grey-level images are the main important problems in computer vision. Model constructions and trainings as well as computational approach for a better and parallel implementation in biologically explicit neural network architectures are discussed in many books [4].

Geometric properties determine the properties of objects in our world. The applications of shape analysis spread over the scientific and technological area in which the values of scale form a hierarchy from the smallest to the largest spatial scale [10].

The important problem in grey-level image and contour analysis is edge detection. Edges characterize object boundaries. So they are useful for segmentation, registration, and identification of objects in a scene in ideal images.

The change in intensity defines the edge and its cross section has the shape of a ramp. Usually, discontinuity characterizes an ideal edge like a ramp with an infinite slope. We are looking for the local maximum at an edge by determining the first derivative. For a continuous image $f(x, y)$, where x and y determine the row and column coordinates respectively, the two directional derivatives $\partial_x f(x, y)$ and $\partial_y f(x, y)$ are taken in account. There are two functions that can be expressed in terms of these directional derivatives, the gradient magnitude and the gradient direction. The gradient magnitude is defined as

$$|\nabla f(x, y)| = \sqrt{(\partial_x f(x, y))^2 + (\partial_y f(x, y))^2} \quad (1.1)$$

and the gradient direction is defined by

$$\angle \nabla f(x, y) = \arctan \left[\frac{\partial_y f(x, y)}{\partial_x f(x, y)} \right] \quad (1.2)$$

To identify edges, the local maxima of the gradient magnitude in $f(x,y)$ are necessary. The second derivative is zero while its first derivative gives the maximum value. For that purpose, there exists an alternative edge detection to locate zeros of the second derivatives of $f(x,y)$. The differential operator used by these strategy so-called zero-crossing edge detectors is the Laplacian edge detector and is defined by

$$\Delta f(x,y) = \partial_{(x,2)} f(x,y) + \partial_{(y,2)} f(x,y) \quad (1.3)$$

In practice, one can use the first-order directional derivatives for finite difference approximations and these are represented by a pair of masks [h_x (horizontal) and h_y (vertical)]. Formally these are linear-phase FIR digital filters. A convolution of the image with these masks provides two directional derivative g_x and g_y (both directions) respectively. The gradient equation is $\nabla = \sqrt{g_x^2 + g_y^2}$ or for simpler implementation using $\nabla = |g_x| + |g_y|$ [52].

Edge location is given by a pixel location when the point value (x, y) exceeds some threshold. The edge map describes the locations of all edge points. The number of important things gives interpretation how the threshold value is selected by the design decision such as image brightness, contrast, level of noise, and edge direction. Sometimes it is very useful to work out the information of the edge direction given by

$$\phi = \arctan \left[-\frac{g_x}{g_y} \right] \quad (1.4)$$

Edges drives to boundaries in images and represent areas with strong intensity contrasts, a varying intensity from one pixel to the other. It is very essential to reduce the amount of data and eliminate the useless information by edge detection of an image, while keeping the necessary structural properties in that image. Different methods exist to do edge detection. However, these methods may be arranged into two classes, gradient and Laplacian. The gradient method works on the first derivative of the image and the

edge is detected by looking for the maximum and minimum, while the Laplacian method uses the second derivative of the image, and by searching for zero crossings the edges are detected. One-dimensional shape of a ramp defines an edge by calculating the derivative of the image, and can highlight its location.

The visual system of the human is very sensitive to details (ex. edges) which are located in the high frequencies of the frequency domain.

As mentioned before, there is only one way to detect edges or variations within a region of an image by using the gradient operator. For example, the gradient G is a vector with two elements, G_x (width direction) and G_y (height direction).

There exist many well known gradient filters (i.e. Roberts, Sobel, Canny and Prewitt...and the others). Performing convolution of the image with some kernels (masks) gives a certain gradient.

In general, each kernel computes the gradient in a typical direction (horizontal and vertical) and later these partial results are combined to produce the final result in which, each of them computes an approximation to the true gradient by either using absolute differences or Euclidean distances. Absolute value computations are easy implementation and faster operations when compared to square and square root operations. Hence to get a faster computing the gradient magnitude, sum the absolute values of the gradients in the X (width or horizontal) and in the Y (height or vertical) directions is performed.

There are a pannel of gradient operators used to detect an edge. By rotating the kernel values some variations are obtained and among them the Roberts, Sobel, Prewitt, and Isotropic gradient kernels as shown in Fig. 1.1.

The small box indicates the kernel origin, while the arrow indicates the direction that each gradient kernel evaluates.

Measuring the difference in grey-levels between a pixel and its surrounding neighbours uses a grey-level metric. The difference in grey-levels is proportional to the metric value.

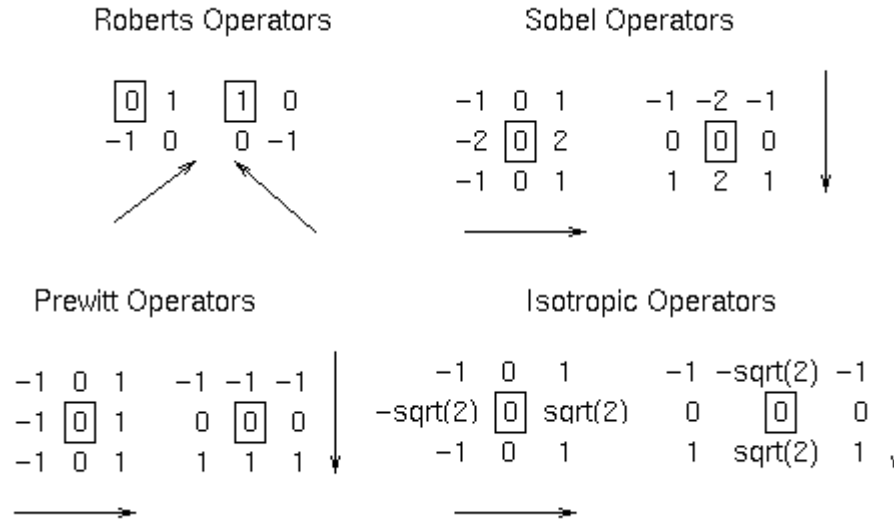


Fig. 1.1 Gradient operators

There exist many grey-level metrics recently proposed to detect an edge for an image which is defined as a collection of traditional rectangular pixels. The detailed information for some of these metrics can be found in [2], [6], [7], [8], and [9]. They are summarized as follows:

1. *Roberts Metric*

In general there are two ways to obtain this metric: by the square root of the sum of the differences of grey-levels of the diagonal neighbours squared, and by the sum of the magnitude of the differences of the diagonal neighbours.

Simplicity and quickness are obtained using the Robert operator to compute the 2-D spatial gradient measurement on an image. The edges are often defined by the highlight regions of high spatial gradient. In general, the input and output of the operator is a grey-scale image. The estimated absolute magnitude of the spatial gradient of the input image at a point therefore, offers for each point, the pixel representation values in the operator output.

Theoretically, the operator is made up of a pair of 2×2 convolution masks as shown in Fig. 1.2. Rotating one mask by 90° gives the other one.

+1	0	0	+1
0	-1	-1	0
G_x		G_y	

Fig. 1. 2 Roberts Cross convolution masks

These masks are designed in a certain way that can respond maximally to edges running at 45° to the pixel grid (Hint: one mask is applied for each of the two perpendicular directions). In order to produce separate measurements of the gradient component output in each orientation (G_x and G_y), the masks can be applied separately to the input image. Then these can be mixed to obtain the absolute magnitude of the gradient at each point and the direction of that gradient.

2. *Sobel Metric and Prewitt Metric*

Both metrics are defined as the square root of the sum of squared two values which are obtained by convolving the image with a row mask and column mask respectively.

Using one-dimensional analysis, the theory can be carried over to two-dimensions whenever it's necessary to get an accurate approximation to reckon up the derivative of the image. The Sobel and Prewitt operators achieve a 2-D spatial gradient measurement on an image and in general, it is used to find the approximate absolute gradient magnitude at each point in an input grey-scale image. The Sobel and Prewitt edge detectors use a pair of 3×3 convolution masks (respectively, the gradient in the x-direction and the y-direction). In general the size of the convolution mask is usually much smaller than the actual image. As a result, manipulating a square of pixels at a time passes the mask through the image.

Theoretically, the operators are made up of a pair of 3×3 convolution masks as shown in Fig. 1.3 (using Sobel metric). Rotating one mask by 90° gives the other one.

-1	0	+1	+1	+2	+1
-2	0	+2	0	0	0
-1	0	+1	-1	-2	-1

G_x G_y

Fig. 1.3 Sobel Cross convolution masks

3. Laplacian Metric

A Laplacian mask is used to obtain this metric. Approximation of the first-order derivative pixel values in an image is used by all of the previous edge detectors. However it is also possible to use second-order derivatives to detect edges of the image. Using the second order operator the most popular one is the Laplacian operator. The Laplacian of a function $f(x,y)$, is given by

$$\nabla^2 f(x,y) = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2} \quad (1.5)$$

As a gain to estimate the derivatives and represent the Laplacian operator with the (3X3) convolution mask (see Fig. 1.4), we can use discrete difference approximations.

0	1	0
1	-4	1
0	1	0

Fig. 1.4 Laplacian operator convolution masks

However there are two main drawbacks using second order derivative that in summary can be defined as:

1. If the first derivative operators exaggerate the effects of noise, the second derivatives will exaggerate noise twice as much.
2. The directional information about the edge is not available.

So we still have a problem when the noise caused is apparent, when using edge detectors and i.e. we should try to reduce the noise in an image prior to or in conjunction with the edge detection process. Another suitable smoothing method for that is Gaussian smoothing, which can be obtained by convolving an image with a Gaussian operator. It is possible to detect edges by using Gaussian smoothing in conjunction with the Laplacian operator, or another Gaussian operator.

The Gaussian smoothing operator is a 2-D convolution operator that is used to ‘blur’ images and remove detail and noise. In general it is similar to the mean filter, but it uses a different kernel that represents the shape of a Gaussian hillock. The amount of smoothing can be controlled by varying the standard deviation σ (width of the Gaussian distribution).

4. *Canny Metric*

Canny metric is optimal for step edges that are infested by white noise. To know the metric efficiency as optimum one to detect edges, Canny has defined three criteria [11], [12], [13,] and [14] for optimal edge detection in a continuous domain:

1. *Right Detection*: the ratio of edge points to non-edge points gives the maximum value on the edge map.
2. *Right Localization*: the true locations must be as close as possible to the detected edge points.
3. *Low-responses Multiplicity*: the distance between two non-edge points on the edge map gives the maximum value.

The Canny operator is made an optimal edge detector. It produces as output an image showing the positions of tracked intensity discontinuities after having taken as input the grey-scale image.

Because the simple edge detection operators previously introduced perform in the discrete domain, the Canny criteria can not be used in this situation and for that reason Demigny and Kamle [13] define the corresponding criteria in the discrete domain.

A lot of problems in edge detection happens when noise occur by imaging and sampling processes.

A multi-stage process is needed for working Canny operator. At the beginning Gaussian convolution is applied to smooth all image. Later, the smoothed image to highlight regions of the image with high first spatial derivatives is performed by a 2-D first derivative operator (like Sobel, Prewitt and Roberts Cross). Increase in rising of the gradient magnitude image is caused by the edges. Then by tracing the algorithm along the top of these ridges, it sets to zero all pixels that are not actually on the ridge top which gives a thin line in the output. This process is known as non-maximal suppression. This process leads to well known hysteresis phenomena which is controlled by two thresholds (T_1 and T_2 where $T_1 > T_2$). If the point on a ridge is higher than T_1 then the tracing can start. In both directions the tracing process is continued from that point until the height of the ridge falls below T_2 . There is one important advantage obtained by using this hysteresis which ensures that noisy edges are not broken up into several edge segments.

Speed, accuracy and reliability are expected in edge detection, which represents one of the fundamental operations in computer vision. To obtain these requirements a parallel implementation of edge detection is done. The parallelizing edge detection process is obtained by splitting the image into segments and then detecting the edges independently in each of the segments [3]. We should be careful and make sure that there is no loose of any edges presented at the segment boundaries during splitting the image and that is why the image should be split with overlapping boundaries.

1.4 Contour extraction using different windows

There is an important role used in image extraction, recognition and analysis which depends on contour extraction from 2D-image.

Multi-usage of contour extraction can be found in visual information processing [15], [16], [17], and [18]. The information that corresponds to shape object can be found in the contour of that object [19]. It is necessary for needs and requirements of practitioners and

researchers in industry, and graduate-level students in computer science to meet the shape analysis objects designs [21].

As mentioned before the important step to obtain some features of object recognition is by using the contour extraction from a given 2D-digital images. Using different windows sizes, there are two main methods of research for contour extraction from a 3D-object in a 2D- image [1]:

1. Object contour following (OCF) which is classified as sequential methods [38] and [39], and
2. Multiple step contour extraction (MSCE) which is classified as parallel methods [1].

Later, an object-oriented contour extraction 'OCE' has been developed which uses 4x4 windows [1] and [40]. Hence, these methods can be classified either as sequential or parallel methods. Sequential means following or tracing and occurring when the result at a point is dependent upon the decision result of the previously processed points. Parallel means simultaneously and occurs when the decision of whether or not a point is on an edge is made on the basis of the grey-level of the point and its surrounding neighbours. As a result we can say that the edge detection operator can be applied at the same time everywhere in the picture. There is another class of contour extraction algorithms which is known as single step parallel contour extraction 'SSPCE' [33] and [41], which uses a 3x3 window, which we will describe in this chapter. A new algorithm exists to be simpler and more suitable for real time applications than the previous ones which use 2x2 windows [41].

Some books have been written in order to improve a generative theory of shape with two properties regarded as fundamental to intelligence - increasing the transfer of structure in order to increase recoverability of the generative operations [28] and [35].

Freeman chain coding [30] and [32] can be used to determine all possible connections for both 8-connectivity and 4-connectivity schemes. An 8-directional chain coding uses eight possible directions while 4-directional chain coding uses four possible directions to

present all possible line segments connecting nearest neighbours according to the 8-connectivity scheme and 4-connectivity scheme respectively as shown in Fig. 1.5.

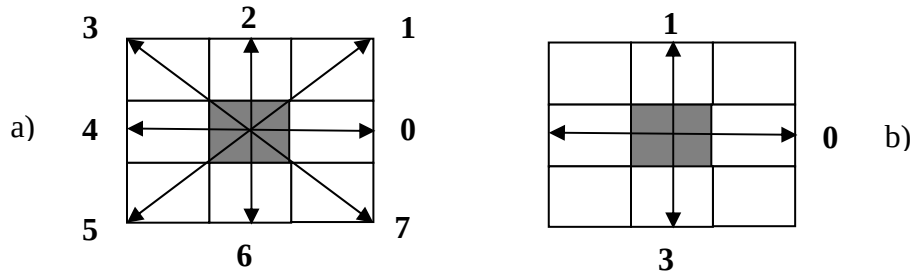


Fig. 1.5 Chain coding using the a) 8-Connectivity scheme, and b) 4- Connectivity scheme

1.4.1 Object contour following 'OCF' method

This method is used to follow the contour edges of a digital image using 4x4 pixel window structures to extract the object contours. The idea of the extraction procedure using this method begins by finding a starting point then crossing edge between the white and black regions, recording the co-ordinates of the black pixel and later turning left continuously until a white pixel is found. Then recording the black pixel co-ordinates as the next contour edge point. Later, begin turning right until a black pixel is found. Finally, end the procedure when the starting point of the contour is reached again. This method has some problems such as searching a proper starting point, which is not always an easy task and hence an additional procedure for selecting an appropriate starting point is required. The second problem is the criterion of termination which is required to determine whether the procedure is finished (in some cases the procedure should be terminated if it is unable to reach the starting point). The idea of these methods is illustrated in Fig. 1.6.

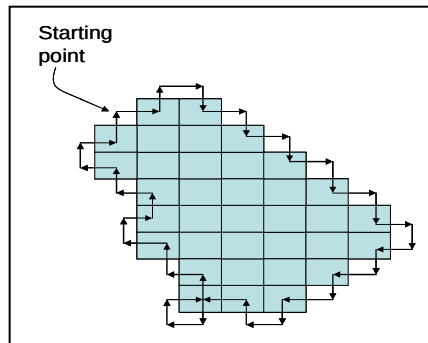


Fig. 1. 6 Object contour follower (OCF)

OCF algorithms have four main drawbacks [1]:

1. Before starting the algorithm, the processed image should be stored;
2. A false contour (implement contour and object starting point) may occur;
3. Extended computation time whenever several contours are to be extracted;
4. The execution time is increased when pixels are visited more than once.

The effect of noise is one of the main problems that increase during extracting object contours in real applications [20]. The effect of the second drawback leads to incomplete contours when the 2D- digital image is noisy [22]. Because of the sequential nature of the OCF algorithms, the last two drawbacks happen.

1.4.2 Multiple step contour extraction 'MSCE' method

There are three essential steps for contour extraction using MSCE methods [1]:

By means of tracing, edges points are connected and then contour points are coded using for instance Freeman chain coding.

Fig. 1.7 shows the steps needed for an object contour extraction using these methods.

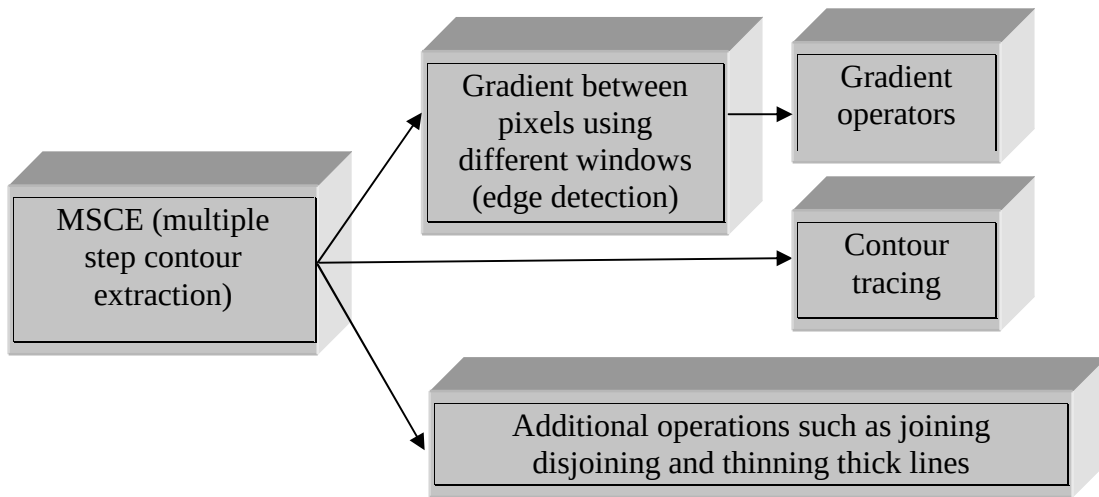


Fig. 1.7 Block diagram of the multiple step contour extraction (MSCE) method

The difference between two pixels is the gradient between them with respect to different grey-scale levels, and the gradient for the pixels which has the same grey-level will be zero. An additional procedure (such as joining disjointed edges and thinning the

thick edges) is necessary to improve the contour structure. This algorithm uses either a 2x2 or 3x3 pixels windows structures to extract the object contours. In spite the fact that the method of the gradient operator for generating edge elements is parallel, the method of connecting (tracing) these extracted edge elements is sequential.

Like the OCF algorithms, the quality of contour extraction is affected by the image noise. Different problems may happen due to the image noise, such as blurred contour points and discontinuous contours [1].

1.4.3 Object-oriented contour extraction 'OCE' method (4x4 windows)

Only a 4x4 pixels window structure is used by this algorithm to extract the object contours by the four central pixels which are processed simultaneously. The algorithm uses the features of both OCF and MSCE methods to overcome most of the disadvantages they have. It features a parallel implement and an effective suppression of noises. OCE method can be realized in real-time [1].

The following three steps are needed for the extraction procedure:

Step1: Framing the image with zeros, **Step2:** Using 8-directional chain-code the eight rules of edge extraction is coded as shown in Listing 1.1.

Listing 1.1
Implementation of the eight rules for contour extraction (4x4 windows)

```

a(i,j) ← 0; i = 1,2,...,N; j = 1,2,...,N;
for i = 2,3,...,N-1; j = 2,3,...,N-1;
{
if b(i,j+1) and b(i+1,j+1) and [b(i,j+2) or b(i+1,j+2)]
then a(i,j+1) ← a(i,j+1) or 20           { edge 0 }
if b(i+1,j) and b(i,j+1) and b(i+1,j+1)
then a(i,j+1) ← a(i,j+1) or 21           { edge 1 }
if b(i+1,j) and b(i+1,j+1) and [b(i+2,j) or b(i+2,j+1)]
then a(i+1,j+1) ← a(i+1,j+1) or 22       { edge 2 }
if b(i,j) and b(i+1,j) and b(i+1,j+1)
then a(i+1,j+1) ← a(i+1,j+1) or 23       { edge 3 }
if b(i,j) and b(i+1,j) and [b(i,j-1) or b(i+1,j-1)]
then a(i+1,j) ← a(i+1,j) or 24           { edge 4 }
if b(i,j) and b(i+1,j) and b(i,j+1)
then a(i+1,j) ← a(i+1,j) or 25           { edge 5 }
if b(i,j) and b(i,j+1) and [b(i-1,j) or b(i-1,j+1)]
then a(i,j) ← a(i,j) or 26               { edge 6 }
if b(i,j) and b(i,j+1) and b(i+1,j+1)
then a(i,j) ← a(i,j) or 27               { edge 7 }
}

```

The extracted edge code is represented by 2^k ($k:0-7$) while $b(i,j)$ is represent the binary value of a pixel point (i,j) .

Step3: The extracted contour edges are sorted and stored or optimized according to the application requirements. The extraction procedure is shown in Fig. 1.8.

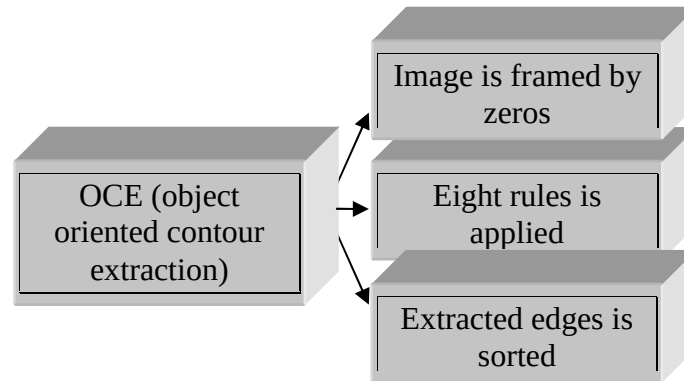


Fig. 1.8 Object-oriented contour extraction (OCE) method

The extracted contours from objects using this procedure are near the image boundary. So, the objects within one pixel distance from the image border are not closed and that is interpreted why the image should be framed with zeros to ensure all of the contours are closed. Fig. 1.9a and Fig. 1.9b shows the extracted edges without framing (one of the extracted contour is not closed) and after framing the image with at least two underground pixels (all extracted contours are closed) respectively.

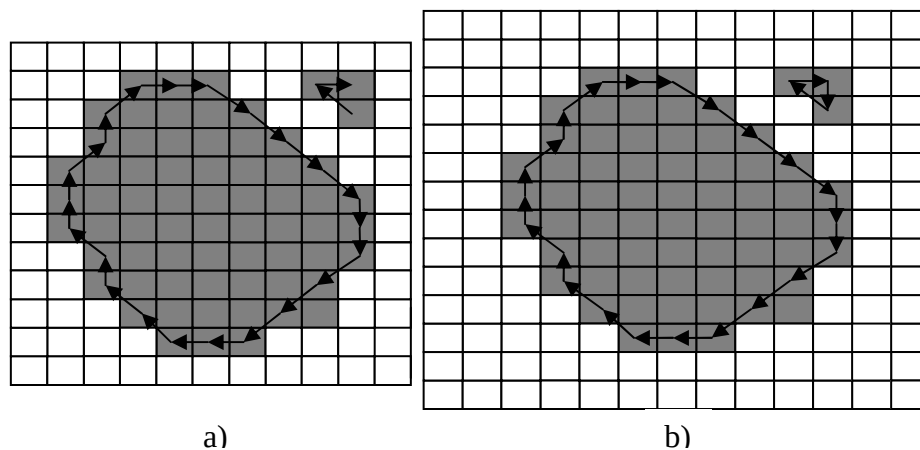


Fig. 1.9 OCE procedure (a) Without correcting the first step, and (b) After correcting the first step

1.4.4 Single step parallel contour extraction 'SSPCE' method (3x3 windows)

There are two algorithms; the first one [41] which uses an 8-connectivity scheme between pixels, and 8-directional Freeman chain coding [30] scheme is used to distinguish all eight possible line segments connecting the nearest neighbours. This algorithm uses the same principle of extraction rules as the OCE algorithm. The second algorithm [41] uses a 4-connectivity scheme between pixels, and 4-directional Freeman chain coding [30] and [32] schemes are used to distinguish all four possible line segments. Both algorithms use a 3x3 pixels window structure to extract the object contours by using the central pixel to find the possible edge direction which connects the central pixel with one of the remaining pixels surrounding it.

The first algorithm is given exactly the same extracted contours as the OCE algorithms and is faster (3.8 times faster) while the second algorithm gives similar contours, but not identical and is faster (4.2 times faster) [41]. The first algorithm explanations are mentioned as follows

The edges can be extracted by applying the definition which says that an object contour edge is a straight line connecting two neighbouring pixels which have both a common neighbouring object pixels which have both a common neighbouring object pixel and a common neighbouring underground pixel [40]. By this definition, no edges can be extracted from the three following cases:

- 1- The window is inside an object region if the all pixels are object pixels;
- 2- The window is inside a background region if all nine pixels are background pixels; and
- 3- Most probable that the centre pixel is a point noise caused by image digitalization if the center pixel is an object pixel surrounded by background pixels.

So, this algorithm uses the same principle and steps of extraction rules as the OCE algorithm, by using the 3x3 window of pixels. The eight rules of edge extraction are applied and are coded using 8-directional chain-code as shown in Listing 1.2. The close contour can be extracted using eight rules and the contour edges are coding instead of 8-directional Freeman chain coding as shown in Fig. 1.10.

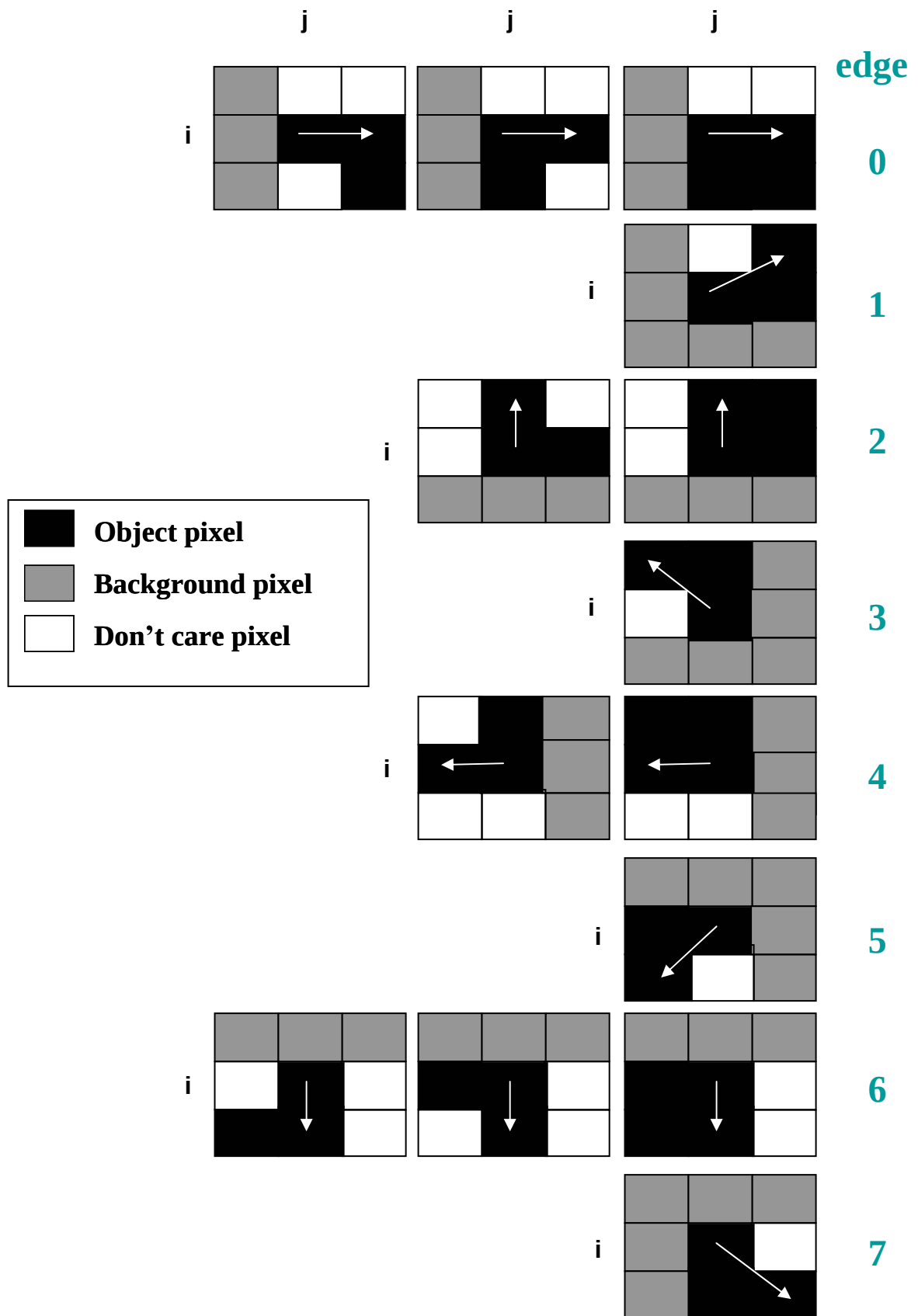


Fig. 1.10 The eight rules to extract closed contour using SSPCE method

Listing 1.2
Implementation of the eight rules for contour extraction (3x3 windows)

```

a(i,j) ← 0; i = 1,2,...,N; j = 1,2,...,N;
for i = 2,3,...,N-1; j = 2,3,...,N-1;
{
if b(i,j) and b(i+1,j) and [b(i,j+1) or b(i+1,j+1)] and [ not [b(i,j-1) or b(i+1,j-1)]]
then a(i,j) ← a(i,j) or 20                                { edge 0 }
if b(i,j) and b(i+1,j) and b(i+1,j-1) and [ not [b(i,j-1)]]
then a(i,j) ← a(i,j) or 21                                { edge 1 }
if b(i,j) and b(i,j-1) and [b(i+1,j) or b(i+1,j-1)] and [ not [b(i-1,j) or b(i-1,j-1)]]
then a(i,j) ← a(i,j) or 22                                { edge 2 }
if b(i,j) and b(i,j-1) and b(i-1,j-1) and [ not [b(i-1,j)]]
then a(i,j) ← a(i,j) or 23                                { edge 3 }
if b(i,j) and b(i-1,j) and [b(i,j-1) or b(i-1,j-1)] and [ not [b(i,j+1) or b(i-1,j+1)]]
then a(i,j) ← a(i,j) or 24                                { edge 4 }
if b(i,j) and b(i-1,j) and b(i-1,j+1) and [ not [b(i,j+1)]]
then a(i,j) ← a(i,j) or 25                                { edge 5 }
if b(i,j) and b(i,j+1) and [b(i-1,j) or b(i-1,j+1)] and [ not [b(i+1,j) or b(i+1,j+1)]]
then a(i,j) ← a(i,j) or 26                                { edge 6 }
if b(i,j) and b(i,j+1) and b(i+1,j+1) and [ not [b(i+1,j)]]
then a(i,j) ← a(i,j) or 27                                { edge 7 }
}

```

1.4.5 Contour extraction based on 2x2 windows

This algorithm is mainly used for grey-scale images [41]. It uses a smaller window for contour extraction, i.e. 2x2 window pixels and their structure and pixel numbering are shown in Fig. 1.11.

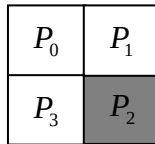


Fig. 1.11 Pixel numbering for 2x2 windows

The processed pixel is the darker one. Two buffers are required for a real time contour extraction system. The first buffer is used for the storage of a previously processed image line and the second one keeps pixel values of the currently processed image line.

The algorithm uses the 8-connectivity scheme, and the extracted edges are coded by using the 8-directional chain coding. It does not require any storage of the scanned (processed) image.

The three main steps of the algorithm are:

- The image is framed by zeros, contour edges are extracted using the eight rules [41], and finally the extracted contour edges are sorted.
- The algorithm does not require the storage of the scanned image, i.e. it can be used for real time applications.
- The eight rules of edge extraction are applied and are coded using 8-directional chain-code as shown in Listing 1.3.

Listing 1.3
Implementation of the eight rules for contour extraction (2x2 windows)

```

a(i,j) ← 0; i = 1,2,...,N; j = 1,2,...,N;
for i = 2,3,...,N-1; j = 2,3,...,N-1;
{
if ( b(i-1,j) = b(i,j) ) ∩ ( b(i-1,j) ≠ b(i-1,j-1) ) ∩ ( b(i-1,j) ≠ b(i,j-1) )
then a(i-1,j) ← a(i-1,j) ∪ b(i-1,j) ∪ 20                                { edge 0 }
if ( b(i-1,j) = b(i,j-1) ) ∩ ( b(i-1,j) ≠ b(i-1,j-1) )
then a(i-1,j) ← a(i-1,j) ∪ b(i-1,j) ∪ 21                                { edge 1 }
if ( b(i,j) = b(i,j-1) ) ∩ ( b(i,j) ≠ b(i-1,j) ) ∩ ( b(i,j) ≠ b(i-1,j-1) )
then a(i,j) ← a(i,j) ∪ b(i,j) ∪ 22                                { edge 2 }
if ( b(i,j) = b(i-1,j-1) ) ∩ ( b(i,j) ≠ b(i-1,j) )
then a(i,j) ← a(i,j) ∪ b(i,j) ∪ 23                                { edge 3 }
if ( b(i,j-1) = b(i-1,j-1) ) ∩ ( b(i,j-1) ≠ b(i,j) ) ∩ ( b(i,j-1) ≠ b(i-1,j) )
then a(i,j-1) ← a(i,j-1) ∪ b(i,j-1) ∪ 24                                { edge 4 }
if ( b(i,j-1) = b(i-1,j) ) ∩ ( b(i,j-1) ≠ b(i,j) )
then a(i,j-1) ← a(i,j-1) ∪ b(i,j-1) ∪ 25                                { edge 5 }
if ( b(i-1,j-1) = b(i-1,j) ) ∩ ( b(i-1,j-1) ≠ b(i,j-1) ) ∩ ( b(i-1,j-1) ≠ b(i,j) )
then a(i-1,j-1) ← a(i-1,j-1) ∪ b(i-1,j-1) ∪ 26                                { edge 6 }
if ( b(i-1,j-1) = b(i,j) ) ∩ ( b(i-1,j-1) ≠ b(i,j-1) )
then a(i-1,j-1) ← a(i-1,j-1) ∪ b(i-1,j-1) ∪ 27                                { edge 7 }
}

```

1.5 Contour descriptors

Generally, object descriptors are referred to object representations which give ways of describing object properties or features [23] and [24]. Object recognition has very

important area in order to obtain fixed object representations to affine transformations, such as translations, rotations and linear scaling [25] and [26]. Many representations such as object contours are obtained using 2D shape features of 3D-objects [23], [27], and [29]. Feature extraction and analysis are so easy that it gives an important advantage of such approach.

It is necessary to learn tools about analysis and statistics shapes for researches, engineers, scientists and medical researchers. Statistics and analysis of shapes are used as textbook of special topics course for a graduate-level in statistics and signal/image analysis [43] and [55].

The description of the extracted contours normally use the chain code, i.e. the x and y co-ordinates of the starting point followed by a string of chain codes representing the contour edges. The contours are described in Cartesian representation, the x and y co-ordinates, or polar representation, using a length of the line l from one point to the next in sequence and the angle α between every two lines [42] and [44]. There is another type of polar representation which describes the contour by using r_i (the length from a reference point inside the contour to the contour vertices) and ϕ_i (the angle between the two lines r_i and r_{i+1}).

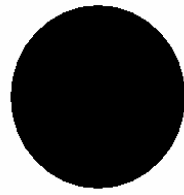
1.6 Comparison between the contour extraction algorithms

The comparison is made between the following three algorithms:

- Contour extraction (CE); it will be referred to as the first algorithm (or 2x2 windows).
- SSPCE method; it will be referred to as the second algorithm (or 3x3 windows).
- OCE method; it will be referred to as the third algorithm (or 4x4 windows).

The comparison is made with respect to speed and number of contour edges. The binary test images are illustrated in Fig. 1.12.

Table 1.1, Table 1.2 and Table 1.3 present the comparison between the three algorithms with respect to the number of operations versus the number of edges for Circle, Rectangle and E letter contours respectively.



