
LAPPEENRANNAN TEKNILLINEN KORKEAKOULU
LAPPEENRANTA UNIVERSITY OF TECHNOLOGY

TIETEELLISIÄ JULKAISUJA 50
RESEARCH PAPERS

PEKKA TOIVANEN

New Distance Transforms for Gray – Level Image Compression

ISBN 978-952-214-754-7 (PDF)

LAPPEENRANNAN TEKNILLINEN KORKEAKOULU
LAPPEENRANTA UNIVERSITY OF TECHNOLOGY

UDK 621.397.3 :
519.72 :
519.712

TIETEELLISIÄ JULKAISUJA
RESEARCH PAPERS

50

PEKKA TOIVANEN

New Distance Transforms for Gray-Level Image Compression

Thesis for the degree of Doctor of Technology to be presented with due permission for public examination and criticism in the auditorium of the House of Student Union at Lappeenranta University of Technology (Lappeenranta, Finland) on the 27th of April, 1996, at 12 o'clock noon.

LAPPEENRANTA
1996

ISBN 951-764-031-5
ISSN 0356-8210

The Lord is my shepherd, I shall not be in want.
He makes me lie down in green pastures,
he leads me beside quiet waters,
he restores my soul.
He guides me in paths of
righteousness
for his name's sake.
Even though I walk
through the valley of the shadow of death,
I will fear no evil,
for you are with me; your rod and your staff,
they comfort me.

You prepare a table before me
in the presence of my enemies.
You anoint my head with oil;
my cup overflows.
Surely goodness and love will
follow me
all the days of my life,
and I will dwell in the house of the
Lord
for ever.

Psalm 23

Preface

The work presented in this thesis has been carried out at the Laboratory of Information Processing in the Department of Information Technology of the Lappeenranta University of Technology, Finland between autumn 1991 and autumn 1994.

I wish to express my deep gratitude to professor Jussi Parkkinen who guided my reasearch work at its final stage with many fruitful discussions and revised the manuscript with many valuable comments. He also provided me excellent facilities for intensive research work.

I am greatly indebted to professor Erkki Oja for taking me as a member of his research personnel in Lappeenranta University of Technology and providing excellent facilities for research work as well as for his supportive guidance and inspiration.

Professor Jaakko Astola and Dr. Ari Visa are also gratefully acknowledged. Their wise criticism helped me to complete the manuscript of this thesis. I am also very thankful to Dr. Ari Vepsäläinen.

Moreover, I also wish to thank all colleagues at the Laboratory of Information Processing for a friendly atmosphere and fruitful discussions throughout the work. Specially I wish to thank Dr. Jouko Lampinen, Dr. Heikki Kälviäinen, Dr. Jukka Heikkonen, Dr. Pasi Koikkalainen, M.Sc. Päivi Ovaska, M.Sc. Jari Mononen, M.Sc. Markku Hauta-Kasari, M.Sc. Jouni Ikonen and M.Sc. Leena Tamminen.

The financial support of the Academy of Finland, Lahja and Lauri Hotinen foundation, Finnish Cultural foundation and Jenny and Antti Wihuri foundation is acknowledged. The scholarship from the Lappeenranta University of Technology was of great value in finishing this thesis.

I want to thank my parents Raili and Pertti for encouragement during all these years.

Finally, I like to express my deepest thankfulness to my wife Jaana. Without her support I would never have been able to complete this thesis. Also, I like to thank our daughter Laura and our sons Tuomas and Markus for never getting bored to my talk on reasearch topics.

Pekka Toivanen
Lappeenranta, January, 1996.

List of Publications

1. A. M. Vepsäläinen and P. J. Toivanen: Two New Image Compression Methods Utilizing Mathematical Morphology", in *Proceedings of the SPIE Visual Communications and Image Processing '91*, Vol. 1606, pp. 282-293, Boston, USA, November 11-13, 1991.

2. P. J. Toivanen: Fast Image Compression Using Distance Function on Curved Space, in *Proceedings of the SPIE Visual Communications and Image Processing '92*, Boston, USA, Vol. 1818, pp. 877-884, November 18-20, 1992.

3. P. J. Toivanen: Image Compression Using DTOCS and Neighborhood Masks along Threshold Boundaries, in *Proceedings of the IEEE Winter Workshop on Nonlinear Digital Signal Processing*, Tampere, Finland, pp. 2.2-6.1 - 2.2-6.6, January 17-20, 1993.

4. P. J. Toivanen: Image Compression by Selecting Control Points Using Distance Function on Curved Space, *Pattern Recognition Letters 14*, pp. 475-482, North-Holland, June 1993.

5. P. J. Toivanen: The Euclidean Distance Transform on Curved Space (EDTOCS) with Application to Image Compression", *in Proceedings of the 7th European Signal Processing Conference (EUSIPCO-94)*, pp. 983-986, Edinburgh, Scotland, U.K, September 13-16, 1994.

6. P. J. Toivanen: Convergence Properties of the Distance Transform on Curved Space (DTOCS), in Proceedings of the *1995 Finnish Signal Processing Symposium (FINSIG'95)*, pp. 75-79, Espoo, Finland, June 2, 1995.

Contents

Abstract

This thesis deals with distance transforms which are a fundamental issue in image processing and computer vision. In this thesis, two new distance transforms for gray-level images are presented. As a new application for distance transforms, they are applied to gray-level image compression.

The new distance transforms are both new extensions of the well-known distance transform algorithm developed by Rosenfeld, Pfaltz and Lay. With some modification their algorithm which calculates a distance transform on binary images with a chosen kernel has been made to calculate a chessboard-like distance transform with integer numbers (DTOCS) and a real-value distance transform (EDTOCS) on gray-level images.

Both distance transforms, the DTOCS and EDTOCS, require only two passes over the gray-level image and are extremely simple to implement. Only two image buffers are needed: The original gray-level image and the binary image which defines the region(s) of calculation. No other image buffers are needed even if more than one iteration round is performed. For large neighborhoods and complicated images the two-pass distance algorithm has to be applied to the image more than once, typically 3-10 times. Different types of kernels can be adopted. It is important to notice that no other existing transform calculates the same kind of distance map as the DTOCS. All the other gray-weighted distance function, GRAYMAT etc. algorithms find the minimum path joining two points by the smallest sum of gray levels or weighting the distance values directly by the gray levels in some manner. The DTOCS does not weight them that way. The DTOCS gives a weighted version of the chessboard distance map. The weights are not constant, but gray-value differences of the original image. The difference between the DTOCS map and other distance transforms for gray-level images is shown. The difference between the DTOCS and EDTOCS is that the EDTOCS calculates these gray-level differences in a different way. It propagates local Euclidean distances inside a kernel. Analytical derivations of some results concerning the DTOCS and the EDTOCS are presented.

Commonly distance transforms are used for feature extraction in pattern recognition and learning. Their use in image compression is very rare. This thesis introduces a new application area for distance transforms. Three new image compression algorithms based on the DTOCS and one based on the EDTOCS are presented. Control points, i.e. points that are considered fundamental for the reconstruction of the image, are selected from the gray-level image using the DTOCS and the EDTOCS. The first group of methods select the maximas of the distance image to new control points and the second group of methods compare the DTOCS distance to binary image chessboard distance. The effect of applying threshold masks of different sizes along the threshold boundaries is studied. The time complexity of the compression algorithms is analyzed both analytically and experimentally. It is shown that the time complexity of the algorithms is independent of the number of control points, i.e. the compression ratio. Also a new morphological image decompression scheme is presented, the 8 kernels' method.

Several decompressed images are presented. The best results are obtained using the Delaunay triangulation. The obtained image quality equals that of the DCT images with a 4×4

normalization matrix.

Keywords: Distance transforms, geodesic distance, gray-level distance transforms, raster scanning, image compression, surface interpolation, feature extraction, mathematical morphology, computer vision, machine vision

List of symbols

Symbol	Explanation
λ, ρ	scalars
x, y, a, b, h	Points in R^n or Z^n
X, Y, Z	Euclidean or digital sets under study
B	structuring element
B^S	symmetric set of B with respect to origin
R^n	Euclidean space of dimension n
Z^n	Digital space of dimension n
$B \subset Z$	Z contains B
$x \in X$	point x belongs to set X
X^c	complement of X
λx	homothetic of X with scaling factor λ
$X \cup Y$	set union, i.e. set of points belonging to X or Y
$X \cap Y$	set intersection
$X \oplus B$	Dilation
$X \ominus B$	Erosion
∂X	boundary of set X
$X \setminus Y$	set difference
$d(x, y)$	distance between x and y
$N_4(p)$	the 4 horizontal and vertical neighbors of pixel p
$N_8(p)$	the 8 neighbors of pixel p

In publications 1,2 and 3, α denotes a new curvature constant which controls the amount in which the curvature of the original image is taken into account when calculating the DTOCS (Distance Transform on Curved Space).

In publications 2 and 3, ϵ denotes a predefined threshold, i.e. the difference between the curved distance (DTOCS) and direct distance (distance if the image was binary).

Chapter 1

Introduction

Vision is clearly the most intriguing sense of living animals. It allows humans to perceive and understand the world in an accurate and effortless way. Computer vision aims to duplicate the effect of human vision by electronically perceiving and understanding an image. Giving computers the ability to see is not an easy task. We live in a three-dimensional world, and when computers try to analyze objects in three-dimensional space, available visual sensors, e.g. TV cameras, usually give two-dimensional images, and this projection to a lower number of dimensions incurs an enormous loss of information. Dynamic scenes such as those to which we are accustomed, with moving objects or a moving camera, make computer vision even more complicated.

Mathematical morphology is a particular discipline in the field of image processing, which has been applied to analyse the structure of materials in various disciplines such as mineralogy, petrography, angiography, cytology, etc. It was born in 1964 when Matheron started to investigate the relationships between the geometry of porous media and their permeabilities, and Serra started to quantify the petrography of iron ores in order to predict their milling properties. This initial research led to the formation of team at the Paris School of Mines at Fontainebleau, the Centre de Morphologie Mathématique, which combined theoretical work with the design of practical applications.

Morphological image processing has grown from a specialized imaging discipline to a major area of study within image processing wordwide in recent years. Growth has been substantial in both theory and application. This recent expansion of interest is evidenced in part by the large number of journal and conference papers presented over the past few years, but it is also reflected in the many industrial applications that have emerged and are currently being developed. These applications range over the entire imaging spectrum, including character recognition, medical imaging, microscopy, inspection, metallurgy, and robot vision, to name a few. Morphology has become a necessary tool for those engineers involved in imaging applications.

All perception processes have one very important common feature, which was observed during the emergence of pattern recognition: patterns forming a class can be gradually deformed without abruptly losing the class membership. This statement refers to all perceptual and sensory pattern classes: images, sounds, odors, etc. It is important to note that this is

not the feature of the natural pattern classes only but also the man-made patterns. Thus, some of the most important man-made patterns, handwritten characters of all languages, possess this relative deformation stability. The early introduction and the critical role in pattern recognition of the terms "similarity" and "dissimilarity" can be attributed to the above observation. The above phenomenon is a sufficient argument in favor of adopting the mathematical concept of distance function as basic in a formal pattern recognition model. Indeed, the concept of distance, or dissimilarity measure, has always played a critical role in pattern recognition. To verify this, it is enough to scan any textbook on the subject. [Gol92]

Albert Einstein believed "that the history of scientific development has shown that of all thinkable theoretical structures a single one has at each stage of advance proved superior to all the others". He also believed that this chosen system is "the simplest possible system of thought which will bind together the observed facts" [Fre79]. In information processing fields, which do not deal directly with "the observed data" and thus, of necessity, must be more "abstract" than the natural sciences, this characterization can be slightly rephrased: the model that combines maximal flexibility and universality in its present and future applications with the simplest analytical structure must be preferable. [Gol92]

The past experience of pattern recognition suggests that at present, and in the foreseeable future, there is no tool for pattern learning model more flexible and universal and yet simpler than the concept of distance between two patterns. [Gol92]

In image processing, the concept of distance is most conveniently calculated using mathematical morphology. It provides the simplest and most flexible tools for performing distance calculations. This thesis presents two new distance transforms for gray-level images. Previously presented distance transforms weight the distance values by the gray values themselves. The new transforms presented in this thesis do not. Therefore, the transforms can be used in calculating minimal distances between points along a curved surface. Furthermore, some basic research is done in applying them to gray-level image compression. The time complexity of the new compression algorithms is analyzed both analytically and experimentally. It is shown that the time complexity of the algorithms is independent of the number of control points, i.e. the compression ratio. Regarding the criteria for choosing a compression technique, the algorithms presented in this thesis could best be used in applications where a rapid flashing image is made more accurate as time goes by. The decompressed image is enhanced iteratively and this can be done only for those parts of the image that the user requires, i.e. independent of the other areas. These methods could be applied to transferring time-varying images over a communications channel. It satisfies to send only some data of those areas that have changed compared to the previous image frame. This is due to the local nature of the operations of mathematical morphology and Delaunay triangulation.