(a)



(b)



(c)

Fig. 1.12 Binary images (a) Circle, (b) Rectangle, and (c) E letter

Table 1.1 Comparison between the algorithms for Circle image

NE	12	31	51	73	AE
<i>1st alg.(NO)</i>	9633	18733	31512	43412	51708
<i>2nd alg.(NO)</i>	1164	19310	55367	86155	89699
<i>3rd alg.(NO)</i>	30406	63753	111193	158618	186514

alg. – algorithm, **NE** – number of edges, **AE** – all edges and *NO* is the number of operations

Table 1.2 Comparison between the algorithms for the Rectangle image

NE	63	113	165	189	AE
<i>1st alg.(NO)</i>	60973	137350	216780	253440	325037
<i>2nd alg.(NO)</i>	1863	191425	388557	471956	476496
<i>3rd alg.(NO)</i>	223687	512887	817477	957375	1,239995

alg. – algorithm, **NE** – number of edges, **AE** – all edges and *NO* is the number of operations

Table 1.3 Comparison between the algorithms for the E letter image

NE	25	57	93	131	160	AE
1 st alg.(NO)	34402	45513	59693	75604	85166	109467
2 nd alg.(NO)	802	15741	28526	41997	56642	56642
3 rd alg.(NO)	123982	159999	217495	277915	312774	407750

alg. – algorithm, **NE** – number of edges, **AE** – all edges and **NO** is the number of operations

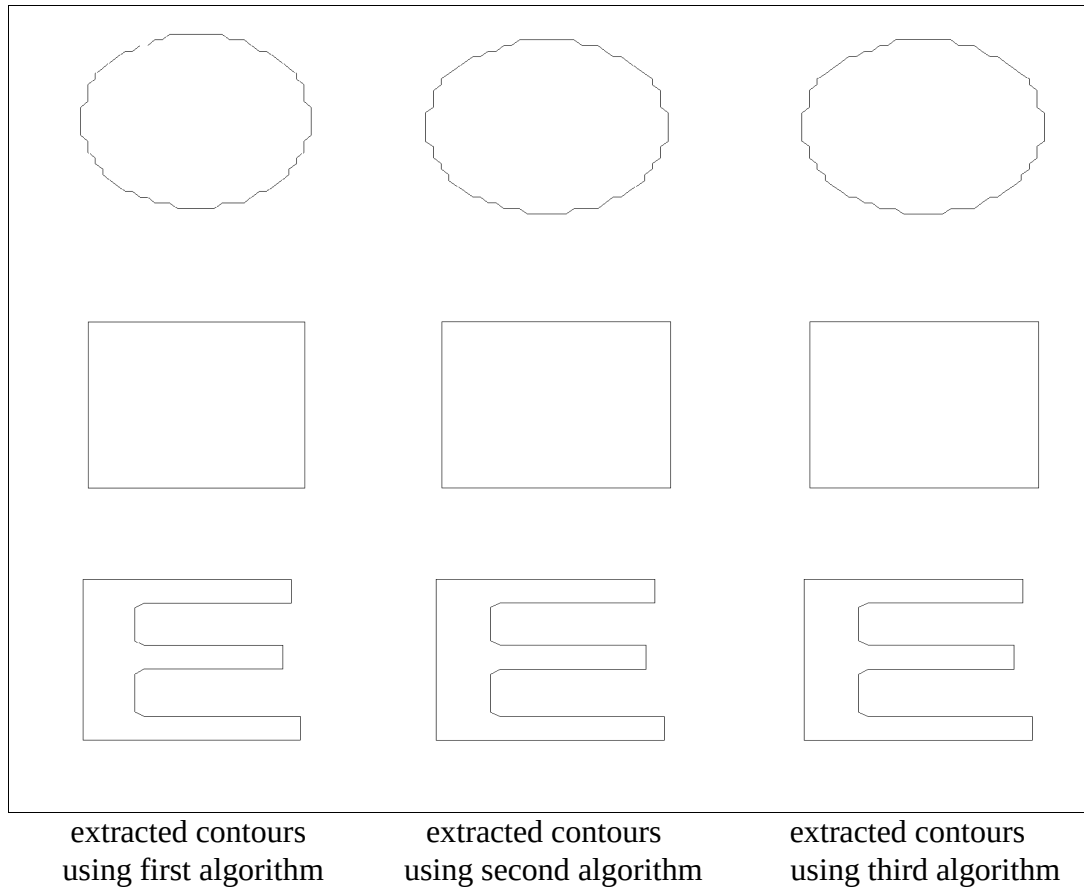
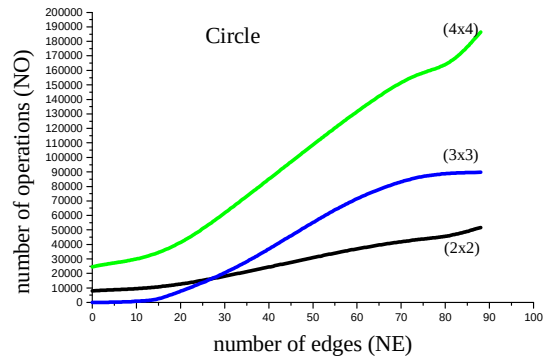


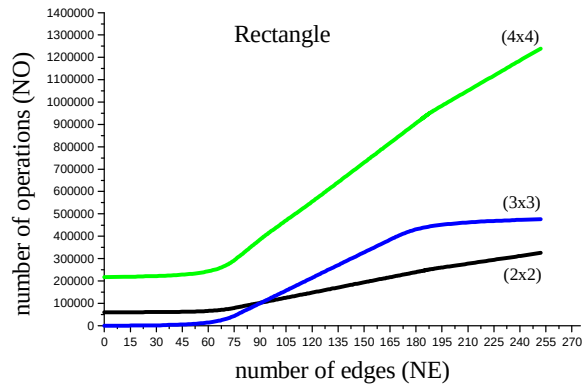
Fig. 1.13 Extracted contours using the three different algorithms

The first column of Fig. 1.13 shows the extracted contours by the first algorithm. The second column of Fig. 1.13 shows the extracted contours by the second algorithm and the third column in Fig. 1.13 shows the extracted contours by the third algorithm.

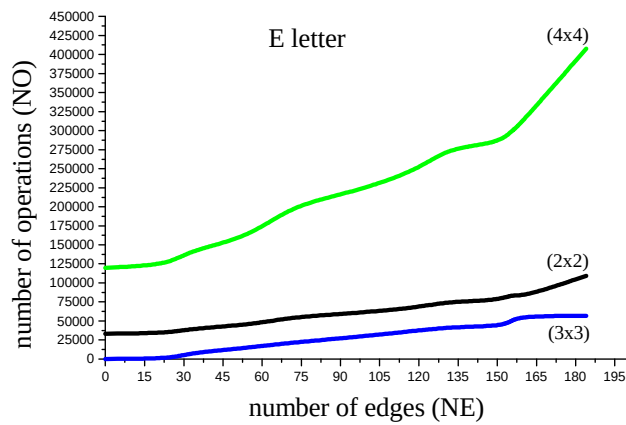
Fig. 1.14 shows the comparison between these algorithms with respect to the number of operations versus the number of edges for the binary images as shown in Fig. 1.12.



(a)



(b)



(c)

Fig. 1.14 Number of operations versus number of edges using the all algorithms for the (a) Circle, (b) Rectangle, and (c) E letter

The results presented in Fig. 1.14 show that the algorithm which had used the smallest size of windows is the fastest algorithm (as in Circle and Rectangle). However the results obtained using these algorithms for the E letter image is not logical.

The experiments are repeated for the binary image of Libya map shape (see Fig. 1.15).

Fig. 1.16 shows the plot between the numbers of operations versus the number of edges for Libya map contour.



Fig. 1.15 Binary image for the Libya map

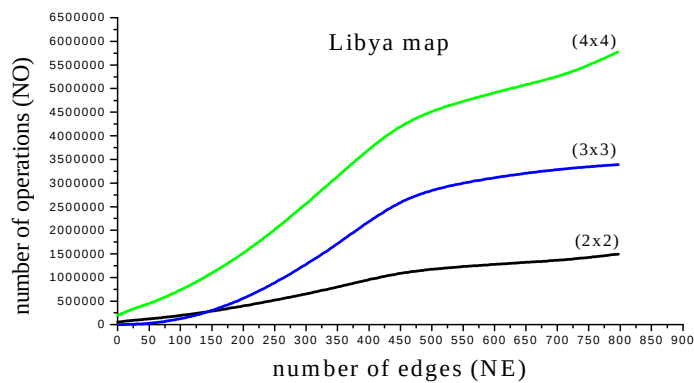


Fig. 1.16 Number of operations versus number of edges using the all algorithms for the Libya map

The comparison between the tested binary images for the number of operations versus the number of edges is shown in Table 1.4.

Table 1.4 Comparison between the algorithms for the all test images

THE METHOD	Circle	Rectangle	E letter	Libya map
1st alg.(NO)	51708	325037	109467	1497205
2nd alg.(NO)	89699	476496	56642	3389720
3rd alg.(NO)	186514	1239995	407750	5774853

alg. – algorithm and NO is the number of operations

The results presented in Table 1.4 show that the contour extraction using 2x2 windows is faster than the other algorithms and the contour extraction using 3x3 windows is faster than the third algorithm (4x4 windows) for the all test images except that for E letter image. So, to obtain logic results and good comparison between different windows for contour extraction, the non-complex shapes are tested.

Chapter2

METHODS OF CONTOUR COMPRESSION IN TIME DOMAIN

2.1 Overview

In many applications of image and contour processing, and analysis it is desirable to obtain a polygonal approximation of an object under consideration. In this chapter we briefly introduce algorithms have been introduced for polygonal approximation of extracted contours. Ramer presented an algorithm which uses the maximum distance of the curve from the approximating polygon as a fit criterion. There exist algorithm was initiated as the tangent algorithm method, which uses the tangent of an angle between two straight lines, called opening and closing lines as the fit criterion. The main purpose of this chapter is to present a new two algorithms of contour approximation polygonal. The first proposed algorithm is known as centroid method. An output of the proposed algorithm is contour in polar (r, Φ) representation. The output contour is limited to the r distances only, and that is the fit criterion of this algorithm. The second proposed algorithm presents the ratio between the direct and real distances for each segment, and uses this ratio as the fit criterion of the algorithm which is referred to as segment distances ratio algorithm. To present the results, the mean square error and signal-to-noise ratio criterions versus compression ratio were used. Computational times of analysed procedures are estimated based on a number of numerical operations. Discussion of good and bad points of the new methods is also included and finally, the comparison with the Ramer method of contour approximation is presented.

2.2 Contour compression methods

There are two approaches to solve the problem of contour compression. The first one is the time domain approach, for which most of its algorithms are concerned with the approximation and/or smoothing of curves such as Ramer [31], Sirjani [45], Montanari [46] and tangent method [34], [36] and [37]. Two algorithms of the same class are proposed and analyzed. The first algorithm [47], for the contour polygonal approximation

known as the centroid method for which the polar representation for the contour output is used. The second algorithm for the contour compression known as the segment distances ratio method for which the ratio of the direct and real distances is used for the segment as the fit criterion. Most of contour approximation methods use Cartesian representation. There are also schemes and applications where polar or Freeman's (also generalised) [30] and [32] chain coding representations are required. A commonly used chain coding representation is the 8-directional chain coding which uses eight possible directions to present all possible line segments connecting the nearest neighbours according to the 8-connectivity scheme.

2.2.1 Ramer method

The algorithm is based on the maximum distance of the curve from the approximating polygon, and this distance is used as the fit criterion. Polygons with a small number of edges for two-dimensional curves are obtained using this algorithm. The segment of the curve is approximated by a straight line segment connecting its start and end. If the fit criterion is not fulfilled, the curve segment is terminated into two segments at the curve point of greatest distance. This process is repeated recursively until for each of the two new lines (curve segments) we don't need to terminate any more. The vertices of an edge of the reconstructed contour are represented by the end of all these curve segments. Finally the lines between these vertices are drawn to obtain the polygonal approximating contour.

Two important disadvantages are obtained using this type of polygonal curve representation. First a very large of edges number that the polygons contains. Second, the noise produced by quantization has influence in the length of the edges. The idea is illustrated in Fig.2.1.

Fig. 2.1 shows an example of polygon generation for the closed contour using Ramer algorithm with approximating straight line segments (intermediate and final).

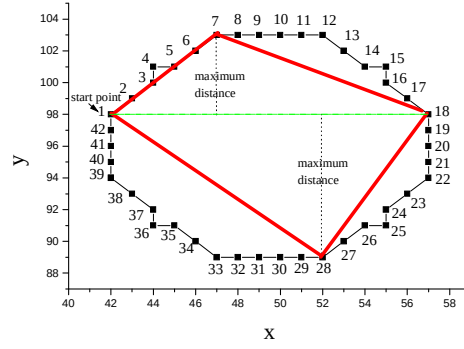


Fig. 2.1 Example for contour approximation using Ramer algorithm

2.2.2 Tangent method

The algorithm is a part of the new approximation methods family called the tangent methods. It is a polygonal method and the tangent of an angle (ϕ) between two straight lines (opening l_o and closing l_c lines) is used as the fit criterion of the algorithm. The fit criterion is illustrated in Fig. 2.2.

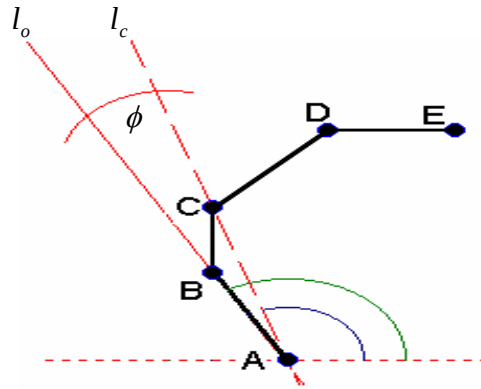


Fig. 2.2 Fit criterion of the tangent algorithm

If the tangent of ϕ is less than a given threshold the end point of the segment is shifted to the right and a new closing line is drawn. If not, the first and end points are stored to determine vertices of an edge of the approximating polygon. Hence the end point opens a new segment and the processes are repeated again.

An additional undesirable thing sometimes created by this algorithm refers to the “cutting effect” which is clearer especially for closed contours. This problem can be avoided by determining the segment length using an additional parameter.

The most typical contour representations method is:

- 1) Generalized representation (θ, l) .

Fig. 2.3 shows the contour representation using the (θ, l) generalized chain coding scheme.

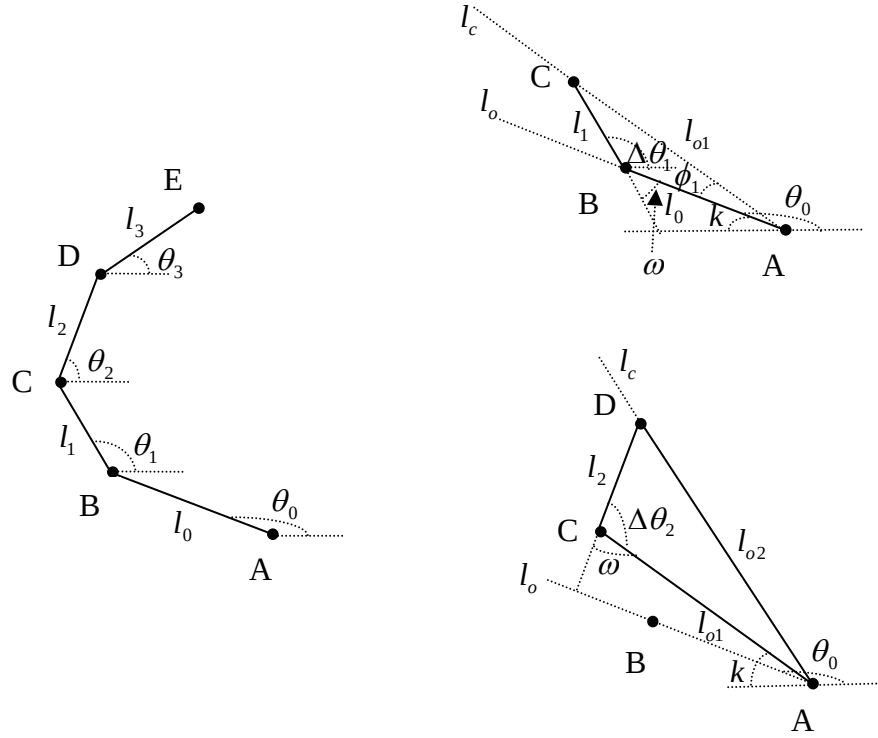


Fig. 2.3 Contour representation using (θ, l)

Analysis of the iterations leads to the following formula

$$\operatorname{tg} \phi_i = \frac{C_{i-1} + \operatorname{tg} \phi_{i-1}}{1 - C_{i-1} \cdot \operatorname{tg} \phi_{i-1}} \quad (2.1)$$

where

$$C_{i-1} = \frac{l_i \cdot \sin \Delta \theta_i}{l_{o(i-1)} - l_i \cdot \cos \Delta \theta_i} \quad (2.2)$$

and

$$l_{oi} = \frac{l_i \cdot \sin \Delta \theta_i}{\sin(\phi_i - \phi_{i-1})} \quad (2.3)$$

where

$$\Delta\theta_i = \pi - (\theta_0 - \theta_i - \phi_{i-1}) \quad (2.4)$$

for $i = 1, 2, \dots, N$, where N - number of segment points; while $\phi_0 = 0$ & $l_{o0} = l_0$

2) Polar representation (α, l) .

Fig. 2.4 shows the contour representation using (α, l) polar chain coding scheme.

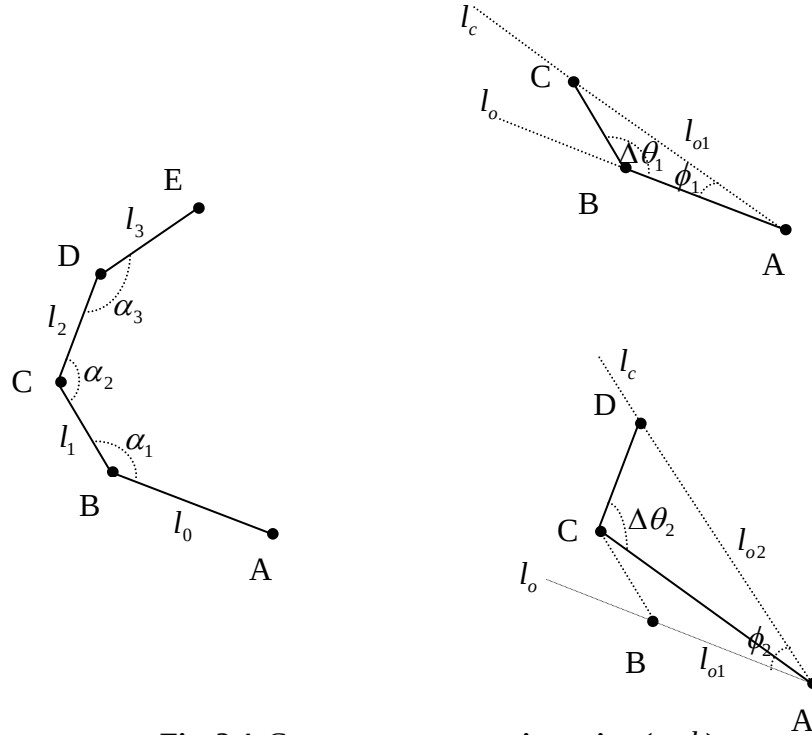


Fig. 2.4 Contour representation using (α, l)

The equation of the contours in (α, l) representation is

$$C_{i-1} = \frac{l_i \cdot \sin \Delta\theta_i}{l_{o(i-1)} - l_i \cdot \cos \Delta\theta_i} \quad (2.5)$$

where

$$l_{oi} = \frac{l_i \cdot \sin \Delta\theta_i}{\sin(\phi_i - \phi_{i-1})} \quad (2.6)$$

and

$$\Delta\theta_i = \theta_{i-1} - \theta_0 + \alpha_i + \phi_{i-1} \quad (2.7)$$

for $i = 1, 2, \dots, N$, where N - number of segment points; while $\phi_0 = 0$ & $l_{o0} = l_0$

3) Cartesian representation.

The Cartesian representation can be obtained from following simple equation

$$\tan\phi = \left| \frac{acl - aop}{1 + (acl \cdot aop)} \right| \quad (2.8)$$

where aop is angle coefficient of the opening line and acl is angle coefficient of the closing line.

Tangent of the ϕ angle is specified if $aop \neq -1$. It means that the ϕ angle is from the range $(-\pi/2, \pi/2)$.

2.2.3 Algorithm for contour approximation using centroid method

The algorithm refers to a recent polygonal approximating method called centroid method (CM).

2.2.3.1 Description of the algorithm

The input contour for the analyzed algorithm is obtained from 256 x 256 grey-scale images by using the SSPCE contour extraction procedure [33]. The input contour is represented by x_s and y_s vectors of Cartesian co-ordinates. The algorithm starts with finding the center of the input contour mass called the reference point $O = (x_m, y_m)$. Co-ordinates x_m, y_m are defined as follows

$$x_m = \frac{1}{N} \sum_{i=0}^{N-1} x_s \quad y_m = \frac{1}{N} \sum_{i=0}^{N-1} y_s \quad (2.9)$$

where N is the number of contour vertices, x_m is the mean value of the x_s vector and y_m is the mean value of the y_s vector.

The input contour is then shifted by x_m and y_m in both axes directions by

$$x_i = x_s - x_m, y_i = y_s - y_m \quad (2.10)$$

Centroid of the shifted contour is placed at (0,0) point, and therefore further computations are vastly simplified. Next, distances r between (0,0) point and shifted contour line are calculated. For that purpose, straight lines are passed through the (0,0) point. Slopes of these lines depend on earlier assignment of an angle between them. The angle between the contour lines is represented by the input parameter Φ . The Φ value is assigned with respect to another input parameter I (accuracy of the procedure). The relation between parameters I and Φ is as follows

$$\Phi = \frac{\pi}{2^{I+1}} \quad (2.11)$$

When accuracy $I = 0$ it means that $\Phi = \pi/2$ and two perpendicular straight lines are passed. In case of $I = 7$ the $\Phi = \pi/256$ and 256 straight lines are passed. The basic idea is illustrated in Fig. 2.5 for $I = 0$ and $I = 1$.

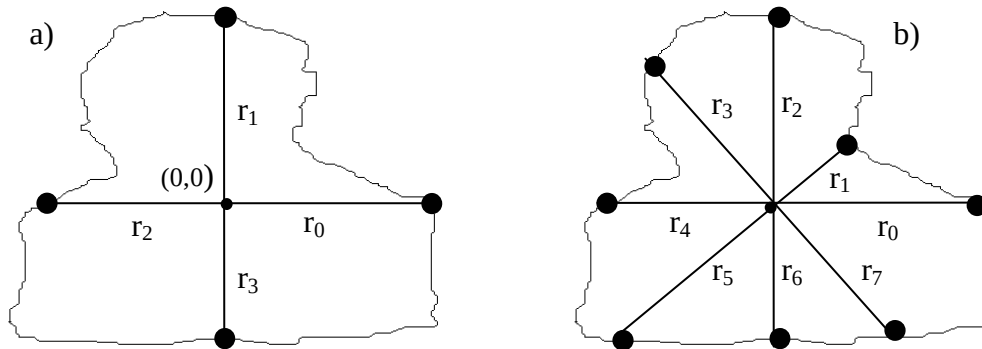


Fig. 2.5 Selection of output vertices a) $I = 0$, and b) $I = 1$

Selected vertices are fully determined by the accuracy of the procedure and sequence of r distances. Therefore, representation of the approximated contour can be one-dimensional and that is the advantage of the presented method.

Distances r are given by the following equation

$$r_i = \sqrt{x_i^2 + y_i^2} \quad (2.12)$$

where (x_i, y_i) is the co-ordinates of selected vertices.

A flowchart of the analyzed algorithm is depicted in Fig. 2.6, where VA is sequence of the output vertices, M is vector of the slopes s , l_M is length of the vector M , (x_i, y_i) is co-ordinates of the input contour points, L_{CC} is length of the input contour and D is counter.

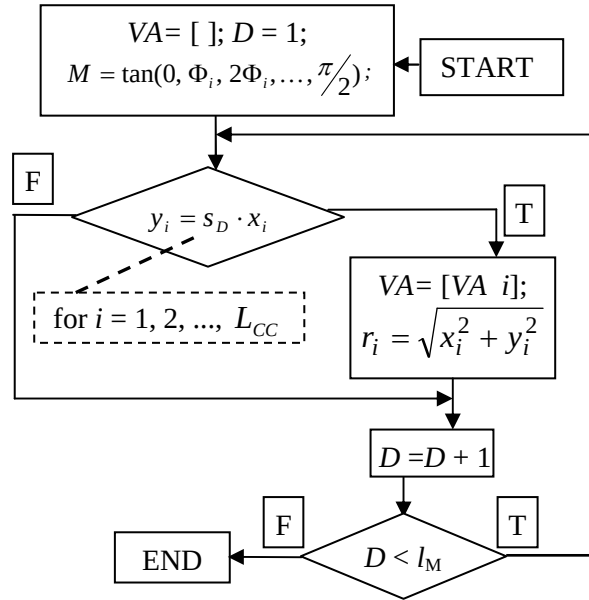


Fig. 2.6 Flowchart of the algorithm

Presented method is related to the data compression problem. To evaluate its compression ability, the following compression ratio was introduced

$$CR = \frac{(B_{CC} - B_{AC})}{B_{CC}} \cdot 100\% \quad (2.13)$$

where B_{CC} is total number of the bits required for the input contour and B_{AC} is total number of the bits required for the approximating contour.

The B_{CC} value depends on length of the input contour and the maximum values of x and y . The value of B_{CC} for contours extracted from 256 x 256 images can be calculated as follows

$$B_{CC} = 2 \cdot L_{CC} \cdot 8 \text{ bits}$$

The full information required for contour reconstruction consists of collecting the following parameters: Φ , r and co-ordinates of the reference point O .

Such output contour representation is sufficient only for regular contour shapes (i.e. straight line is crossing only two points), as shown in Fig. 2.5. General contours (i.e. straight line is crossing more than two points as shown in Fig. 2.7 for Serpent contour) require additional information.

The mean square error (MSE) and signal-to-noise ratio (SNR) criterions were used as measures of quality of approximation. The MSE criterion is defined by the following equation

$$MSE = \frac{1}{L_{CC}} \sum_{i=1}^{L_{CC}} d_i^2 \quad (2.14)$$

where d_i is the distance from i point of the curve segment and straight line between starting and ending points (vertices) of that segment.

The SNR is defined by the following formula

$$SNR = -10 \cdot \log_{10} \left(\frac{MSE}{VAR} \right) \quad (2.15)$$

where VAR is variance of the input sequence.

From the practical point of view the values of MSE and SNR can not exceed the 4.0 and 30 dB, respectively. Otherwise, the details of contours are eliminated and level of introduced distortion can not be accepted.

2.2.3.2 Results of the experiments

To visualise the experimental results a set of two test contours was selected. Selected contours are depicted in Fig. 2.7. Selected results of the compression of the test contours are illustrated in Fig. 2.8 and Fig. 2.9. Fig. 2.8 shows the results of compression performed for the test contour Serpent. Fig. 2.8a presents the reconstruction obtained when the additional information is not used.

Proper reconstruction of the general shape contours requires the additional information - succession of selected vertices. This information is collected during the procedure as the VA sequence and needs for every piece as many bits as is required for its maximum value. To minimise these requirements, VA sequence is normalised by its length.

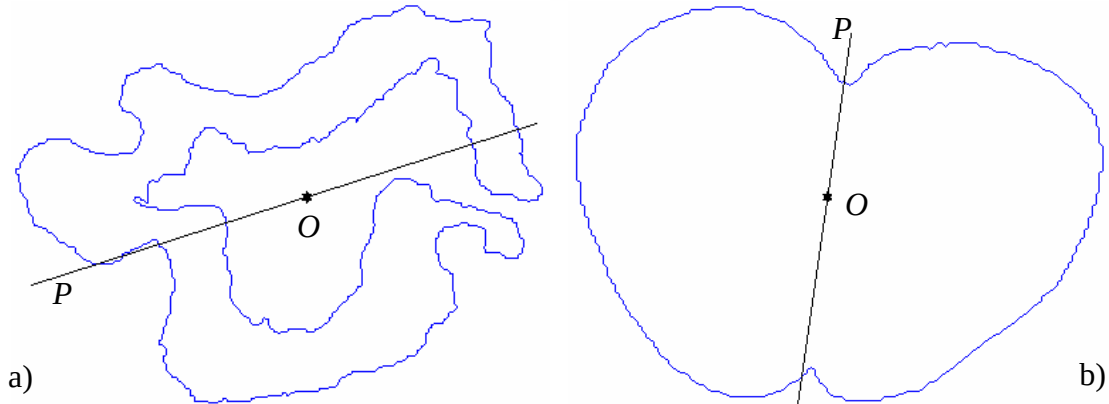


Fig. 2.7 Test contours: a) Serpent, and b) Apple

Some exemplary reconstructions of general shape contour Serpent, without and with transmitting the additional information about the vertex succession are depicted in Fig. 2.8 (related results are in Table 2.1).

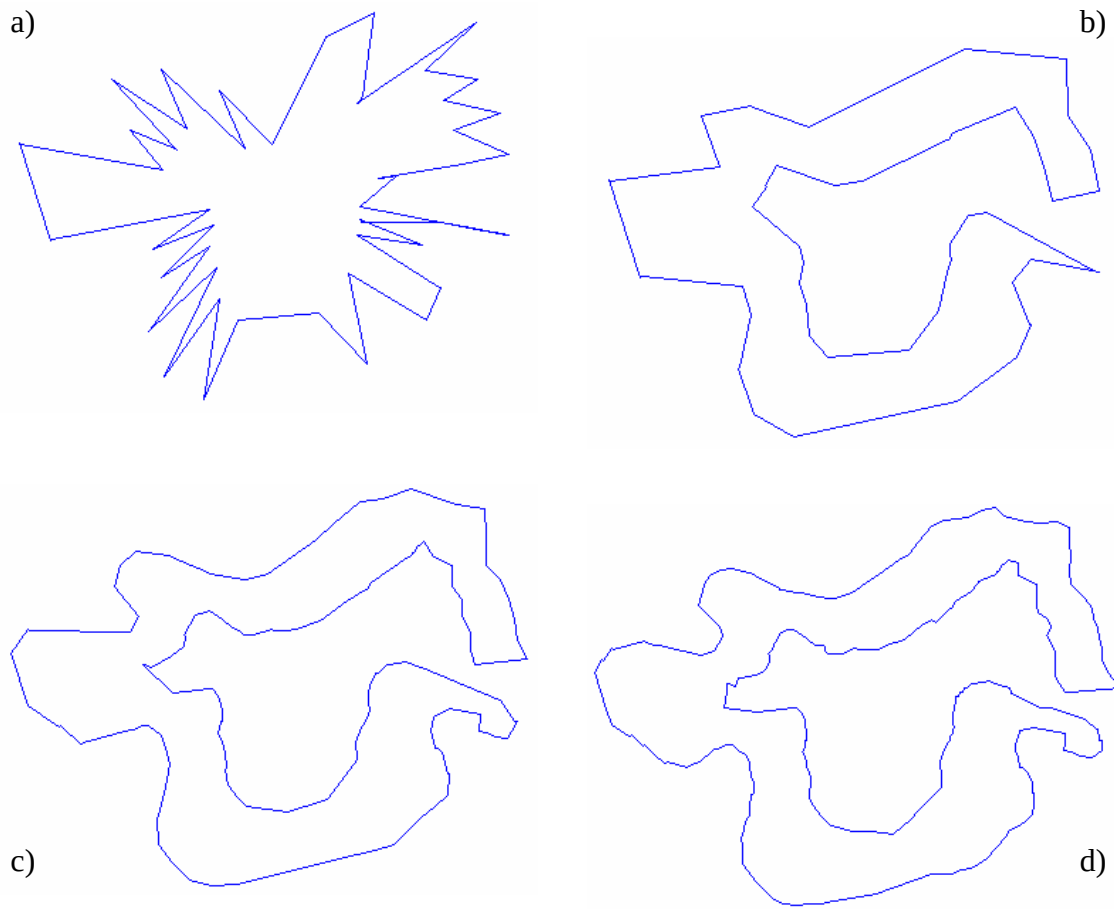


Fig. 2.8 Selected results of Serpent contour test approximation

Table 2.1 Results after approximation for the Serpent contour

Case	AI	Level	MSE	SNR	CR [%]
a)	Not transmitted	3	-	-	97,61
b)	Transmitted	3	32,39	22,09	96,45
c)	Transmitted	4	5,33	29,93	90,48
d)	Transmitted	5	1,14	36,63	79,30

AI – additional information

Some selected results of regular shape Apple contour without transmitting the vertex succession information are depicted in Fig. 2.9 (related results are in Table 2.2).

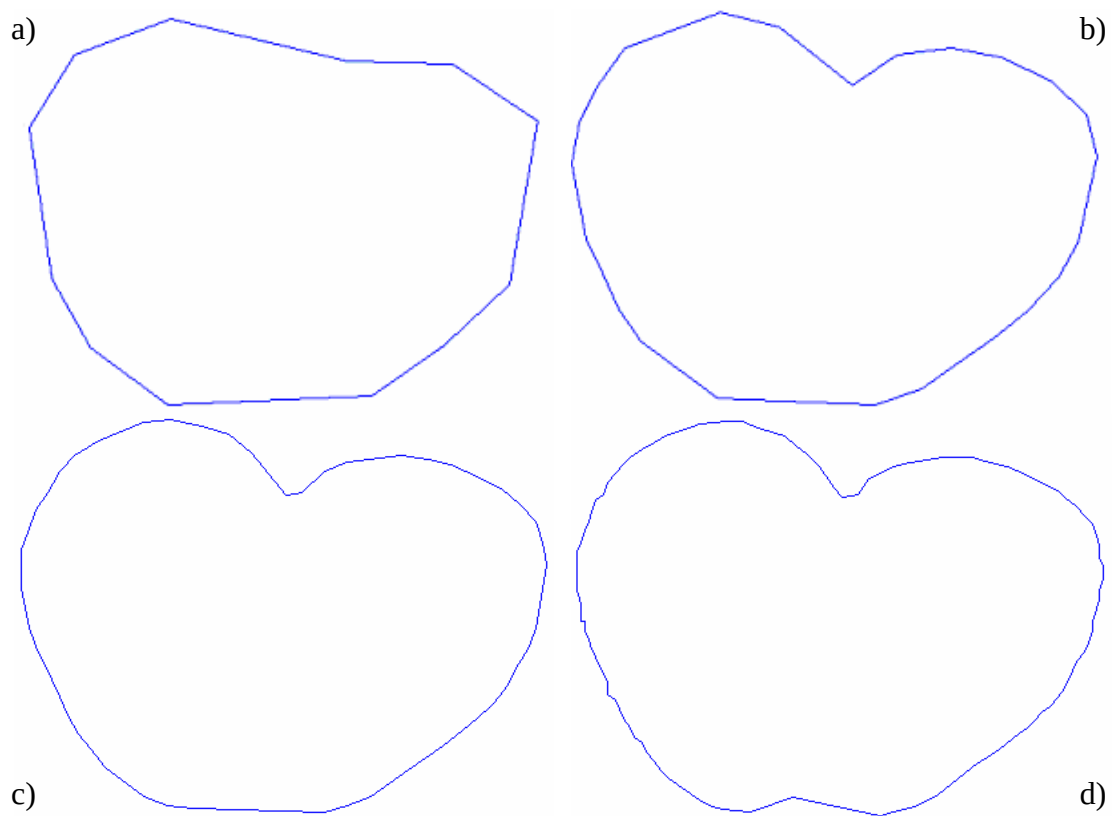


Fig. 2.9 Selected results of Apple contour test approximation

Table 2.2 Results after approximation for the Apple contour

Case	AI	Level	MSE	SNR	CR [%]
a)	not transmitted (not required)	2	12,93	22,18	97,89
b)		3	2,61	29,14	95,86
c)		4	2,11	30,05	91,67
d)		5	0,59	35,61	83,74

AI – additional information

The results presented in Fig. 2.8 and 2.9 show that the analyzed method has good compression abilities, especially for contours of regular shape. It is seen that the compression ratio for such type of contours can be even greater than 95% with some significant visible losses of approximation quality.

Comparison of the compression abilities of the analyzed method and the Ramer algorithm is presented in Fig. 2.10. Plots depicted in Fig. 2.10 were prepared for both types of shapes using the test contours.

Fig. 2.10 shows however that the Ramer algorithm can be more useful for contours of general shapes. The maximum acceptable compression ratio for this algorithm is about 10% greater than the proposed method.

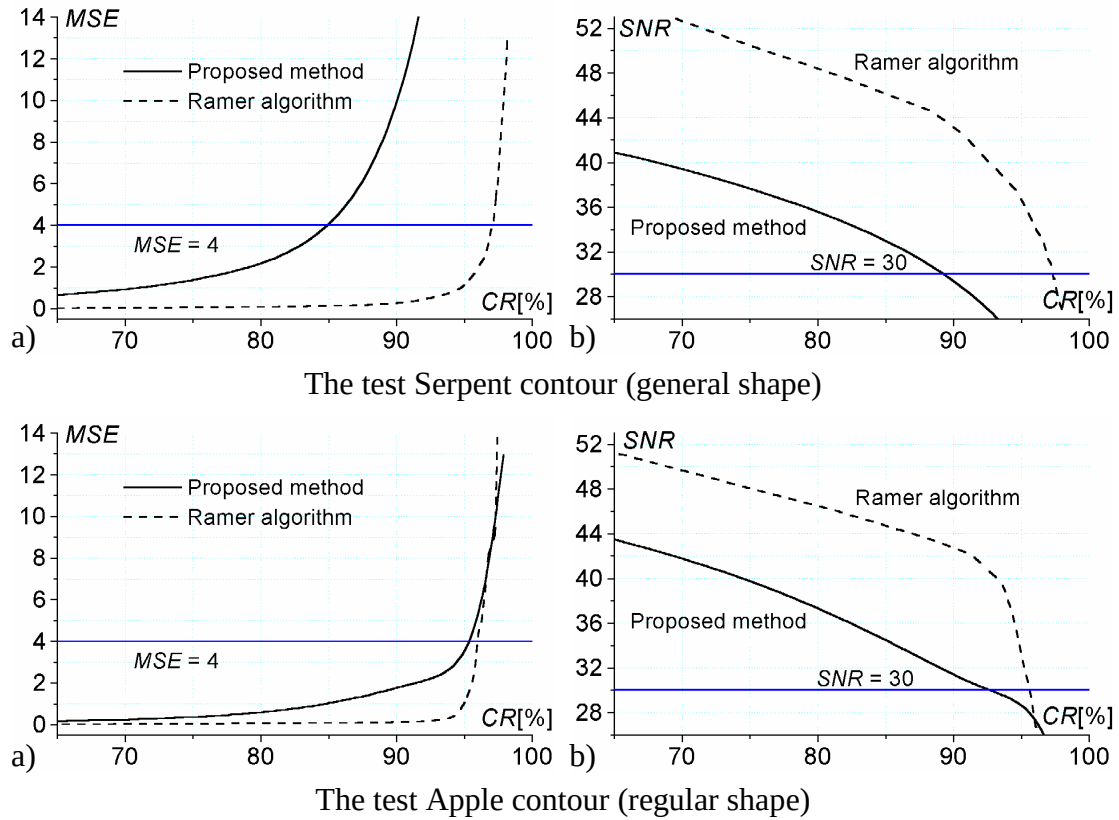


Fig. 2.10 Comparison of the analyzed method with the Ramer algorithm:
a) MSE versus CR, and b) SNR versus CR

2.2.4 Algorithm for contour compression using segment distances ratio method

A new approach for contour data approximating algorithm proposed by Prof. A. Dziech is presented and developed in this section. Cartesian co-ordinates of an input contour are processed in such a manner that finally contours are presented by a set of selected vertices from the edge of the contour. The idea includes explanation of the final vertices selection rule, main points of the procedure, results obtained during the experiments and comparison with the Ramer method of contour approximation. To present the results, the mean square error and signal-to-noise ratio criterions were used. Computational times of analysed procedures are estimated based on a number of

numerical operations. The simplicity and speed are the main advantages of the analyzed algorithm.

The algorithm refers to a recent polygonal approximating method called segment distances ratio (SR).

2.2.4.1 Description of the algorithm

An input contour for the algorithm is extracted from 256 x 256 grey-scale images using single step parallel contour extraction (SSPCE) method [33]. The algorithm is a polygon method. The fit criterion of the algorithm is the ratio between the direct (LD) and real (LR) distances for each segment and is defined by the following equation

$$\frac{LD}{LR} > th \quad (2.16)$$

where th is given threshold value of the ratio.

The first point of each segment is called starting point (SP) and the last one is called the ending point (EP). To calculate the direct distance; the trigonometric formula is used.

If the ratio is greater than a given threshold; the EP point is equal to the length of the segment divided by the scaling factor. Otherwise; the SP point is shifted to the EP point and the EP point is equal to the length of the segment multiplied by the scaling factor and a new segment is drawn. The SP and EP points determine vertices of an edge of the approximating polygon.

The algorithm scans contour points only once; i.e. it does not require the storage of the received contour points, and original points of the contour are discarded as soon as they are processed. Only the co-ordinates of the starting point of the contour segment, and the last processed point are stored.

The idea of the algorithm is illustrated in Fig. 2.11.

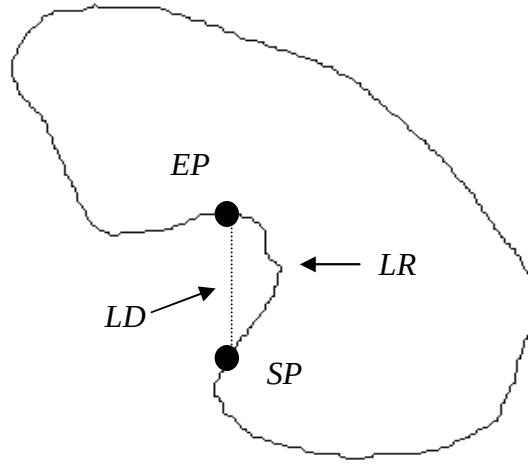


Fig. 2.11 Fit criterion of the algorithm for one segment

where LD is the direct distance of the segment and LR is the real distance of the segment according to the SSPCE method.

The approximation procedure starts at the time, when the first and last points of a segment are received. This algorithm can be treated as a very fast one.

The criterion depends on the input contour representation method. The algorithm use an 3x3 pixels window structure to extract the object contours by using the central pixel to find the possible edge direction which connects the central pixel with one of the remaining pixels surrounding it.

A flowchart of the algorithm is depicted in Fig. 2.12, where VA is sequence of the output vertices, CC is sequence of the input contour, SP is starting point, EP is ending point, N is length of the segment, $divN$ is scaling factor, LD & LR is as mentioned above and th is threshold value of the ratio.

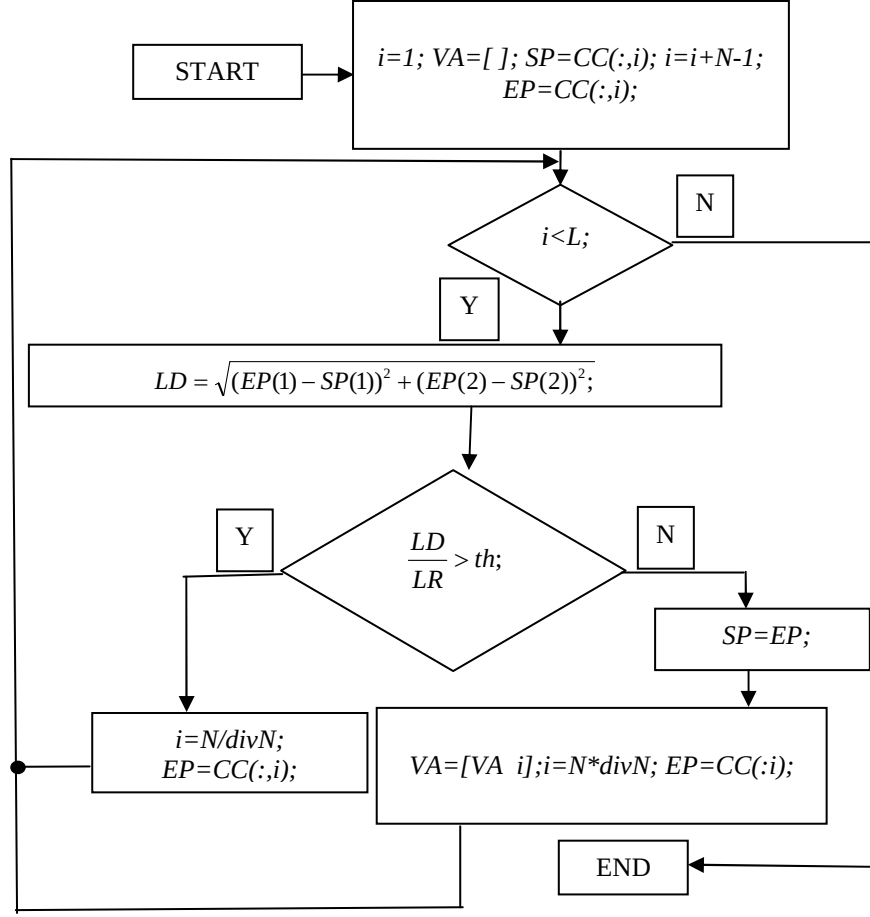


Fig. 2.12 Flowchart of the algorithm

2.2.4.2 Applied measurements

The approximating method is related to the data compression problem. To evaluate its compression ability, the following compression ratio was introduced

$$CR = \frac{(L_{CC} - L_{AC})}{L_{CC}} \cdot 100\% \quad (2.17)$$

where L_{CC} is length of the input contour and L_{AC} is length of the approximating polygon.

The mean square error (MSE) and signal-to-noise ratio (SNR) criterions were used as measures of quality of approximation using Equation (2.14) and Equation (2.15) respectively.

From the practical point of view the values of MSE and SNR can not exceed the 4,0 and 30 dB, respectively. Otherwise, the details of contours are eliminated and level of introduced distortion can not be accepted.

2.2.4.3 Results of the experiments

To visualise the experimental results a set of two test contours was selected. Selected contours are depicted in Fig. 2.13.

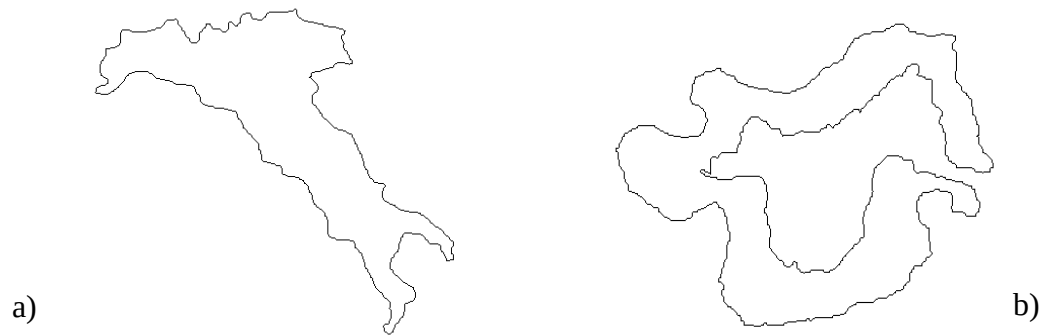


Fig. 2.13 Test contours a) Italy, and b) Serpent

Some selected results of the compression for the Italy contour are illustrated in Fig. 2.14 (related results are in Table 2.3).

Table 2.3 Results of approximation for the Italy contour

	MSE	SNR	CR [%]	NO
a)	0.64	40.57	64.62	131728
b)	3.64	33.02	94.04	12103
c)	8.78	29.28	95.90	7358
d)	10.89	28.26	96.28	5746

NO – number of operations



Fig. 2.14 Results of approximation for the Italy contour

Some selected results of the compression for the Serpent contour are depicted in Fig. 2.15 (related results are in Table 2.4). We are looking for good quality (i.e. less distortion) and at the same time we need a high speed (i.e. number of arithmetic operations has small number).

Table 2.4 Results of approximation for the Serpent contour

	<i>MSE</i>	<i>SNR</i>	<i>CR [%]</i>	<i>NO</i>
a)	0.15	45.57	59.60	178026
b)	1.54	35.31	91.21	28372
c)	3.47	31.79	93.98	16933
d)	7.53	28.43	96.17	5634

NO – number of operations



Fig. 2.15 Results of approximation for the Serpent contour

The results presented in Fig. 2.14 and Fig. 2.15 show that the analyzed method has good compression abilities. We can state, maintaining assumptions related to the allowed values of MSE and SNR, that compression ratio for such contours can be even greater than 95%. However, the high compression (more than 95%) can be obtained with some visible distortion of approximation quality.

The results show that the contour can be compressed using the analyzed method at 94% and the level of distortion is acceptable (i.e. the signal-to-noise value is a round 30 db).

Comparison of the compression abilities of the analyzed method and the Ramer algorithm is presented in Fig. 2.16 for the test contours.

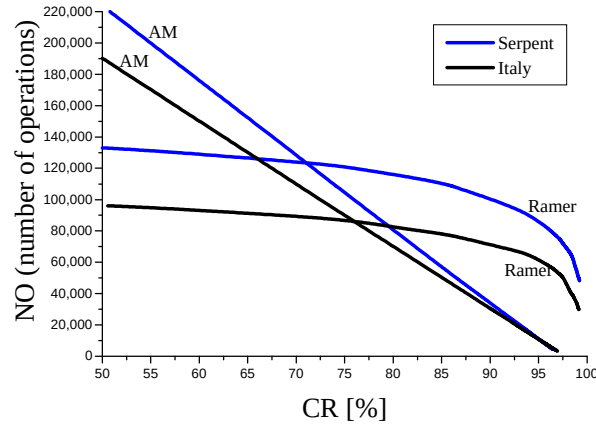


Fig. 2.16 Comparison of the analyzed method with the Ramer algorithm for NO versus CR of the tested contours

The comparison presented confirms above considerations about the compression abilities of the proposed method. Results obtained for both analysed methods are comparable. Both, the analyzed method and the Ramer algorithm allow obtaining the compression ratio that is even greater than 95% with some significant details losses of approximation quality.

Fig. 2.17 and Fig. 2.18 show the arithmetic operations number (NO) versus mean square error (MSE) and signal- to- noise-ratio (SNR) respectively.

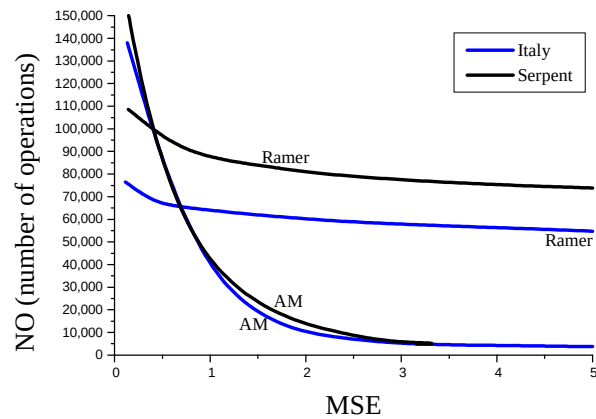
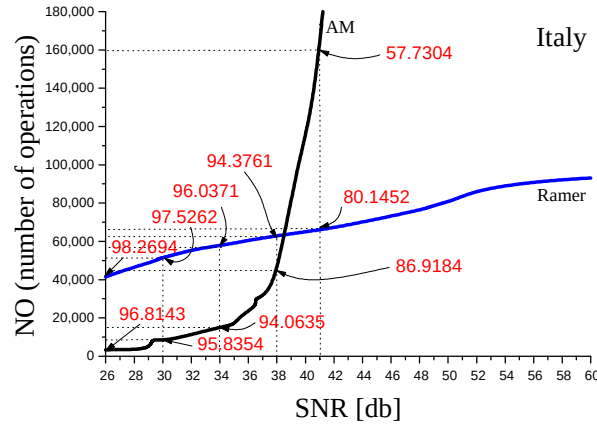
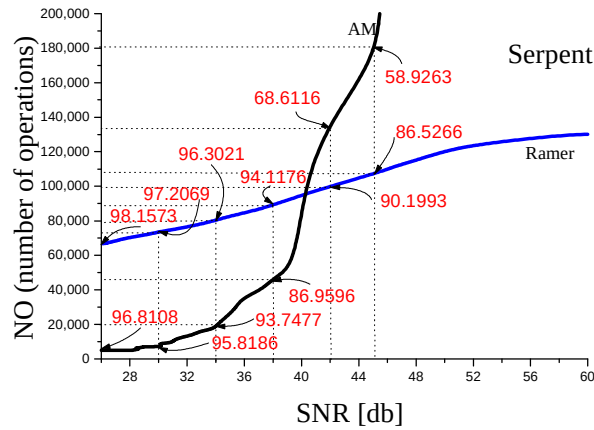


Fig. 2.17 Comparison of the analyzed method with the Ramer algorithm for NO versus MSE of the tested contours



(a)



(b)

Fig. 2.18 Comparison of the analyzed method with the Ramer algorithm for NO versus SNR: (a) Italy contour, and (b) Serpent contour

Fig. 2.19 shows the main advantage of the analyzed algorithm, which is much faster than the Ramer algorithm. The number of the operations using Ramer algorithm is much higher than in the analyzed method even if the compression ratio is fixed on a certain value. So, the computational time is much less and gives the analyzed algorithm to be used in a wide application for the contours where the speed is necessity. However, the SNR using the Ramer algorithm gives much better approximation quality; but the analyzed algorithm is much faster. The number of arithmetic operations for the Ramer method is many times greater than the analyzed method for some contours.

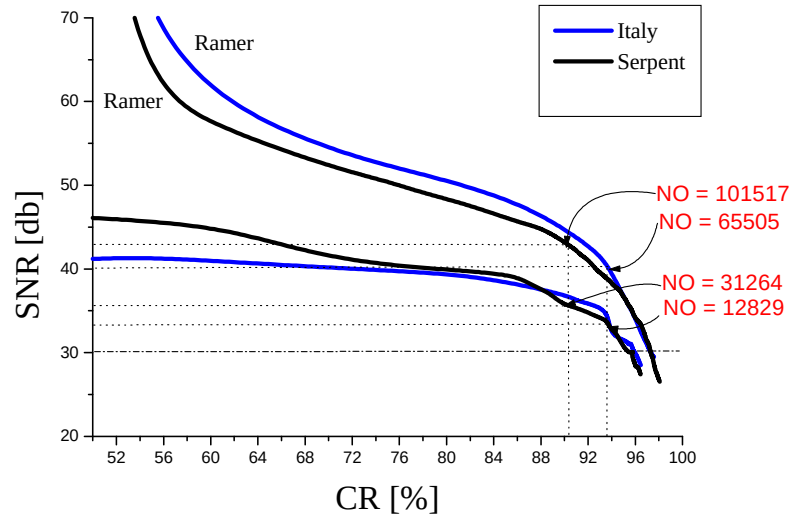


Fig. 2.19 Comparison of the analyzed method with the Ramer algorithm for SNR versus CR of the selected contours

2.3 Comparison between the contour approximation algorithms

The comparison is done for some test contour as shown in Fig. 2.20a which were extracted by the usage of the “SSPCE” (single step parallel contour extraction). The comparison is made between the following three algorithms:

- Ramer algorithm method; it will be referred to as the first algorithm.
- First proposed algorithm method (centroid method or CM); it will be referred to as the second algorithm.
- Second proposed algorithm method (segment distances ratio method or SR); it will be referred to as the third algorithm.

To present the results of the comparison, the mean square error and signal-to-noise ratio criterions versus compression ratio were used.

Fig. 2.20 (related results are in Table 2.5) show the contour compression by the three algorithms for the selected contour with compression ratio is about 95%.

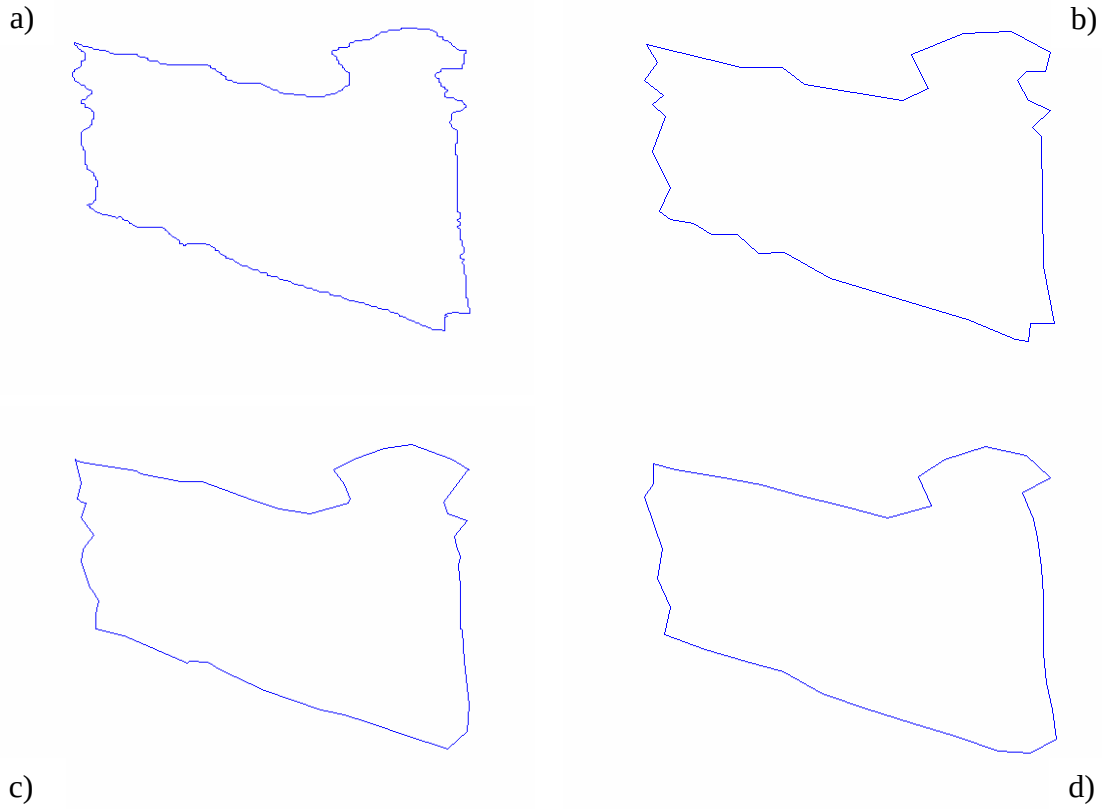


Fig. 2.20 a) Original contour, b) Approximated contour using the first algorithm, c) Approximated contour using the second algorithm, and d) Approximated contour using the third algorithm

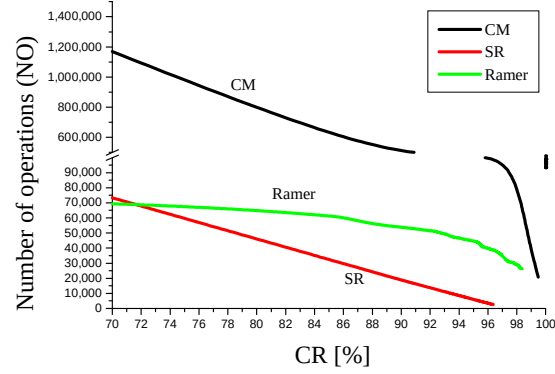
Table 2.5 Approximated contour using the three algorithms for the Libya map contour

	<i>MSE</i>	<i>SNR</i>	<i>CR [%]</i>	<i>NO</i>
b)	0.89	39.61	95.11	44442
c)	2.26	35.54	94.93	204883
d)	3.81	33.26	94.99	2712

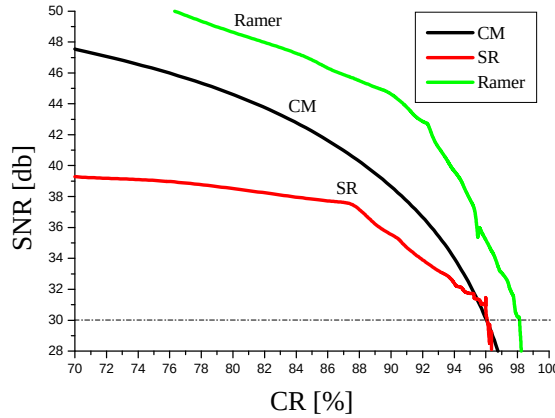
NO – number of operations

From Fig. 2.20; it is clear that the Ramer method quality is much better than the other algorithms; but the third algorithm is much faster.

Fig. 2.21 (a & b) shows the comparison between the algorithms for number of the operations versus compression ratio and signal-to-noise-ratio versus compression ratio respectively.



(a)



(b)

Fig. 2.21 Plot comparing algorithms
a) NO versus CR, and b) SNR versus CR

The compression ratio using the second and third analyzed algorithms can be even greater than 96% with some details losses, and their complexity is much less than that of the Ramer algorithm. However the Ramer algorithm can be more useful for contours of general shapes than that of the analyzed algorithms.

The main advantage of the third analyzed algorithm in comparison with the other methods is very short computational time of the approximating procedure (i.e. many times faster than that of Ramer and centroid methods).

Chapter3

TRIANGLE FAMILY METHODS OF CONTOUR COMPRESSION IN TIME DOMAIN

3.1 Overview

The main purpose of this chapter is to present four algorithms (called triangle family) of contour approximation polygonal. The idea of these algorithms has been created and proposed by Prof. A. Dziech. The first proposed algorithm presents the ratio between the triangle height and length distances for each segment, and uses this ratio as the fit criterion of the algorithm which is referred to as the “triangle height over length ratio algorithm”. The second proposed algorithm presents the triangle height distance for each segment as the fit criterion of the algorithm which is referred to as the “triangle height algorithm”. The third proposed algorithm presents the square of the triangle height distance for each segment as the fit criterion of the algorithm which is referred to as the “triangle height square algorithm”. The fourth proposed algorithm presents the area for each triangle segment as the fit criterion of the algorithm which is referred to as the “triangle area algorithm”. Computational times of analysed procedures are estimated based on a number of numerical operations.

Discussion of good and bad points of the new methods is also included and finally, the comparison with the Ramer and tangent methods of contour approximation is presented. Number of operations that is necessary to perform approximation and compression procedures is also presented. The simplicity and the small number of operations are the main advantages of all proposed algorithms.

The proposed algorithms belong to a family of polygonal methods of approximation. An input contour for the proposed algorithms is extracted from 256 x 256 grey-scale images using single step parallel contour extraction (SSPCE) method [33].

The approximation procedure starts at the time, when the first and last points of the segment are determined. The analyzed criterion can be modified depending on contour

representation methods. The contour extraction by the proposed algorithms is used SSPCE method.

The analyzed algorithms scan contour points only once; i.e. it does not require the storage of the analysed contour points. The original points of the contour are discarded as soon as they are processed. Only the co-ordinates of the starting point of the contour segment, and the last processed point are stored. [58].

3.2 Applied measures

The analyzed approximating methods are related to the data compression problem. To evaluate its compression ability the Equation 2.17 is used. The mean square error (MSE) and signal-to-noise ratio (SNR) criterions were used to evaluate the distortion introduced during the analyzed approximating procedures by using Equation 2.14 and Equation 2.15 respectively.

Performed analysis and experiments for the analyzed algorithms show that SNR should be greater than 30 dB to obtain the expected compromise between compression ratio and quality of reconstruction. In the case of high threshold level, the contour details are eliminated and level of introduced distortion can not be accepted. The range of the threshold values depends on the length between two points of the triangle.

To visualise the experimental results a set of two test contours was selected. Selected contours are shown in Fig. 3.1.



Fig. 3.1 Test contours a) Italy, and b) Rose

3.3 Algorithm for contour approximation using the triangle height over length ratio method

This algorithm refers to a recent polygonal approximating method called height over length ratio triangle method [57] and [58].

3.3.1 Description of the algorithm

The idea of this method consist a segmentation of the contour points to get a triangle shape. The ratio of the height of the triangle (h) to the length of the base of the triangle (b) is then compared with the given threshold value as follows

$$h/b < th \quad (3.1)$$

where h is height of the triangle, b is length of the triangle and th is given threshold value.

The first point of each segment is called the starting point (SP) and the last one is called the ending point (EP). To calculate these values, a simple trigonometric formula is used. If the ratio value is smaller than the threshold according to Equation (3.1) the EP point of the triangle is stored and SP point is shifted to the EP point, then a new segment is drawn. Otherwise B point is stored and the SP point is shifted to the B point of the triangle. Then a new segment is drawn. The stored points determine the vertices of an edge of the approximating polygon.

The idea of the analyzed algorithm is illustrated in Fig. 3.2.

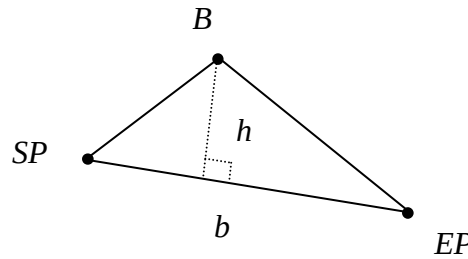


Fig. 3.2 Illustration of the basic triangle for the analyzed algorithm

A flowchart of the analyzed algorithm is depicted in Fig. 3.3, where VA is sequence of output contour, CC is sequence of the input contour, SP is starting point, EP is ending point, $(h, b$ and $th)$ is as mentioned before (see Fig. 3.2 and Equation 3.1) and f is length between each two points of the triangle.

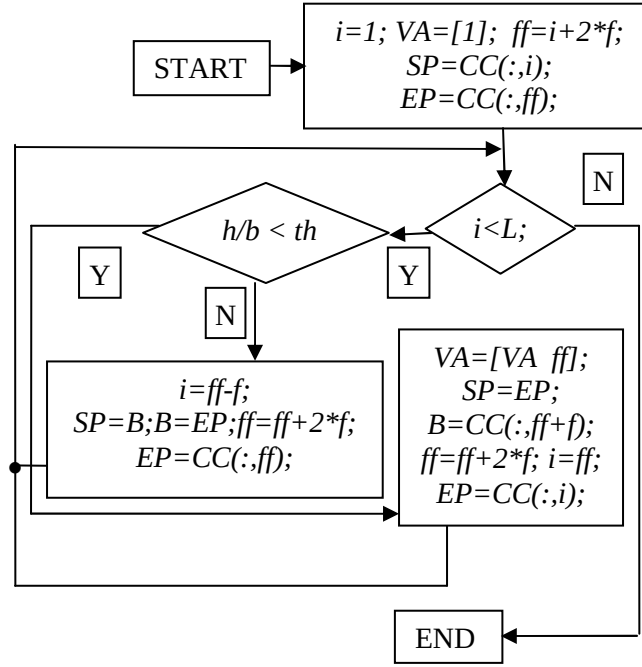


Fig. 3.3 Flowchart of the analyzed algorithm

We must find the answer to the following important question: how is the threshold for any one of the triangle family methods determined? The answer is in the next section.

3.3.2 Threshold determination

The performed analysis and experiments show that SNR should be greater than 30 dB to obtain the expected compromise between compression ratio and quality of reconstruction. In the case of a high threshold level, the contour details are eliminated and level of introduced distortion can not be accepted. The range of the threshold values depends on the length between two points of the triangle.

Usually we are interested in obtaining higher compression ratio (CR) with the accepted level of the reconstruction quality. If, for example, we are interested in

obtaining $CR = 97.4\%$ for the Rose contour (Fig. 3.1 b). Fig. 3.4 shows the comparison between CR versus N (number of points between each two points in the triangle shape) for the Rose contour.

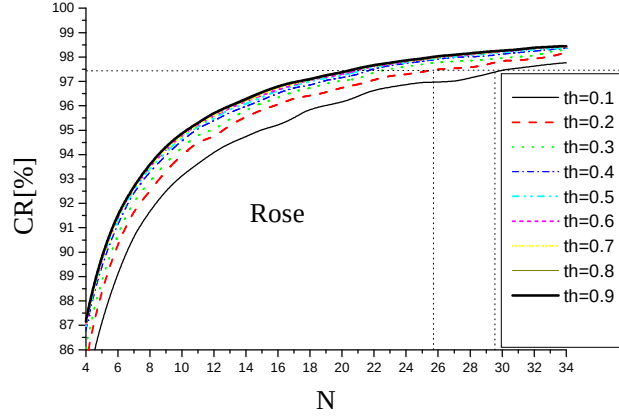


Fig. 3.4 CR versus N for different thresholds of Rose contour

From Fig. 3.4 it is clear that to obtain the $CR = 97.4\%$ the N should be around 30 at threshold ($th = 0.1$) and N is around 26 at threshold ($th = 0.2$). But Fig. 3.4 does not show whether the level of reconstruction quality is acceptable or not !!! In this case the SNR versus N plot is needed for different threshold values as shown in Fig. 3.5.

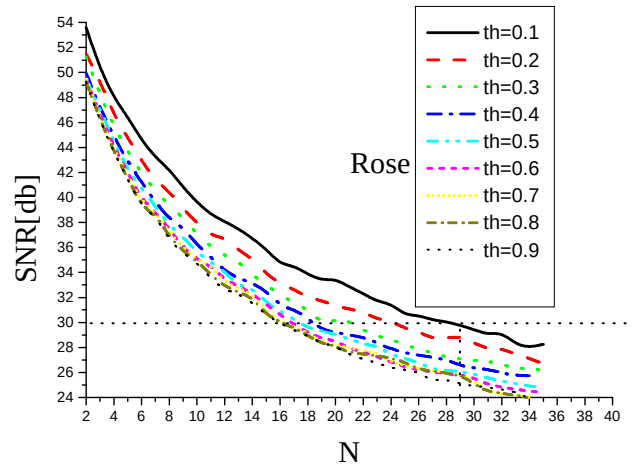


Fig. 3.5 SNR versus N for different thresholds of Rose contour

To determine the optimum threshold for higher CR we have to look for the threshold which has maximum number of N with the conservation of the accepted level of the reconstruction quality. From Fig. 3.5, it is clear that the threshold value equal to 0.1 is the optimum one. For the $CR = 97.4\%$ the N is at around 29 and the threshold is around 0.1 (i.e. SNR is greater than 30 db) and hence the level of the distortion is accepted as shown in Fig. 3.6 (related results are shown in Table 3.1) for the Rose contour.

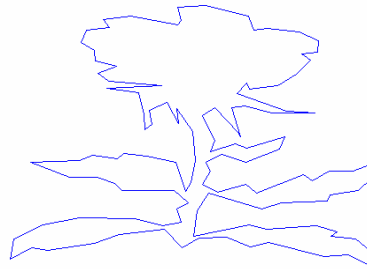


Fig. 3.6 Approximated contour using triangle height over length method for the Rose contour

Table 3.1 Approximated contour using triangle height over length method for the Rose contour

th	N	MSE	SNR	$CR [\%]$	NO
0.12	29	14.01	30.01	97.39	3477

NO – number of operations; th – threshold

Fig. 3.7 (related results are shown in Table 3.2) shows the Rose contour approximation using Ramer algorithm for the similar previous values of MSE and SNR (see Table 3.1).

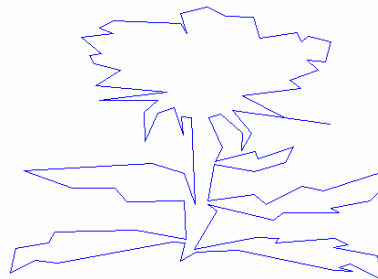


Fig. 3.7 Approximated contour using Ramer method for the Rose contour

Table 3.2 Approximated contour using Ramer method for the Rose contour

<i>Th</i>	<i>MSE</i>	<i>SNR</i>	<i>CR [%]</i>	<i>NO</i>
9.79	13.90	30.04	97.80	311380

NO – number of operations; *th* – threshold

The test contours for an Arabic name and E letter are shown in Fig. 3.8.

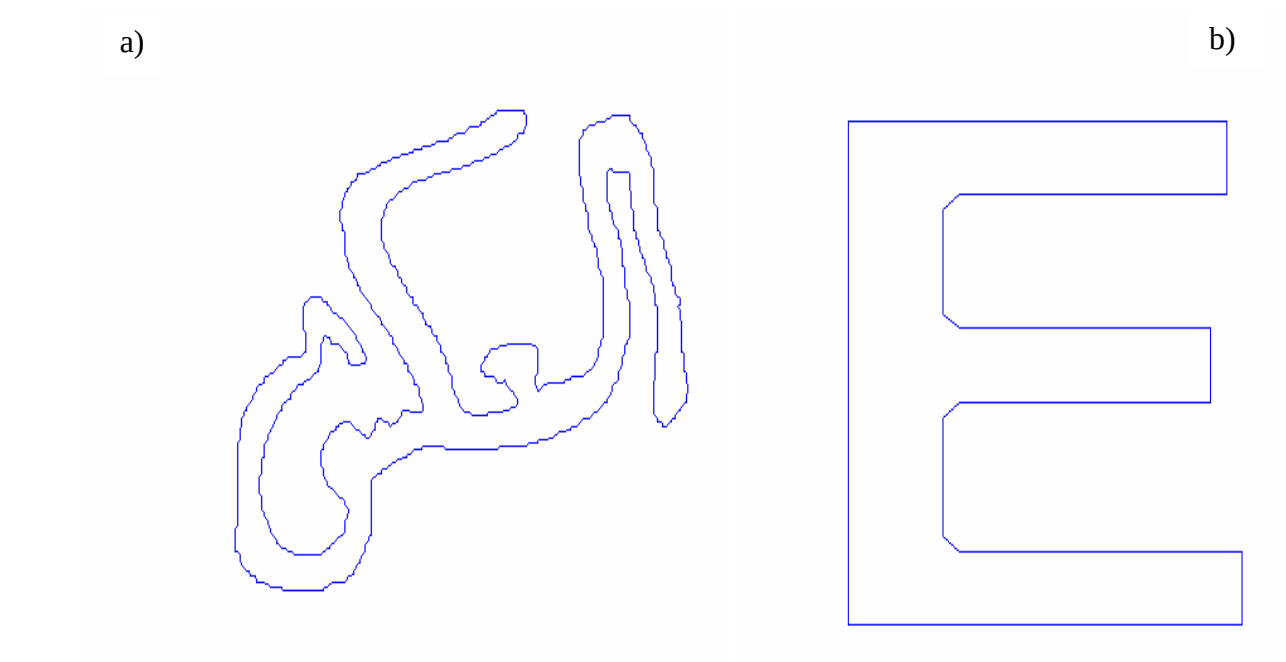


Fig. 3.8 Test contours for a) Arabic name, and b) E letter

Fig. 3.9 shows the SNR versus N of different threshold (*th*) for the Italy, Arabic and E letter contours respectively.