The aims of this thesis can be summarized as follows:

1. to present two new and fast distance transforms for gray-level images, the Distance Transform on Curved Space (DTOCS) which is an integer transform, and the Euclidean Distance Transform on Curved Space (EDTOCS) which is a real number transform.

2. to present, as a new application for distance transforms, image compression algorithms for gray-level images based on the DTOCS and the EDTOCS. The distance transforms

are used in finding control points, i.e. points that are considered fundamental for the reconstruction of the image, in the original gray-level image.

In this thesis, all the concepts are presented in discrete framework. This thesis begins with a list of publications, contents and a list of symbols used in this thesis. Chapter 1 is an introduction to the field of image analysis presenting the aim, scope and motivation of this thesis. Chapter 2 presents a brief introduction to the mathematical background of this thesis. It presents the basic operations of mathematical morphology and a definition for the skeleton based on successive erosions. Distance transforms for binary and gray-level images and are also covered. This chapter presents the Rosenfeld-Pfaltz-Lay algorithm, which is the forefather of the DTOCS and the EDTOCS. A list of other distance transform types and techniques for calculating them are presented. Also the GRAYMAT and the gray-weighted distance transform are presented, to which the DTOCS is compared in Publication 6. Delaunay triangulation is presented, which is used in the decompression of the compression part of this thesis. Chapter 3 deals with DTOCS and EDTOCS applied to image compression. The definitions of the DTOCS and EDTOCS are given. An overview on existing compression methods is given. Four compression algorithms are presented with several test images. The time complexity of the new compression algorithms is derived both analytically and experimentally. Chapter 4 is a discussion on the methods presented in this thesis and possible future research topics. Summary of the publications with contributions by the author and errata are found in Chapter 5. Chapter 6 presents an Appendix with decompressed images. The publications are found in Chapter 7.

Chapter 2

Mathematical Background

2.1 Mathematical Morphology

In the following, some some basic concepts of mathematical morphology are defined. Let Z^k be the set of k-tuples of integers. k is a positive integer, the dimension of the set.

Definition 2.1. Let $B \subset Z^k$. The symmetric set B^S is the set

$$B^S = \{-x \mid x \in B\} \tag{2.1}$$

Definition 2.2. Let $B \subset Z^k$. The translated set B_x, where the set B is translated by $x \in Z^k$, is defined by

$$B_x = \{x + y \mid y \in B\} \tag{2.2}$$

Definition 2.3. Let $A \subset Z^k$ and the structuring element $B \subset Z^k$. Dilation is denoted by $A \oplus B$ and is defined by

$$A \oplus B = \{a + b \mid a \in A, b \in B\} \tag{2.3}$$

Definition 2.4. Let $A \subset Z^k$ and the structuring element $B \subset Z^k$. Erosion of A by B is denoted by $A \ominus B$ and is defined by

$$A \ominus B = \{z \in Z^k \mid B_z \subseteq A\} \tag{2.4}$$

Definition 2.5. Let $A \subset Z^k$ and the structuring element $B \subset Z^k$. Opening of A by B is denoted by $A \circ B$ and is defined by

$$A \circ B = (A \ominus B) \oplus B \tag{2.5}$$

Definition 2.6. Let $A \subset Z^k$ and the structuring element $B \subset Z^k$. Closing of A by B is denoted by $A \bullet B$ and is defined by

$$A \bullet B = (A \oplus B) \ominus B \tag{2.6}$$

Definition 2.7. Let $A \subset Z^k$ and the structuring element $B \subset Z^k$. Skeleton $SK(A)$ of a set A is defined as follows:

$$S_n(A) = ((A \ominus nB) \setminus (A \ominus nB)) \circ B,\ n = 0, 1, 2, ..., N \tag{2.7}$$

$$SK(A) = \cup_{n=0}^{N} S_n(A) \tag{2.8}$$

where $N = max\{n | (A \ominus nB) \neq \emptyset\}$. Set difference is denoted by $\setminus$. $nB = \underbrace{B \oplus B \oplus ... \oplus B}_{n \text{ times}}$, $n = 1, 2, ...$. $S_n(A)$ denotes the nth skeleton subset of A.

According to [Vin91b] several types of skeletons, or medial axes, exist. Also a considerable number of different skeleton algorithms based on morphological thinnigs, local maxima of distance functions [Ser88], and contours have been published. [Vin91b]

Several generalizations of the medial axis transform for gray-level images have been presented, including the Spatial Piecewise Approximation by Neighborhoods, SPAN, [Ahu78] [Wan79], the gray-weighted medial axis, GRAYMAT, [Lev70], the gradient medial axis, GRADMAT, [Wan79] and the min-max medial axis transform, MMMAT, [Nak78].

2.2 Distance Transforms

Let Z^k be the set of k-tuples of integers. k is a positive integer, the dimension of the image. An image is a function $f_R : Z^k \rightarrow R$, where R is the set of real numbers. A discrete image is a function $f_N : Z^k \rightarrow N$, where N is the set of positive integers. f_R and f_N are called gray-level images. If $N = \{0, 1\}$ the image f_N is a binary image.

2.2.1 Binary Images

Definition 2.8. Let $A \subset Z^2$. Let $x \in A$ and $y \in A$. Function $d_A(x, y)$ is called a distance function if it satifies the following three criteria:

$$\textit{Symmetricity} : d_A(x, y) = d_A(y, x) \tag{2.9}$$

$$\textit{Positive definiteness} : d_A(x, y) = 0 \Leftrightarrow x = y \tag{2.10}$$

$$\textit{Triangle inequality} : d_A(x, z) \leq d_A(x, y) + d_A(y, z) \tag{2.11}$$

Let $N_4(e)$ denote the set of 4 diagonal and horizontal neighbors of pixel e. Let $N_8(e)$ denote the set of 8 neighbors of pixel e. Let us define a 3×3 square kernel K, $K = N_8(e) \cup \{e\} = \{a, b, c, d, e, f, g, h, k\}$. Table 2.1.(a) presents this kernel, called the chessboard kernel, and Table 2.1.(b) the city-block kernel, in which the diagonal corner points $\{a, c, g, k\}$ are omitted.

a	b	c
d	e	f
g	h	k

(a)

	b	
d	e	f
	h	

(b)

Table 2.1. (a) The 3×3 chessboard kernel used in the Rosenfeld-Pfaltz-Laŷ algorithm. (b) The city-block kernel used in the Rosenfeld-Pfaltz-Laŷ algorithm.

There are several types of distance functions. A distance function with an algorithm to calculate it is called a distance transform. Among the distance functions are the following:

Euclidean. Let Z^2 be the set of 2-tuples of integers. Let $A \subset Z^2$. $x = (x_1, x_2) \in A$ and $y = (y_1, y_2) \in A$. The Euclidean distance between x and y is

$$d_A(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} \tag{2.12}$$

Danielsson [Dan80] proposed an algorithm for the vector Euclidean distance transform, which allows the generation of distance maps with no significant errors for binary images. This

sequential algorithm uses four passes of a 3×3 kernel over the image; the sum of two squares must be computed for each of the nine kernel-pixels; and it needs two extra images of the same size as the original one to store intermediate results. Ye [Ye88] presented a version of the Danielsson's Euclidean distance transform which produces a distance map in which each pixel is a vector of two integer components. A parallel Euclidean distance transform algorithm that always gives correct results has been published in [Yam84]. This algorithms also uses a 3×3 kernel; the sum of two squares must be computed for each kernel-pixel; and it needs two extra images to store intermediate results: the signed number of steps to the nearest feature pixel. The number of iterations is proportional to the longest distance in the image. In [Rag90] a refined version is presented to generate completely error-free Euclidean distance maps. Vincent [Vin91c] presented an algorithm which produces an error-free Euclidean distance map. His method uses chains in encoding the objects boundaries and propagates these structures in the image using rewriting rules.

The Euclidean distance transform has been also extended to arbitrary dimensions [Rag93]. In [Rag93] some raster scanning, sequential algorithms, are proposed. In [Sai94] fast sequential algorthms for n-dimensional Euclidean distance transform are presented which are faster than the one presented in [Yam84]. Mullikin [Mul92] presented vector distance transform algorithms which are derived from the Danielsson's and Ye's algorithms for anisotropically sampled images.

In [Bor91], a new algorithm for the extraction of local maxima from the Euclidean distance map is presented. Reconstruction of the shape from the local maxima is achieved using a reverse Euclidean distance transform. The resulting shape is exactly the original one, but the distance values are different from the original distance transform.

Chessboard. Let Z^2 be the set of 2-tuples of integers. Let $A \subset Z^2$. $x = (x_1, x_2) \in A$ and $y = (y_1, y_2) \in A$. The Chessboard distance [Bar77] and [Bor84] is defined by the following formula:

$$d_A(x, y) = max\{|x_1 - y_1|, |x_2 - y_2|\} \tag{2.13}$$

The chessboard distance transform tends to give too small values compared to the Euclidean distance transform.

City Block. Let Z^2 be the set of 2-tuples of integers. Let $A \subset Z^2$. $x = (x_1, x_2) \in A$ and $y = (y_1, y_2) \in A$. The city block distance is defined as:

$$d_A(x, y) = |x_1 - y_1| + |x_2 - y_2| \tag{2.14}$$

It is the simplest and fastest of all distances. It is, however, also the worst approximation of the Euclidean distance and tends to give too big values compared to the Euclidean distance transform. It is, among other places, found in [Bar77]. As for all algorithms described by the mask, the computation can be either parallel [Bor84], or sequential [Ros66b].

Other distance distances include the octagonal distance which uses both the city block and the chessboard distance. Presentation of it, and a parallel algorithm for computing it are found in [Ros68]. Other octagonal algorithms are also described in [Ros68]. Chamfer distances and algorithms for their calculation are found in [Ros82],[Bor83], [Bor84] and [Bor86].

$\sqrt{8}$	$\sqrt{5}$	2	$\sqrt{5}$	$\sqrt{8}$
$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{3}$
2	1	0	1	2
$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$
$\sqrt{8}$	$\sqrt{5}$	2	$\sqrt{3}$	$\sqrt{8}$

(a)

2	2	2	2	2
2	1	1	1	2
2	1	0	1	2
2	1	1	1	2
2	2	2	2	2

(b)

4	3	2	3	4
3	2	1	2	3
2	1	0	1	2
3	2	1	2	3
4	3	2	3	4

(c)

Table 2.2. (a) The Euclidean distance for each pixel from the center point. (b) The chessboard distance for each pixel from the center point. (c) The city-block distance for each pixel from the center point.

Digital distance transforms that use only a small image neighborhood at a time are based on the idea of approximating global distances by propagating local distances, i.e. distances between neighboring pixels. This propagation can be done either parallelly or sequentially. Sequential distance transforms for a set $A \subset Z^2$ were first published in 1966 [Ros66b], and parallel ones in 1968 [Ros68]. The sequential algorithms are further divided to raster-scanning and contour-processing algorithms [Rag92a].

In the Rosenfeld-Pfaltz-Laỹ algorithm, [Ros66b], [Lay84], [Ser88], it is assumed that the input image is a binary image or a binarized version of a gray level one. It gives the distance image, whose maximum is always in the middle of a homogenious region. It gives the distance function of a set $A \subset Z^2$ after two iterations. The algorithm is initialized by assigning the value 0 to all points of A^C and the value max, i.e. the maximal representative value of the memory, to all points of A. These points holding the value max are denoted by $a, b, ..., k$ and the changed values $a^*, b^*, ..., k^*$.

The following formulas are for the chessboard kernel of Table 2.1 (a).

1st iteration: The first application of this sequential algorithm proceeds in the "direct video order" (rowwise from left to right, and from top to bottom) substituting the middle point e with new e^* according to the following formula:

$$e^* = min[e, 1 + min(a^*, b^*, c^*, d^*)] \tag{2.15}$$

2nd iteration: The second iteration is applied in the "inverse video order" (rowwise from right to left, and from bottom to top) substituting the middle point e^* with new e^* according to the following formula:

$$e^* = min[e^*, 1 + min(f^*, g^*, h^*, k^*)] \tag{2.16}$$

The Rosenfeld-Pfaltz-Lay algorithm produces a chessboard distance function of Table 2.2.(b) when the chessboard kernel of Table 2.1.(a) is used. Similarly, it produces a city-block distance function of Table 2.2.(c) when the city-block kernel of Table 2.1.(b) is used. From the distance function image generated by the above algorithm it is easy to derive a skeleton [Ser88].

Definition 2.9. Let $A \subset Z^k$ and the structuring element $B \subset Z^k$. $nB = \underbrace{B \oplus B \oplus ... \oplus B}_{n \text{ times}}$, $n = 1, 2, ...$ For each $x \in A$, the distance transform of A with respect to B by successive erosions is denoted by $d[A, B](x)$ and is defined by

$$d[A, B](x) = \begin{cases} max\ \{n \mid x \in A \ominus (n-1)B\} & , x \in A \\ 0 & , x \notin A \end{cases} \tag{2.17}$$

In [Wan88] it is proved that the city block distance function can be calculated either using successive erosions, see Definition 2.9., or using the Rosenfeld-Pfaltz-Lay algorithm.

In early papers about distance transforms, [Ros66b], [Ros68], no attempt was made to optimize the local distances used in the distance transform. This optimization was accomplished for a 3×3 neighborhood in [Bor83] and [Bor84]. Naturally, the the approximation to the Euclidean distance transform becomes better the larger the size of the neighborhood that is used in the algorithm is. The values for the mask coefficients for different neighborhood sizes for both the sequential and parallel algorithms are derived in [Bor86]. In many digital image processing applications real-valued pixels are undesirable. A thorough analysis of integer approximations for different neighborhood sizes is found in [Bor86]. Verwer [Ver91] presented optimal mask coefficients for local distances in two and three dimensions. Integer approximations for the local distances were developed for neighborhoos sizes of three and five.

The distance function can also be obtained by chains and loops, also called contour processing and ordered propagation, as presented in [Vli88],[Vin91a],[Rag92a] and [Rag92b]. Contour processed distance transforms in its basic form have been suggested by Piper and Granum [Pip87]. Verwer et al. [Ver89] used bucket sorting to process the propagation front in perfect order. A simple method for achieving approximately ordered propagation as well as algorithms for Euclidean distance mapping was suggested by Ragnemalm [Rag90] [Rag92a].

Vincent [Vin91a] presented simple algorithms based on queues of pixels, which are more general than those based on chains and loops.

Geodesic distance

The new distance transforms presented in this thesis are derived from the following definition of geodesic distance.

Definition 2.10. Pixels $p \in Z^2$ and $q \in Z^2$ are 8-connected if $q \in N_8(p)$. A discrete 8-path from pixel $r = (x_0, y_0)$ to pixel $s = (x_n, y_n)$ is a sequence of pixels $(x_0, y_0), (x_1, y_1), ..., (x_n, y_n)$ where (x_i, y_i) is 8-connected to (x_{i-1}, y_{i-1}), $i = 1, 2, ..., n$.

Definition 2.11. Let $A \subset Z^2$ and $x \in A$ and $y \in A$. Let's denote by $\mathcal{P}_A(x,y)$ the set of discrete 8-paths in A linking x and y. The number $\delta_A(x,y)$ is defined as follows: $\delta_A(x,y)$ is the minimum of the lengths of the paths γ of $\mathcal{P}_A(x,y)$, if such paths exist and $+\infty$ if not. Mathematically, it can be expressed by [Pre91]:

$$\delta_A(x,y) = \begin{cases} min\ \mathcal{L}(\gamma), \gamma \in \mathcal{P}_A(x,y) & \text{, if } \mathcal{P}_A(x,y) \neq \emptyset \\ +\infty & \text{, otherwise} \end{cases} . \tag{2.18}$$

where $\mathcal{L}(\gamma)$ is the length of the path γ. The length of the path $\mathcal{L}(\gamma)$ is the sum of distances between 8-connected pixels along a discrete 8-path. Here, $\delta_A(x,y)$ is called the geodesic distance from x to y with respect to A. The distance $\delta_A(x,y)$ satisfies the three properties of a distance function: separability, symmetry and triangle inequality. $\delta_A(x,y)$ can take infinite values especially if x and y belong to two different connected components of A. It should be noted that a geodesic path of minimal length between two points x and y is not necessary unique when A is not simply connected. [Pre91] [Sko89]

2.2.2 Gray-Level Images

Rutovitz [Rut68] proposed an algorithm for obtaining a gray-weighted distance function, in which gray-value is identified with a concept of "height" and gray-weighted distance is defined in such a way that it is less along paths with low gray-value pixels. In his method, he proposed only two iterations. The time complexity of his algorithm is of the order of $O(n^2)$ for an $n \times n$ image. Rosenfeld [Ros69] presented a parallel version of this algorithm.

The algorithm of Rutovitz proceeds as follows. The first iteration is performed on the original image and it proceeds rowwise from left to right and from up to bottom, calculating the new value $F^*(e)$. The second iteration is performed on a copy of the original image and it proceeds rowwise from right to left and from bottom to up, calculating the new value $F^*(e)$.

The final distance map is obtained by taking the minimum of the transformed original image and the transformed copy image pixel by pixel. The 3×3 kernel of Table 2.1.(a) is used. $F^*(e)$ denotes the new point at each iteration step, i.e. the transformed point, and $G(a), G(b), \ldots$ denote the gray values of the original gray-level image at the kernel points. β is a parameter for which $0 < \beta \leq 1$.

1st iteration:
The first iteration proceeds in the direct video order, i.e. rowwise from left to right and from top to bottom, calculating the new value

$$F^*(e) = \beta G(e) + min(G(a), G(b), G(c), G(d)) \tag{2.19}$$

2nd iteration:
The second iteration proceeds in the inverse video order, i.e. rowwise from right to left and from bottom to top, calculating the new value

$$F^*(e) = \beta G(e) + min(G(f), G(g), G(h), G(k)) \tag{2.20}$$

Fall-distance [Rut78] [Vos79] is another modification of distance, in which the only permitted paths from the reference set are those with falling, i.e. strictly decreasing, gray values. The set of points reached by such strictly decreasing paths is known as the fall-set of the reference set. In [Rut78] a sequential two-pass algorithm is used to calculate the fall-distance.

Another generalization, the GRAYMAT [Lev70] defines the gray-weighted distance between two points as the smallest sum of gray levels along any path joining the points. The algorithm is obtained by a suitable generalization of the algorithms that have been used in the black-and-white case, e.g. Rosenfeld and Pfaltz [Ros66b], Montanari [Mon68]. The same algorithm is also presented in [Pip87] and is used as the first stage of a cost algorithm in [Ver90]

Let $A \subset Z^2$. Let $x \in A$ and let $G(x)$ be the original gray-level image. The 3×3 kernel K of Table 2.1.(a) is applied to the calculation, but also a 5×5 or city-block kernel of Table 2.1.(b) could be applied with appropriate modifications to the Eq. (2.22.) and Eq. (2.23.). The distance between the center point $e \in K$ and other kernel points $x \in N_8(e) \subset K$ is

$$t(e, x) = \frac{G(e) + G(x)}{2} d(e, x) \tag{2.21}$$

where $d(e, x) = 1$ if $x \in N_4(e)$ and $d(e, x) = \sqrt{2}$ if $x \in (N_8(c) \setminus N_4(e))$

Each pixel of the image $B(x)$ has a high constant value, i.e. the maximal representative number of the memory, and the boundary points of A have a value 0. The pixels of the image are considered sequentially in the following way.

1st iteration: The first iteration proceeds rowwise form left to right, and form top to bottom.

$$B^*(e) = min(B(e), B(a) + t(e, a), B(b) + t(e, b), B(c) + t(e, c), B(d) + t(e, d)) \tag{2.22}$$

2nd iteration: The second iteration proceeds rowwise from right to left, and from bottom to top.

$$B^*(e) = min(B(e), B(f) + t(e, f), B(g) + t(e, g), B(h) + t(e, h), B(k) + t(e, k)) \tag{2.23}$$

The above algorithm presented in [Lev70] has a time complexity of the order of $O(n^2)$ for an $n \times n$ image. The gray-weighted medial axis (GRAYMAT) is defined as the set of points whose gray-weighted distances to the set of 0's are local maxima.

In [Pre91] two new distance transforms are defined, namely the topographical distance, which is defined using a function called the connection cost, and the differential distance transform, which is defined using the deviation cost function.

2.3 Delaunay Triangulation

Surface representation and reconstruction are important problems in a variety of disciplines including geographic data processing, computer vision, computer graphics, and computer aided design.

Let us consider S, a set of points in a plane. The points of S are connected to each other to form triangles. The triangles are not overlapping with each other. Each point belongs to at least one triangle. T is a Delaunay triangulaton of S if the circle connecting all vertices of any of its triangles does not contain any other point of S in its interior.

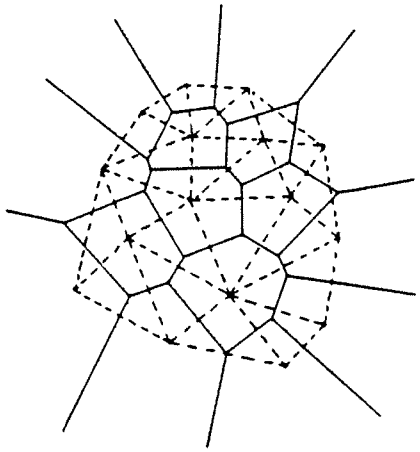

Figure 2.1. The Voronoi tesselation (solid line) and the Delaunay triangulation (dashed line). [Lee80]

The use of the city-block distance and the Euclidean distance in Delaunay triangulation is discussed in [Che85] and [Che86]. Lee presented a recursive algorithm to construct the Delaunay triangulation using the Euclidean distance [Lee80]. The Delaunay triangulation in both two and three dimensions has been used by different authors as the basis for constructing object-centered surface descriptions [Bar77] [Boi84] [Boi88] [Flo89] [Bru91] [Flo92]. Triangle-based representations are invariant under translations and rotations. They adapt to the variable density of the data distribution, and they can be easily updated. In particular, among all possible triangulations, the Delaunay triangulation is considered the most appropriate for surface approximation, because of the equilateral shape of its triangles [Bab76]. [Raj94]

Chapter 3

Image Compression

3.1 Overview on Existing Compression Methods

In recent years, there has been a dramatically increasing demand for handling images in digital form. Owing to performance improvements and significant reductions in the cost of image scanners, photographis, printed text, and other media can now be easily converted into digital form. Direct acquisition of digital images is also becoming more common as sensors and associated electronics improve; the use of satellite imaging, e.g. LANDSAT, in remote sensing of the earth and the advent of electronic still-cameras in the consumer market are good examples. In addition, many different imaging modalities in medicine, such as computed tomography (CT) or magnetic resonance imaging (MRI), generate images directly in digital form. Computer-generated images are also becoming a major source of digital data. [Rab91]

However, there is one problem with digital images, namely, the large number of bits required to represent them. Fortunately, digital images, in their canonical representation, generally contain a significant amount of redundancy. Image compression aims at taking advantage of this redundancy to reduce the number of bits required to represent an image. This can result in significant savings in the memory needed for image storage or in the channel capacity required for image transmission. Several excellent review articles on image compression have been published in the literature [Kun87], [Jai81], [Net80]. There have also been several special journal issues dedicated to image coding and visual communications [Hsi89], [Hsi87], [Net85]. A few books are solely devoted to the subject [Net88], [Cla85], [Jay84], and numerous textbooks on image processing also contain detailed coverage of image compression [Lim90], [Jai89], [Gon87], [Ros82]. [Rab91]

The need for image compression becomes apparent when one computes the number of bits per image resulting from typical sampling rates and quantization schemes. For example, consider the amount of storage required for the following types of images [Rab91]:

- a low-resolution, TV quality, color video image: 512×512 pixels/color, 8 bits/pixel, and 3 colors $\Rightarrow \approx 6 \times 10^6$ bits,

- a 24×36 mm (35-mm) negative photograph scanned at $12\mu m$: 3000×2000 pixels/color, 8 bits/pixel, and 3 colors $\Rightarrow \approx 144 \times 10^6$ bits,

- a 14×17 -inch radiograph scanned at $70\mu m$: 5000×6000 pixels, 12 bits/pixel $\Rightarrow \approx 360 \times 10^6$ bits,

- a LANDSAT Thematic Mapper scene (used in remote sensing): approximately 6000×6000 pixels/spectral band, 8 bits/pixel, and 6 nonthermal spectral bands $\Rightarrow \approx 1.7 \times 10^9$ bits.

Obviously, the storage of even a few images could pose a problem. As another example of the need for image compression, consider the transmission of a low-resolution $512 \times 512 \times 8$ bits/pixel $\times$3-color video image over telephone lines. Using a 9600 baud (bits/s) modem, the transmission would take approximately 11 minutes for just a single image, which is unacceptable for most applications. [Rab91]

Let N_G denote the number of bits in the original image and N_C the number of bits in the compressed image. The compression ratio R is defined as follows [Rab91]:

$$R = \frac{N_G}{N_C} \tag{3.1}$$

In general, three types of redundancy in digital images can be identified [Rab91]:

- spatial redundancy, which is due to the correlation or dependence between neighboring pixel values,

- spectral redundancy, which is due to the correlation between different color planes, e.g. in an RGB color image, or spectral bands, e.g. aerial photographs in remote sensing,

- temporal redundancy, which is due to the correlation between different frames in a sequence of images.