Finally we can say to obtain higher compression ratio without any significant visible distortion in the reconstruction quality; the accepted level of the reconstruction quality is determined. The optimum range value of the threshold is around (0.1 - 0.2) as shown in Fig. 3.5 and Fig. 3.9.

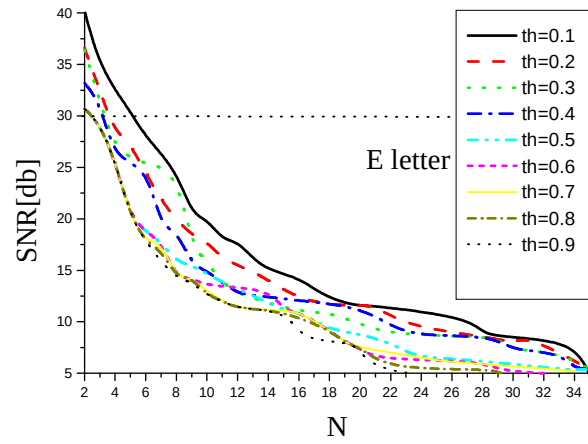
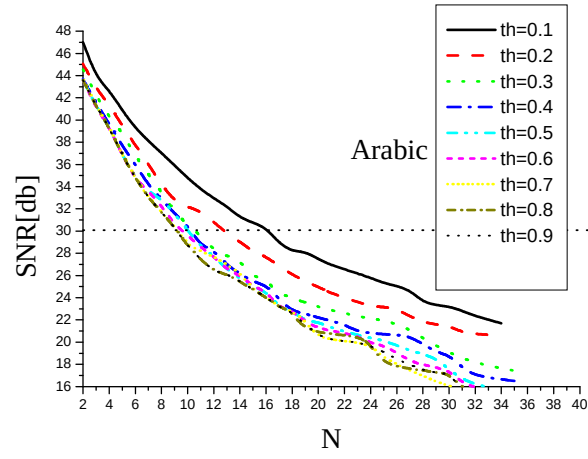
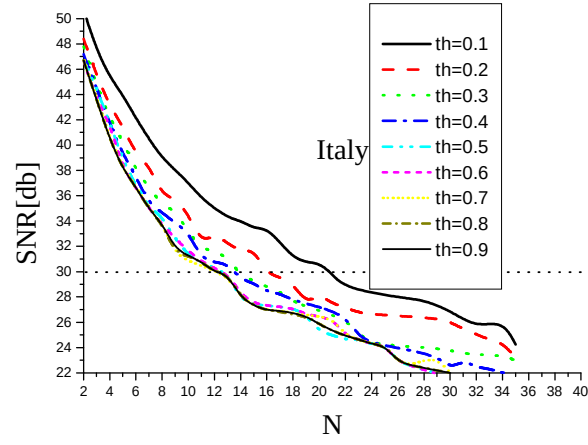


Fig. 3.9 SNR versus N for different thresholds of Italy, Arabic and E letter contours

3.3.3 Results of the experiments

Some exemplary reconstructions of Italy contour, are shown in Fig. 3.10 (related results are shown in Table 3.3).



Fig. 3.10 Results of approximation for the Italy contour

Table 3.3 Results of approximation for the Italy contour

	<i>MSE</i>	<i>SNR</i>	<i>CR [%]</i>	<i>NO</i>
a)	0.73	40.01	89.39	3413
b)	2.10	35.41	92.46	2424
c)	3.47	33.22	94.41	1738
d)	6.58	30.45	96.18	1180

NO – number of operations

Some selected results of Rose contour are shown in Fig. 3.11 (related results are shown in Table 3.4).



Fig. 3.11 Results of approximation for the Rose contour

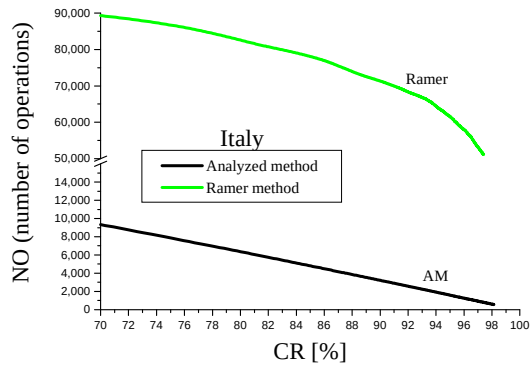
Table 3.4 Results of approximation for the Rose contour

	MSE	SNR	CR [%]	NO
a)	0.32	46.45	87.23	17220
b)	1.62	39.38	92.89	9743
c)	3.60	35.91	94.90	7033
d)	11.93	30.70	96.50	4816

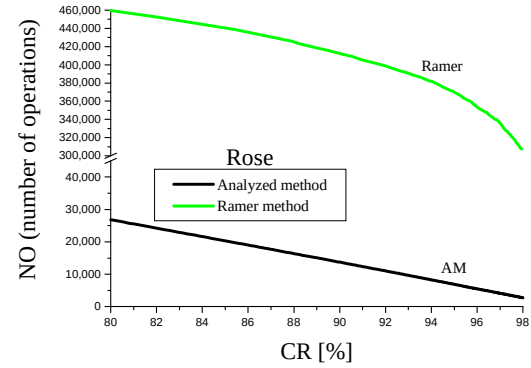
NO – number of operations

The results presented in Fig. 3.10 and Fig. 3.11 show that the analyzed method has good compression abilities. We can state, that compression ratio for analysed contours can be greater than 96.49% with some significant losses of approximation quality.

Comparison of the compression abilities of the analyzed method with the Ramer method is presented in Fig. 3.12 for the analysed contours of Italy and Rose.



(a)

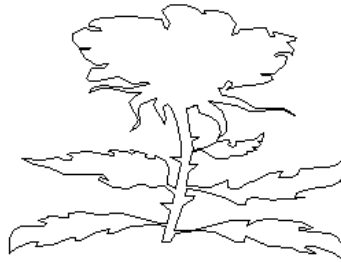


(b)

Fig. 3.12 Comparison of the analyzed method with the Ramer method for the contours of (a) Italy, and (b) Rose

It is shown that the analyzed algorithm is much faster than that of Ramer. The approximated Rose contour using Ramer and analyzed method for CR = 93.32% is shown in Fig. 3.13 (related results are shown in Table 3.5).

a)



b)

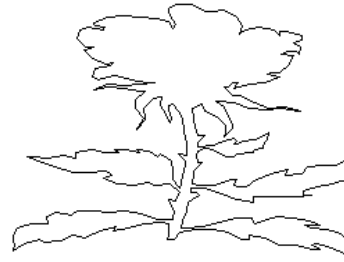


Fig. 3.13 Approximated Rose contour using a) Ramer method, and b) Analyzed method

Table 3.5 Results of approximated Rose contour using Ramer and analyzed methods

THE METHOD	MSE	SNR	CR [%]	NO
Ramer	0.63	43.50	93.32	388031
analyzed method	1.70	39.17	93.32	9067

NO – number of operations

The comparison of the analyzed method with the tangent and Ramer methods has also been performed. The plots of MSE and SNR versus CR are shown in Fig. 3.14 and Fig. 3.15 respectively.

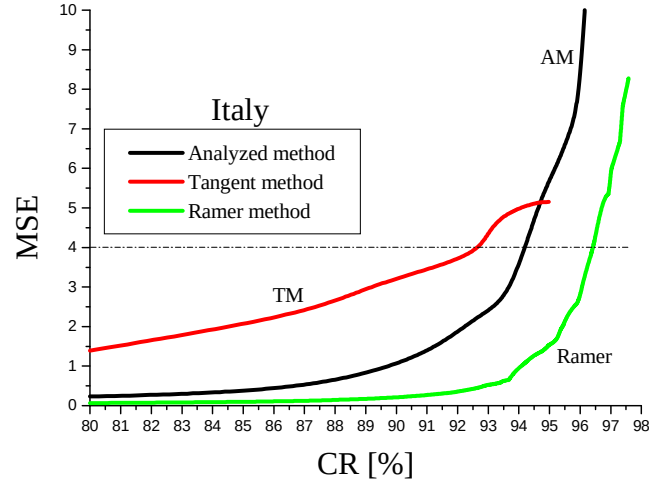


Fig. 3.14 Comparison of the analyzed method with the Ramer and tangent methods of MSE versus CR for the Italy contour

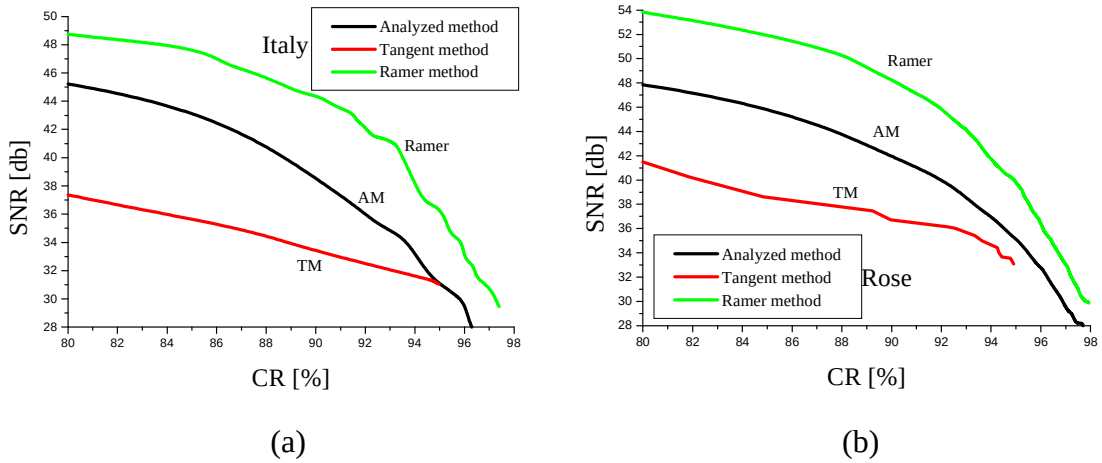


Fig. 3.15 Comparison of the analyzed method with the Ramer and tangent methods of SNR versus CR for the contours of (a) Italy, and (b) Rose

The plots show that SNR using the Ramer algorithm is better than that of tangent method and close to the analyzed method. The quality of the analyzed algorithm is better and also

faster than the tangent method. The presented results show that the analyzed method is faster than that of Ramer in all analysed cases.

The reason for the large number of operations in Ramer method is related to the searching of the maximum distance from the curve segment and straight line between starting and ending points of contour segment.

3.4 Algorithm for contour approximation using the triangle height method

This algorithm refers to a recent polygonal approximating method called height triangle method.

3.4.1 Description of the algorithm

The height of the triangle (h) is compared with the given threshold value using Equation (3.1). The idea of the analyzed algorithm is as illustrated before in Fig. 3.2. The description of the algorithm can be found in the subsection (3.3.1).

A flowchart of the analyzed algorithm is as depicted before in Fig. 3.3 (the triangle height is compared with the threshold value).

3.4.2 Results of the experiments

Some exemplary reconstructions of Italy contour, are shown in Fig. 3.16 (related results are shown Table 3.6).

Table 3.6 Results of approximation for the Italy contour

	<i>MSE</i>	<i>SNR</i>	<i>CR [%]</i>	<i>NO</i>
a)	0.66	40.46	87.34	3276
b)	1.59	36.63	91.34	2208
c)	2.34	34.94	92.74	1893
d)	7.17	30.08	95.53	1135

NO – number of operations

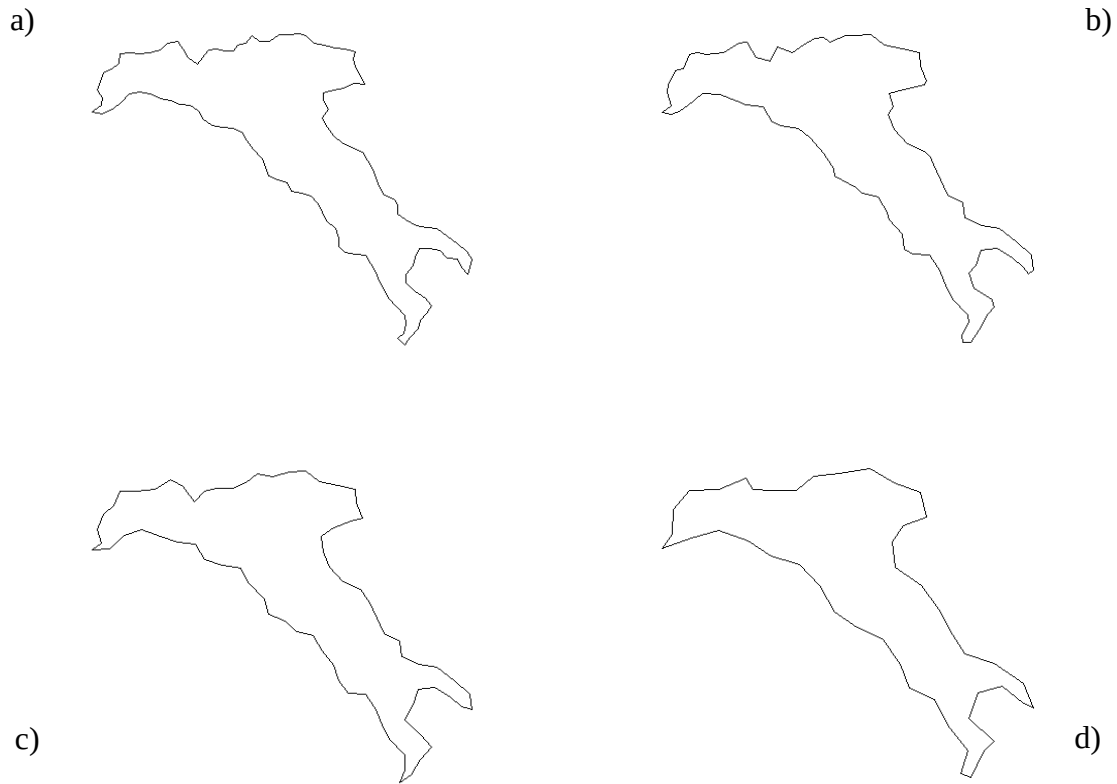


Fig. 3.16 Results of approximation for the Italy contour

Some selected results of Rose contour are shown in Fig. 3.17 (related results are shown in Table 3.7).

Table 3.7 Results of approximation for the Rose contour

	<i>MSE</i>	<i>SNR</i>	<i>CR [%]</i>	<i>NO</i>
a)	0.54	44.18	87.14	14025
b)	1.65	39.29	91.58	9277
c)	3.61	35.90	94.21	6381
d)	12.09	30.65	96.48	3909

NO – number of operations



Fig. 3.17 Results of approximation for the Rose contour

The results presented in Figs. 3.16 and 3.17 show that the analyzed method has good compression abilities. We can state, that compression ratio for analysed contours can be greater than 96.45% with some significant loses of approximation quality.

Comparison of the compression abilities of the analyzed method with the Ramer method is presented in Fig. 3.18 for the analysed contours of Italy and Rose.

The plot shows that the number of arithmetic operations for the analyzed method is less than that of Ramer method. So, the analyzed algorithm is much faster than that of Ramer. From Fig. 3.18 (Rose plot) we can see that when the $CR = 96\%$ the number of operations for the analyzed method is only 5000 while the number of operations using Ramer method is 350000 (i.e. more than seventy times); and that is the main advantage of the analyzed algorithm.

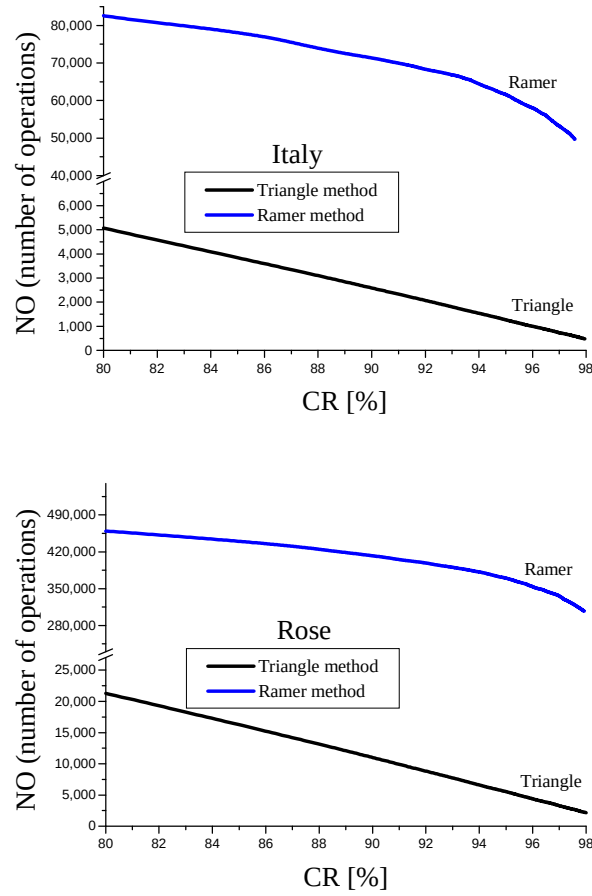
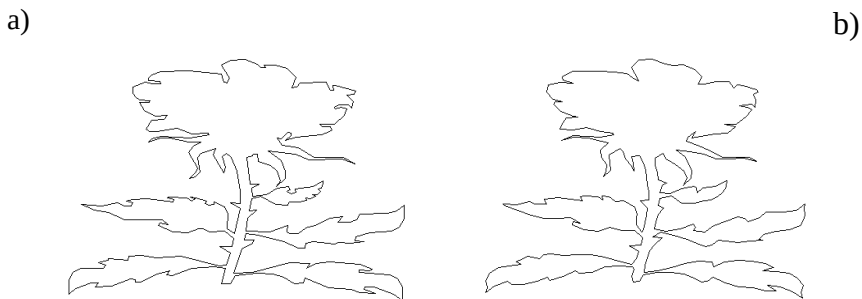


Fig. 3.18 Comparison of the analyzed method with the Ramer method for the contours of Italy and Rose

It is shown that the analyzed algorithm is much faster than that of Ramer. The approximated Rose contour using Ramer and analyzed method for CR = 93.32% is shown in Fig. 3.19 (related results are shown in Table 3.8).



**Fig. 3.19 Approximated Rose contour using
a) Ramer method, and b) Analyzed method**

**Table 3.8 Results of approximated
Rose contour using Ramer and analyzed methods**

THE METHOD	MSE	SNR	CR [%]	NO
Ramer	0.63	43.50	93.32	388031
analyzed method	2.54	37.42	93.32	7315

NO – number of operations

The comparison of the analyzed method with the tangent and Ramer methods has also been performed. The plots of MSE and SNR versus CR are shown in Fig. 3.20 and Fig. 3.21 respectively.

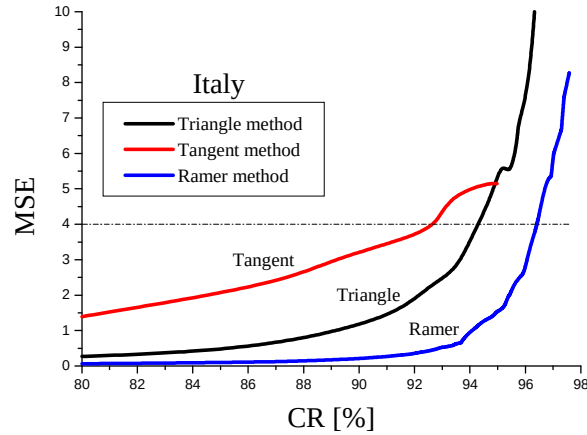


Fig. 3.20 Comparison of the analyzed method with the Ramer and tangent methods for MSE versus CR of the Italy contour

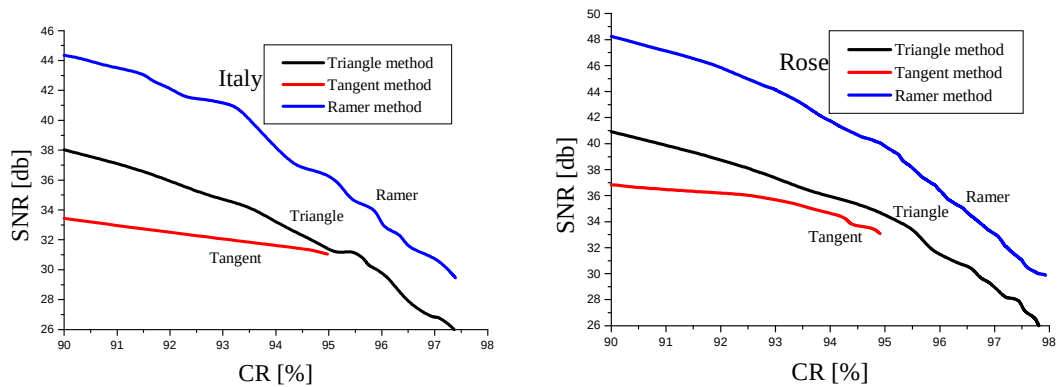


Fig. 3.21 Comparison of the analyzed method with the Ramer and tangent methods of SNR versus CR for the contours of Italy and Rose

The plots show that SNR using the Ramer algorithm is better than that of tangent method and close to the analyzed method. The quality of the analyzed algorithm is better and also faster than the tangent method. The presented results show that the analyzed method is faster than that of Ramer in all analysed cases.

3.5 Algorithm for contour approximation using the triangle height square method

This algorithm refers to a recent polygonal approximating method called height square triangle method.

3.5.1 Description of the algorithm

The squared of the triangle height (h^2) is compared with the given threshold value using Equation (3.1). The idea of the analyzed algorithm is as illustrated before in Fig. 3.2. The description of the algorithm can be found in the subsection (3.3.1).

A flowchart of the analyzed algorithm is as depicted before in Fig.3.3 (the square of the triangle height is compared with the threshold value).

3.5.2 Results of the experiments

Some exemplary reconstructions of Italy contour, are shown in Fig. 3.22 (related results are shown in Table 3.9).

Table 3.9 Results of approximation for the Italy contour

	<i>MSE</i>	<i>SNR</i>	<i>CR [%]</i>	<i>NO</i>
a)	0.77	39.77	90.78	2372
b)	2.07	35.46	93.48	1676
c)	3.51	33.17	95.44	1154
d)	6.23	30.69	96.37	921

NO – number of operations

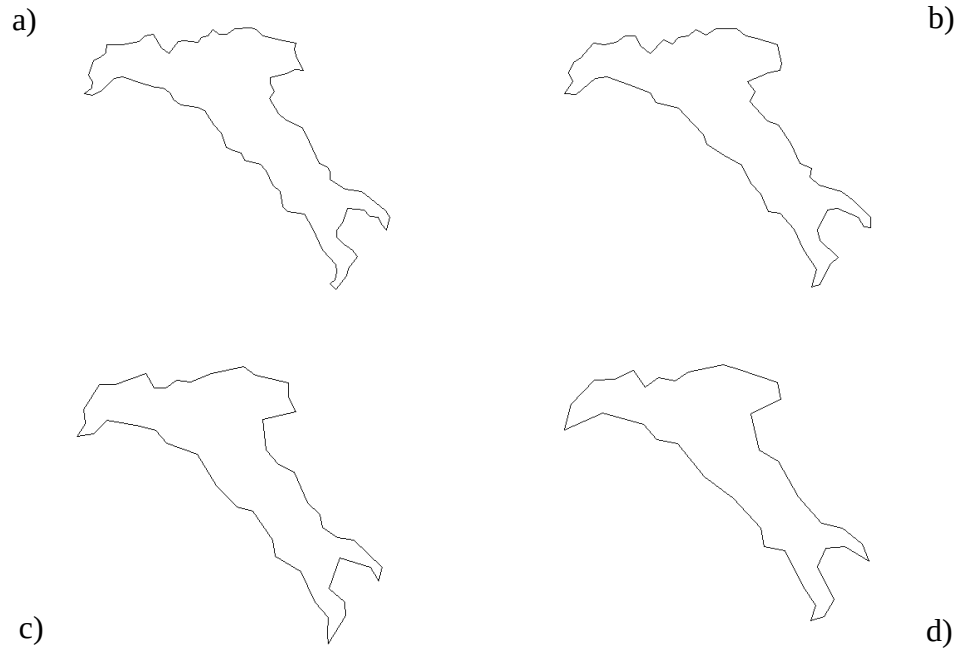


Fig. 3.22 Results of approximation for the Italy contour

Some selected results of Rose contour are shown in Fig. 3.23 (related results are shown in Table 3.10).



Fig. 3.23 Results of approximation for the Rose contour

Table 3.10 Results of approximation for the Rose contour

	<i>MSE</i>	<i>SNR</i>	<i>CR [%]</i>	<i>NO</i>
a)	0.31	46.51	87.16	14369
b)	1.63	39.36	92.93	7802
c)	3.59	35.92	95.32	5253
d)	12.00	30.70	97.08	3239

NO – number of operations

The results presented in Fig. 3.22 and Fig. 3.23 show that analyzed method has good compression abilities. We can state, that compression ratio for analysed contours can be greater than 97% with some significant loses of approximation quality.

Comparison of the compression abilities of the analyzed method with the Ramer method is presented in Fig. 3.24 for the analysed contours of Italy and Rose.

The plots show that the analyzed algorithm is much faster than that of Ramer. The approximated Rose contour using Ramer and analyzed method for $CR = 93.32\%$ is shown in Fig. 3.25 (related results are shown in Table 3.11).

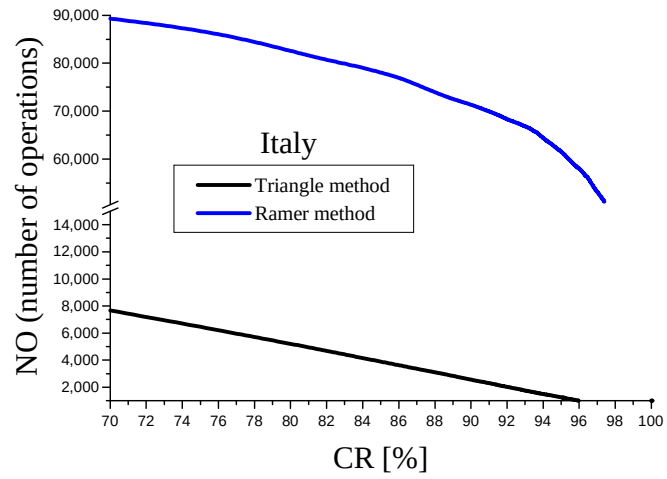


**Fig. 3.25 Approximated Rose contour using
a) Ramer method, and b) Analyzed method**

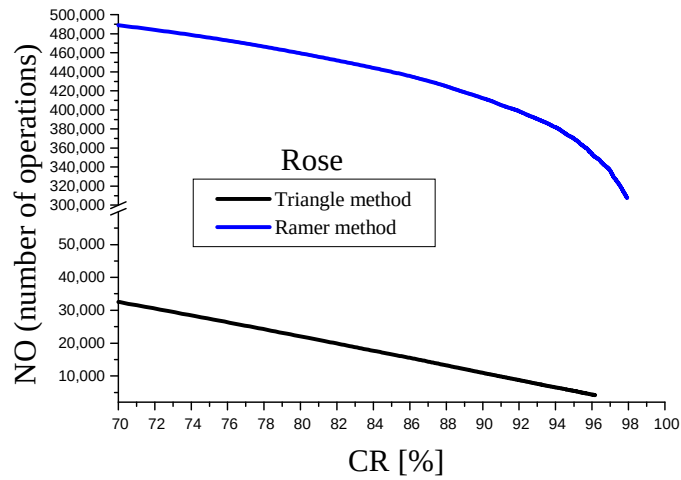
**Table 3.11 Results of approximated
Rose contour using Ramer and analyzed methods**

THE METHOD	MSE	SNR	CR [%]	NO
Ramer	0.63	43.50	93.32	388031
analyzed method	1.56	39.55	93.32	7458

NO – number of operations



(a)



(b)

**Fig. 3.24 Comparison of the analyzed method with the
Ramer method for the contours of (a) Italy, and (b) Rose**

The comparison of the analyzed method with the tangent and Ramer methods has also been performed. The plots of SNR and MSE versus CR are shown in Figs. 3.26 and Fig. 3.27 respectively.

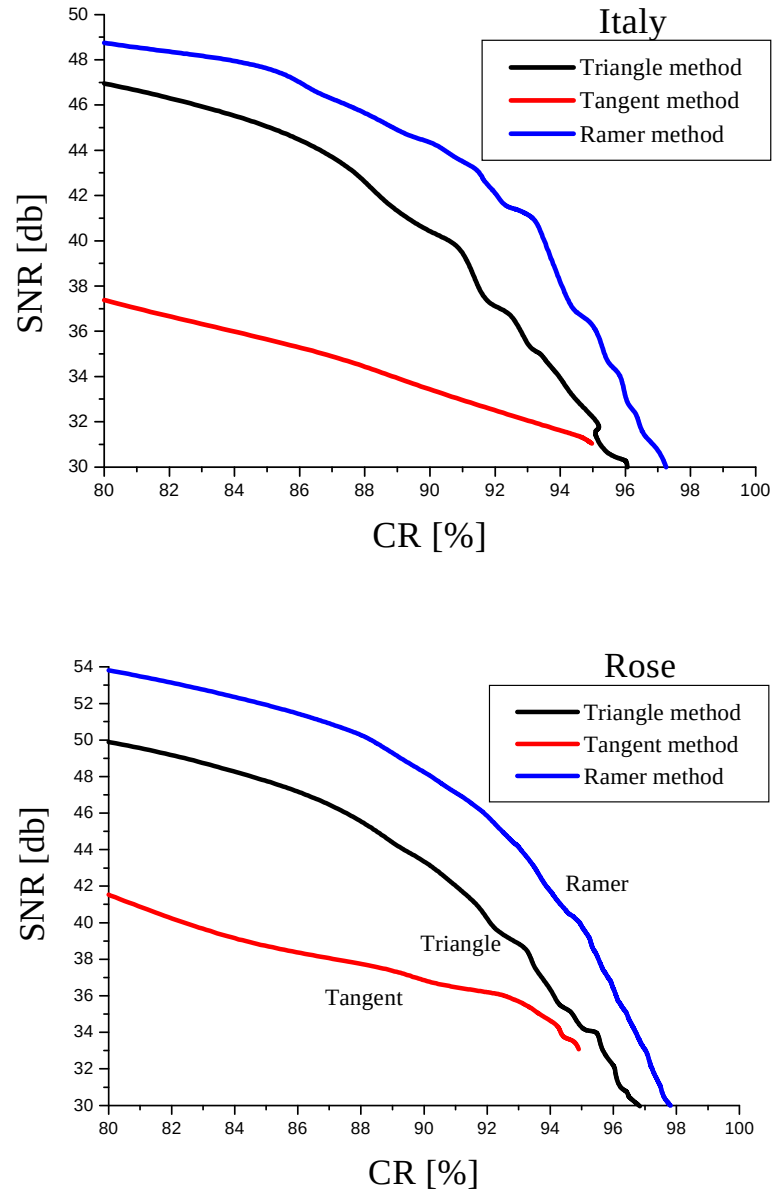


Fig. 3.26 Comparison of the analyzed method with the Ramer and tangent methods of SNR versus CR for the contours of Italy and Rose

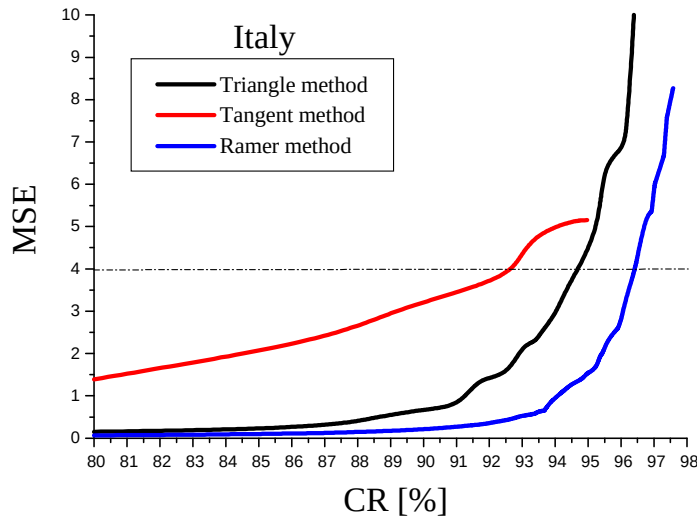


Fig. 3.27 Comparison of the analyzed method with the Ramer and tangent methods for MSE versus CR of the Italy contour

The plots show that SNR using the Ramer algorithm is better than the tangent method and close to the analyzed method. The quality of the analyzed algorithm is better and also faster than that of tangent method. The presented results show that the analyzed method is faster than that of Ramer in all analysed cases.

3.6 Algorithm for contour approximation using the triangle area method

This algorithm refers to a recent polygonal approximation method called triangle area method.

3.6.1 Description of the algorithm

The area of the triangle ($\frac{b \cdot h}{2}$) is compared with the given threshold value using Equation (3.1). The idea of the analyzed algorithm is as illustrated before in Fig. 3.2. The description of the algorithm can be found in the subsection (3.3.1).

A flowchart of the analyzed algorithm is as depicted before in Fig. 3.3 (the triangle area is compared with the threshold value).

3.6.2 Results of the experiments

Some exemplary reconstructions of Italy contour, are shown in Fig. 3.28 (related results are shown in Table 3.12).

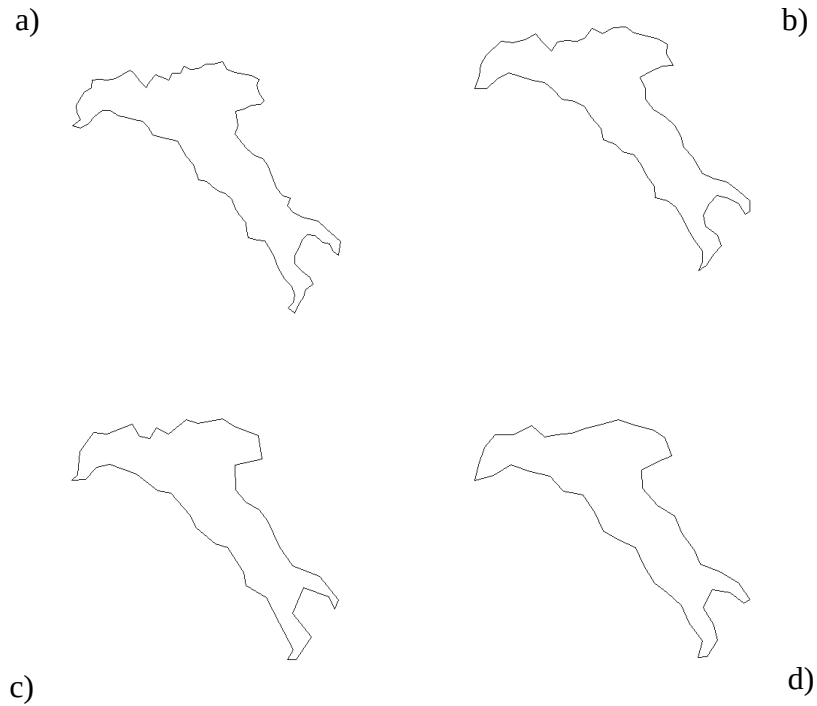


Fig. 3.28 Results of approximation for the Italy contour

Table 3.12 Results of approximation for the Italy contour

	<i>MSE</i>	<i>SNR</i>	<i>CR [%]</i>	<i>NO</i>
a)	0.72	40.04	90.13	3136
b)	2.04	35.53	92.46	2354
c)	3.52	33.17	95.53	1431
d)	6.55	30.50	95.16	1504

NO – number of operations

Some selected results of Rose contour are shown in Fig. 3.29 (related results are shown in Table 3.13).



Fig. 3.29 Results of approximation for the Rose contour

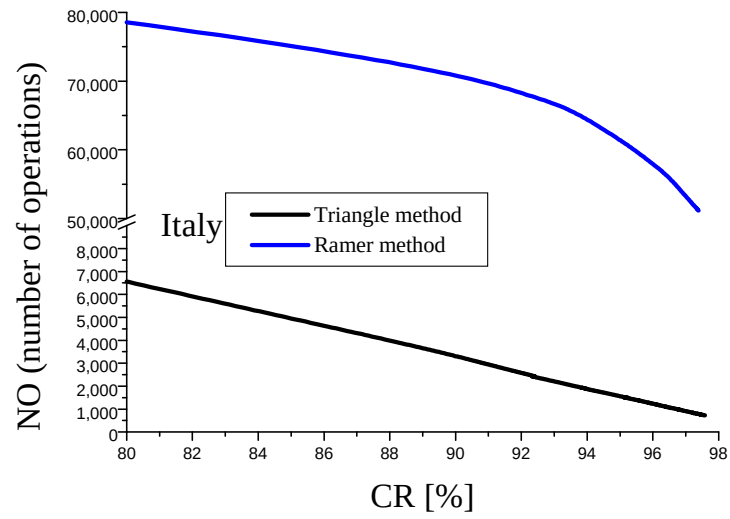
Table 3.13 Results of approximation for the Rose contour

	<i>MSE</i>	<i>SNR</i>	<i>CR [%]</i>	<i>NO</i>
a)	0.32	46.42	87.00	18119
b)	1.63	39.35	92.90	9804
c)	3.61	35.90	94.71	7244
d)	11.94	30.70	96.87	4298

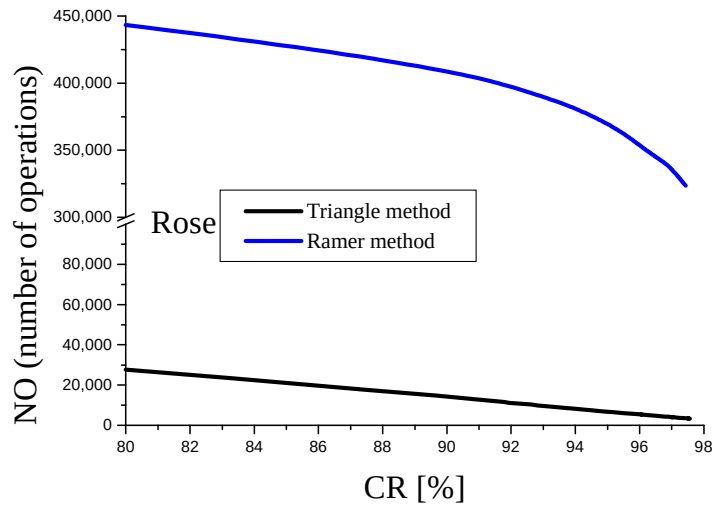
NO – number of operations

The results presented in Fig. 3.28 and Fig. 3.29 show that the analyzed method has good compression abilities. We can state, that compression ratio for analysed contours can be greater than 96.5% with some significant loses of approximation quality.