The compression methods can be categorized into two groups: *lossless* and *lossy*. In lossless compression, also known as *bit-preserving* or *reversible* compression, the reconstructed image after compression is numerically identical to the original image on a pixel-by-pixel basis. Obviously, lossless compression is ideally desired since no information is compromised. However, only a modest amount of compression is possible.

In lossy compression, also known as *irreversible* compression, the reconstructed image contains degradations relative to the original. As a result, much higher compression can be achieved as compared to lossless compression. In general, more compression is obtained at the expense of more distortion. It is important to note that these degradations may or may not be visually apparent. In fact, the term *visually lossless* has often been used to characterize lossy compression schemes that result in no visible loss under normal viewing

conditions. Unfortunately, the definition of visually lossless is quite subjective and extreme caution should be taken in its interpretation. [Rab91]

A comprehensive list of factors that can affect the choice of a lossy compression algorithm is presented in [Rab91].

Among the first generation spatial methods are Pulse-Code Modulation (PCM) [Rob62], Predictive Coding [Kun85], Differential Pulse-Code Modulation (DPCM) [Hab71], [Mus79], [Net77], [Lim78], Delta Modulation [Kun85], Interpolative Coding [Rab91] and Bit-Plane Coding [IEE80] [Kun85].

The basic motivation behind transform coding is to pack the relevant information into a small number of coefficients. The inverse transform recovers the original image. Most commonly used transformations are linear transformations implemented with fast algorithms for computational efficiency. [Kun85] [Kun87] [Rab91] Among the first generation transform methods are the Karhunen-Loéve transform [Pra78] [Rab91], the Fourier, Hadamard, Haar, sine, cosine, and slant transforms [Lee84], [Nga84], [Vet84], [Vet85], [Gon88], [Ara88], [Fei90].

Hybrid methods combine predictive coding and transform coding having the advantages of hardware simplicity (DPCM) and robust performances (transform coding). [Kun85], [Bay90].

The second generation methods can be divided into two groups. The first group is characterized by the use of local operators. Pyramidal coding [Bur83] and anisotropic nonstationary predictive coding [Wil83] [Knu83] are the main examples of this group of methods. [Kun87] The second group methods attempt to describe an image in terms of contour and texture. Directional decomposition-based coding [Iko85] and segmentation-based coding [Koc83] are two major examples of this second group of methods.

A recently introduced image compression method is based on the linear wavelet theory. With this method is possible to obtain both time and space resolution at the same time giving better compression ratios than classical methods [DeV92] [Sri93].

In [Wan81] the min-max-Medial Axis Transform is used for image compression giving 1.0 and 2.5 bits per pixel for noisy chromosome images. Such rates are comparable with those typically obtained in interpolative and transform coding schemes.

3.2 The New Compression Algorithms

The existing distance transforms for gray-level images presented in [Rut68] and [Lev70] find the minimum path joining two points by the smallest sum of gray levels or weighting the distance values directly by the gray levels in some manner. Because the distance values do not depend on gray value differences, but the values themselves, they cannot be used in compression algorithms, in which the obtained distance values for the gray-level image are compared to the corresponding binary image values. This is useful in finding the changes in the curvature of the gray-level image and using this information in the placement of control, or dominant points. In addition to this, the new transforms can be used in feature extraction in cases where conventional methods based on binarizing the gray-level images would give no distinction between the pattern classes. This thesis presents two new distance transforms in which the distance values are weighted by the gray value differences of the original image.

Definition 3.1 Let $X \subset Z^2$. Let $B \subset Z^2$ be the structuring element. Let the external boundary of X be denoted by ∂X and be defined by $\partial X = (X \oplus B) \setminus X$. $\partial X \subset X^C$. [Gia88]

In the definitions below we will use the following notation. $x \in X$ and $y \in \partial X$. Let $\Psi_X(x,y)$ be the set of digital 8-paths in $(X \cup \partial X)$ linking x and y. Let $\gamma \in \Psi_X(x,y)$ and let γ have n pixels. Let $a_i \in \gamma$ and $a_{i+1} \in \gamma$ be two adjacent pixels in the path γ. Let $\mathcal{G}_X(a_i)$ denote the gray value of the pixel a_i.

The Distance Transform on Curved Space (DTOCS) is defined as follows.

Definition 3.2. Let the distance between a_i and a_{i+1} be $d_X(a_i, a_{i+1}) = |\mathcal{G}_X(a_i) - \mathcal{G}_X(a_{i+1})| + 1$, $i = 1,2,...,n-1$. The length of the path γ is defined by $\Lambda(\gamma) = \sum_{i=1}^{n-1} d_X(a_i, a_{i+1})$. The DTOCS distance image is defined by

$$\mathcal{F}_X(x) = min(\Lambda(\gamma), \gamma \in \Psi_X(x,y)), \quad y \in \partial X, \quad \Psi_X(x,y) \neq \emptyset \tag{3.2}$$

$$\mathcal{F}_{\partial X}(y) = 0 \tag{3.3}$$

The Euclidean Distance Transform on Curved Space (EDTOCS) is defined as follows.

Definition 3.3. Let the distance between a_i and a_{i+1} be $d_X(a_i, a_{i+1}) = \sqrt{(\mathcal{G}(a_i) - \mathcal{G}(a_{i+1}))^2 + 1}$, $i = 1,2,...,n-1$, if $a_{i+1} \in N_4(a_i)$. $d_X(a_i, a_{i+1}) = \sqrt{(\mathcal{G}(a_i) - \mathcal{G}(a_{i+1}))^2 + 2}$, $i = 1,2,...,n-1$, if $a_{i+1} \in (N_8(a_i) \setminus N_4(a_i))$. The length of the path γ is $\Lambda(\gamma) = \sum_{i=1}^{n-1} d_X(a_i, a_{i+1})$. The EDTOCS distance image is defined by

$$\mathcal{F}_X(x) = min(\Lambda(\gamma), \gamma \in \Psi_X(x,y)), \quad y \in \partial X, \quad \Psi_X(x,y) \neq \emptyset \tag{3.4}$$

$$\mathcal{F}_{\partial X}(y) = 0 \tag{3.5}$$

The DTOCS is used in gray-level image compression in Publications 1,2,3 and 4. The EDTOCS is used in image compression in Publication 5.

Let us make the following definitions for the compression algorithms. The algorithms use the DTOCS both in one and two dimensions. Let the distance function generated by the DTOCS when $\alpha = 0$ on image G be d_G and the distance function generated by DTOCS when $\alpha = 1$ be D_G correspondingly. Let G be the original gray-level image and F be a binary image which determines the region or regions of calculation and contains control points. $rnd(G)$ is an operation which picks a random number of points from the image G. $Edge(G)$ is an operation which produces an edge image from G.

Let's define a square, kernel $K = \{k_1, k_2, ..., k_n\}$. Then, F_j^k denotes a point in the kernel area when the kernel is placed on the image F, $j = 1, ..., n$. Denote $R = \{r_i\}$ the boundary curve with m points in which $\mid D_G - d_G \mid > \epsilon$. r_i^F refers to the ith point in the curve R in the image F. Let $M = \{m_*\}$ be the set of points which hold local maximas of the distance map generated by the DTOCS at each iteration step. $C = \{c_1, c_2, ...\}$, i.e. the set of already picked control points. At the initialization of the algorithms the initial size of C is set to N. That value is normally $5 - 10$. As the algorithms proceed, N is increased at each iteration phase.

DTOCS denotes performing the DTOCS equations with $\alpha = 1$ once, which are found in Publications 1,2,3,4 and 6. DTOCS-alg denotes calculating DTOCS with $\alpha = 0$ and $\alpha = 1$ and comparing the two distance maps pixel by pixel.

The following image compression algorithms are presented in the publications of this thesis. Algorithm 1 is introduced in Publications 1 and 4. Algorithm 2 is introduced in Publication 3. Algorithm 3 is introduced in Publication 2 and algorithm 4 in Publication 5.

Algorithm 1 calculates the DTOCS with $\alpha = 0$ and $\alpha = 1$ and compares the obtained two distance images pixel by pixel. A curve is formed in those pixels where the difference between the two pixels is bigger than a predetermined threshold. The curve is handled so that DTOCS is calculated in one dimension and a new control point is put in positions where the difference between the two DTOCS distances is bigger than the predetermined threshold. This procedure is repeated until no curves are found. Algorithm 2 equals the algorithm 1 except that the curves are handled in a different way. Square masks of different sizes are adopted. A new control point is put in a position where the difference between the two DTOCS distances is bigger than the predetermined threshold and where at the same time there are no previously put control points in the mask area. The basic idea in algorithm 3 is to calculate DTOCS and select the maximas of the obtained distance function as new control points, thereby removing them from the area of calculation. The algorithm is initialized by picking randomly a few points from an edge image of the original image. The algorithm 4 is the same as algorithm 3 except that the EDTOCS is used instead of the DTOCS.

Algorithm 1:

1. **begin** Algorithm 1
2. Set ϵ.

3. Set N.
4. **for** $i = 1$ **to** N **begin**
5. $C \leftarrow rnd(G)$
6. $c_i \leftarrow 0$
7. **endfor**
8. $F \setminus C \leftarrow maxint$
9. **repeat** DTOCS-alg
10. **until** $\mid D_G - d_G \mid > \epsilon$
11. **if** $\{r_i\} \neq \emptyset$ **then begin**
12. $r_0^F \leftarrow 0$
13. $C \leftarrow r_0^F$
14. **endif**
15. **else** Huffman **end.**
16. **for** $i = 1$ **to** m **begin**
17. **if** $\mid D_r(i) - d_r(i) \mid > \epsilon$ **begin**
18. $r_i^F \leftarrow 0$
19. $C \leftarrow r_i^F$
20. **endif**
21. **endfor**
22. **goto** 9.
23. **end**

Algorithm 2:

1. **begin** Algorithm 2
2. Set ϵ.
3. Set N.
4. **for** i=1 **to** N **begin**
5. $C \leftarrow rnd(G)$
6. $c_i \leftarrow 0$
7. **endfor**
8. $F \setminus C \leftarrow maxint$
9. **repeat** DTOCS-alg
10. **until** $\mid D_G - d_G \mid > \epsilon$
11. **if** $\{r_i\} \neq \emptyset$ **then begin**
12. $r_0^F \leftarrow 0$
13. $C \leftarrow r_0^F$
14. **endif**
15. **else** Huffman **end.**
16. **for** $i = 1$ **to** m **begin**
17. place K to r_i^F
18. **if** $\mid D_r(i) - d_r(i) \mid > \epsilon$ **and** $F_j^K \notin C$ **and not** $(F_1^K = F_2^K = ... = F_n^K)$ **then begin**

19. $r_i^F \leftarrow 0$
20. $C \leftarrow r_i^F$
21. **endif**
22. **endfor**
23. **goto** 9.
24. **end**

Algorithm 3:

1. **begin** Algorithm 3
2. $F = Edge(G)$
3. $F = F \oplus B$
4. Set N
5. **for** i=1 **to** N **begin**
6. $C \leftarrow rnd(F)$
7. $c_i \leftarrow 0$
8. **endfor**
9. $F \setminus C \leftarrow maxint$
10. **repeat** DTOCS
11. $\forall s: C \leftarrow m_s$
12. **until** $\{m_s\} = \emptyset$
13. **end**

Algorithm 4:

1. **begin** Algorithm 4
2. $F = Edge(G)$
3. $F = F \oplus B$
4. Set N
5. **for** i=1 **to** N **begin**
6. $C \leftarrow rnd(F)$
7. $c_i \leftarrow 0$
8. **endfor**
9. $F \setminus C \leftarrow maxreal$
10. **repeat** EDTOCS
11. $\forall s: C \leftarrow m_s$
12. **until** $\{m_s\} = \emptyset$
13. **end**

The control point image is Huffman-coded both for the relative position of points from each other and for their gray values. The signal-to-noise ratio (SNR) is given by the following equation:

$$SNR = -10log_{10}\frac{1}{512}\sum_{j=0}^{511}\sum_{k=0}^{511}\frac{(f(j,k)-\hat{f}(j,k))^2}{255^2} \tag{3.6}$$

where $f(j,k)$ and $\hat{f}(j,k)$ are the original and reconstructed pixel values, respectively.

The decompression is performed using the 8 kernels's method described in [Vep91], [Toi93a] and [Toi93b] or using Delaunay triangulation. The gray value of each point inside a Delaunay triangle is given by the equation:

$$p = d_A p_A + d_B p_B(1 - d_A - d_B)p_C, \tag{3.7}$$

where p_A, p_B p_C and denote the gray values of the 3 triangle points and d_A, d_B and $1-d_A-d_B$ denote the normalized Euclidean distances from the pixel to the triangle corner points, i.e. the control points. The new pixel value p is rounded to the nearest integer.