Comparison (Ramer and analyzed methods) of the compression abilities versus the number of operations is presented in Fig. 3.30 for the analysed contours of Italy and Rose. The plots show that the analyzed algorithm is much faster than that of Ramer.



(a)



(b)

Fig. 3.30 Comparison of the analyzed method with the Ramer method for the contours of (a) Italy, and (b) Rose

The approximated Rose contour using Ramer and analyzed method for $CR = 93.32\%$ is shown in Fig. 3.31 (related results are shown in Table 3.14).

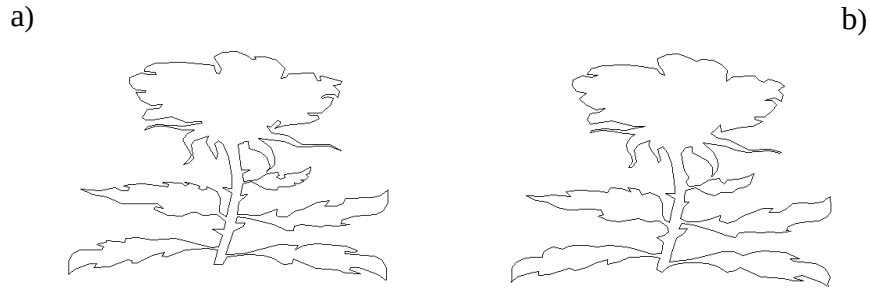


Fig. 3.31 Approximated Rose contour using
a) Ramer method, and b) Analyzed method

Table 3.14 Results of aapproximated
Rose contour using Ramer and analyzed methods

THE METHOD	MSE	SNR	CR [%]	NO
Ramer	0.63	43.50	93.32	388031
analyzed method	1.99	38.48	93.32	9097

NO – number of operations

The comparison of the analyzed method with the tangent and Ramer methods has also been performed. The plots of MSE and SNR versus CR are shown in Fig. 3.32 and Fig. 3.33 respectively.

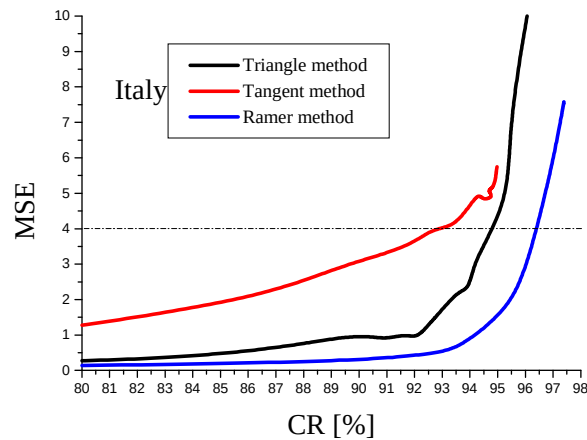
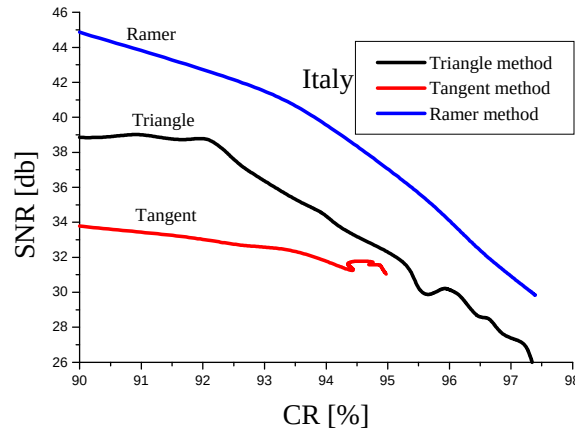
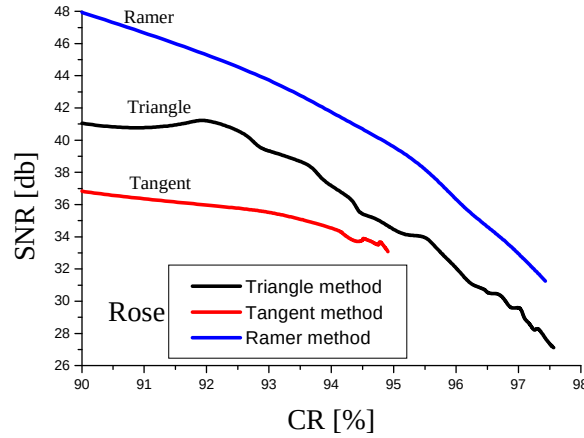


Fig.3.32 Comparison of the analyzed method with the
Ramer and tangent methods for MSE
versus CR of the Italy contour



(a)



(b)

Fig. 3.33 Comparison of the analyzed method with the Ramer and tangent methods for SNR versus CR of the contours of (a) Italy, and (b) Rose

The plots show that SNR using the Ramer algorithm is better than that of tangent method and close to the analyzed method. The quality of the analyzed algorithm is better and also faster than that of tangent method. The presented results show that the analyzed method is faster than that of Ramer in all analysed cases.

3.7 Comparison between the triangle family algorithms

High computational complexity leads to high implementation cost. Computational time is evaluated by calculating the number of operations that the algorithm takes; and

that is the important factor for comparison. The MSE and SNR criteria versus compression ratio are also used to evaluate the distortion between the purpose comparisons.

The comparison is done for some test contours (Italy & Rose) which was extracted by using the “SSPCE” (single step parallel contour extraction). The comparison is made between the following five algorithms:

- Triangle height over base (or hb) method; it will be referred to as the first algorithm.
- Triangle height (or h) method; it will be referred to as the second algorithm.
- Triangle height square (or hs) method; it will be referred to as the third algorithm.
- Triangle area (or area) method; it will be referred to as the fourth algorithm.
- Ramer method; it will be referred to as the fifth algorithm.

Comparison of the compression abilities versus the MSE & SNR are shown in Fig. 3.34 & Fig. 3.35 respectively.

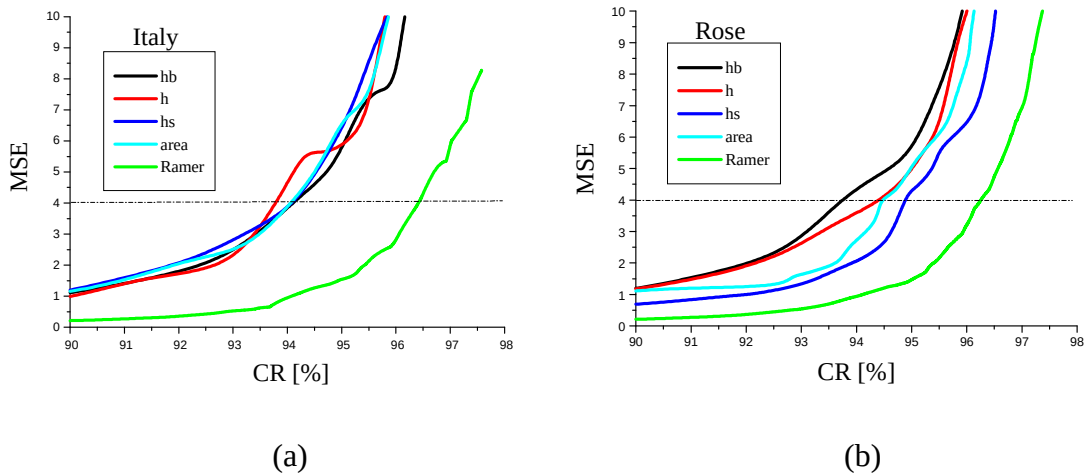
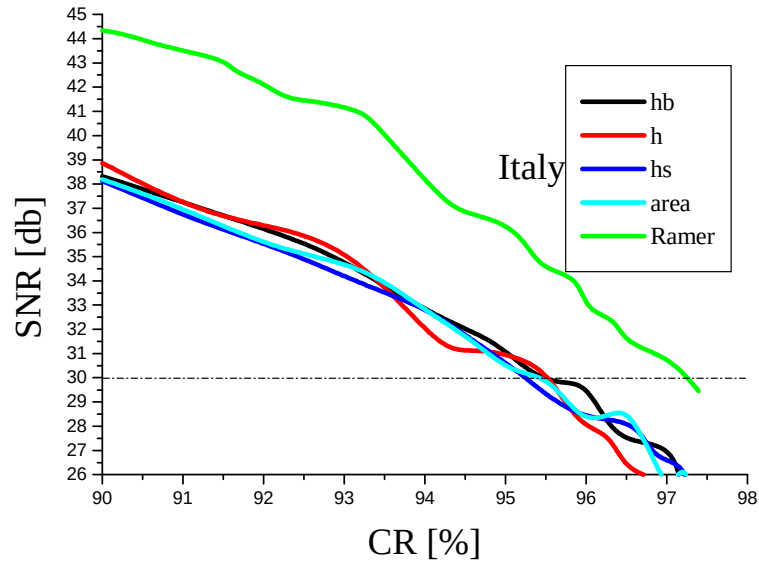
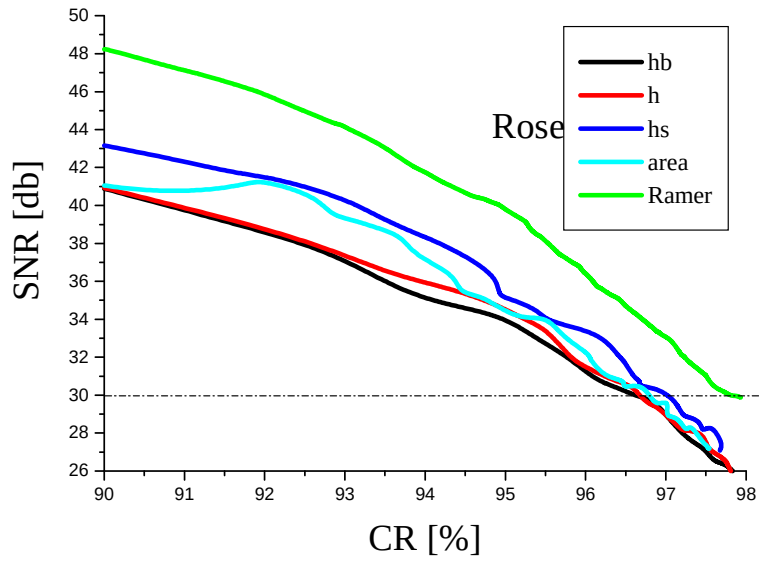


Fig. 3.34 MSE versus CR for (a) Italy contour, and (b) Rose contour



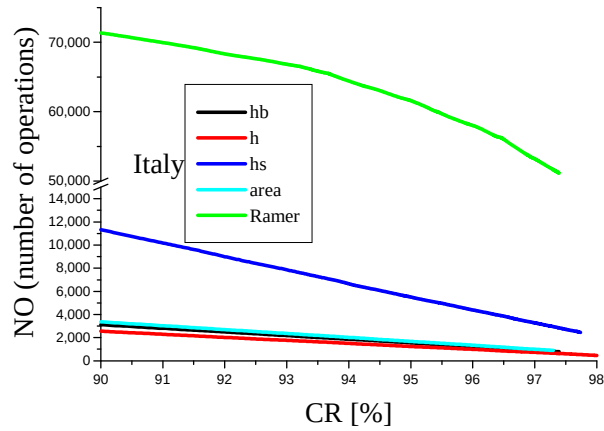
(a)



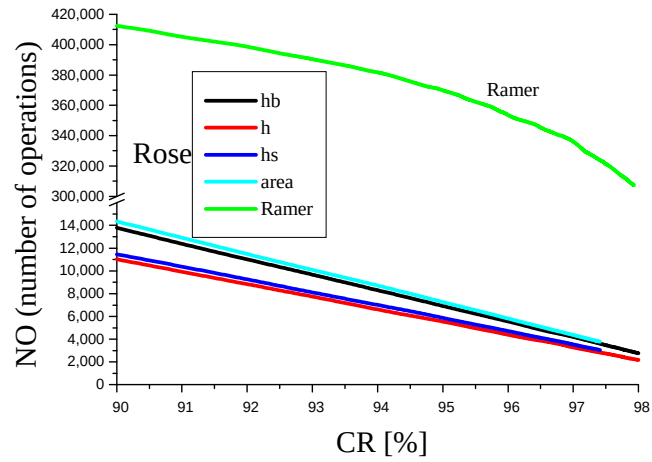
(b)

Fig. 3.35 SNR versus CR for (a) Italy contour, and (b) Rose contour

Comparison of the compression abilities versus the number of operations is presented in Fig. 3.36.



(a)



(b)

Fig. 3.36 NO versus CR for (a) Italy contour, and (b) Rose contour

The plots show that SNR using the Ramer algorithm is close to the triangle family methods if the contour is not as complicated as in Italy contour; the reconstruction quality by the triangle family algorithms are very similar but the (hs) method is much better for complicated contours as in Rose contour. The number of operations is very similar between the triangle family algorithms at high compression. The triangle family algorithms are many times faster than that of Ramer method. The compression ratio using triangle family methods can be even greater than 97% with some visible distortion and the complexity is much less than that of the Ramer algorithm.

Chapter 4

METHODS OF CONTOUR COMPRESSION IN TRANSFORM DOMAIN

4.1 Overview

The idea behind transform coding method is to represent the original signal with a small number of transform coefficients. In typical images, transform coding using rules says that a large amount of signal energy is concentrated in a few numbers of coefficients. Transform coding has an important property relevant for keeping distortion at an acceptable level while minimizing the number of transform coefficients and it is one of the important parts using in JPEG (Joint Photographic Experts Group) standard which is one of the most widely known standards for lossy image compression.

In this chapter we look at a number of different transforms, including the popular Walsh transform [48], [49], [50], [51] and [52]. There exists a class of transforms; based on the integration of Haar functions, known as periodic Haar piecewise-linear (PHL) transform [53]. Also a high pass filters are used to extract contours of real images and then by using the correlation functions these images can easily combined.

The main purpose of this chapter is to make comparisons for contour compression between time domain algorithms (centroid, segment distances ratio, triangle height square and Ramer) and transform domain algorithms (Walsh, DCT [54], Haar [56] and PHL). To present the results, the mean square error and signal-to-noise ratio criterions versus compression ratio are used. Computational times of analysed procedures are estimated based on a number of numerical operations.

First, the extraction steps are done then the compression steps follow. Fig. 4.1 shows two stages. The first stage is the extracted contour; while the second one is the compressed contour. The input of the first stage is colour image and the extracted contour is the output of that stage. The input of the second stage is the contour signal ($X(n)$ and $Y(n)$); and the output is the approximated contour ($X(n)$ and $Y(n)$).

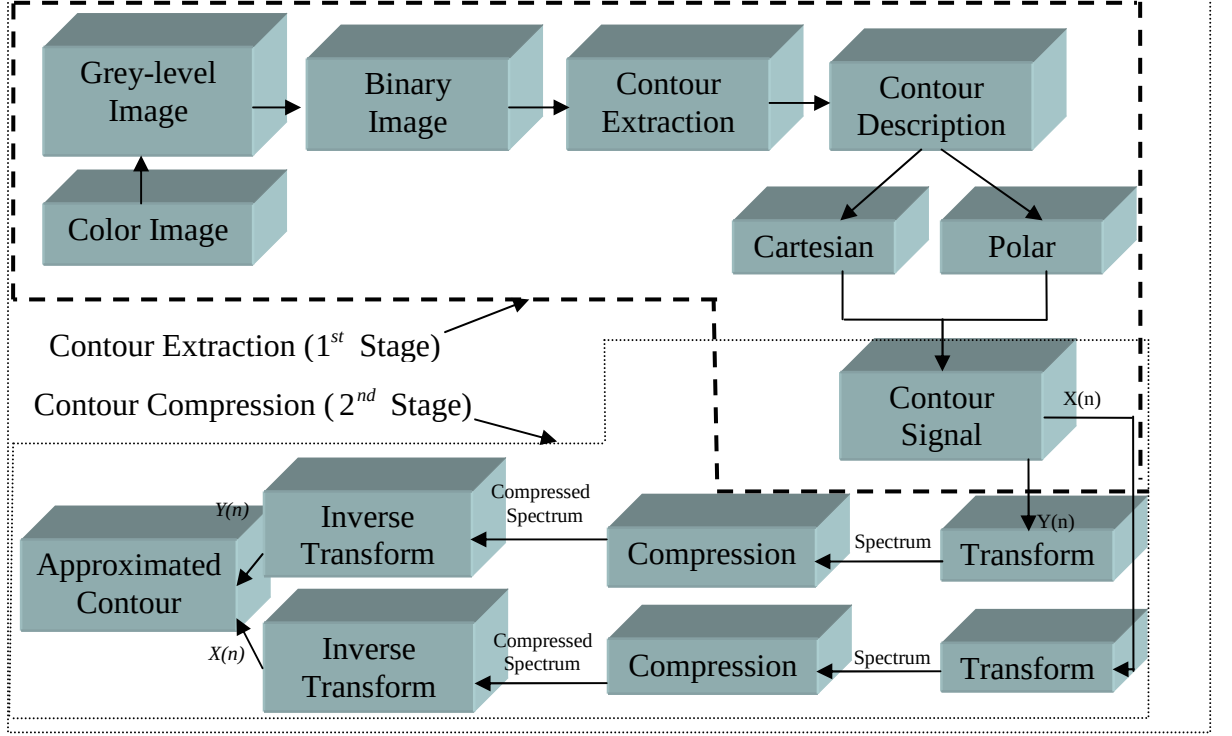


Fig 4.1 Block diagram for contour extraction and compression in spectral domain

4.2 Periodic Haar piecewise-linear (PHL) transform

The set of N Haar functions is defined by [53]

$$har(0, t) = 1 \quad \text{for } t \in [0, T], \quad (4.1)$$

usually $T = 1$.

$$har(i, t) = \begin{cases} 2^{\frac{g-1}{2}} & \text{for } [\frac{i}{2^{g-1}} - 1] \leq t < [\frac{i+1}{2^{g-1}} - 1] \\ -2^{\frac{g-1}{2}} & \text{for } [\frac{i+1}{2^{g-1}} - 1] \leq t < [\frac{i+2}{2^{g-1}} - 1] \\ 0 & \text{otherwise} \end{cases} \quad (4.2)$$

where $0 < g \leq \log_2 N$ and $1 \leq i < 2^g$.

The set of periodic Haar piecewise linear functions are determined by

$$\text{PHL}(0,t) = 1, \quad t \in (-\infty, \infty) \quad (4.3)$$

$$\text{PHL}(1,t) = \left[\frac{2}{T} \int_{uT}^{t+uT} \text{har}(1,\tau) d\tau \right] + \frac{1}{2}$$

$$\text{PHL}(i+1,t) = \frac{2^{g+1}}{T} \int_{uT}^{t+uT} \text{har}(i+1,\tau) d\tau \quad (4.4)$$

where $i = 1, 2, \dots, N-2$, $g = 1, 2, \dots, (\log_2 N) - 1$, $u = 0, 1, 2, \dots, g$ is index of group of PHL functions and u is number of period.

The normalization factor (2^{g+1}) is applied to normalize the maximum amplitude of the PHL functions. The set of PHL functions is linearly independent but not orthogonal.

A continuous signal $x(t)$, defined over the interval $[0, T]$, with finite energy, can be expanded into the following PHL series

$$x(t) = \sum_{i=0}^{\infty} c_i \cdot \text{PHL}(i,t), \quad t \in (-\infty, \infty) \quad (4.5)$$

The coefficients of expansion c_i are defined by

$$c_0 = x(0) \quad (4.6)$$

$$c_i = -\frac{1}{2^{g+1}} \int_0^T x(t) \cdot \text{har}'(i,t) dt \quad (4.7)$$

where $i = 1, 2, \dots, N-1$; $g = 1, 2, \dots, (\log_2 N) - 1$ and $\text{har}'(i,t)$ are all derivatives of 1D Haar functions, i.e. $\text{har}'(i,t) = \lim_{\Delta t \rightarrow 0} [\text{har}(i, t + \Delta t) - \text{har}(i, t)]$.

Since the derivatives of Haar functions are represented by discrete values, thus for a signal $x(t)$ sampled in N points, we can write the PHL transform coefficients as

$$c_0 = x(0) \quad (4.8a)$$

$$c_i = -\frac{1}{2^{g+1}} \sum_{n=0}^{N-1} x(n) \cdot har'(i, n) \quad (4.8b)$$

where $x(n)$ is the discrete signal obtained by sampling $x(t)$.

Equation (4.8) lead to the PHL approximation of the signal $x(t)$. The approximated signal may be expressed as follows

$$x_a(t) = \sum_{i=0}^{N-1} c_i \cdot \text{PHL}(i, t) \quad (4.9)$$

To obtain the PHL spectrum (set of c_i coefficients), we form the matrix equation of the forward and inverse PHL transform as follows

a) Forward transform

$$[C(N)] = \left[-\frac{1}{2^{g+1}}\right] [\text{PHL}(N)][X(N)] \quad (4.10)$$

b) Inverse transform

$$[X(N)] = [\text{IPHL}(N)][C(N)] \quad (4.11)$$

where $[C(N)]$ is vector of PHL coefficients (PHL spectrum), $[X(N)]$ is vector of sampled signal, $[\text{PHL}(N)]$ is matrix of forward transform, $[\text{IPHL}(N)]$ is matrix of inverse transform and $\left[-\frac{1}{2^{g+1}}\right]$ is diagonal matrix of normalization.

$$[-\frac{1}{2^{g+1}}] = \text{diag}[1, -\frac{1}{2^1}, -\frac{1}{2^2}(2 \cdot \text{times}), -\frac{1}{2^3}(4 \cdot \text{times}), \dots, -\frac{1}{2^g}(2^{g-1} \cdot \text{times})] \quad (4.12)$$

The [PHL(N)] and [IPHL(N)] matrices, for N=8, are shown below

$$[\text{PHL}(8)] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & -2 & 0 & 0 & 0 \\ \sqrt{2} & 0 & -2\sqrt{2} & 0 & \sqrt{2} & 0 & 0 & 0 \\ \sqrt{2} & 0 & 0 & 0 & \sqrt{2} & 0 & -2\sqrt{2} & 0 \\ 2 & -4 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -4 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -4 & 2 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 2 & -4 \end{bmatrix} \quad (4.13)$$

$$[\text{IPHL}(8)] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1/4 & \sqrt{2}/2 & 0 & 2 & 0 & 0 & 0 \\ 1 & 1/2 & \sqrt{2} & 0 & 0 & 0 & 0 & 0 \\ 1 & 3/4 & \sqrt{2}/2 & 0 & 0 & 2 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 3/4 & 0 & \sqrt{2}/2 & 0 & 0 & 2 & 0 \\ 1 & 1/2 & 0 & \sqrt{2} & 0 & 0 & 0 & 0 \\ 1 & 1/4 & 0 & \sqrt{2}/2 & 0 & 0 & 0 & 2 \end{bmatrix} \quad (4.14)$$

The diagonal matrix of normalization will, in this case, have the following values:

$$[-\frac{1}{2^{g+1}}] = \text{diag}[1, -\frac{1}{2}, -\frac{1}{4}, -\frac{1}{4}, -\frac{1}{8}, -\frac{1}{8}, -\frac{1}{8}, -\frac{1}{8}] \quad (4.15)$$

It can be seen that

$$[-\frac{1}{2^{g+1}}][\text{PHL}(N)][\text{IPHL}(N)] = [\text{I}(N)] \quad (4.16)$$

where [I(N)] is the identity matrix.

Fig. 4.2 shows that the PHL transform is performed for each of $X(n)$ and $Y(n)$, then the threshold sampling is done and finally the IPHL is applied for each X and Y of PHL transforms to reconstruct the approximated contour (the related results are shown in Table 4.1).

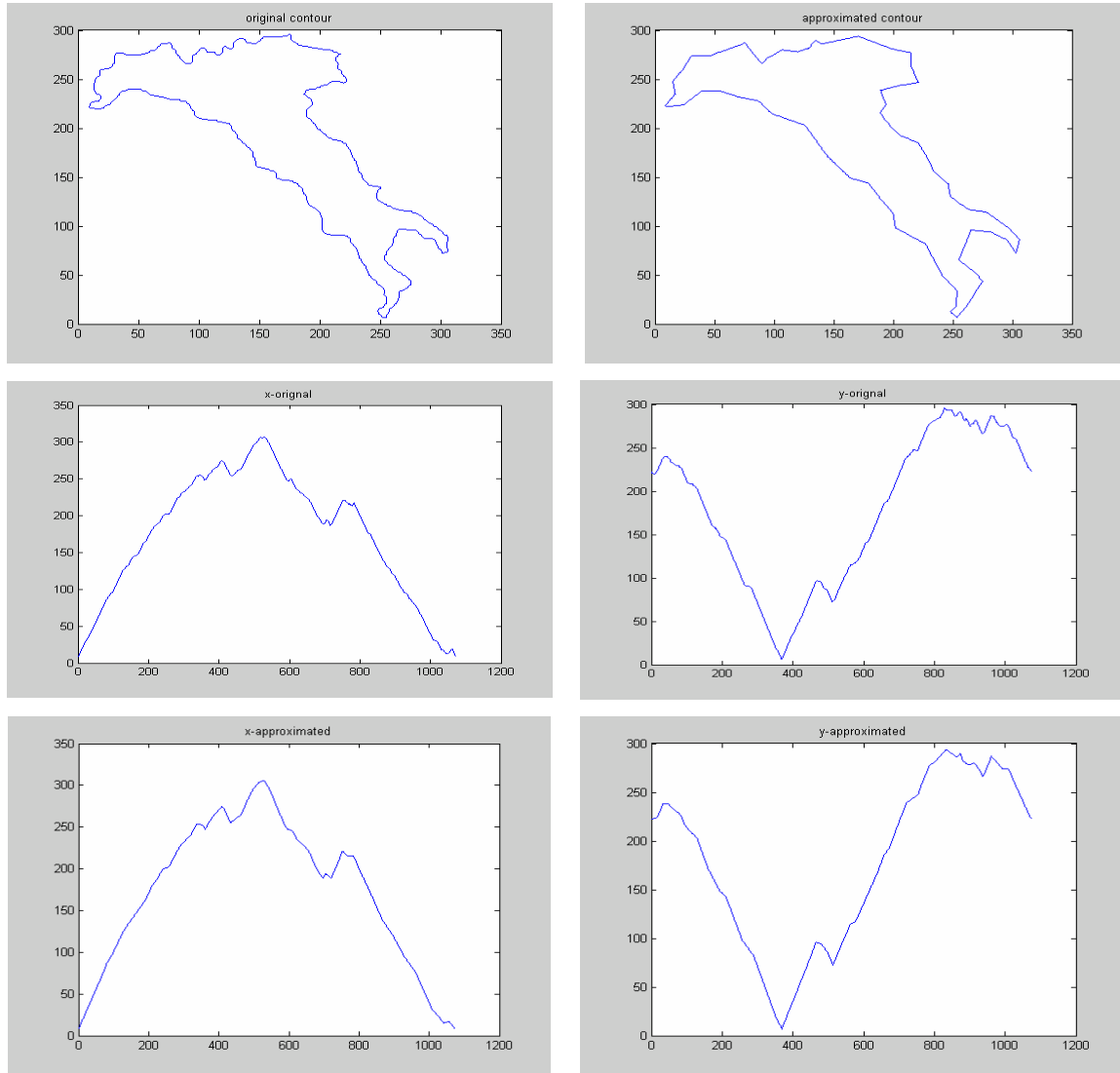


Fig. 4.2 Compressed contour using PHL transform

**Table 4.1 Results of approximation
for the Italy contour using PHL transform**

Threshold	MSE	SNR	CR [%]	NO
0.37	3.32	33.43	94.41	38638

NO – number of operations

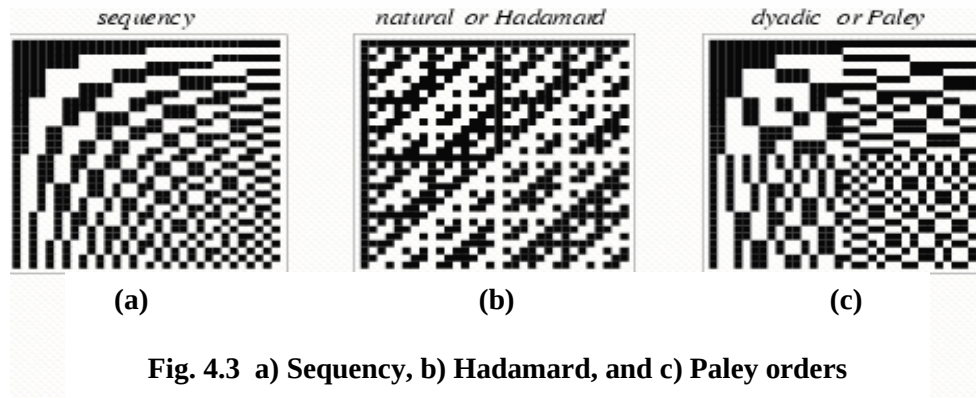
4.3 Walsh transform

Orthogonally of the all rows and all columns in the Hadamard matrix is by ordering the values +1 and -1 in such a way. The Kronecker product of any two Hadamard matrices gives a new Hadamard matrix. The (2x2) is the smallest size of the Hadamard matrix:

$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (4.17)$$

Ordering the rows in a certain way according to the number of zero crossings for the Hadamard matrix, the Walsh transform can be obtained. A Hadamard transform is discrete by definition. A Hadamard matrix has an orthogonal set when the all rows (columns) are ordering [50].

Fig. 4.3a represents the sequency order which is in each row, has one more color change than the preceding row. Fig. 4.3b represents the natural (or Hadamard) order in which the Walsh functions display a mixed construction. There is another order using the Gray code to recording the rows, known as dyadic (or Paley) [50] as shown in Fig. 4.3c.



The Hadamard transform coefficients are called sequence components and the Walsh functions are obtained by ordered the number of their zero-crossings. The Hadamard and Walsh ordered is shown in Fig. 4.4.

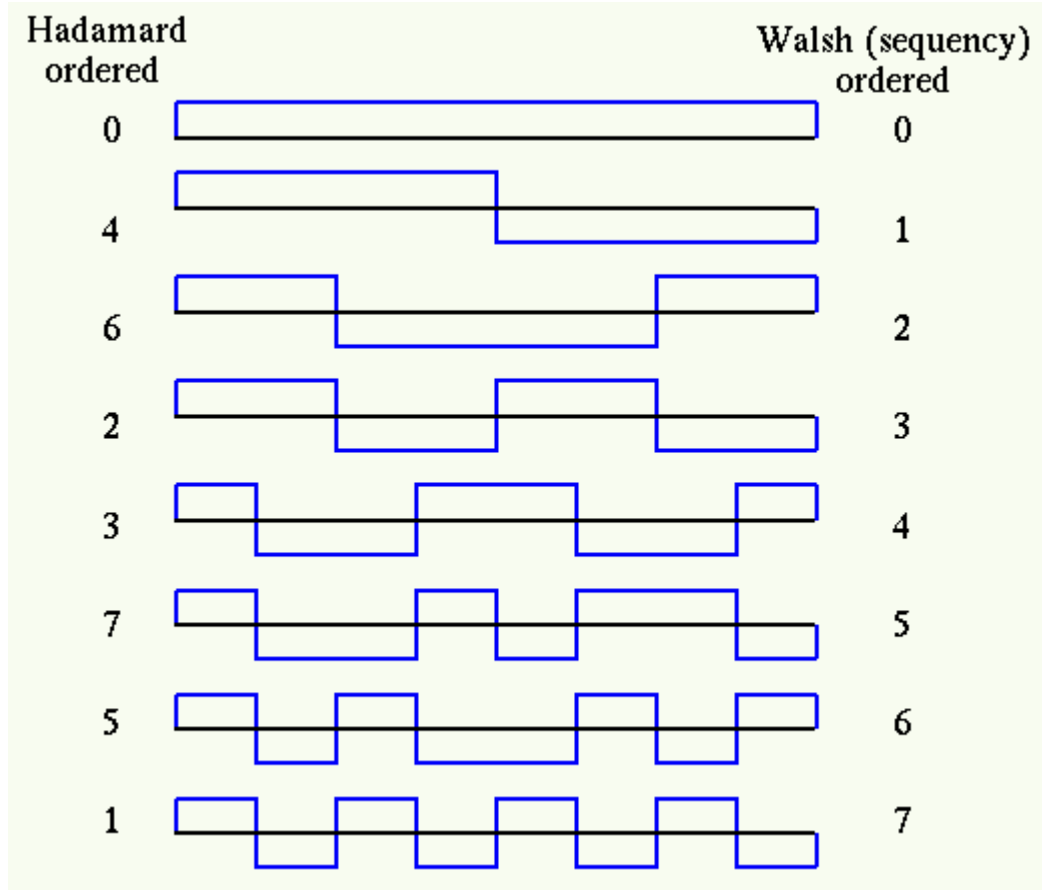


Fig. 4.4 Hadamard and Walsh ordered

The Walsh transform using Hadamard matrix (often referred as Hadamard transform) and its inverse are given by

$$[S(n)] = \frac{1}{N} [H(n)] [X(n)] \quad (4.18)$$

$$[X(n)] = [H(n)]^{-1} [S(n)]$$

where $[S(n)]$ is vector of spectral components and $[H(n)]$ is Hadamard matrix.

Fig. 4.5 shows the Walsh transform is performed for each of $X(n)$ and $Y(n)$, then the threshold sampling is done and finally the inverse Walsh is applied for each X and Y of Walsh transform to reconstruct the approximated contour (the related results are shown in Table 4.2).

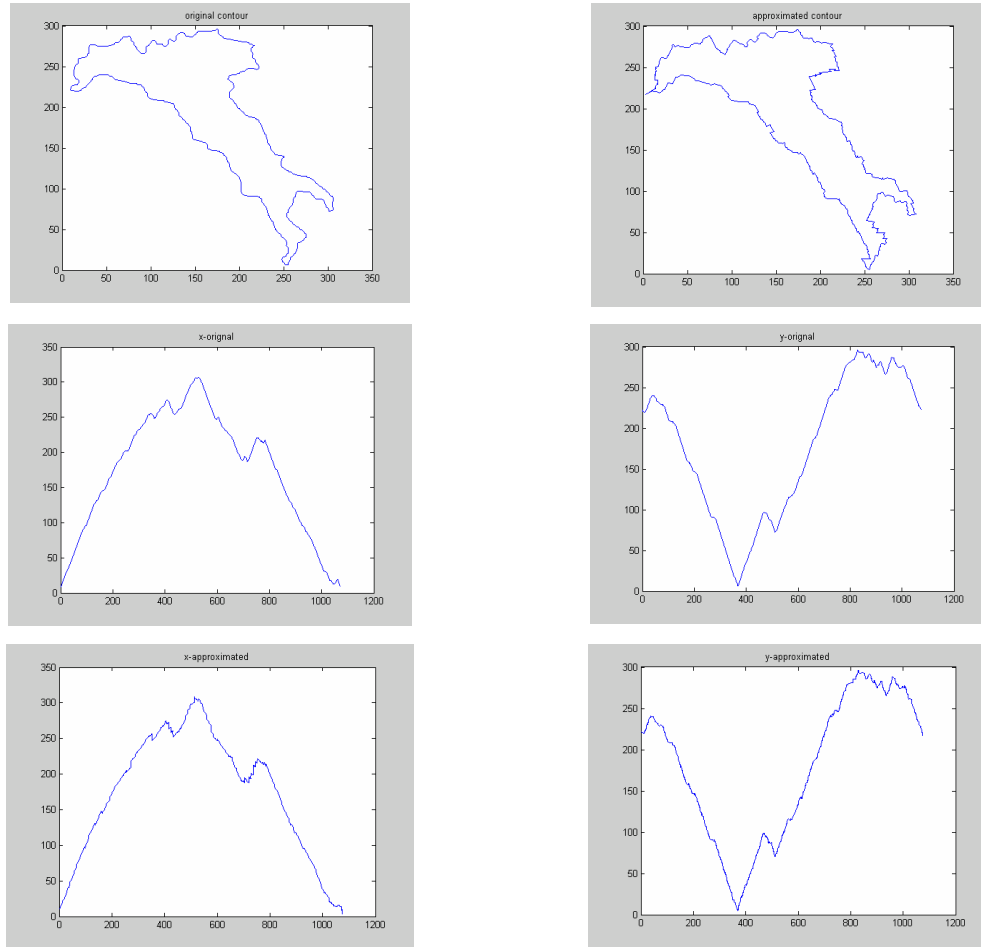


Fig. 4.5 Compressed contour using Walsh transform

**Table 4.2 Results of approximation
for the Italy contour using Walsh transform**

Threshold	MSE	SNR	CR [%]	NO
0.15	1.54	36.75	49.21	302486

NO – number of operations

4.4 Discrete cosine transform (DCT)

The amount of variations (image's visual quality) is different from one basis to another using discrete cosine transform (DCT). The DCT is similar to the discrete Fourier

transform (DFT) in order to transform a signal or image from the spatial domain to the frequency domain, but it gives better results for lines approximation and energy compaction.