3.3 Test Results of the New Algorithms

All the figures which are referred to in this chapter are found in Chapter 6: Appendix. Figure 6.1 shows the original Leena image of size 512 × 512 × 8 bits. Figure 6.2 depicts an image which has been compressed using algorithm 2 and decompressed using the 8 kernels' method. In this image, the control points were allowed only to be in even positions resulting in a substantially low kB value for positional information, 1.89 kB. See Table 3.1. Figure 6.3 is compressed using algorithm 1 and decompressed using Delaunay triangulation among the control points. In this image, the control points are free to take both even and odd positions yielding a high value for positional information, 8.08 kB. Visually this image looks better than the previous one, which is also confirmed by the corresponding signal-to-noise ratios. If the control points are forced to be only in even positions of the image buffer, a compression ratio of 1:26 is achieved with 6.83 KB for positional information and 2.98 kB for gray values. Figure 6.4 depicts an image which has been compressed using the Discrete Cosine Transform (DCT) and Huffman coding. A normalization matrix of size 4 × 4 is used. Figure 6.5 shows a DCT image with an 8 × 8 normalization matrix and a compression ratio 1 : 19. In Figure 6.6 the ratio is 1 : 30.

Name	Ratio	Points	Pos kB	Value kB	SNR dB
Fig. 6.2	1:18	11100	1.89	12.56	22.49
Fig. 6.3	1:17/1:26	11495	8.08/6.83	7.19/2.98	25.05
Fig. 6.4	1:16				26.51
Fig. 6.5	1:19				29.77
Fig. 6.6	1:30				27.9

Table 3.1. The obtained data for the Leena image.

Figure 6.7 shows the original peppers image of size $512 \times 512 \times 8$ bits. Figure 6.8 depicts an image which is compressed using the algorithm 1 and decompressed using the 8 kernels' method. In Figure 6.9, the same control point image is decompressed using Delaunay triangulation. In both Figure 6.7 and 6.8 the control points are allowed to to take both even and odd positions. Figure 6.10 depicts an image which has been compressed using the Discrete Cosine Transform (DCT) and Huffman coding. A normalization matrix of size 4×4 is used. Figure 6.11 shows a DCT image with a 8×8 normalization matrix. Figures 6.12 - 6.16 illustrate the same methods for the goldhill image and Figures 6.17 - 6.21 for the airplane image. Figure 6.22 shows the same Delaunay decompressed image as Fig. 6.3 with a compression ratio of $1 : 26$ with even positions of the control points. It can be compared to a DCT image with a 4×4 normalization matrix in Fig. 6.23 and to the standard JPEG DCT image in Fig. 6.24, which has a signal-to-noise ratio 34.69 dB. They both have a compression ratio $1 : 16$. Figure 6.25 shows a JPEG DCT image with a ratio $1 : 26$. Figures 6.26 and 6.27 depict the airplane image.

Name	Ratio	Points	Pos kB	Value kB	SNR dB
Fig. 6.8	1:16	11258	8.04	8.08	24.63
Fig. 6.9	1:16	11258	8.04	8.08	27.19
Fig. 6.10	1:17				26.82
Fig. 6.11	1:19				29.46

Table 3.2. The obtained data for the peppers image.

Name	Ratio	Points	Pos kB	Value kB	SNR dB
Fig. 6.13	1:15	11258	8.04	8.08	24.51
Fig. 6.14	1:16	11671	8.21	7.47	25.32
Fig. 6.15	1:16				17.06
Fig. 6.16	1:19				28.78

Table 3.3. The obtained data for the goldhill image.

Name	Ratio	Points	Pos kB	Value kB	SNR dB
Fig. 6.18	1:17	10736	7.20	8.00	22.21
Fig. 6.19	1:17	10736	7.20	8.00	24.68
Fig. 6.20	1:16				26.60
Fig. 6.21	1:18				28.02

Table 3.4. The obtained data for the airplane image.

To conclude, features of the proposed compression methods can be summarized as follows:

1. The best compression results are achieved using the algorithms 1 and 2 described above.

2. The 8 kernels' decompression method gives the best results with algorithm 2.

3. The Delaunay triangulation decompression scheme gives the best results with algorithm 1. This is due to the fact that a sparse distribution of control points given by algorithm 2 and decompressed by Delaunay triangulation will results in a "mosaic look".

4. The overall performance of the proposed methods is of the same category as the DCT method with a 4×4 normalization matrix. In rather smooth areas where the pixel values change slowly these methods give better results than the DCT method with a 4×4 normalization matrix. This is clearly visible in the smooth areas of the test images. See images 6.22, 6.23, 8.26 and 6.27. In areas with rapid change, i.e. high frequency areas and edges, the DCT method is better.

5. The JPEG DCT method gives a somewhat better reconstruction quality than the methods presented in this thesis, specially near the edges. See figures 6.24 and 6.25.

The limitations and restrictions of these compression algorithms can be summarized as follows:

- The compression algorithms presented in this thesis are not suitable for noisy images, because noise points will results in false distance values for DTOCS and thereby deviating the positions of nearby control points from their optimum.

- Images with only a few gray levels are problematic. For thin edges which are 1-2 pixels wide, the results are rather poor because the nearest control point in some direction may be far away on the opposite side of a smooth region. Therefore, for such images, the algorithms must be modified so that adequate thickness for the edges is achieved.

3.4 Complexity of the New Algorithms

Let N_G denote the number pixels in the gray-level image. b, d, e and f denote constant scalars. Let other definitions be as in Chapter 3.2. An upper bound for the time complexity of algorithm 1 can be derived as follows.

Algorithm 1:

Lines:	Time
2:	$O(1)$
3:	$O(1)$
4-7:	$O(2N) = O(N)$
8:	$O(N_G - N)$
9-10:	$O(max(aN_G, aN_G, aN_G)) = O(N_G)$
11-14:	$O(3a)$
15:	Huffman is left out
16-21:	$O(3am) = O(am)$

Let a be a repetition counter, a scalar. Since $m = bN_G$ and $N < N_G$, it follows that an upper bound for the total number of execution steps is then

$$O(max(1, 1, N, N_G - N, N_G, 3, abN_G)) = O(N_G) \tag{3.8}$$

Algorithm 2:

Lines:	Time
2:	$O(1)$
3:	$O(1)$
4-7:	$O(2N) = O(N)$
8:	$O(N_G - N)$
9-10:	$O(max(aN_G, aN_G, aN_G)) = O(N_G)$
11-14:	$O(3a)$
15:	Huffman is left out
16-21:	$O(4am2n$ = O(am)

Let a be a repetition counter, a scalar. Since $am = abN_G$, and $N < N_G$, it follows that an upper bound for the total number of execution steps is then

$$O(max(1, 1, N, N_G - N, N_G, 3, abN_G)) = O(N_G) \tag{3.9}$$

Algorithm 3:

Lines:	Time
2:	$O(N_G)$
3:	$O(N_G)$
4:	$O(1)$
5-8:	$O(N)$ operations
9:	$O(N_G - N)$
10-12:	$O(max(aN_G, aN_G, aeN_G, dN_G)) = O(N_G)$

Let a be a repetition counter, a scalar. It follows that

$$O(max(N_G, N_G, 1, N, N_G - N, N_G)) = O(N_G) \tag{3.10}$$

Algorithm 4:

Lines:	Time
2:	$O(N_G)$
3:	$O(N_G)$
4:	$O(1)$
5-8:	$O(N)$ operations
9:	$O(N_G - N)$
10-12:	$O(max(aN_G, aN_G, aeN_G, dN_G)) = O(N_G)$

Let a be a repetition counter, a scalar. It follows that

$$O(max(N_G, N_G, 1, N, N_G - N, N_G)) = O(N_G) \tag{3.11}$$

The CPU time tests for the compression algorithms were run on a SUN 5 RISC workstation using the UNIX *time* command. Table 3.5 shows the CPU time of compression algorithms 1 and 2 versus threshold and number of control points. The tests point out that the CPU time of these algorithms is about constant. The compression ratio, i.e. the number of control points, does not have much effect on the CPU time. Table 3.6 shows the CPU times of algorithms 3 and 4 versus distance constant and number of control points. Again, the CPU times remain about constant.

Name	Threshold	Image size	Number of control points	CPU time in sec.
Algorithm 1	20	512x512	15940	0.49
Algorithm 1	30	512x512	10266	0.40
Algorithm 1	40	512x512	7036	0.49
Algorithm 1	50	512x512	5215	0.41
Algorithm 1	60	512x512	3968	0.44
Algorithm 2	20	512x512	13013	0.51
Algorithm 2	30	512x512	9090	0.40
Algorithm 2	40	512x512	6919	0.48
Algorithm 2	50	512x512	5321	0.41
Algorithm 2	60	512x512	4440	0.43

Table 3.5. CPU times for the compression algorithms 1 and 2.

Name	Dist.const.	Image size	Number of control points	CPU time in sec.
Algorithm 3	30	512x512	12619	0.46
Algorithm 3	40	512x512	11722	0.44
Algorithm 3	50	512x512	11186	0.39
Algorithm 3	60	512x512	10743	0.42
Algorithm 3	80	512x512	9738	0.46
Algorithm 3	100	512x512	8501	0.43
Algorithm 3	120	512x512	7157	0.42
Algorithm 4	100	512x512	13377	0.50
Algorithm 4	110	512x512	11623	0.50
Algorithm 4	120	512x512	10018	0.52
Algorithm 4	130	512x512	8281	0.47
Algorithm 4	140	512x512	6066	0.44

Table 3.6. CPU times for the compression algorithms 3 and 4.

Chapter 4

Discussion

In this thesis, two new distance transforms for gray-level images are presented. As a new application for distance transforms, the new transforms are applied to grey-level image compression.

The new distance transforms are both new extensions of the well-known distance transform algorithm developed by Rosenfeld, Pfaltz and Laÿ. With a little modification their algorithm, which calculates a distance transform on binary images with a chosen kernel, has been made to calculate a chessboard-like distance transform with integer numbers (DTOCS) and a weighted real number distance transform (EDTOCS) on gray-level images. Both distance transforms, the DTOCS and EDTOCS, require only two passes over the gray-level image and are extremely simple to implement. Only two image buffers are needed: The original gray-level image and the binary image which defines the region(s) of calculation. No other image buffers are needed even if more than one iteration round is performed. For large neighborhoods and complicated images the two-pass distance algorithms have to be applied to the image more than once, typically some 3-10 times. Different types of kernels can be adopted. It is important to notice that no other existing transform calculates the same kind of distance map as the DTOCS and EDTOCS do. The difference between the DTOCS map and the distance transform for the GRAYMAT [Lev70] and the gray-weighted distance transform [Rut68] is shown. The EDTOCS differs from the DTOCS by calculating the propagating weights inside a given kernel in a different way. The EDTOCS propagates local Euclidean subdistances, whereas in the DTOCS the propagating distances are local chessboard distances added to integer gray value differences. The time complexity of both the DTOCS and EDTOCS is $O(N)$, where N is the number of pixels of the gray-level image.

The DTOCS and EDTOCS fulfill the basic requirements for morphological algorithms presented in [Vin91a]. These requirements are speed, accuracy and flexibility.

The DTOCS and EDTOCS algorithms can easily be applied to 5×5 and 7×7 kernels. Now the minimal paths have to be calculated first before the DTOCS and EDTOCS algorithms can be applied. This is because, contrary to binary cases, the minimal distances from the points which are farther away from pixel p than $N_8(p)$ are not known in forehand. The DTOCS and EDTOCS algorithms can easily be converted to 3-pass and 4-pass versions. Some preliminary tests were made with a 4-pass DTOCS algorithm with the kernels presented

in [Pip87], but the results were not better than those obtained with a 2-pass algorithm presented in this thesis. The sequential two-pass algorithms of DTOCS and EDTOCS can quite easily be converted to parallel versions in the same manner as presented by Borgefors [Bor86]. This remains to be done in the future, as well as a thorough analysis of 3- and 4-pass versions of the algorithms.

Commonly distance transforms are used for feature extraction in pattern recognition and learning. Their use in image compression is very rare. In this thesis, as a new application area, the DTOCS and EDTOCS were applied to image compression. Four compression algorithms are presented and the results are compared to conventional DCT images and JPEG images. It is quite obvious that the biggest problems of the compression algorithms lie near the edges of the images. To improve the quality of the reconstructed images it would be better to have more control points along the curve segments and to have more parallel control point curves at the edges. This should be done somehow independent of the number of curve segments and control points at the smooth areas. Otherwise the total number of points will be too high from the compression ratio point of view.

As shown both analytically and experimentally in this thesis, the time complexity of the new compression algorithms is independent of the number of control points, i.e. the compression ratio. It is approximately $O(N)$, where N is the number of pixels in the gray-level image.

Regarding the criteria for choosing a compression technique, the algorithms presented in this thesis could best be used in applications where a rapid flashing image is made more accurate as time goes by. First, a modest number of control points is sent just to make the subject of the image visible to the viewer, and more control points are sent as soon as possible so that the decompressed image is enhanced iteratively. These methods could be applied to transferring time-varying images over a communications channel. It satisfies to send only control points of those areas that have changed compared to the previous image frame. Also, some applications require that an image undergoes the compression-decompression cycle many times. For example, an image may be compressed and transmitted to a destination where it is decompressed and viewed. An operator may alter a small portion of the image and then compress the whole image and send it to another destination where this process is further repeated. In such applications, it is essential that the repeated compression and decompression of the unaltered portions does not result in any additional degradation beyond the first stage of compression. With the methods presented in this thesis it is possible to alter portions of the image of arbitrary size so that the above mentioned cycle only affects that portion and does not result in any degradation in other parts of the image.

Furthermore, the new transforms can be used in feature extraction in cases where conventional methods based on binarizing the gray-level images would give no distinction between the pattern classes.

To conclude, the contributions of this thesis can be summarized as follows:

- to present a new and fast integer distance transform for gray-level images (the DTOCS), and a new and fast real number distance transform for gray-level images (the EDTOCS)

- to present, as a new application for gray-level distance transforms, image compression

algorithms for gray-level images based on the DTOCS and the EDTOCS.

Chapter 5

Summary of the Publications

This thesis consists of six publications, two articles in an international journal and four conference papers.

Publication 1 presents a new image compression scheme based on the Distance Transform on Curved Space (DTOCS). In this method, the curved distance produced by the DTOCS is compared to the direct chessboard distance obtained by considering the set X in which the calculation is performed a binary image. In regions where the difference between the distances exceeds a predefined threshold, a boundary curve is formed. Along the curve, a one-dimensionsl DTOCS is calculated and compared to direct distance. Again, if the difference between the distances exceeds the threshold, a control point is put on the curve. Also a new surface interpolation scheme is presented in this paper. It uses 8 kernels, or 8 structuring elements, corresponding to the 8 directions on the image plane.

Publication 1 is based on paper Pekka J. Toivanen: "Fast Surface Model Generation Using Distance Function on Curved Space", which was accepted for publication in SPIE/SPSE's Symposium on Electronic Imaging Science & Technology, San Jose, USA, 24 February - 1 March 1991. Paper number in advance program is 1451-27. The author of this thesis wrote the extended abstract and made the tests based on which it was accepted. The mathematical formulation of the skeleton defined by the DTOCS was developed and written by the author of this thesis. The algorithm to calculate the DTOCS was developed together with Dr. Ari Vepsäläinen. The computer program to calculate the DTOCS was written by the author of this thesis. The compression algorithm which utilizes the DTOCS was developed together with Dr. Ari Vepsäläinen.

Publication 2 presents a new distance transform, called the Distance Transform on Curved Space (DTOCS) and a sequential two-pass algorithm to calculate it. The introduced transform is used to select control points, i.e. points that are considered fundamental for the reconstruction of the image, from a given gray-level image. The new distance transform requires a gray-level image and a binary image to determine the region in which the distance calculation is performed. The maximas in the distance image are directly selected as new control points and the new distance image is calculated for the set from which the control points have been excluded. This paper also introduces a parameter α which controls the

amount in which the curvature is taken into account in the two-pass algorithm. The images in this paper are decompressed using a method which applies 8 neighboring points and calculates first the MD-skeleton among the control points and then triangulates among the control and MD-skeleton points. The value given to the new point is the weighted mean of the three triangel points, i.e. 1 control point and 2 MD-skeleton points.

Publication 3 deals with the selection of control points on the threshold boundaries in the image. This paper introduces neighborhood masks of different sizes moving along the boundary curves given by the difference between the DTOCS and the binary image chessboard distance. The idea is to avoid control point boundaries being too near each other. The size of the mask needed to achieve this goal is dependent on the image. Specially at low numbers of control points, i.e. at high compression ratios, the improvement is clear compared to the method presented in Publication 2. The signal-to-noise ratios are better than in Publication 2 and also visually the images look better.

Publication 4 presents the same image compression algorithm as Publication 1 and introduces smoothing to the original gray-level images before comression. The smoothing is done with a 5×5 averaging kernel. The signal-to-noise ratios are compared between images with smoothing and without it. Also the advantages of smoothing are discussed with example images. The obtained control point images are decompressed using the 8 kernel's method introduced in Publication 1, which uses 8 kernels, or structuring elements, corresponding to the 8 directions in the image plane.

Publication 5 presents a new distance transform for gray-level images, called the Euclidean Distance Transform on Curved Space (EDTOCS). It calculates a weighted real number distance map for an arbitrary gray-level image of arbitrary size. Every pixel holds a value corresponding to the length of the shortest discrete 8-path to the nearest point in the background. The area in which the transform is performed can consist of several disjoint regions. Some analytical results are derived regarding the behaviour of the parameter α. Proofs for two theorems are given. The EDTOCS is also applied to image compression using a similar algorithm to the one presented in Publication 2. The results were approximately of the same quality as in Publication 2.

Publication 6 presents a new distance transforms for gray-level images. It is called the Distance Transform on Curved Space (DTOCS) and it performs the distance calculation with integer numbers and gives a distance map, in which the value of every pixel is the length of the shortest path to the nearest background pixel. Along this path, each subpath between adjacent pixels is calculated as the gray-level difference between the pixels + 1. The area in which the transform is calculated may consist of several disjoint regions.

Analytical results for DTOCS are derived regarding the behaviour of the curvature parameter α in the algorithm formulas. The effect of applying kernels of different shapes and sizes are discussed. Furthermore, the convergence properties of the transform are analyzed. The convergence analysis is performed on a real world image of different sizes showing that the transform algorithm converges to the correct distance map with respect to its definition. The computational complexity of the transform is dicussed.

Errata

In **Publications 1, 2 and 3** Definition 2 says: ...The DTOCS -skeleton of an Euclidean set $X : R^{n-1}$... It should be: ...The DTOCS -skeleton of an Euclidean set $X \in R^{n-1}$...

In **Publications 1,2 and 3** Definition 3 says: $SK(X) = \cup S_n(X)$. It should be: $SK(X) = \cup_{n=0}^{N} S_n(X)$.

In **Publication 1** on page 289 it says:
$da = \alpha|F(e) - F^*(a)|$
$db = \alpha|F(e) - F^*(b)|$
$dc = \alpha|F(e) - F^*(c)|$
$dd = \alpha|F(e) - F^*(d)|$

$df = \alpha|F(e) - F^*(f)|$
$dg = \alpha|F(e) - F^*(g)|$
$dh = \alpha|F(e) - F^*(h)|$
$dk = \alpha|F(e) - F^*(k)|$

It should be:
$da = \alpha|G(e) - G^*(a)|$
$db = \alpha|G(e) - G^*(b)|$
$dc = \alpha|G(e) - G^*(c)|$
$dd = \alpha|G(e) - G^*(d)|$

$df = \alpha|G(e) - G^*(f)|$
$dg = \alpha|G(e) - G^*(g)|$
$dh = \alpha|G(e) - G^*(h)|$
$dk = \alpha|G(e) - G^*(k)|,$

where $G(a), G(b), ...$ denote the pixels in the kernel area in the original gray-level image $G(x)$. This notational correction applies also to **Publication 3** on pages 2.2 - 6.2 and 2.2 - 6.3 and to **Publication 4** on page 477.

In **Publication 2** it says at the end of page 879 and beginning of page 880: The value given to the new point is the weighted mean of the 3 triangle points (i.e. control points). It should be: The value given to the new point is the weighted mean of the 3 triangle points (i.e. 1 control point and 2 MD-skeleton points).

In **Publication 4** it says starting from the end of page 482: It gives visually better images than Delaunay triangulation, as shown in Vepsäläinen and Toivanen (1991). It should be: The obtained images are approximately of the same quality as Delaunay triangulated images. Some parts are better, specially edges and regions of high frequency texture, and some parts are worse, typically regions of low frequencies.

In **Publication 5** in the proof of Proposition 1 it says:
$min[\alpha\sqrt{(G(e) - G(a))^2} + 2 + ...]$ and $min[\alpha\sqrt{(G(e) - G(a))^2} + 1 + ...]$.

It should be:
$min[a\sqrt{(G(e)-G(a))^2+2}+...$ and $min[a\sqrt{(G(e)-G(a))^2+1}+....$

In the same proof it says:
Since 255 is the maxint, and maxint + 1 = 0, it follows that ...
It should be: Since every term has maxint added to the square root, it has no effect in the order of magnitude of the terms, and can therefore be omitted from the equations.

In **Publication 5** in the Abstract it says:
It gives the minimum of all the possible exact Euclidean distances for every point measured from the background.
It should be: It gives the minimum of all the possible piecewise Euclidean distances of the discrete 8-paths for every point measured from the background.

In **Publication 5** on page 984 it says:
1) It gives the exact Euclidean distance transform over the gray-value image.
It should be: 1) It gives a weighted distance transform over the gray-value image, in which every minimal path is a discrete 8-path and is piecewise Euclidean.

Bibliography

[Ahu78] N. Ahuja, L.S. Davis, D.L. Milgram and A. Rosenfeld, "Piecewise Approximation of Pictures Using Maximal Neighborhoods", *IEEE Trans. Comput.*, Vol. C-27, pp. 375-379, 1978.

[Ara88] Y. Arai, T. Agui and M. Nakajima, "A fast DCT-SQ scheme for images", Trans. IEICE, E-71, pp. 1095-1097, 1988.

[Arc78] C. Arcelli and G. Sanniti di Baja, "On the Sequential Approach to Medial Line Transformation", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-8, 1978, pp. 139-144.

[Arc81] C. Arcelli, L. P. Cordella and S. Levialdi, "From Local Maxima to Connected Skeletons", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 3, No. 2, pp. 134-143, March 1981.

[Arc89] C. Arcelli and G. Sanniti di Baja, "A One-Pass Two-Operation Process to Detect the Skeletal Pixels on the 4-Distance Transform, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 11, 1989, pp. 411-414.

[Bab76] I. Babuska and A. K. Aziz, "On the angle condition in the finite element method", *SIAM J. Numer. Anal.*, 13, 1, 1976, pp. 214-226.

[Bar77] R. E. Barnhill, "Representation and approximation of surfaces, *Mathematical Software III (J.R. Rice Eds.)*, Academic Press, San Diego, CA, 1977, pp.69-120.

[Bay90] D. M. Baylon and J. S. Lim, "Transform/subband analysis and synthesis of signals", MIT Research Laboratory of Electronics Technical Report, June 1990.

[Ber84] G. Bertrand, "Skeletons in Derived Grids", *Proceedings of the Seventh International Conference on Pattern Recognition*, Montreal, 1984, pp. 326-329.

[Beu90] S. Beucher and L. Vincent, "Introduction aux outils mophologiques de segmentation", in *Traitement d'images en microscopie à balayage et en microanalyse par sonde électronique*, ANRT ed., Paris, pp. F1-F43, March 1990.

[Blu62] H. Blum, "An Associative Machine for Dealing with the Visual Field and Some of its Biological Implications", in E. E. Bernard and M. R. Kare (Eds.), *Biological Prototypes and Synthetic Systems*, Vol. 1, Plenum Press, New York, 1962.

[Boi84] J. D. Boissonnat, Geometric Structures for Three Dimensional Shape Representation, ACM Trans. Graphics 3, 4, 1984, pp. 266-286.

[Boi88] J. D. Boissonnat, Shape Reconstruction from Planar Cross Sections, *Computer Vision, Graphics and Image Processing*, 44, 1988, pp. 1-29.

[Bor83] G. Borgefors, "Chamfering: A Fast Method for Obtaining Approximations of the Euclidean Distance in N Dimensions, *3rd Scandinavian Conference on Image Analysis*, Copenhagen, Denmark, pp. 250-255, 1983.

[Bor84] G. Borgefors, "Distance Transformations in Arbitrary Dimensions", *Computer Vision, Graphics and Image Processing* 27, pp. 321-345, 1984.

[Bor86] G. Borgefors, "Distance Transformations in Digital Images", *Computer Vision, Graphics and Image Processing*, Vol. 34, pp. 344-371, 1986.

[Bor91] G. Borgefors, I. Ragnemalm, G. Sanniti di Baja, "The Euclidean Distance Transform: Finding the Local Maxima and Reconstructing the Shape", *in Proceedings of the 7th Scandinavian Conference on Image Analysis (SCIA91)*, Aalborg, Denmark, pp. 974-981, 1991.

[Bru91] E. Bruzzone, G. Garibotto and F. Mangili, Three-dimensional Surface Reconstruction Using Delaunay Triangulation in the Image Plane, Proceedings, International Workshop on Visual Form, Capri, May 1991.