Its name comes from the fact that the basis vectors of the transform matrix C are obtained as a function of sampled cosines

$$[C]_{i,j} = \begin{cases} \sqrt{\frac{1}{N}} & \text{for } i=0, j=0,1,\dots,N-1 \\ \sqrt{\frac{2}{N}} \cos \frac{(2j+1)i\pi}{2N} & \text{for } i=1,2,\dots,N-1, j=0,1,\dots,N-1 \end{cases} \quad (4.19)$$

Simplest implementation is obtained using the DCT as a set of basis functions which given a known input array size (for example 8 x 8 windows) for pre-computed and stored processes. Applying the convolution of the mask (8 x 8 windows) across all rows/columns of the image exhibits simple computations values. Calculations of these values are easier by using the DCT formula. Fig. 4.6 represents a 64 (8 x 8) DCT basis functions.

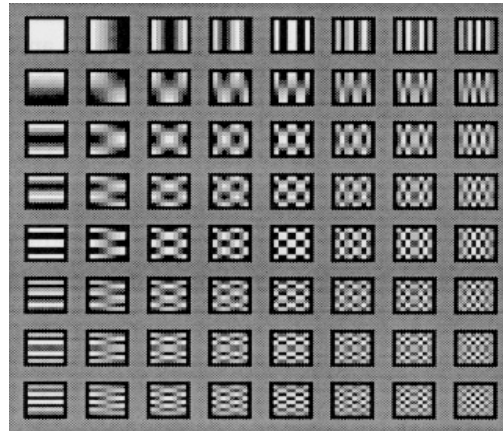


Fig. 4.6 The basis functions for the DCT transform

The two-dimensional block of the input signal giving by the following equation

$$F(m, n) = \frac{2}{\sqrt{MN}} C(m)C(n) \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \cos \frac{(2x+1)m\pi}{2M} \cos \frac{(2y+1)n\pi}{2N} \quad (4.20)$$

where $C(m), C(n) = \frac{1}{\sqrt{2}}$ for $m, n = 0$ and $C(m), C(n) = 1$ otherwise.

Performing 1D DCT (vertically for columns) and 1D DCT (horizontally) to resultant vertical DCT or alternately horizontal to vertical; gives the output array of DCT coefficients.

Most software implementations use fixed point arithmetic. All multiplies in some of the fast implementations contains shifts and adds for approximated coefficients. Because all multiplications using DCT are real, the number of required multiplications are minimized compared to the discrete Fourier transform. If for example the DCT input is an 8 by 8 array of integers in which each pixel contains the grey-scale level; the range of the levels is from 0 to 255 if the pixel is presented by 8 bit. In this case the output array of DCT coefficients contains integers in the range (from -1024 to 1023).

In typical images the energy of the signal concentrated at low frequencies and usually appears in the upper left corner of the DCT image. Higher frequencies are represented by the lower right values and are often small enough to be neglected with small visible distortion.

Fig. 4.7 shows the DCT transform is performed for each of $X(n)$ and $Y(n)$, then the threshold sampling is done and finally the IDCT is applied for each X and Y of DCT to reconstruct the approximated contour (the related results are shown in Table 4.3).

**Table 4.3 Results of approximation
for the Italy contour using DCT transform**

Threshold	MSE	SNR	CR [%]	NO
22.40	3.32	33.42	92.37	720727

NO – number of operations

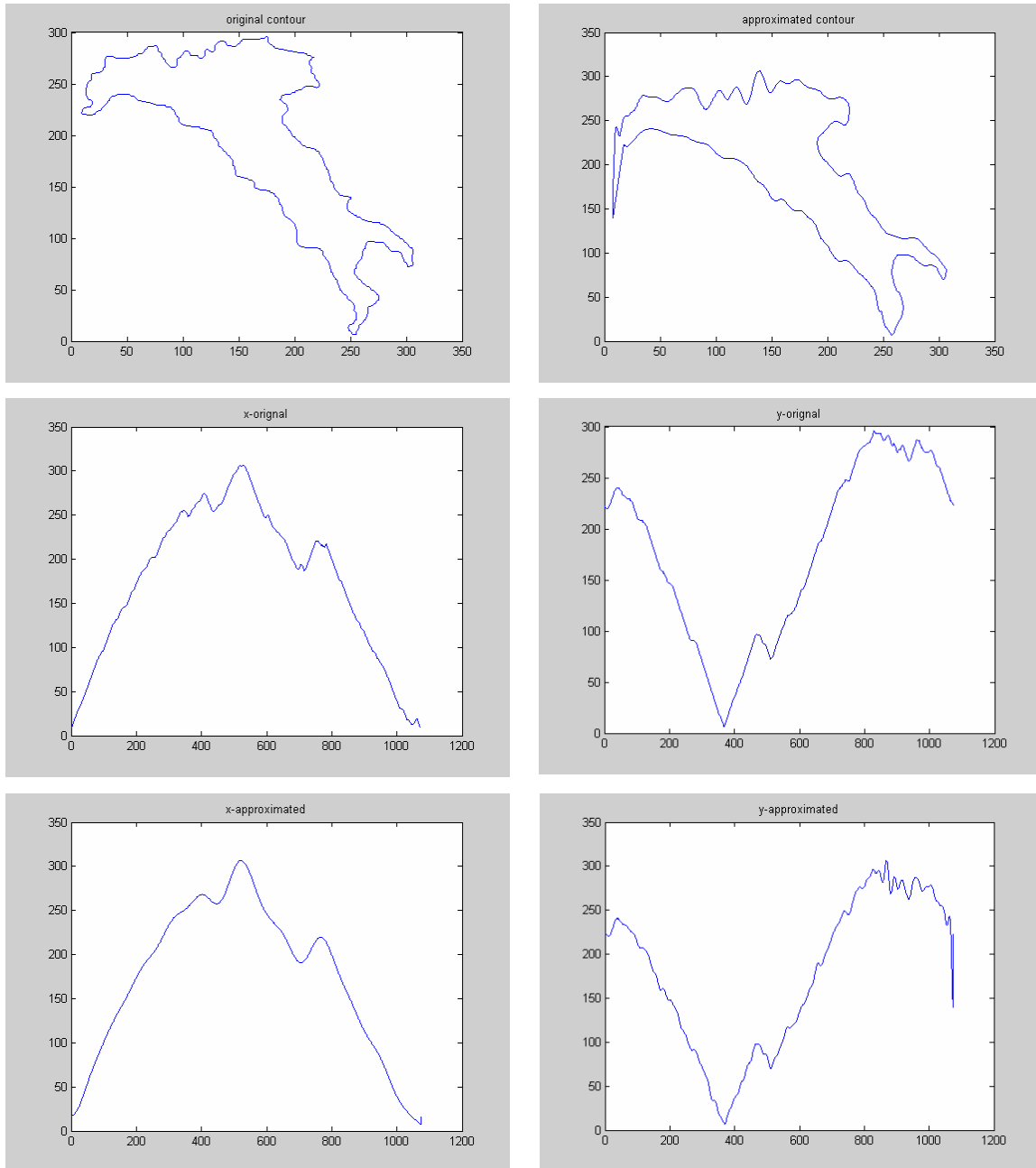


Fig. 4.7 Compressed contour using DCT transform

4.5 Haar transform

The Haar transform is the simplest type of the wavelet transform which uses rectangular basis functions.

The basis functions can be generated from the relations

$$H(0,0,x) = \frac{1}{\sqrt{N}}$$

$$H(g,z,x) = \begin{cases} \frac{2^{\frac{g}{2}}}{\sqrt{N}} & \text{for } \frac{z-1}{2^g} \leq x < \frac{z+\frac{1}{2}}{2^g} \\ -\frac{2^{\frac{g}{2}}}{\sqrt{N}} & \text{for } \frac{z-\frac{1}{2}}{2^g} \leq x < \frac{z}{2^g} \\ 0 & \text{elsewhere} \end{cases} \quad (4.21)$$

where $0 \leq g < \log_2 N$ and $1 \leq z < 2^g$ over the range $0 \leq x < 1$.

Functions of the Haar transform are orthonormal. The Haar transform matrix with size 8×8 is shown below

$$\frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{bmatrix}$$

As we can see that all elements are 1, -1 and 0 except apart from powers of $\sqrt{2}$. When multiplication of the Haar matrix with a signal is interpreted as a signal sampling in hierarchical way from low to high frequencies; the first sample corresponds to the average while the second corresponds to the mean difference between the first and last four neighbour pixels, and the last four transform samples represent differences of two

neighbouring pixels. Haar transform has a good high- and low-frequency response because most of the transform coefficients depend only on their direct neighbours.

The Haar functions are shown in Fig. 4.8.

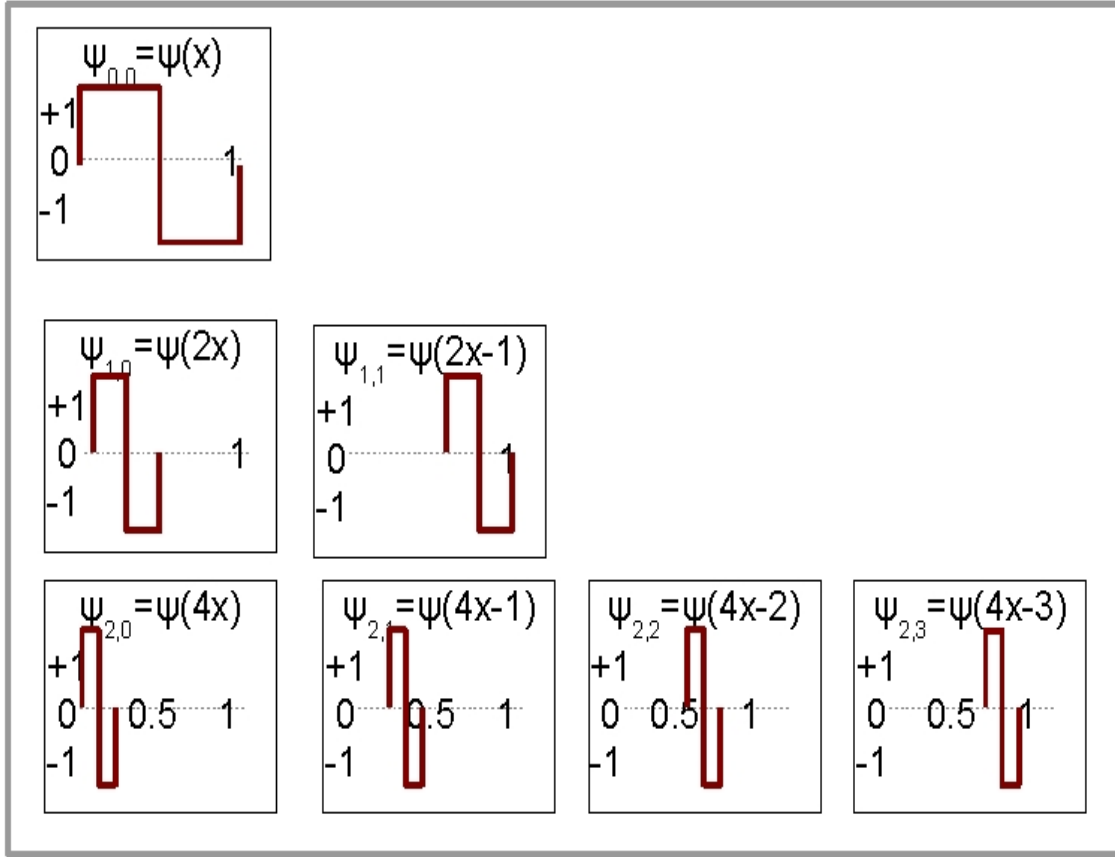


Fig. 4.8 Set of Haar functions

The Haar Transform equations is as follows

$$[S(n)] = \frac{1}{N} [Har(n)] [X(n)] \quad \text{for } n = \log_2 N \quad (4.22)$$

$$[X(n)] = [Har(n)]^{-1} [S(n)] \quad (4.23)$$

where $[S(n)]$ is vector of spectral components and $[Har(n)]$ is Haar matrix.

Fig. 4.9 shows that the Haar transform is performed for each of $X(n)$ and $Y(n)$, then the threshold sampling is done and finally the inverse Haar is applied for each X and Y of Haar to reconstruct the approximated contour (the related results are shown in Table 4.4).

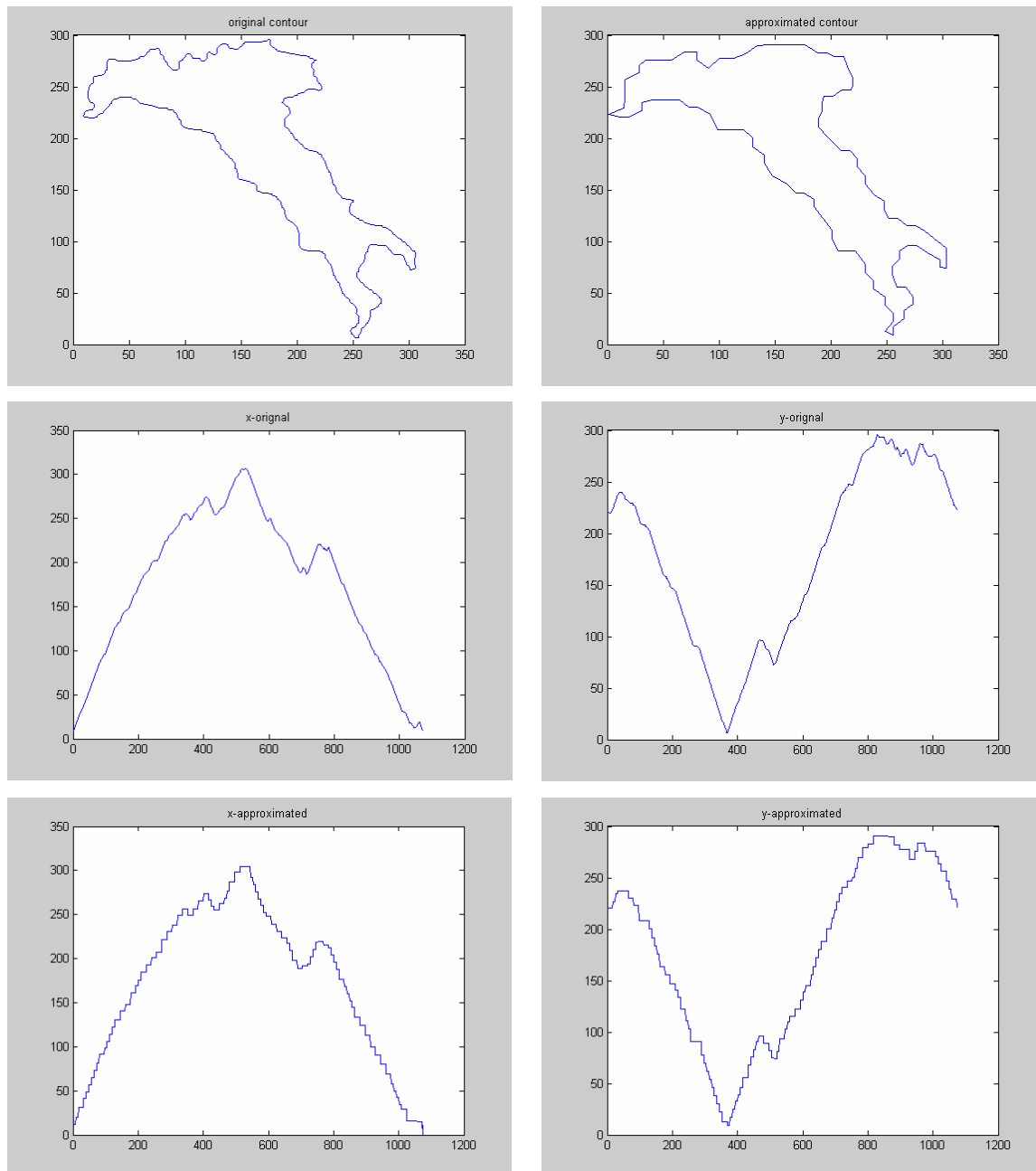


Fig. 4.9 Compressed contour using Haar transform

**Table 4.4 Results of approximation
for the Italy contour using Haar transform**

Threshold	MSE	SNR	CR [%]	NO
0.33	3.32	33.41	91.48	99151

NO – number of operations

4.6 Comparison between the fast transforms and time domain methods

The computational complexity is one of the most important factors in evaluating a transform. High computational complexity leads to high implementation cost.

The MSE and SNR criterions versus compression ratio are also used to evaluate the distortion between the purpose comparisons.

The comparison is done for Italy contour which was extracted by the using of the “SSPCE” (single step parallel contour extraction) [33]. The comparison is made between the following algorithms:

- Ramer method; it will be referred to as the first algorithm.
- Segment distances ratio (or SR) method; it will be referred to as the second algorithm.
- Triangle height square (or Trihs) method; it will be referred to as the third algorithm.
- PHL transform; it will be referred to as the fourth algorithm.
- Walsh transform; it will be referred to as the fifth algorithm.
- DCT transform; it will be referred to as the sixth algorithm.
- Haar transform; it will be referred to as the seventh algorithm.

The Comparison of the compression abilities versus the MSE & SNR are shown in Fig. 4.10 & Fig. 4.11 respectively.

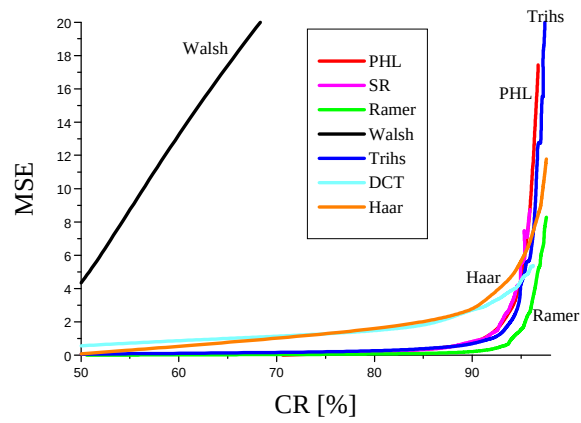


Fig. 4.10 MSE versus CR for the Italy contour

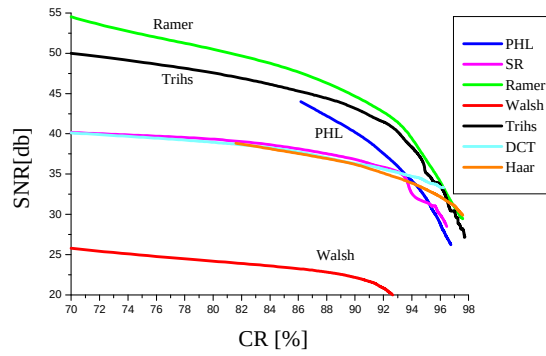


Fig. 4.11 SNR versus CR for the Italy contour

Comparison of the compression abilities versus the number of operations is presented in Fig. 4.12 for the Italy contour.

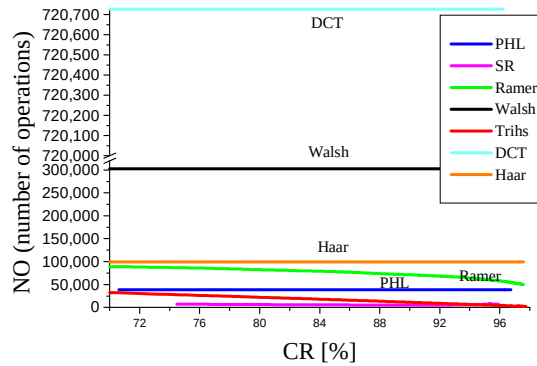


Fig. 4.12 NO versus CR for the Italy contour

From Fig. 4.10 to Fig 4.12; PHL, SR and Trihs algorithms are faster than that of Ramer method. The Walsh transform is faster than DCT (which is the slowest one). The SR and Trihs are the fastest algorithms with high compression ratio. Trihs is very close to the Ramer method for high compression ratio. The reconstruction of PHL and SR are very similar when the CR around 96%. The reconstruction quality by DCT is not so good at the higher compression ratio but is still better than that of Walsh transform.

4.7 Algorithms for contour extraction and image compression in transform domain using High Pass Filter (HPF)

Three different algorithms of contour extraction and image compression using low-pass filter (LPF) and high-pass filter (HPF) proposed by Prof. A. Dziech are investigated and compared with Sobel and Canny detectors in this section. The algorithms use periodic Haar piecewise-linear (PHL) transform, discrete cosine transform (DCT) and Haar Transform. Effectiveness of the contour extraction for different classes of images is evaluated. The main idea of the procedure for both contour extraction and image compression are performed. To compare the results, the mean square error and signal-to-noise ratio criteria were used. The number of operations that is necessary to perform contour extraction and image compression procedures is also presented. The simplicity and the small number of operations are the main advantages of the proposed algorithms.

A high pass filter is a filter that passes high frequencies while low frequencies are prevented. Getting the objective using high pass filter is done by looking for slowly changing areas (i.e. low frequency) of the image and bring out the fast changing details (i.e. high frequency) in the image. This means that if we were to high pass filter the box image we would only see an outline of the box. The edge of the box is the only place where the neighbouring pixels are different from one another. Contour representation and compression are required in many applications.

The main purpose of these algorithms is to obtain the extracted contour and compressed image using a high-pass filter by using PHL [53] or DCT [54] or Haar [56] transforms.

The algorithms show that the analyses are based on the number of operations and are imprecise and probably underdeveloped and we can see their image quality is quite good compared with the other methods.

The results are compared with Sobel and Canny edge detectors for the contour extraction [2, 5, 6, 7 and 8].

4.7.1 Description of the algorithms

The fit criterion of the first algorithm [59] consists of selecting one square block of the spectral images while leaving the other coefficients rejected (i.e. non shadow region) as shown in Fig. 4.13 a.

The fit criterion of the second algorithm [59] is based on selection of the square block (i.e. shadow region) represented by high frequency components of compressed image (e.g. 128x128) as shown in Fig. 4.13 b.

The fit criterion of the third algorithm [59] consists in rejecting one of the square block of the spectral images (e.g. 128x128) and the other coefficients will be taken into account in the contour reconstruction stage (e.g. shadow region) as shown in Fig. 4.13 c.

The forward PHL or DCT or Haar transforms (which were described in the previous sections) are applied to the grey-level image. By using low- and high-pass filters after the zonal procedure the two spectral sub-images are obtained. The threshold is done for the obtained image by HPF to get the extracted contour and then compared with the Sobel and Canny edge detectors. The compressed image and extracted contour are obtained by using the inverse transform for each of the two sub-images respectively. These two sub-images are combined together to reconstruct the original grey-level image.

A flowchart of the all analyzed algorithms is depicted in Fig.4.14.

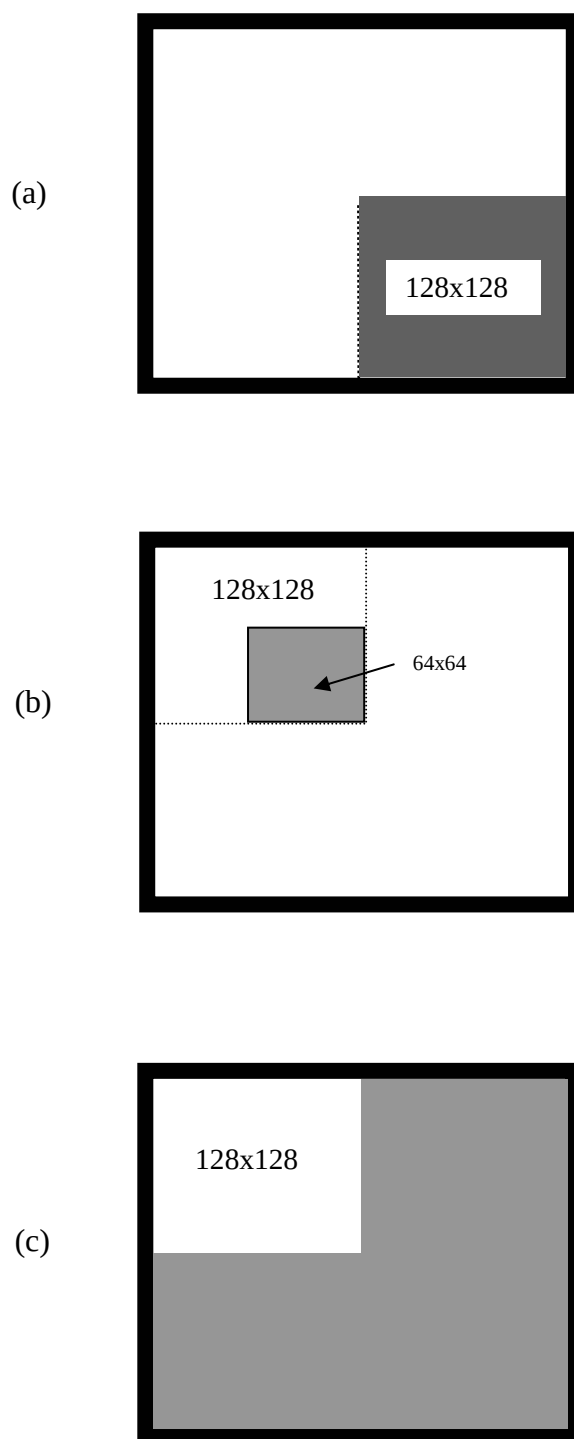


Fig. 4.13 HPF zonal method for the spectral image (256x256) using
(a) The first analyzed algorithm,
(b) The second analyzed algorithm, and
(c) The third analyzed algorithm

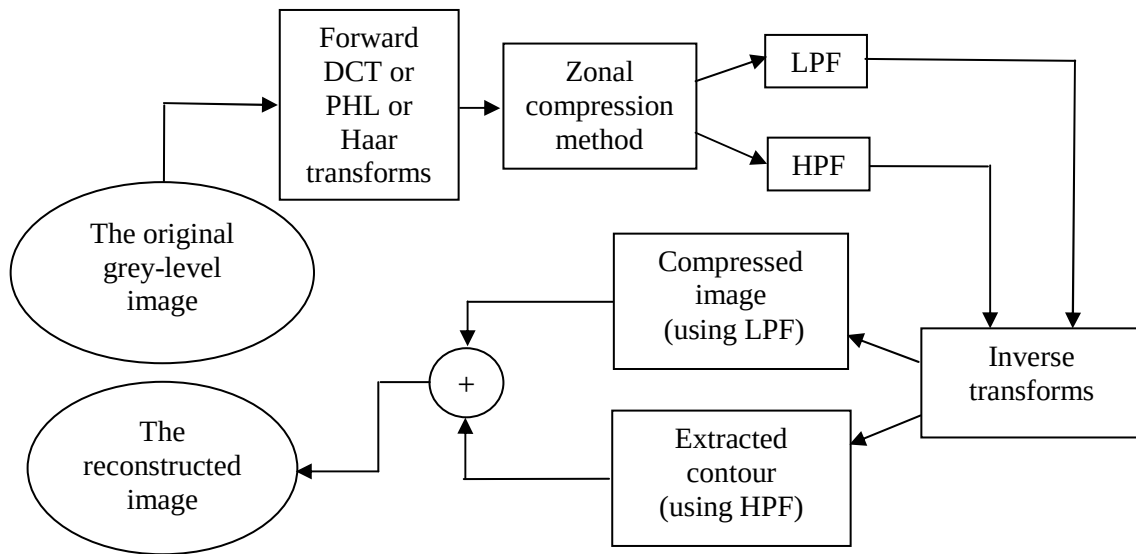


Fig 4.14 Block diagram of the analyzed algorithms

4.7.2 Experimental results

To visualise the experimental results a set of four test grey-levels images were selected. Selected images are shown in Fig. 4.15 (related results are shown from Fig. 4.16 to Fig. 4.21).

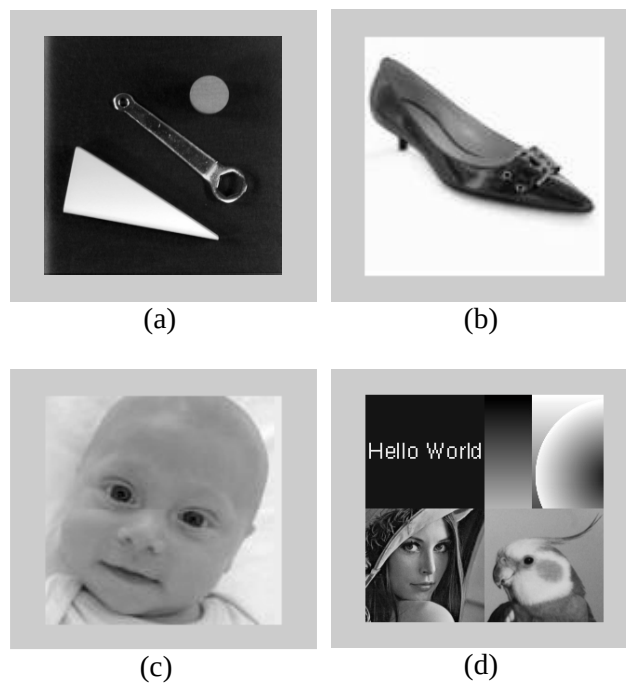


Fig. 4.15 Test images: a) Tools (256x256), (b) Shoe (128x128), (c) Baby (128x128), and (d) Montage (256x256)

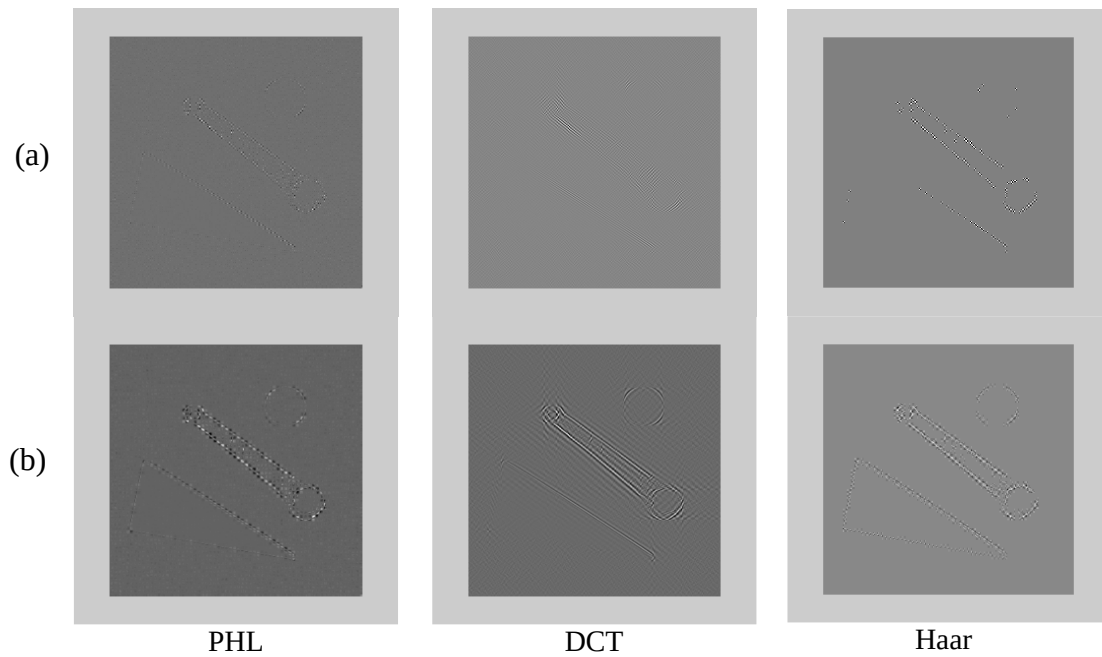


Fig. 4.16 Reconstruction of Tools image using I algorithm
(a) Using the lower right block (128x128), and
(b) Using the lower right block (192x192)

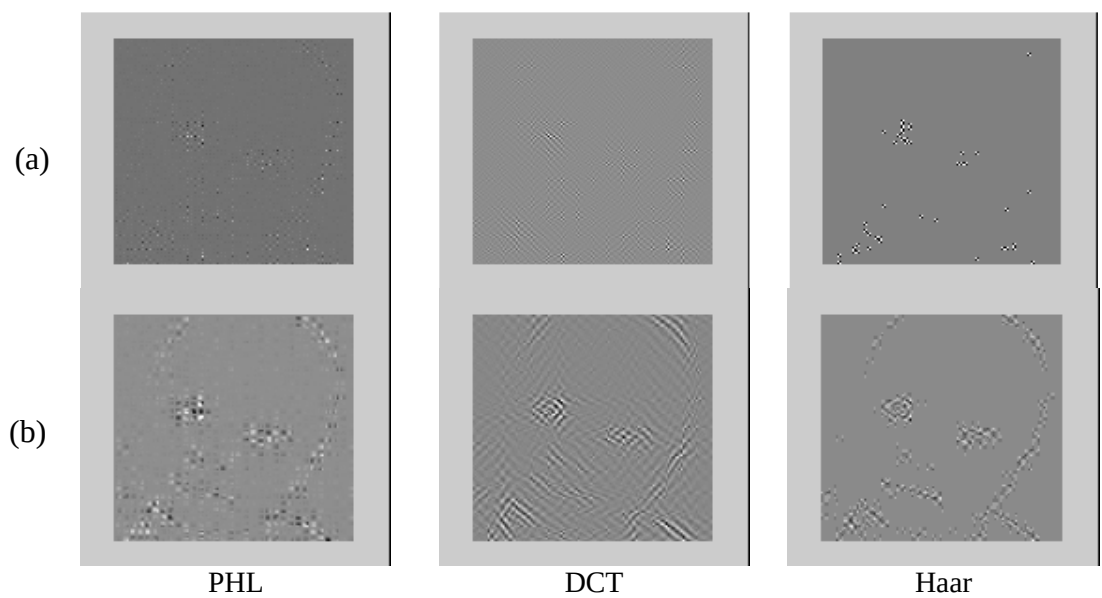


Fig. 4.17 Reconstruction of Baby image using I algorithm
(a) Using the lower right block (64x64), and
(b) Using the lower right block (96x96)

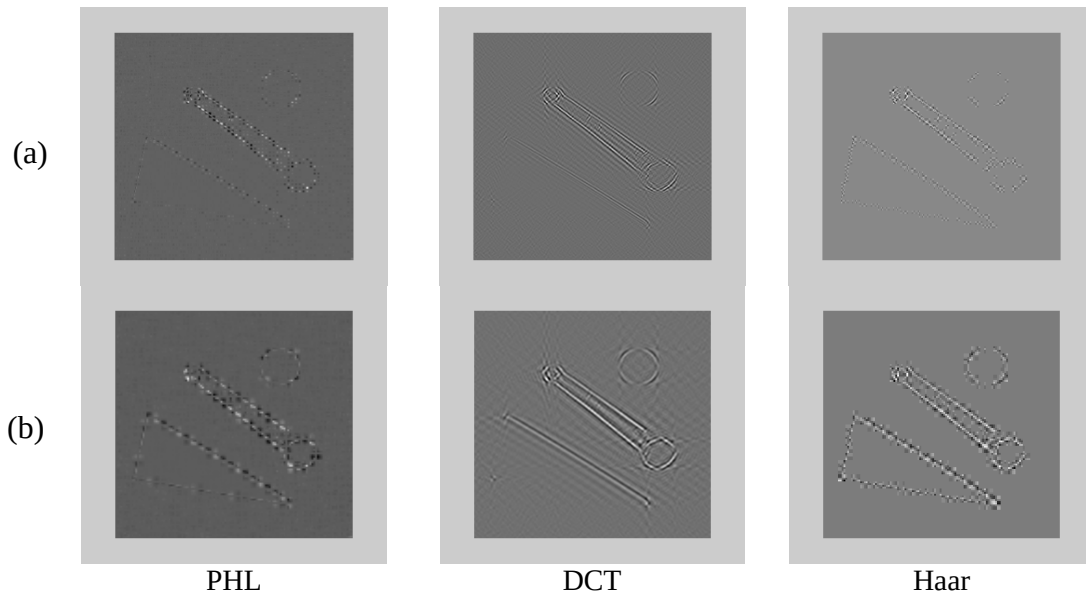


Fig. 4.18 Reconstruction of Tools image using II algorithm
 (a) 128x128 block represented by the upper left block (192x192), and
 (b) 96x96 block represented by the upper left block (128x128)