[Bur83] P.J. Burt and E.H. Adelson, "The Laplacian Pyramid as a Compact Image Code", IEEE Trans. Commun., pp. 532-540, 1983.

[Cla85] R. J. Clarke, *Transform Coding of Images*, Academic Press, London, 1985.

[Che85] L. P. Chew and R. L. Drysdale, "Voronoi diagrams based on convex distance functions", Proceedings of the 1st Symposium on Computational Geometry, Baltimore 1985, pp. 235-244.

[Che86] L. P. Chew, "There is a planar graph almost as good as the complete graph", Proceedings of the Second Annual Symposium on Computational Geometry, Yorktown Heights 1986, pp. 169-177.

[Dan80] P. E. Danielsson, "Euclidean Distance Mapping", Computer Graphics and Image Processing 14, pp. 227-248, 1980.

[DeV92] R. A. DeVore, B. Jawerth and B. J. Lucier, "Image Compression Through Wavelet Transform Coding", *IEEE Transactions in Information Theory*, Vol. 38, No. 2, March 1992.

[Gia88] C. R. Giardina and E. R. Dougherty, "Morphological Methods in Image and Signal Processing", Prentice-Hall, Englewood Cliffs, NJ, USA, 1988.

[Ede87] H. Edelsbrunner, "Algorithms in Combinatorial Geometry", Springer-Verlag, New York, 1987.

[Fei90] E. Feig, "A fast scaled-DCT algorithm", in Proc SPIE Image Processing Algorithms and Techniques, 1244, 2-13, 1990.

[Flo89] L. De Floriani, "A pyramidal data structure for triangle-based surface description", *IEEE Comput. Graphics Appl.*, Mar., 1989.

[Flo92] "An on-line algorithm for constrained Delaunay triangulation", *CVGIP: Graphical Models and Image Processing*, Vol. 54, No 3, July 1992, pp. 290-300.

[Fre79] "A. P. French, ed., *Einstein: A Centenary Volume.*, Heinemann, London, 1979.

[Gol69] M. Golay, "Hexagonal Pattern Transforms", Proc. IEEE Trans. on Comp., Vol. C18, No. 8, August 1969.

[Gol92] L. Goldfarb, "What is distance and why do we need the metric model for pattern learning?", *Pattern Recognition*, Vol. 25, No. 4, pp. 431-438, 1992.

[Gon87] R. C. Gonzalez and P. Wintz, *Digital Image Processing*, 2nd Edition, Addison-Wesley, Reading, MA, 1987.

[Gon88] C. A. Gonzales and J. L. Mitchell, "A note on DCT algorithms with low multiplicative complexity", JPEG-150, 1988.

[Hab71] A. Habibi, "Comparison of nth-order DPCM encoder with linear transformations and block quantization techniques"; IEEE Trans. Commun. Tech., COM-19(6), 948-956, 1971.

[Har92] R. M. Haralick and L. Shapiro, "Computer and Robot Vision", Vol. 1. (Electrical and Computer Engineering Ser.). 608p. (ISBN 0-201-10877-1). Addison-Wesley Publishing Company, 1992.

[Hil68] J. Hilditch, "An Application of Graph Theory in Pattern Recognition", in D. Michie (Ed.), *Machine Intelligence*, Vol. 3, Edinburgh University Press, Edinburgh, 1968, pp. 325-347.

[Hil69] J. Hilditch, "Linear Skeletons from Square Cupboards", in B. Meltzer and D. Michie (Eds.), *Machine Intelligence*, Vol. 4, Edinburgh University Press, Edinburgh, 1969, pp. 403-420.

[Hsi87] T. R. Hsing, editor, Special Issue on Visual Communications and Image Processing, Opt. Eng., 26(7), 1987.

[Hsi89] T. R. Hsing and K.-H. Tzou, editors, Special Issue on Visual Communications and Image Processing, Opt. Eng., 28(7), 1989.

[IEE80] A. N. Netravali, Guest Ed., Proc. IEEE (Special Issue on Digital Encoding of Graphics), Vol. 68, pp. 757-944, July 1980.

[Iko85] A. Ikonomopoulos and M. Kunt, "High compression image coding via directional filtering" *Signal Processing*, Vol. 8, No. 2, pp. 179-203, April 1985.

[Jai81] A. K. Jain, "Image data compression: a review", Proc. IEEE, 69(3), 349-389, 1981.

[Jai89] A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[Jay84] N. S. Jayant and P. Noll, *Digital Coding of Waveforms*, Prentice-Hall, Englewood Cliffs, NJ, 1984.

[Jen92] J.-F. Jenq and S. Sahni, "Serial and Parallel Algorithms for the Medial Axis Transform", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 12, pp. 1218-1224, December 1992.

[Knu83] H. E. Knutsson, R. Wilson and G. H. Granlund, "Anisotropic nonstationary image estimation and its applications: Part I -Restoration of noisy images", *IEEE Trans. Commun.*, Vol. COM-31, pp. 388-397, March 1983.

[Koc83] M. Kocher and M. Kunt, "Image data compression by contour-texture modelling", in Proc. SPIE Int. Conf. on the Applications of Digital Image Processing, Geneva, April 1983, pp. 131-139.

[Kun85] M. Kunt, A. Ikonomopoulos and M. Kocher, "Second-generation image-coding techniques", Proc. IEEE, 73(4), 549-574, 1985.

[Kun87] M. Kunt, M. Benard and R. Leonardi, "Recent Results in High-Compression Image Coding", *IEEE Trans. on Circuits and Systems*, Vol. 34, No. 11, 1987, pp. 1306-1336.

[Lan84] C. Lantuejoul and F. Maisonneuve, "Geodesic Methods in Image Analysis", *Pattern Recognition*, Vol. 17, pp. 117-187, 1984.

[Lay84] B. Laÿ, "Descriptors of the Programs of the Software Package Morpholog., Ecole des Mines, Paris, 1984.

[Lee80] D. T. Lee and B, J, Schacter, "Two algorithms for constructing a Delaunay triangulation", *Internat. Comput. Inform. Sci.* 9, 3, 1980, pp. 219-242.

[Lee84] B. G. Lee, "A new algorithm to compute the discrete cosine transform", IEEE Trans. Acous., Speech, Signal Proc., ASSP-32(6), pp. 1243-1245, 1984.

[Lev70] G. Levi and U. Montanari, "A Grey-Weighted Skeleton", *Inform. Contr.*, Vol. 17, pp. 62-91, 1970.

[Lim78] J. O. Limb and C. B. Rubinstein, "On the design of quantizers for DPCM coders: a functional relationship between visibility, probability, and masking", IEEE Trans. Commun., COM-26(5), 573-578, 1978.

[Lim90] J. S. Lim, *Two-dimensional Signal and Image Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1990.

[Mar86] P. A. Maragos, "Morphological Skeleton Representation and Coding of Binary Images", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-34, No. 5, October 1986.

[Mat75] G. Matheron, "*Random sets and intergral geometry*, Wiley, New York, 1975.

[Mey86] F. Meyer, "Automatic Screening of Cytological Specimens", *Computer Vision, Graphics, and Image Processing*, Vol. 35, 1986, pp. 356-369.

[Mey89] F. Meyer, "Skeletons and Perceptual Graphs", *Signal Processing*, Vol. 16, No. 4, pp. 335-363, April 1989.

[Mon68] U. Montanari, "A method for obtaining skeletons using a quasi-euclidean distance", *J. ACM*, 15, pp. 600-624, 1968.

[Mon69] U. Montanari, "Continuous Skeletons from Digitized Images", *Journal of the Association for Computing Machinery*, Vol. 16, 1969, pp. 534-549.

[Mon87] A. Montanvert, Contribution au traitement de formes discrètes. Squelettes et codage par graphe de la ligne médiane, Ph.D. Thesis, Université Scientifique, Technologique et Médicale de Grenoble, France, October 1987.

[Mot70] J.C. Mott-Smith, "Medial Axis Transformations", in *Picture processing and Psychopictorics*, B.S. Lipkin and A. Rosenfeld Eds., New York:Academic, pp. 267-278, 1970.

[Mul92] J. C. Mullikin, "The Vector Distance Transform in Two and Three Dimensions", *CVGIP: Graphical Models and Image Processing*, Vol. 54, No. 6, pp. 526-535, November 1992.

[Mus79] H. G. Musmann, "Predictive image coding", in *Image Transmission Techniques*, W. K. Pratt ed., *Advances in Electronics and Electron Physics*, Supplement 12, 73-112, Academic Press, Orlando, FL, 1979.

[Nak78] Y. Nakagawa and A. Rosenfeld, "A Note on the Use of Local Min and Max Operations in Digital Picture Processing", *IEEE Trans. Syst., Man, Cybern.*, Vol. SMC-8, pp. 632-635, 1978.

[Net77] A. N. Netravali, "On quantizers for DPCM coding of picture signals", IEEE Trans. Info. Theory, IT-23(3), 360-370, 1977.

[Net80] A. N. Netravali and J. O. Limb, "Picture coding: A review", Proc. IEEE, 68(3), 366-406, 1980.

[Net85] A. N. Netravali and B. Prasada, editors, Special Issue on Visual Communication Systems, Proc. IEEE, 73(4), pp. 497-848, 1985.

[Net88] A. N. Netravali and B. G. Haskell, *Digital Pictures: Representation and Compression*, Plenum Press, New York, 1988.

[Nga84] K. N. Ngan, "Image display techniques using the cosine transform", *IEEE Trans. Acous., Speech, Signal Proc.*, ASSP-32(1), pp. 173-177, 1984.

[Pel81] S. Peleg and A. Rosenfeld, "A Min-Max Medial Axis Transformation", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-3, pp. 208-210, 1981.

[Pip87] J. Piper and E. Granum, "Computing Distance Transformations in Convex and Non-Convex Domains", Pattern Recognition 20 (6), pp. 599-615, 1987.

[Pra78] W.K. Pratt, *Digital Image Processing*, New York: Wiley, p. 729, 1978.

[Pre91] F. Preteux and N. Merlet, "New Concepts in Mathematical Morphology: The Topographical Distance Functions", P.D. Gader, E.R. Dougherty, Editors, Proc. SPIE 1568. pp. 66-77, 1991.

[Rab91] M. Rabbani and P. W. Jones, Digital Image Compression Techniques, SPIE Optical Engineering Press, 2nd printing, 1991.

[Rag90] I. Ragnemalm, "Contour processing distance transforms", *Progress in Image Analysis and Processing*, World Scientific, Singapore, pp. 204-212, 1990.

[Rag92a] I. Ragnemalm, "Fast Erosion and Dilation by Contour Processing and Thresholding of Distance Maps", Pattern Recognition Letters, Number 13, pp. 161-166, March 1992.

[Rag92b] I. Ragnemalm, "Neighborhoods for Distance Transformations Using Ordered Propagation", *CVGIP: Image Understanding*, Vol. 56, No. 3, pp. 399-409, November 1992.

[Rag93] I. Ragnemalm, "The Euclidean distance transform in arbitrary dimensions", *Pattern Recognition Letters*, Number 14, PATREC 1121, pp. 883-888, November 1993.

[Raj94] V. T. Rajan, "Optimality of the Delaunay Triangulation in R^d", *Discrete and Computational Geometry*, 12, pp. 189-202, 1994

[Rea85] T. C. Rearick, "Syntactical Methods for Improvement of the Medial Axis Transformation", *SPIE Applications of Artificial Intelligence II*, Vol. 548, 1985, pp. 110-115.

[Rob62] L. G. Roberts, "Picture coding using pseudo-random noise", IRE Trans. Informat. Theory, Vol. IT-8, pp. 145-154, Jan. 1962.

[Ros66b] A. Rosenfeld and J.L. Pfaltz, "Sequential Operations in Digital Picture Processing", *J. Ass. Comput. Mach.*, Vol. 13, pp.471-494, 1966.

[Ros68] A. Rosenfeld and J. Pfaltz, "Distance functions on digital pictures", *Pattern Recognition*, 1, No. 1, pp. 33-61, 1968.

[Ros69] "Picture Processing by Computer", p. 142, Academic Press, New York, 1969.

[Ros82] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, Vol. 1, Academic Press, Orlando, FL, 1982.

[Rut68] D. Rutovitz, "Data sructures for operations on digital images", *Pictorial Pattern Recognition* (G. C. Cheng, R. S. Ledley, D. K. Pollock and A. Rosenfeld eds.), pp. 105-133, Thompson Book, Washington, 1968.

[Rut78] D. Rutovitz, "Expanding picture components to natural density boundaries by propagation methods, The notions of fall-set and fall-distance.", *Proc. 4th International Joint Conference on Pattern Recognition*, Kyoto, Japan, Nov. 7-10, pp. 657-664, 1978.