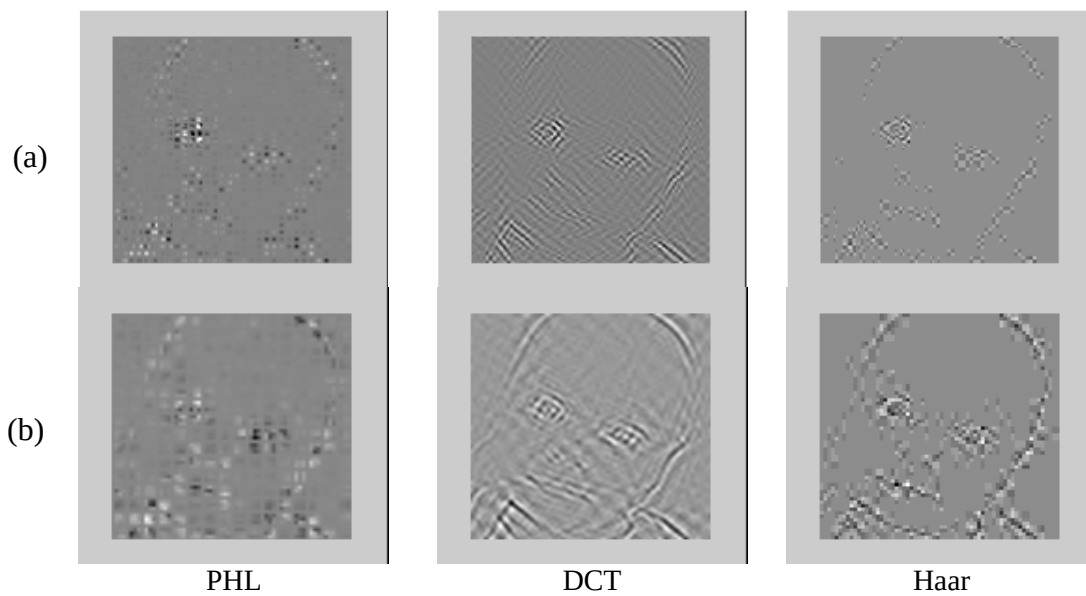


Fig. 4.19 Reconstruction of Baby image using II algorithm
 (a) 64x64 block represented by the upper left block (96x96), and
 (b) 48x48 block represented by the upper left block (64x64)

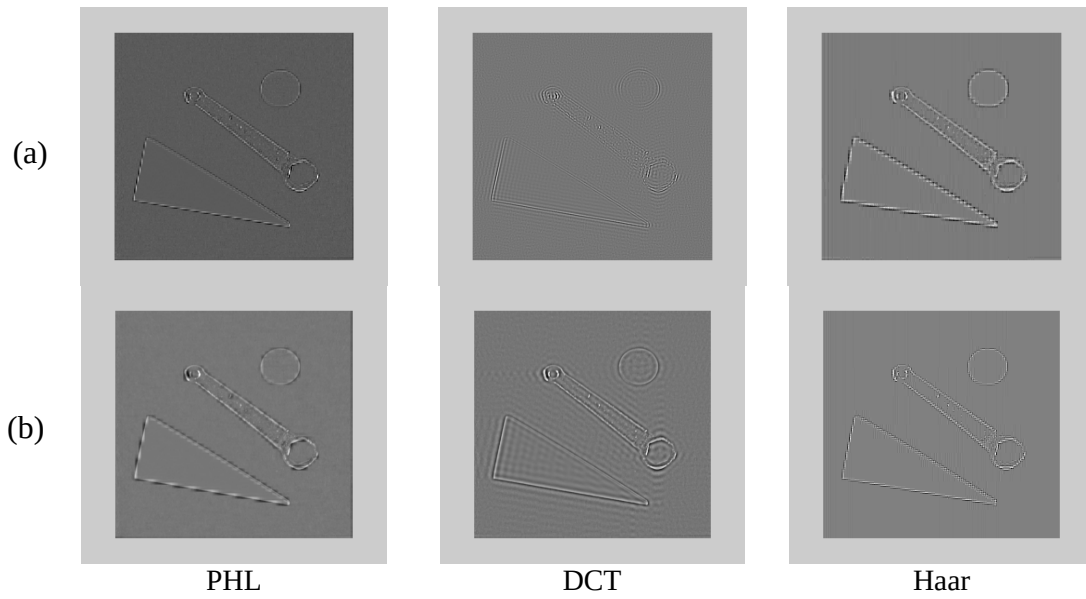


Fig. 4.20 Reconstruction of Tools image using III algorithm
(a) By rejecting the upper left block (128x128), and
(b) By rejecting the upper left block (64x64)

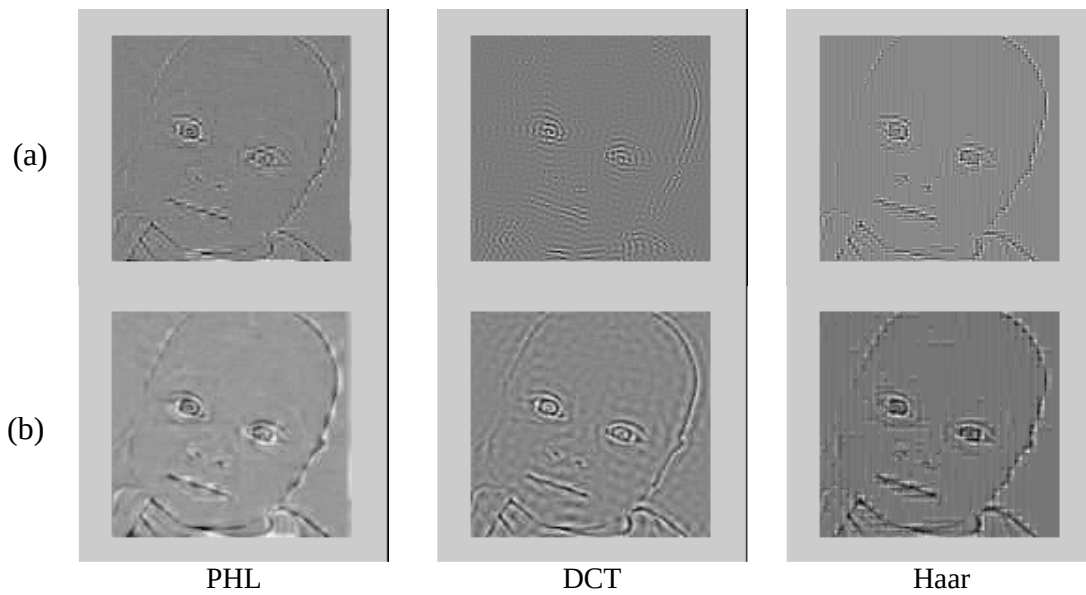


Fig. 4.21 Reconstruction of Baby image using III algorithm
(a) By rejecting the upper left block (64x64), and
(b) By rejecting the upper left block (32x32)

The results presented in Fig. 4.16 to Fig. 4.21 show that the third analyzed algorithm has the best extraction property; and it is used in the next experiments.

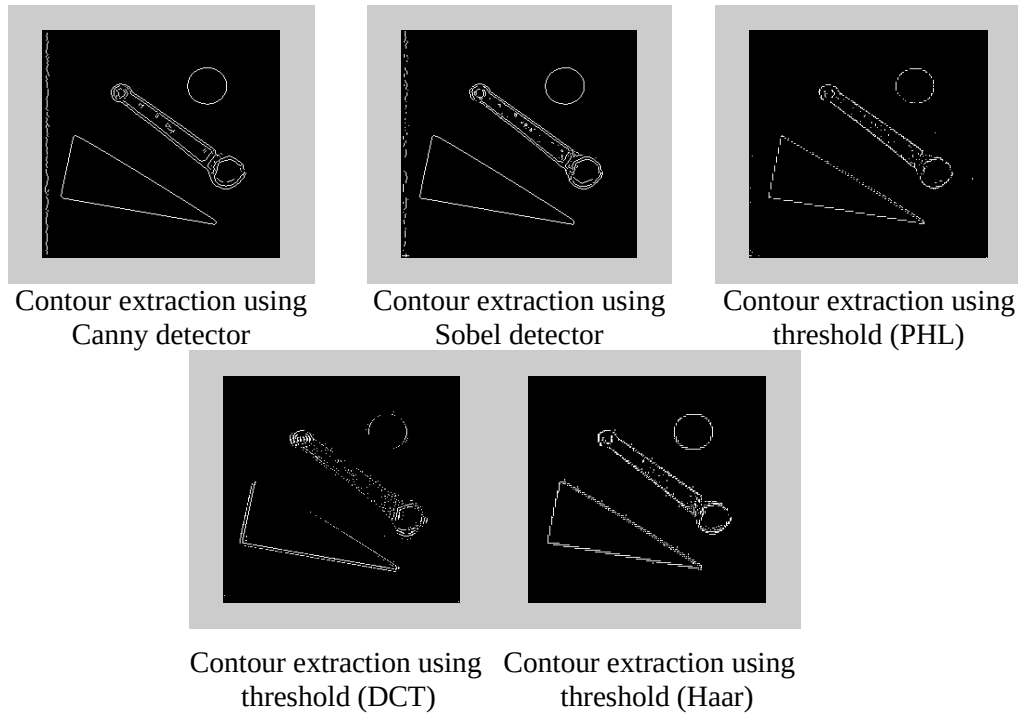


Fig. 4.22 Contour extraction using III algorithm for the Tools image

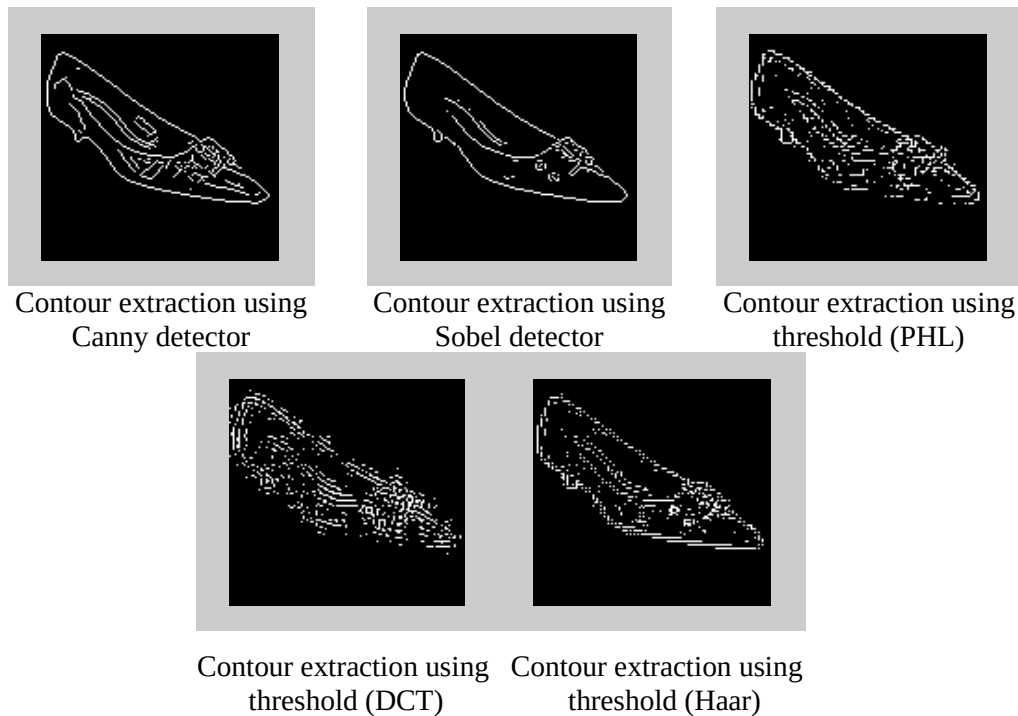


Fig. 4.23 Contour extraction using III algorithm for the Shoe image

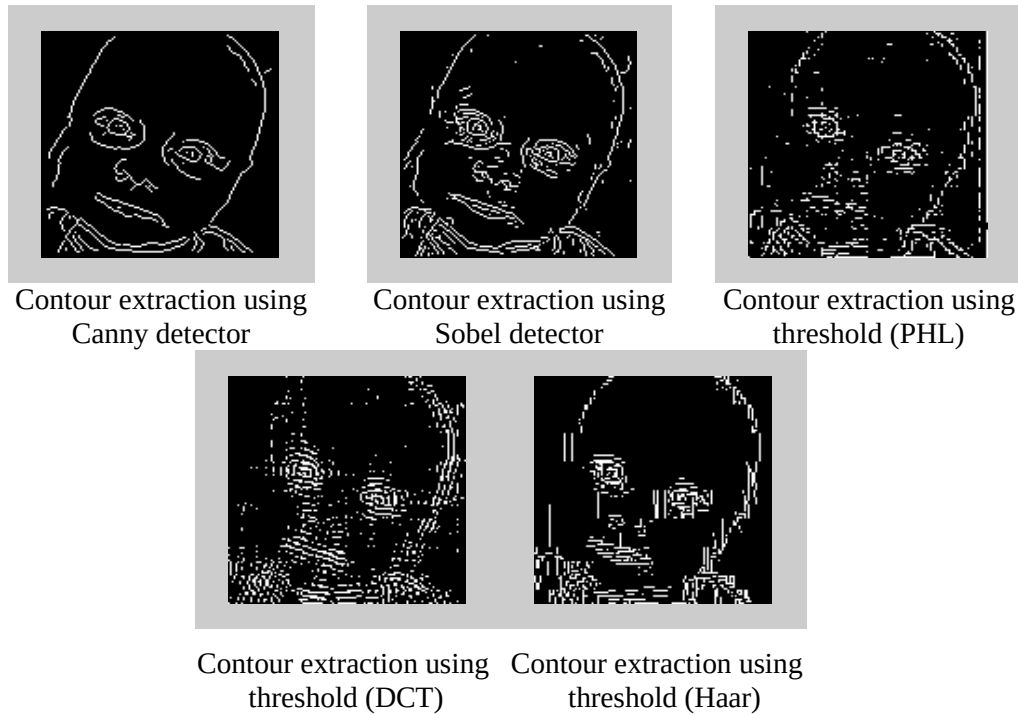


Fig. 4.24 Contour extraction using III algorithm for the Baby image

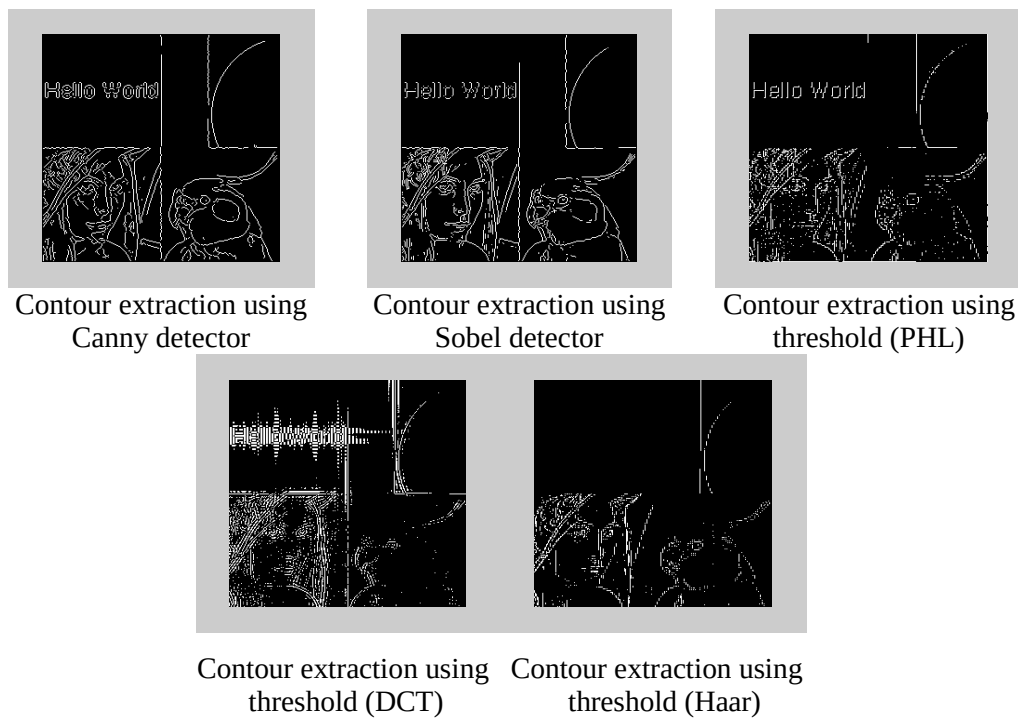


Fig. 4.25 Contour extraction using III algorithm for the Montage image

The results presented from Fig. 4.22 to Fig. 4.25 show that the contour detection quality using Sobel has the best quality followed by Canny, PHL, Haar, and DCT respectively.

The proposed image compression and contour extraction method is related to the data compression and extraction problems. To evaluate its compression ability, the following compression ratio was introduced

$$CR = \frac{NOZ}{(n \cdot m)} \cdot 100\% \quad (4.24)$$

where NOZ is number of zero coefficients in the spectral domain and $(n \cdot m)$ is the image size.

The mean square error (MSE) and peak signal-to-noise ratio ($PSNR$) criteria were used to evaluate the distortion introduced during the image compression and contour extraction procedures. The MSE criterion is defined by the following equation

$$MSE(I, \tilde{I}) = \frac{1}{(n \cdot m)} \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} (I(i, j) - \tilde{I}(i, j))^2 \quad (4.25)$$

where I and $\tilde{I}$ are the grey-level and reconstructed images respectively.

The $PSNR$ is defined by the following formula

$$PSNR(I, \tilde{I}) = 10 \cdot \log_{10} \frac{(L-1)^2}{MSE(I, \tilde{I})} \quad (4.26)$$

where L is the grey-level number.

Fig. 4.26 shows the illustration of the main idea by using the third analyzed algorithm for the Montage image to obtain both the extracted contours and compressed image (related results are shown in Table 4.5).

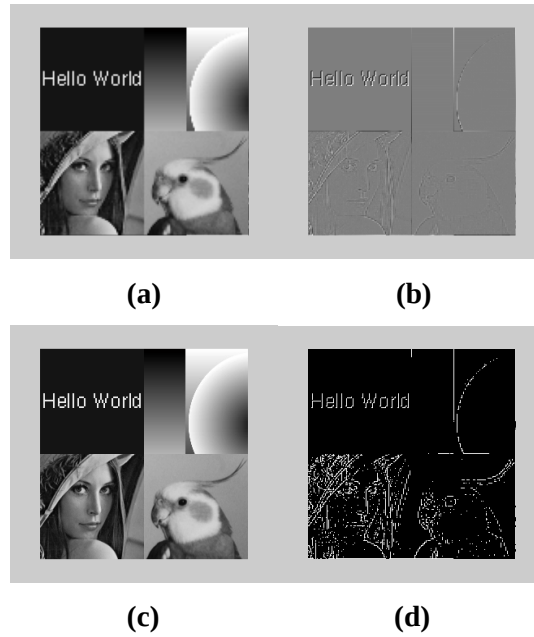


Fig. 4.26 Using III algorithm and PHL transform (a) Image compression using LPF, (b) Image reconstruction using HPF, (c) Image reconstruction using LPF & HPF filters, and (d) Contour extraction from (b)

Table 4.5 Results of reconstruction for Montage image

	<i>MSE</i>	<i>PSNR [db]</i>	<i>CR [%]</i>
(a)	207.46	24.96	82.37
(c)	3.60e-027	312.62	not use

The results of comparison between the used transforms and Sobel and Canny edge detectors using the third proposed algorithm with respect to the number of operations are shown in Table 4.6.

Table 4.6 Comparison using the third algorithm

THE METHOD	Tools	Shoe	Baby	Montage
Canny	6193039	1587695	1489574	6456175
Sobel	1972618	494164	478160	1972618
PHL	1357836	393740	393740	1357836
DCT	11531840	2672194	2672194	11531840
Haar	2380300	557836	557836	2380300

Table 4.6 shows that the PHL requires the smallest number of operations in comparison to the other methods. Therefore the PHL extraction method is the fastest.

The compression ratio obtained by this method is greater than 82% using LPF without significant visible distortion and also has good contour detection using HPF as shown in Fig. 4.26. The high reconstruction quality of the grey-level image can be obtained by combining these two sub-images (e.g. 312.62 db as shown in Fig. 4.26 c). An important advantage of the analyzed method is the simplicity of implementation both in terms of memory requirement and fit criterion complexity.

4.8 Algorithm for combining two images using contour extraction in transform domain by High Pass Filter (HPF)

This algorithm shows the application of contour extraction method using HPF and compares it with other methods (edge detectors). The algorithm used periodic Haar piecewise-linear PHL transform, discrete cosine transform (DCT) and Haar transform. Effectiveness of the contour extraction for different classes of images is evaluated. In the zonal procedure, the third analyzed algorithm which was introduced in the previous section is used in the analyzed algorithm (see Fig. 4.13 c).

The main idea of the analyzed procedure for contour extraction is utilized. To compare the results, the mean square error and signal-to-noise ratio criteria were used. The simplicity and good combining quality are the main advantages of the analyzed algorithms. The analyzed algorithm is very useful in many applications (ex. image compression).

The main purpose of the algorithm is to combine the original grey-level images by combining their extracted contours which have been obtained from the sub-images using high-pass filters and by using one of the different transforms (PHL or DCT or Haar).

4.8.1 Description of the algorithm

Fit criterion of the analyzed algorithm consists of two grey-level images. The size of the original images is squared and should fulfil the 2^n where n is an integer number. The common points are communed between the two sub-images.

The forward PHL [53], or DCT [53], or Haar [56] transforms (described in the previous sections) are applied to the each one of sub-images. By using high-pass filters after the zonal procedure the two spectral sub-images are obtained. The inverse transform

is applied to each of the two spectral sub-images, and then the threshold is done for each of them to obtain the extracted contours. Then the correlation function is done using both extracted contours to found the common edge between them. The maximum correlation function indicates the common points of two images. The additional procedure is necessary to get precise common points between the first lines. Hence, the common edge points (either in X or Y axes) for both sub-images is determined and the original grey-level image can be easily combined.

A flowchart of the all analyzed algorithm is depicted in Fig.4.27.

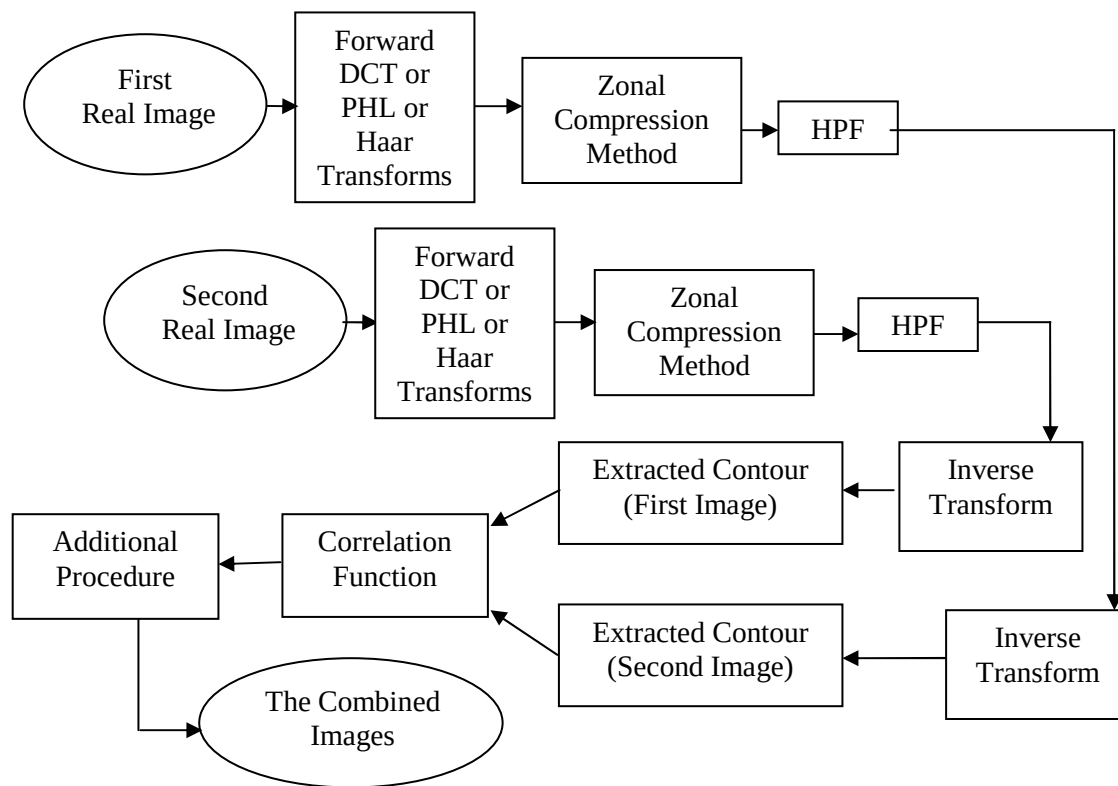


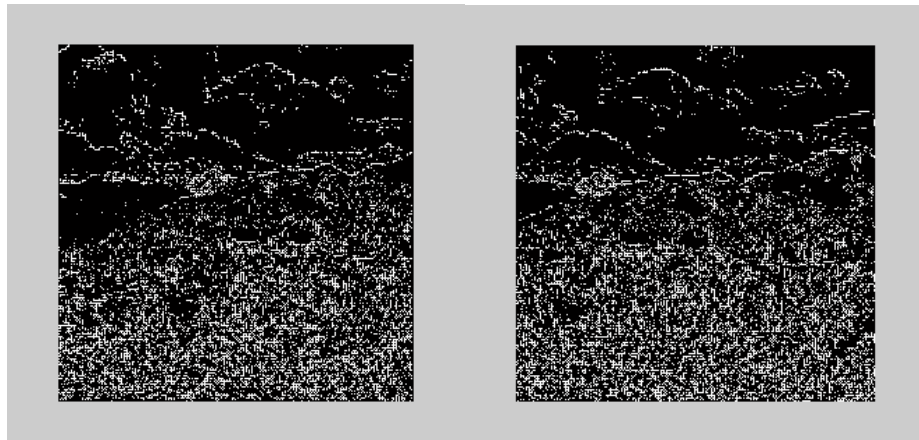
Fig 4.27 Block diagram of the analyzed algorithm

4.8.2 Experimental results

To visualise the experimental results a set of six test grey-levels images were selected. The related results are shown from Fig.4.28 to Fig. 4.32.



(a)



(b)

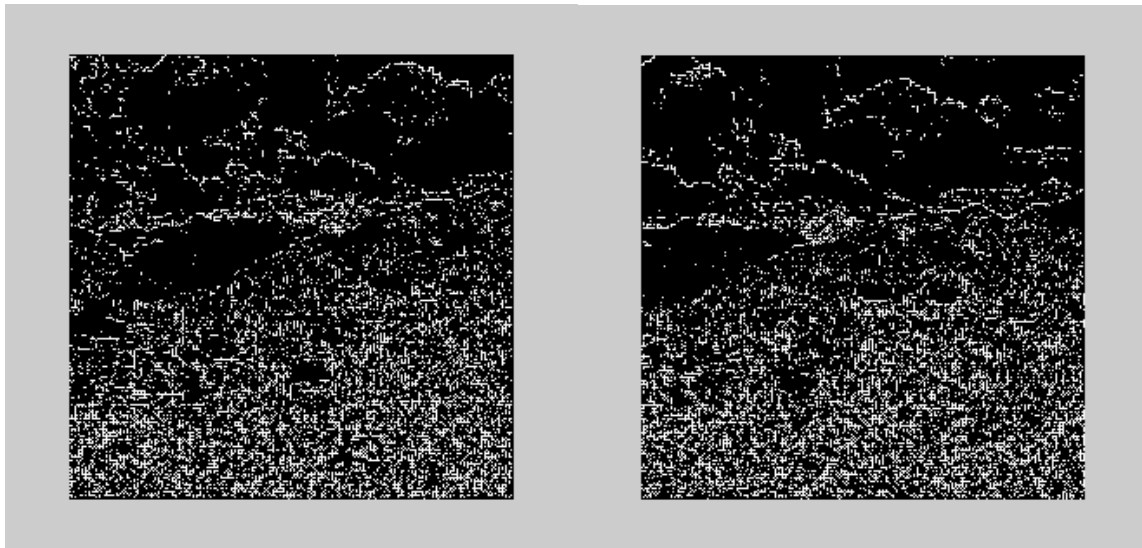


(c)

Fig. 4.28 (a) Real test images (256x256), (b) Extracted contours from both images, and (c) Image combination before and after the additional procedure



(a)



(b)

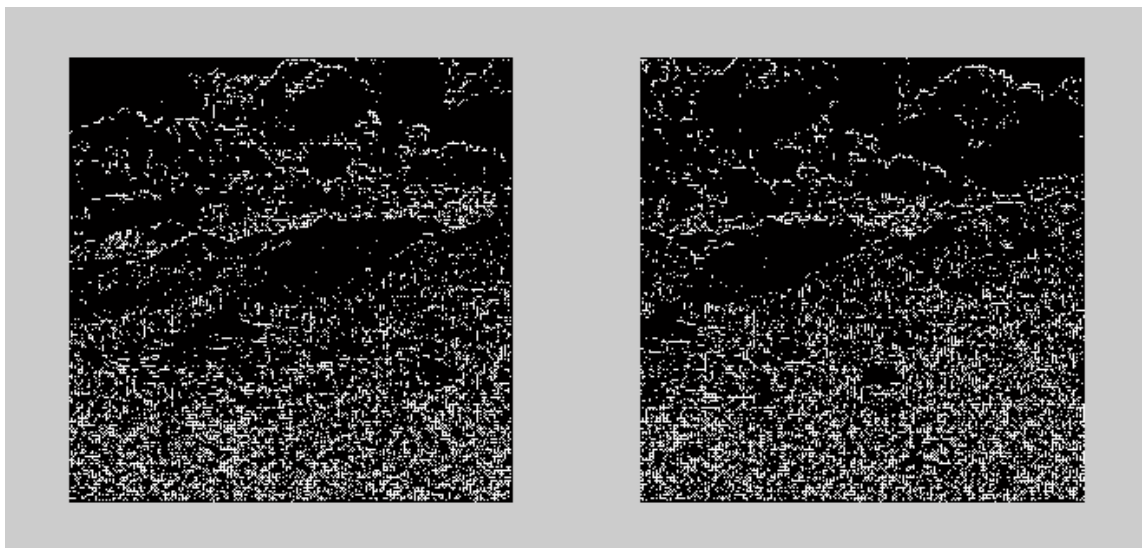


(c)

Fig. 4.29 (a) Real test images (256x256), (b) Extracted contours from both images, and (c) Image combination before and after the additional procedure



(a)



(b)

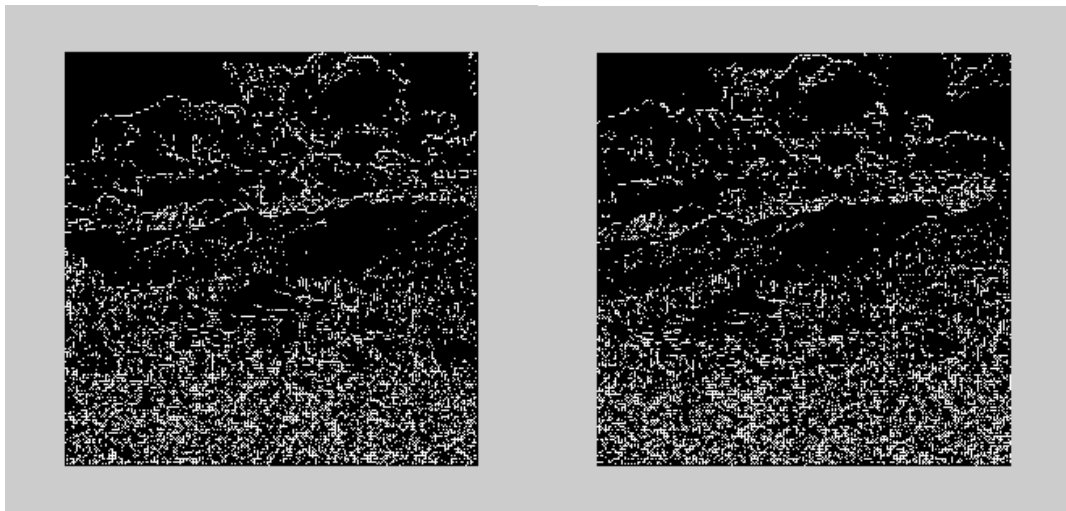


(c)

Fig. 4.30 (a) Real test images (256x256), (b) Extracted contours from both images, and (c) Image combination before and after the additional procedure



(a)



(b)

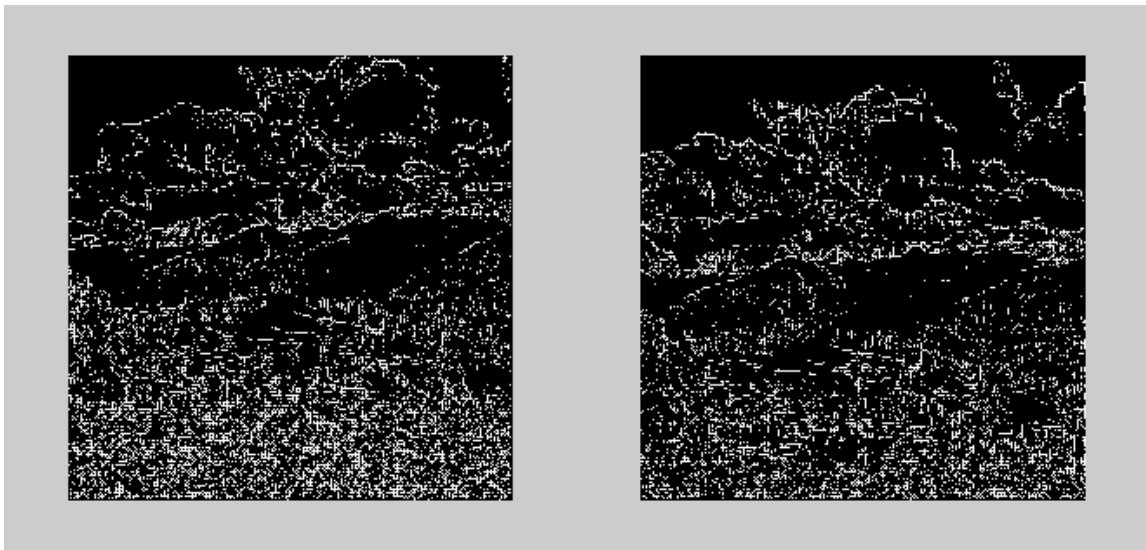


(c)

Fig. 4.31 (a) Real test images (256x256), (b) Extracted contours from both images, and (c) Image combination before and after the additional procedure



(a)



(b)



(c)

Fig. 4.32 (a) Real test images (256x256), (b) Extracted contours from both images, and (c) Image combination before and after the additional procedure

The results using the proposed algorithm for the all test images is shown in Fig. 4.33.



Fig. 4.33 Results using the analyzed algorithm

As a conclusion we can say the presented results show that the common data between two images can be detected using contour extraction (high pass filter) and correlation function.

4.9 Algorithms for reconstructing the image using contour extraction in transform domain by High Pass Filter (HPF) and bit-planes decomposition methods

Two different algorithms of contour extraction using low and high-pass filters and bit-planes decomposition respectively are presented and compared in this section. The first algorithm used periodic Haar piecewise-linear (PHL) transform. Effectiveness of the contour extraction for different classes of images is evaluated. In this section the main idea of the analyzed procedures for contour extraction is evaluated. For comparison purpose we have most significant bit (MSB) as a digital image (the second algorithm). To compare the results, the mean square error and signal-to-noise ratio criterions were used. The simplicity is the main advantage of the analyzed algorithms.

4.9.1 Description of HPF algorithm

The algorithm has used HPF zonal sampling (the third algorithm) [59] which consist of rejecting one of the squared blocks of the spectral images and the other coefficients to be taken into account in the contour reconstruction. The forward and inverse 2-dimensional periodic Haar piecewise-linear (PHL) [53] transform is applied to obtain the extracted contours. The forward 1-dimensional PHL transform and later threshold is done for the extracted contours. Finally, the reconstructed image is obtained using the inverse 1-dimensional PHL transform.

A flowchart of the first analyzed algorithm is depicted in Fig. 4.34.

4.9.2 Description of bit-planes decomposition algorithm

The analyzed algorithm is applied to decompose the original image into bit-planes whose number depends on the distribution of the grey-levels it may contain. For 256 grey-levels images, eight bit-planes are required. The contours are extracted using single

step parallel contour extraction (SSPCE) method [33]. The contour reconstruction is done using forward and inverse PHL transforms respectively for the most significant bit.

A flowchart of the second analyzed algorithm is depicted in Fig. 4.35.