[Sai94] T. Saito and J.-I. Toriwaki, "New Algorithms for Euclidean Distance Transformation of an n-Dimensional Digitized Picture with Applications", *Pattern Recognition*, Vol. 27, No. 11, pp. 1551-1565, 1994.

[Sch89] M. Schmitt, Des algorithmes morphologiques à l'intelligence artificielle, Ph.D. Thesis, School of Mines, Paris, February 1989.

[Ser82] J. Serra, *Image Analysis and Mathematical Morphology*, Academic Press, New York, 1982.

[Ser88] J. Serra, *Image Analysis and Mathematical Morphology*, Academic Press, London, 1989.

[Sha75] M. I. Shamos and D. Hoey, "Closest-point problems", Proc. 16th Ann. IEEE Symp. on Foundations of Computer Science, New York, 1975, pp. 151-162.

[Shi92] Y.-C. Shih, O. R. Mitchell, "A Mathematical Morphology Approach to Euclidean Distance Transformation" *IEEE Transactions on Image Processing*, Vol. 1, No. 2, pp. 197-204, April 1992.

[Sko89] M. M. Skolnick, L. Vincent and S. S. Wilson, Short Course Notes, SPIE's 1989 Symposium on Advances in Intelligent Robotics Systems with Visual Communications and Image Processing, SC4, Philadelphia, USA, Nov. 1989.

[Sri93] P. Sriram and M. W. Marcellin, "Image Coding Using Wavelet Transforms and Entyropy-Constrained Trellis Coded Quantization", Proc. International Conference on Acoustics, Speech and Signal Processing -93, Vol. 5, pp. 554-557, 1993.

[Toi92] P. J. Toivanen, "Fast Image Compression Using Distance Function on Curved Space", in Proceedings of the SPIE Visual Communications and Image Processing '92, Boston, USA, Vol. 1818, November 18-20, 1992.

[Toi93a] P. J. Toivanen: "Image Compression Using DTOCS and Neighborhood Masks along Threshold Boundaries", in *Proceedings of the IEEE Winter Workshop on Nonlinear Digital Signal Processing '93*, pp. 2.2-6.1 - 2.2-6.6, Tampere, Finland, January 17-20, 1993 .

[Toi93b] P. J. Toivanen, "Image Compression by Selecting Control Points Using Distance Function on Curved Space", *Pattern Recognition Letters 14*, PATREC 1070, pp. 475-482, North-Holland, June 1993.

[Toi94] P. J. Toivanen: "The Euclidean Distance Transform on Curved Space (EDTOCS) with Application to Image Compression", in Proceedings of the *7th European Signal Processing Conference (EUSIPCO-94)*, pp.983-986, Edinburgh, Scotland, U.K, September 13-16, 1994

[Toi95] P. J. Toivanen, "Convergence Properties of the Distance Transform on Curved Space (DTOCS)", in *Proceedings of the 1995 Finnish Signal Processing Symposium (FINSIG'95)*, Helsinki, Finland, June 2, 1995, pp. 75-79.

[Vep91] A. M. Vepsalainen and P. J. Toivanen, "Two New Image Compression Methods Utilizing Mathematical Morphology", in Proceedings of the SPIE Visual Communications and Image Processing '91, Boston, USA, 1606, November 11-13, 1991, pp. 282 - 293.

[Ver89] B. J. H. Verwer, P. W. Verbeck and S. T. Dekker, "An Efficient Uniform Cost Algorithm Applied to Distance Transforms", IEEE Trans. Pattern Anal. and Machine Intelligence 11, pp. 425-429, 1992.

[Ver90] P. W. Verbeck and B. J. H. Verwer, "Shading from shape, the eikonal equation solved by grey-weighted distance transform", *Pattern Recognition Letters 11*, pp. 681-690, October, 1990.

[Ver91] B. J. H. Verwer, "Local distances for distance transformations in two and three dimensions", *Pattern Recognition Letters*, 12, pp. 671-682, November 1991.

[Vet84] M. Vetterli and H. Nussbaumer, "Simple FFT and DCT algorithms with reduced number of operations", *Signal Processing*, 6, pp. 267-278, 1984.

[Vet85] M. Vetterli, "Fast 2-D discrete cosine transform", in Proc. ICASSP, pp. 1538-1541, 1985.

[Vin89] L. Vincent and S. Beucher, "The Morphological Approach to Segmentation: an Introduction", Technical Report CMM, School of Mines, Paris, July, 1989.

[Vin90] L. Vincent, *Algorithmes morphologiques à base de files d'attente et de lacets. Extension aux graphes.*, Ph.D. Thesis, School of Mines, Paris, May 1990.

[Vin91a] L. Vincent, "New Trends in Morphological Algorithms", in Proc. SPIE Nonlinear Image Processing II, 1451, pp.158-170, 1991.

[Vin91b] L. Vincent, "Efficient computation of various types of skeletons", in Proc. Vol. 1445 Image Processing, 1991, pp. 297-311.

[Vin91c] L. Vincent, "Exact Euclidean Distance Function by Chain Propagations", *Proc. IEEE*, pp. 520-525, 1991.

[Vli88] L. J. Van Vliet and B. J. H. Verwer, "A Contour Processing Method for Fast Binary Operations", Pattern Recognition Letters 7, pp. 27-36, 1988.

[Vos79] A. M. Vossepoel, A. W. M. Smeulders and K. Van der Broek, "DIODA: delineation and feature extraction of microscopical objects", *Comp. Prog. Biomedicine* 10, pp. 231-244, 1979.

[Wan79] S. Wang, A. Rosenfeld and A.Y. Wu, "A Medial Axis Transformation for Grayscale Pictures", Comput. Vis. Lab., Comput. Sci. Center, Univ. of Maryland, College Park, MD, *Tech. Rep. TR-843*, Dec., 1979.

[Wan81] S. Wang, A. Y. Wu and A. Rosenfeld, "Image Approximation from Gray Scale Medial Axes", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. PAMI-3, No. 6, Nov. 1981.

[Wan88] X. Wang and G. Bertrand, "An Algorithm fo a Generalized Distance Transformation Based on Minkowski Operators", *Proceedings of Ninth International Conference on Pattern Recognition*, Rome, pp. 1164-1168, 1988.

[Wil83] R. Wilson, H.E. Knutsson and G.H. Granlund, "Anisotropic Nonstationary Image Estimation and Its Applications: Part II -Predictive Image Coding", IEEE Trans. Commun., Vol. 31, pp. 398-406, 1983.

[Xia89] Y. Xia, "Skeletonization via the Realization of the Fire Front's Propagation and Extinction in Digital Binary Shapes", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 11, No. 10, pp. 1076-1086, October 1989.

[Yam84] H. Yamada, "Complete Euclidean Distance Transformation by Parallel Operation", *Proc. 7th Int. Conf. on Pattern Recognition*, Montreal, Canada, pp. 69-71, 1984.

[Ye88] Q.-Z. Ye, "The Signed Euclidean Distance Transform and Its Applications", in *Proc. 9th International Conference on Pattern Recognition*, pp. 495-499, 1988.

Chapter 6

Appendix

Figure 6.1. The original Leena image of size 512x512x8 bits.

Figure 6.3. Delaunay decompressed image. Compression ratio 1:17 or 1:26 with even positions.

Figure 6.5. DCT image. 8×8 matrix. Compression ratio 1:19.

Figure 6.2. 8 kernels' method. Compression ratio 1:18.

Figure 6.4. DCT image. 4×4 matrix. Compression ratio 1:16.

Figure 6.6. DCT image. 8×8 matrix. Compression ratio 1:30.

Figure 6.7. The original peppers image.

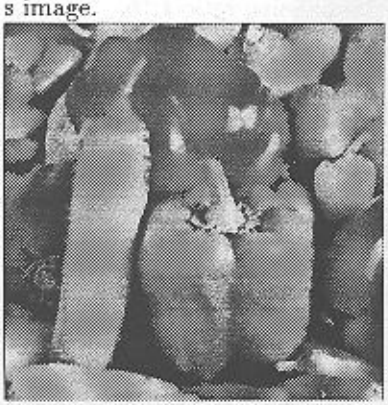

Figure 6.9. Delaunay decompressed image. Compression ratio 1:16.

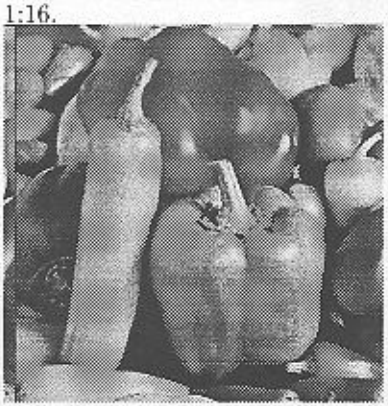

Figure 6.11. DCT image. 8×8 matrix. Compression ratio 1:19.

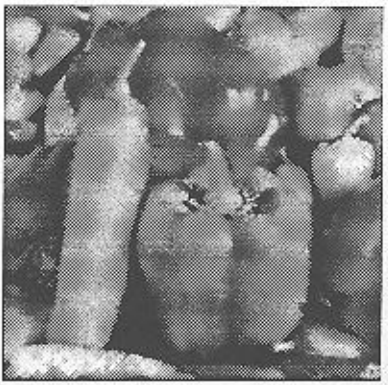

Figure 6.8. 8 kernels' method. Compression ratio 1:16.

Figure 6.10. DCT image. 4×4 matrix. Compression ratio 1:17.

Figure 6.12. The original Goldhill image.

Figure 6.14. Delaunay decompressed image. Compression ratio 1:16.

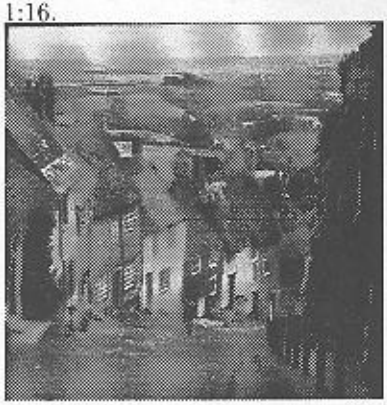

Figure 6.16. DCT image. 8×8 matrix. Compression ratio 1:18.

Figure 6.13. 8 kernels' method. Compression ratio 1:16.

Figure 6.15. DCT image. 4×4 matrix. Compression ratio 1:16.

Figure 6.17. The original airplane image.

Figure 6.19. Delaunay decompressed image. Compression ratio 1:17.

Figure 6.21. DCT image. 8×8 matrix. Compression ratio 1:18.

Figure 6.18. 8 kernels' method. Compression ratio 1:17.

Figure 6.20. DCT image. 4×4 matrix. Compression ratio 1:16.

Figure 6.22. Delaunay decompressed image. Compression ratio 1:17 or 1:26 with even positions.

Figure 6.23. DCT image. 4×4 matrix. Compression ratio 1:16.

Figure 6.24. JPEG DCT image. Compression ratio 1:16.

Figure 6.25. JPEG DCT image. Compression ratio 1:26.

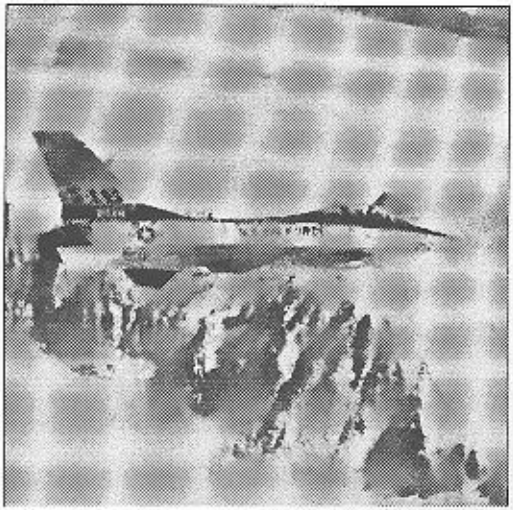

Figure 6.26. Delaunay decompressed image. Compression ratio 1:17.

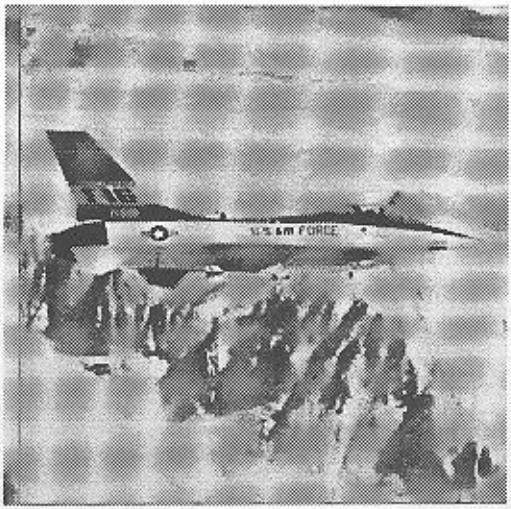

Figure 6.27. DCT image. 4×4 matrix. Compression ratio 1:16.