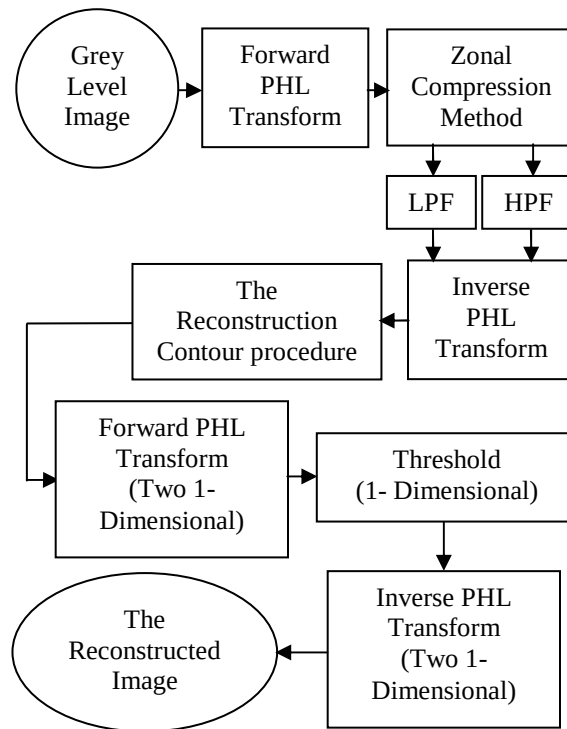


Fig. 4.34 Block diagram of the HPF algorithm

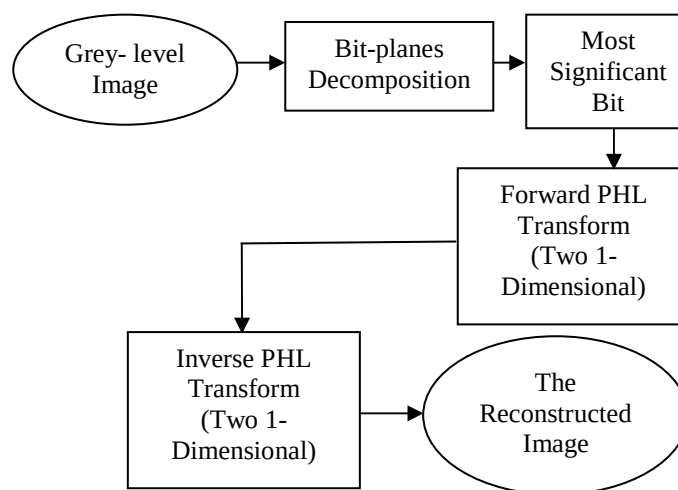


Fig. 4.35 Block diagram of the bit-planes decomposition algorithm

4.9.3 Applied measures

The contour extraction method is related to the data compression and extraction problem. To evaluate its compression ability, the following compression ratio was introduced

$$CR = \frac{(B_{ori} - B_{com})}{B_{ori}} \cdot 100\% \quad (4.27)$$

where B_{ori} is total number of bits required for the original image and B_{com} is total number of bits required for the compressed image.

The mean square error (*MSE*) and peak signal-to-noise ratio (*PSNR*) criterions were used to evaluate the distortion introduced during the contour compression procedure. The *MSE* and *PSNR* can be calculated using Equation (4.25) and Equation (4.26) respectively.

4.9.4 Experimental results

To visualise the experimental results a set of two test grey-levels images were selected. Selected images are shown in Fig. 4.36.

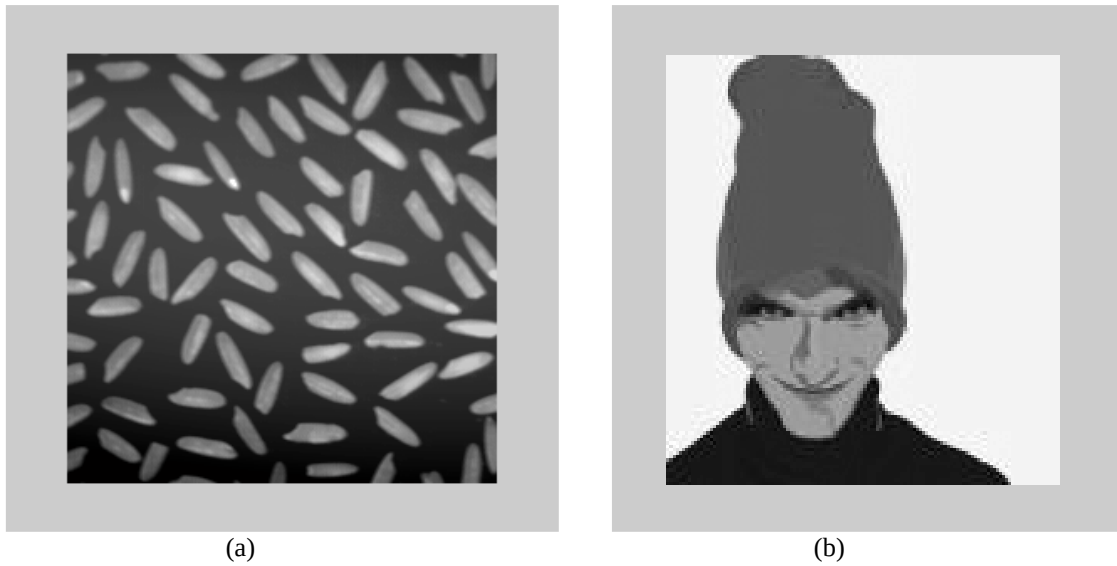


Fig. 4.36 Test images: (a) Rice (256x256), and (b) Man face (128x128)

The related results are shown from Fig.4.37 to Fig. 4.42.

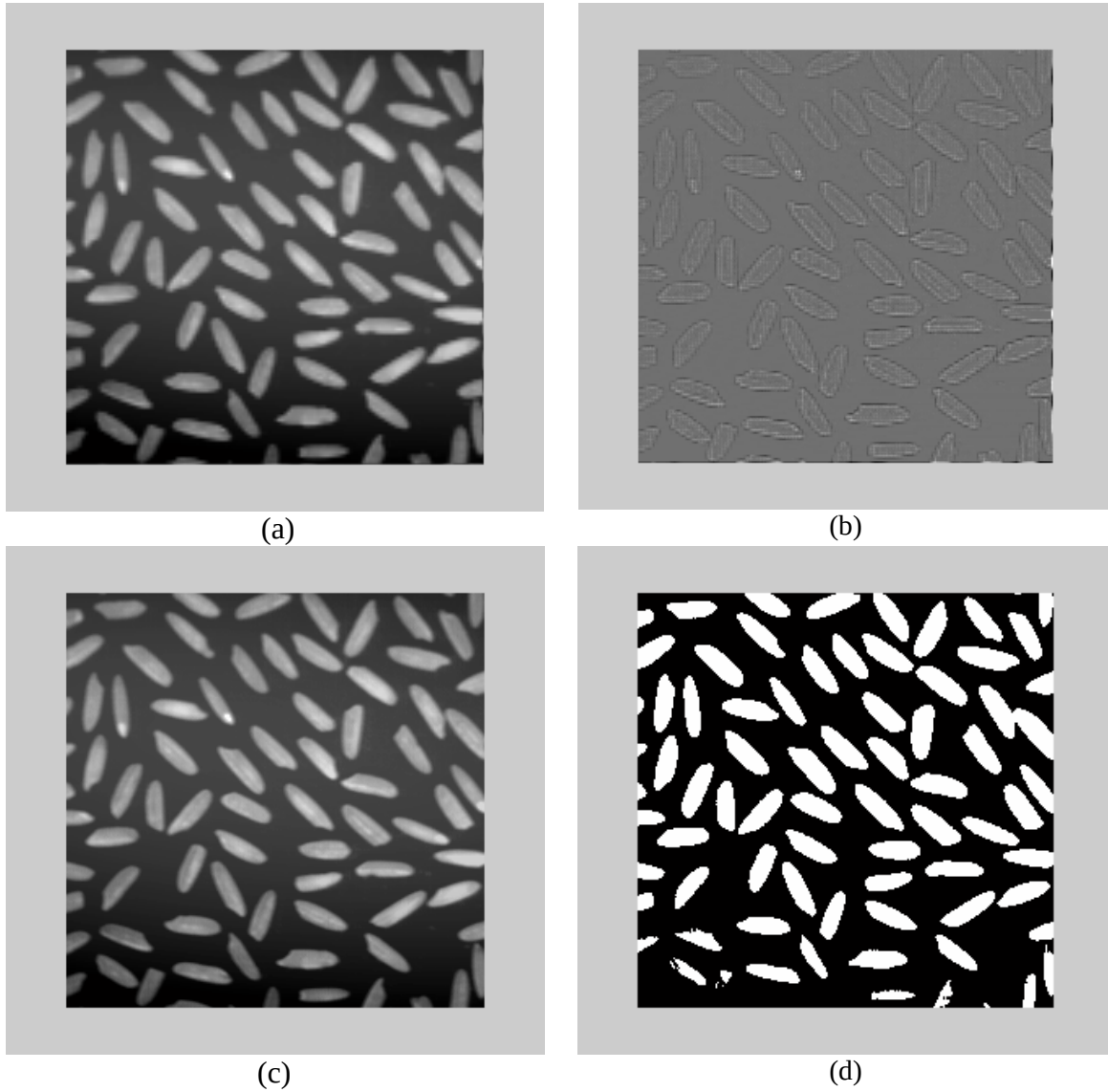


Fig. 4.37 Reconstruction for Rice image using the HPF algorithm by (a) LPF, (b) HPF, (c) LPF and HPF, and (d) Binary image using the threshold for (c)

Table 4.7 Results for the Rice image of Fig. 4.38

The Method	MSE	SNR [db]	CR [%]	NO
HPF	372.21	22.42	67.84	16473243
BP	380.93	22.32	67.95	15864941

HPF is high pass filter, BP is bit planes, and NO is number of operations

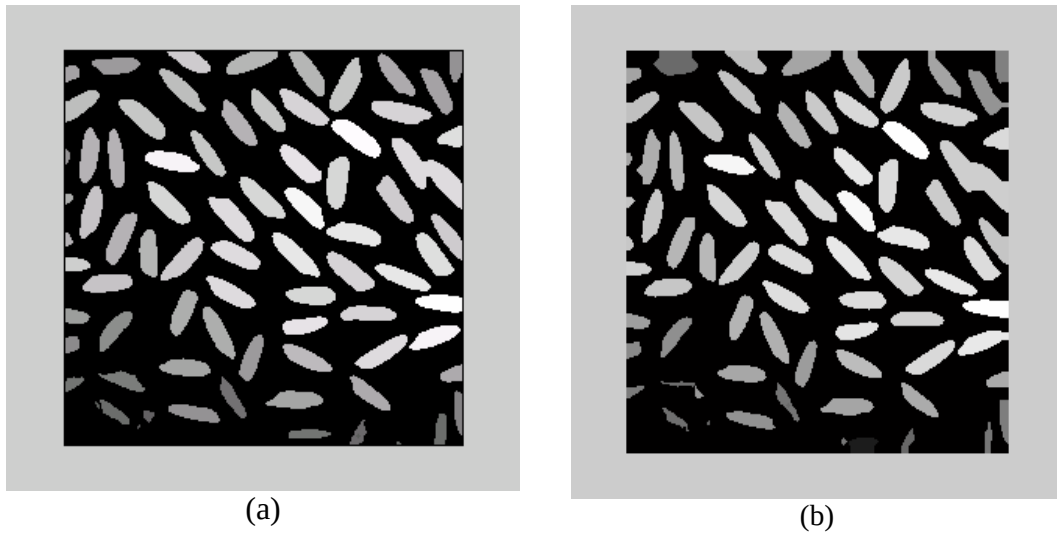


Fig. 4.38 Image reconstruction for Rice image using the (a) HPF algorithm, and (b) Bit-planes algorithm (number eight)

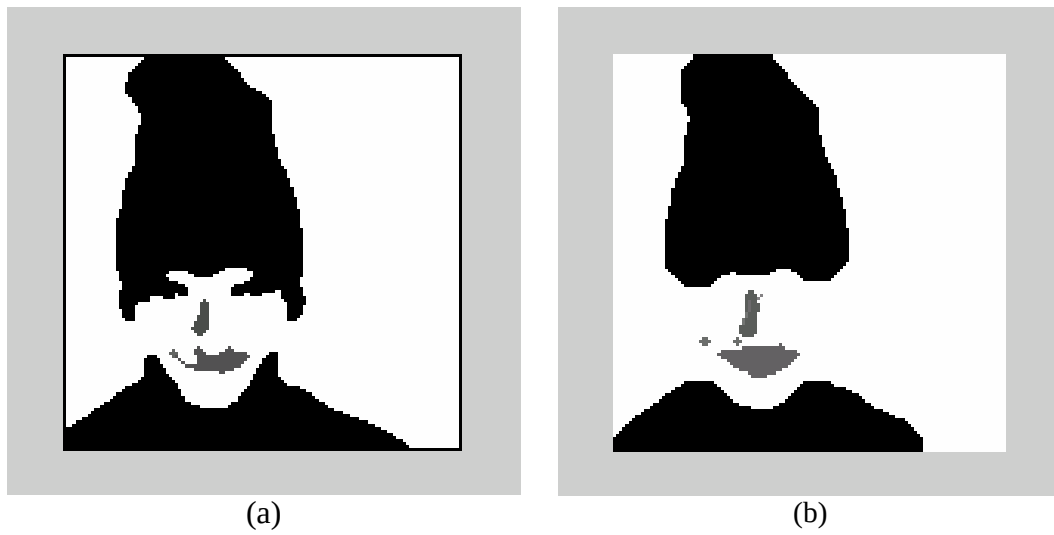


Fig. 4.39 Reconstruction of Man face image using the (a) HPF algorithm, and (b) Bit-planes algorithm (number eight)

Table 4.8 Results for the Man face image

<i>The Method</i>	<i>MSE</i>	<i>SNR [db]</i>	<i>CR [%]</i>	<i>NO</i>
HPF	1.50e+03	16.49	76.42	1315941
BP	1.60e+03	16.17	76.07	476103

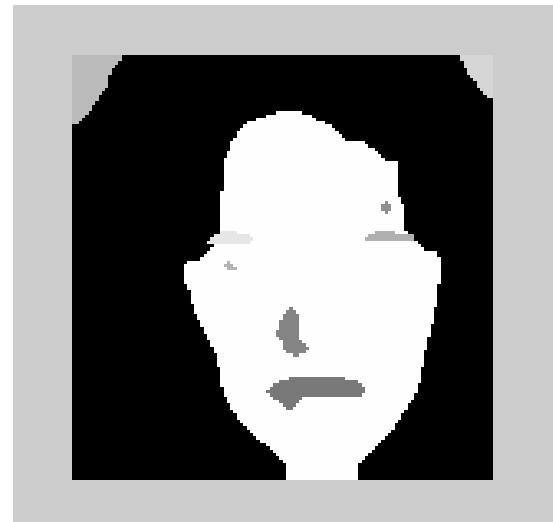
HPF is high pass filter, BP is bit planes, and NO is number of operations



(a) Women face image



(b)



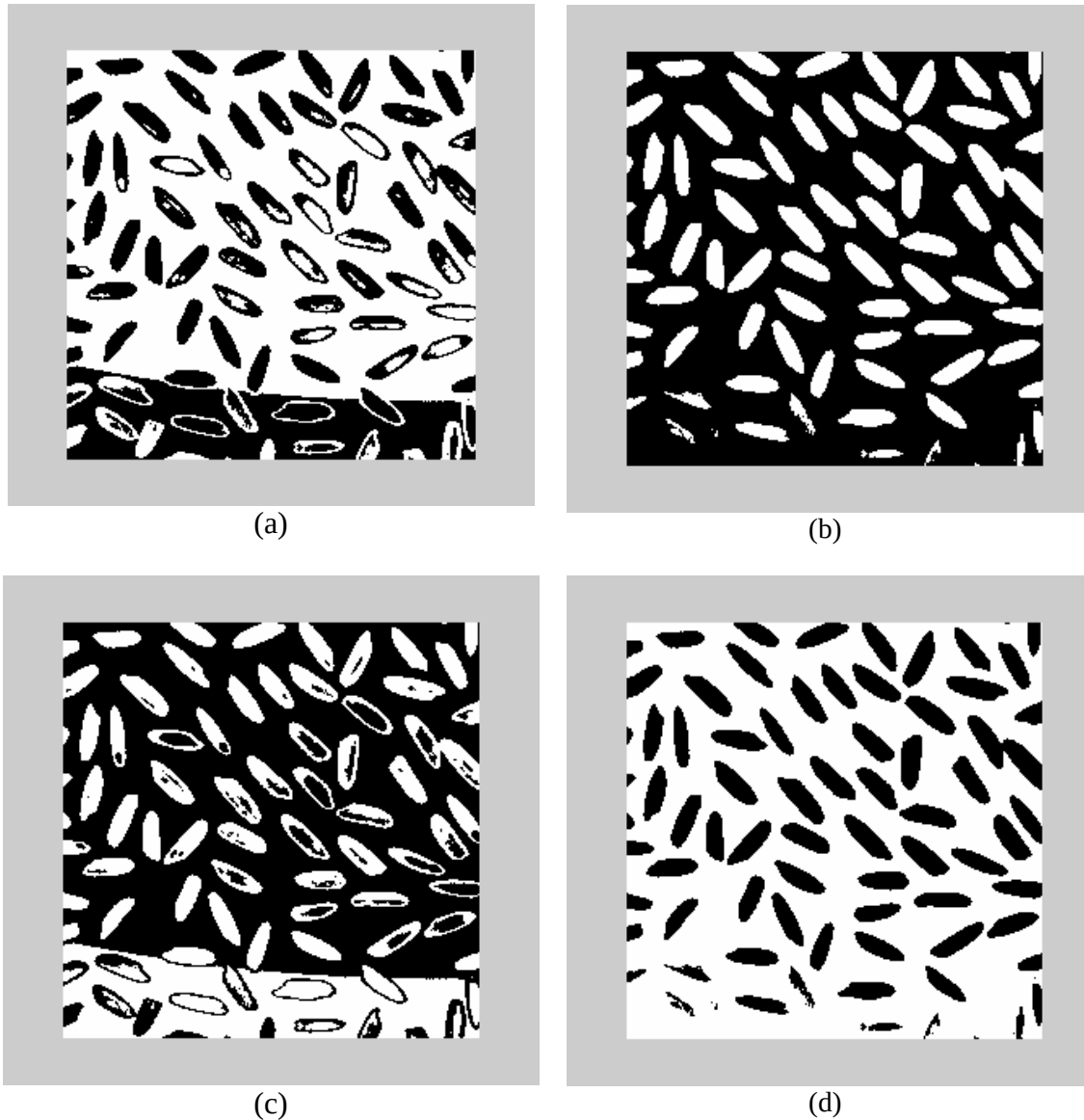
(c)

Fig. 4.40 Reconstruction of Women face image using the (b) HPF algorithm, and (c) Bit-planes algorithm (number eight)

Table 4.9 Results for the Women face image

<i>The Method</i>	<i>MSE</i>	<i>SNR [db]</i>	<i>CR [%]</i>	<i>NO</i>
HPF (b)	686.27	19.77	69.07	782732
BP (c)	1.17e+03	17.45	66.88	475830

HPF is high pass filter, BP is bit planes, and NO is number of operations



**Fig. 4.41 Test image of Rice image using the bit-planes algorithm
 (a) Bit-plane number seven, (b) The most significant bit
 (number eight), and (c and d) The complement
 for (a and b) respectively**

As a conclusion we can say the presented results show that the image can be reconstructed by extracted contours using double transforms (the HPF algorithm) and using only most significant bit-planes (the bit-planes algorithm). Results show however that SNR is not too high (i.e. some significant loses of approximation quality); but the reconstruction image using HPF algorithm is a little bit acceptable more than using the bit-plane algorithm.

The outer contour using bit-plane method is quite good but the inner ones have bad visuals. Low quality with respect to the inner contours gives this algorithm much faster than that of HPF algorithm.



Fig. 4.42 Test image of Man face image using the bit-planes algorithm
(a) Bit-plane number seven, (b) The most significant bit (number eight), and (c and d) The complement for (a and b) respectively

CONCLUSIONS

The main results that have been obtained in this dissertation and the conclusions drawn from these results are presented below:

1. Detailed analysis of the methods for contour data extraction has been given including the comparison between them using the number of operations versus the number of edges in the contour. It used different size of windows and the results approve that the fastest algorithm is the one which has the smallest size of windows. In this thesis all the test contours used were extracted using the “SSPCE” (single step parallel contour extraction) algorithm by using 8-directional chain coding scheme.
2. Two algorithms for contour data approximation in the time domain were proposed. The first algorithm, “contour approximation using centroid method”, and the second algorithm, “contour compression using segment distances ratio”. The comparison has been done using mean square error, signal-to-noise ratio, compression ratio and the number of operations. The comparison between these two algorithms and the well known Ramer algorithm has showed that:
 - The compression ratio using the first and second proposed algorithms can be even greater than 96% and their complexity is much less than that of the Ramer algorithm.
 - The second proposed algorithm in comparison with the other algorithms has very short computational time of the approximating procedure (i.e. many times faster than that of Ramer).
3. Four algorithms reefered to as the “triangle family” for contour data approximation in the time domain are developed. The first algorithm is, “a new method for contour compression using height over length ratio of the triangle”; the second algorithm is, “a new method for contour compression using height of the triangle”; the third algorithm is, “a new method for contour compression using height square of the triangle”; the fourth algorithm is, “a new method for contour

compression using the area of the triangle”. The comparison between these algorithms and the well known Ramer algorithm has showed that:

- The compression ratio using the proposed algorithms can be even greater than 97% with some details losses and their complexity is much less than that of the Ramer algorithm.
 - In general, it can be said that the SNR using the Ramer algorithm is very close to the triangle family algorithms. The reconstruction quality using the third algorithm is very similar to the Ramer algorithm.
 - The number of operations is very similar between the triangle family algorithms at high compression. The triangle family algorithms are many times faster than that of Ramer method (more than 50 times faster).
4. Three algorithms for contour extraction (using high-pass filter) and image compression (using low-pass filter) were developed and compared with Sobel and Canny detectors. The results show that the third proposed algorithm has the best extraction property out of the others. Also it is shown that the algorithms for contour extraction using periodic Haar piecewise-linear (PHL) transform by using different zonal compression method is faster than that of discrete cosine transform (DCT) and Haar transform and Sobel and Canny contour detectors as well.
 5. Combination of the original real grey-level images by a developed algorithm was presented. The algorithm combines their extracted contours that are obtained using high-pass filters and by using one of the transforms (PHL or DCT or Haar).
 6. Two algorithms were presented for image reconstruction by contours extraction. The first algorithm using HPF and double transforms (two 1-dimensional and 2-dimensional PHL transform) while the second using two 1-dimensional PHL transform and only the most significant bit of the image. However, the SNR is not too high (i.e. some significant loses of approximation quality).

REFERENCES

- [1] A. Nabout and H. A. Nour Eldin, “The topological contour closure requirement for object extraction from 2D-digital images”, *Proceedings 1993 IEEE International Conference on Systems, Man and Cybernetics (Le Touquet, France), 1993*, pp. 120-125.
- [2] Scott E. Umbaugh, “Computer vision and image processing”, Prentice-Hall, 1998.
- [3] Nalini K. Ratha, Tolga Acar, Muhittin Gokmen, and Anil K. Jain, “A distributed Edge detection and surface reconstruction algorithm”, *Proceedings Computer Architectures for machine Perception (Como, Italy), 1995*, pp. 149-154.
- [4] Yali Amit, “2D Object Detection and Recognition”, MIT Press, 2002.
- [5] G. Economou, S. Fotopoulos, and M. Vemis, “A novel edge detector based on non – linear local operations”, *Proceedings IEEE International Symposium on Circuits and Systems (London), 1994*, pp. 293-296.
- [6] D.H. Ballard and C.M. Brown, “Computer vision”, Prentice Hall, Englewood Cliffs, NJ, 1982.
- [7] R.M. Haralick and L. G. Shapiro, “Computer and robot vision”, Addison-Wesley Publishing Co., 1992.
- [8] B.K.P. Horn, “Robot vision”, The MIT Press, Cambridge, MA, 1986.
- [9] W.K. Pratt, “Digital image processing”, Wiley, New York, 1991.
- [10] Luciano da Fontoura Da Costa And Roberto Marcondes Cesar Junior., “Shape Analysis and Classification: Theory and Practice”, 1st edition, CRC Press, Inc. Boca Raton, FL, USA, 2000.
- [11] Kim L. Boyer and Sudeep Sarkar., “On the localization performance measure and optimum edge detection”, *Proceedings IEEE Transactions on Pattern Analysis and Machine Intelligence 16 (1994)*, pp. 106-108.

- [12] J. F. Canny, "A computational approach to edge detection", *Proceedings IEEE Transactions on Pattern Analysis and Machine Intelligence* 8 (1986), pp. 679-698.
- [13] Didier Demigny and Tawfik Kamle, "A discrete expression of cranny's criteria for step edge detector performances evaluation", *Proceedings IEEE Transactions on Pattern Analysis and Machine Intelligence* 19 (1997), pp. 1199-1211.
- [14] Hemant D. Tagare and Rui J.P. deFigueiredo, "Reply to on the localization performance measure and optimal edge detection", *Proceedings IEEE Transactions on Pattern Analysis and Machine Intelligence* 16 (1994), pp. 108-110.
- [15] L.S. Davis, "A survey of edge detection techniques", *Computer Vision, Graphics, and Image Processing* 4 (1975), pp. 248-270.
- [16] R.M. Haralick, "Edge and region analysis for digital image data", *Computer Vision, Graphics and Image Processing* (1980), pp. 60-73.
- [17] Sadayuki Hongo, Mitsuo Kawato, Toshi Inui and Sei Miyake, "Contour extraction of images on parallel computer-local, Parallel and stochastic algorithm which learns energy parameters", *International Joint Conference on Neural Networks (Washington D.C.)*, 1989, pp. 161-168.
- [18] P.K. Sahoo, S. Soltani, A. K.C. Wong and Y.C. Chen, "A survey of thresholding techniques", *Computer Vision, Graphics and Image Processing* (1988), pp. 233-260.
- [19] Marie-Pierre Dubuisson and Anil K. Jain, "Object contour extraction using color and motion", *Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition (New York)*, 1993, pp. 471-476.
- [20] M. Bennamoun, B. Boashash, and A. Bower, "Object contour extraction from motion parameters", *Proceedings 2nd Australian and New Zealand Conference on intelligent Information Systems (Brisbane, Australia)*, pp. 337-341.
- [21] Safar, Maytham H., Shahabi, Cyrus, "Shape Analysis and Retrieval of Multimedia Objects", Springer Berlin Heidelberg New York, 2003, Hardcover ISBN: 1-4020-7252-X.

- [22] M. Fleck, "Some defects in finite-difference edge finders", *Proceedings IEEE Transactions on Pattern Analysis and Machine Intelligence* 14 (1992), pp. 337-345.
- [23] P. Parodi and G. Piccioli, "A feature-based recognition scheme for traffic scenes", *Proceedings of the Intelligent Vehicles (Detroit, USA)*, 1995.
- [24] Gerhard X. Ritter and Joseph N. Wilson, "Computer vision algorithms in image algebra", CRC Press, New York, 1996.
- [25] Giovanni Marola and Anna Vaccarelli, "An algebra method for detecting three-dimensional objects in a picture", *Proceedings IAPR International Conference on Pattern Recognition (The Hague, The Netherlands)*, 1992, pp. 415-419.
- [26] Peter J. Olver, Guillermo Sapiro and Allen Tannenbaum, "Affine Invariant Invariant detection: Edges, active contours and segments", *Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition (San Francisco, California)*, 1996, pp. 520-525.
- [27] R. Mohrotra, "Similar-shape retrieval in shape data management", *Computer* 28 (1995), pp. 57-62.
- [28] Jean-Paul Zolesio, Marwan Moubachir, "Moving Shape Analysis And Control : Applications to Fluid Structure Interactions", CRC Pr I Llc, 2006.
- [29] Wei Xiong and Christine Graffigne, "A hierarchical method for detection of moving objects", *Proceedings IEEE International Conference on Image Processing (Austin, Texas)*, 1994, pp. 795-799.
- [30] H. Freeman, "Techniques for The Digital Computer Analysis of Chain Encoded Arbitrary Plane Curves", *Proceedings of the National Electrician Conference*, 1961, pp. 421-432.
- [31] U. Ramer, "An Iterative Procedure for the Polygonal Approximation of Plane Curves", *Computer Graphics and Image Processing*, Academic Press, 1972, pp. 244-256.
- [32] A. K. Jain, "Fundamentals of Digital Image Processing", New Jersey: Prentice Hall International, 1989.

- [33] A. Dziech, W. S. Besbas, "Fast Algorithm for Closed Contour Extraction", *Proceedings of the Int. Workshop on Systems, Signals and Image Processing*, Poznań, Poland, 1997, pp. 203-206.
- [34] B. Tibken, A. Dziech, R. Baran, "Spatial Methods of Contour Approximation", *International Conference on Computational Intelligence for Modeling, Control and Automation*, Vienna, Austria, 2003, pp. 354-362.
- [35] Michael Leyton , "A Generative Theory of Shape", Springer-Verlag, October 2001.
- [36] R. Baran, W. Dziech, "Tangent Method of Contour Approximation", *International Workshop on Multimedia Communication and Services*, Kielce, Poland, 2003, pp. 11-16.
- [37] A. Dziech, R. Baran, "The Tangent Method Algorithms for Contour Approximation", *International Workshop on Multimedia Communication and Services*, Cracov, Poland, 2004, pp. 11-16.
- [38] Duda R.O., and Hart P.E., "Pattern Classification and Scene Analysis", John Wiley & Sons, New York, 1973.
- [39] Heijden F., "Image Based Measurement Systems", John Wiley, 1995.
- [40] Nabout A., Su B., Nour Eldin H. A., "A Novel Closed Contour Extractor, Principle and algorithm", *Proceedings of the International IEEE/ISCAS Conf. on Circuits and Systems, April 29 – May 3, Seattle, USA, Vol.1*, pp. 445-448, 1995.
- [41] W. Besbas, "Contour Extraction, Processing and Recognition", Poznan University of Technology, Ph. D. Thesis, 1998.
- [42] Batchelor B. G., Laing S. G., "Polar-Vector Representations of edges in Pictures" *Electronics Letters*, Vol.13, No.24, pp.727-729, 1977.
- [43] JL Starck and F Murtagh, "Astronomical Image and Data Analysis", Springer, 2006 (2nd edition).
- [44] Freeman H., "Boundary Encoding and processing", in *Picture Processing And Psychopictorics (B. S. Lipkin and Rosenfeld, Eds)*, pp.241-266, Academic

Press, NY, 1970.

- [45] A. Sirjani, G. R. Cross Dixon, "An Algorithm for Polygonal Approximation of Digital Objects", *Pattern Recognition Letters, Elsevier Science Publishers*, PP.299-303, 1998.
- [46] U. Montanari, "A Note on Minimal Length Polygonal Approximation to a Digitized Contour", *Comm. ACM, Vol. 13*, PP.41-47, 1970.
- [47] Andrzej Dziech, Remigiusz Baran and Ali Ukasha, "Contour Compression Using Centroid Method", *WSEAS Int.Conf. on Electronics, Signal Processing and Control (ESPOCO 2005)*, Copacabana, Rio de Janeiro, Brazil, April 25-27, 2005, pp.225-229.
- [48] Vetterli, Martin & Kovacevic, Jelena, *Wavelets and Subband Coding*, Printice Hall Inc., 1995.
- [49] Walsh, J. L. "A Closed Set of Normal Orthogonal Functions", *Amer. J. Math.* 45, 5-24, 1923.
- [50] Wolfram, S., "A New kind of Science", Champaign, IL: Wolfram Media, pp. 573 and 1072-1073, 2002.
- [51] Harmuth, H. F., "Applications of Walsh Functions in Communications", *IEEE Spectrum* 6, pp.82-91, 1969.
- [52] Gonzalez, R.C. and Woods, R. E., "Digital Image Processing", Addison – Wesley, Reading, MA., 1992.
- [53] A. Dziech, F. Belgasse & H. J. Nern, Image data compression using zonal sampling and piecewise-linear transforms, *Journal of Intelligent And Robotic Systems. Theory & Applications*, 28(1-2), Kluwer Academic Publishers, June 2000, 61-68.
- [54] Clarke R. J., "Transform Coding of Images", Academic Press, 1985.
- [55] Krim, Hamid; Yexxi, Jr., Anthony, "Statistics and Analysis of Shapes", 2006, XL 395 p., 143 illus., Hardcover ISBN: 0-8176-4376-1.

- [56] Brigham E.O., "The Fast Fourier Transform", Prentice-Hall, Englewood Cliffs, 1974.
- [57] Andrzej Dziech, Ali Ukasha and Remigiusz Baran, "A New Method For Contour Compression", *WSEAS Int. Conf. on Signal, Speech and Signal Processing (SSIP 2005)*, Corfu Island, Greece, August 17-19, 2005, pp.282-286.
- [58] Andrzej Dziech, Ali Ukasha and Remigiusz Baran, "Fast Method For Contour Approximation And Compression", *WSEAS Transactions on Communications Issue 1, Volume 5*, January 2006, pp. 49 - 56.
- [59] Dziech A., Ukasha A. and Wassermann J., "A New Method for Contour Extraction and Image Compression in Spectral Domain", *International Conference on Multimedia Signal Processing and Communications*, Zadar, Croatia, 2006, pp. 41- 44.

WNIOSKI

Najważniejsze rezultaty uzyskane w pracy doktorskiej oraz wyciągnięte z nich wnioski zaprezentowane są poniżej:

1. Przedstawiono szczegółową analizę istniejących metod dotyczących ekstrakcji krawędzi obrazu, jak również porównanie tych metod poprzez zestawienie liczby operacji i liczby krawędzi w konturze obrazu. W tej metodzie wykorzystuje się różne rozmiary okien, w wyniku czego najszybszym algorytmem jest ten z najmniejszym rozmiarem okien. W tej pracy doktorskiej wszystkie testy na konturach były przeprowadzane z użyciem algorytmu „SSPCE” (pojedynczy krok równoległych ekstrakcji krawędzi). Algorytm ten wykorzystuje 8 (ośmio-) kierunkowe kodowanie łańcuchowe.
2. W pracy doktorskiej zaprezentowano również dwa algorytmy aproksymacji krawędzi w dziedzinie czasu. Pierwszy z tych algorytmów to: „kompresja krawędzi z użyciem centralnej metody”, zaś drugim algorytmem jest: „kompresja krawędzi z użyciem częściowych odległości”. Do ich porównania wykorzystano błąd średniokwadratowy, SNR (stosunek sygnału do szumu), stosunek kompresji oraz liczbę operacji. Owo porównanie tych dwóch algorytmów oraz algorytmu Ramer wykazało następujące zależności:
 - Stosunek kompresji z użyciem pierwszego oraz drugiego zaproponowanego algorytmu może być znacznie większy niż 96% a ich złożoność jest dużo mniejsza niż w przypadku algorytmu Ramer.
 - Drugi algorytm porównywany z resztą algorytmów posiada krótki czas obliczeniowy w metodach przybliżonych. (to jest: bardzo często znacznie szybszy czas niż w algorytmie Ramer).

3. Przedstawiono cztery algorytmy aproksymacji krawędzi należące do tak zwanej „rodziny trójkątnej”. Pierwszy algorytm to: „nowa metoda kompresji krawędzi w wykorzystaniu stosunku wysokości nad długością trójkąta”; drugi algorytm to: „nowa metoda kompresji krawędzi z użyciem wysokości trójkąta”; z kolei trzeci algorytm to: „nowa metoda kompresji krawędzi z użyciem kąta trójkąta”; zaś czwarty algorytm to „nowa metoda kompresji krawędzi z wykorzystaniem powierzchni trójkąta”. Porównanie tych algorytmów oraz algorytmu Ramer wykazało następujące zależności:

- Stosunek kompresji z użyciem zaproponowanych algorytmów może być znacznie większy niż 97% bez znaczącej widocznej straty, natomiast złożoność tych algorytmów jest znacznie mniejsza niż algorytm Ramer.
- Powszechnie uważa się, że stosunek sygnału do szumu (SNR) z wykorzystaniem algorytmu Ramer jest bardzo zbliżony do algorytmów z rodziny trójkątnej. Odtworzenie jakości z wykorzystaniem trzeciego algorytmu następuje w podobny sposób jak w przypadku algorytmu Ramer.
- Liczba operacji jest zbliżona do algorytmów z rodziny trójkątnej o wysokiej kompresji. Algorytmy z rodziny trójkątnej są bardzo często szybsze niż algorytm Ramer (ponad 50 [pięćdziesiąt] razy szybsze).

4. Udoskonalono trzy algorytmy do ekstrakcji krawędzi (z użyciem filtra górno-przepustowego) oraz do kompresji obrazu (z użyciem filtra dolno-przepustowego). Porównano je z detektorem Sobel oraz Canny. W rezultacie stwierdzono, że trzeci z przedstawionych algorytmów w stosunku do reszty posiada najlepszą ekstrakcję. Ponadto wykazano, że algorytmy ekstrakcji krawędzi wykorzystujące transformatę PHL z różnymi strefowymi metodami kompresji są szybsze niż dyskretna transformacja kosinusoidowa (DCT), transformacja Haara oraz detektory kontur Sobel i Canny.

5. Zaprezentowano zrekonstruowany obraz w odcieniach szarości z udoskonalonym algorytmem. Algorytm ten łączy kontury z obrazu, które zostały uzyskane za pomocą filtrów górno-przepustowych oraz jednej z transformacji (PHL, DCT lub Haara).
6. Do zrekonstruowanego obrazu z ekstrakcją krawędzi użyto dwóch algorytmów. Pierwszy z nich wykorzystuje HPF oraz podwójną transformację (1[jedno-]wymiarowej oraz 2[dwu-]wymiarowej transformacji PHL), natomiast drugi algorytm używa 1-wymiarowej transformacji PHL oraz najważniejszego bitu z danego obrazu. SNR (stosunek sygnału do szumu) nie jest zbyt wysoki, a zrekonstruowany obraz jest zadawalający.