



Budapest University of Technology and Economics
Department of Mechatronics, Optics and Engineering Informatics

**ENGINEERING APPLICATION OF BIOLOGICAL MODELS;
Development of Entropy Based
Image Compression Methods**

PhD Thesis

Sándor Piros

Supervisor:

Péter Korondi PhD, DSc.

Budapest, 2014.

“Dedicated

*To those who seek evidence of
Nature’s universal Method of Creation
And those who find the story
Of inexhaustible interest.”*

Creation by Evolution

Frances Mason 1928

Contents

List of Figures	vii
List of Tables.....	x
GLOSSARY	xi
INTRODUCTION.....	1
Molecular Biology	1
Developmental Biology	2
Digital Signal Processing (DSP)	2
How Mechatronics is Related to Biology and Vice Versa?.....	2
1 BIOLOGICAL BACKGROUND	5
1.1 The Route from DNA to Protein (DNA-RNA-Protein)	5
1.2 DNA Replication.....	6
1.3 RNA Transcription	6
1.4 Protein Translation	7
2 COMPUTATIONAL METHODS	8
2.1 General Discrete Wavelet Transformation (DWT)	8
2.2 Second Generation Wavelet Transformation of 1D (one-dimensional) Signal	9
2.3 Spatial 2D System	10
2.4 Matlab Modeling and Simulation.....	11
3 UNIVERSAL CHROMOSOME NUMBERING SYSTEM FOR CELL LINEAGES, BACTERIA COLONIES ETC.	12
3.1 Cell Numbering System	12
3.2 Cell Lineage	13
3.3 Identification of DNA Strands	14
3.4 Binary Tree; Cell Numbering System	15
3.5 Correlation between Cell Lineage and Numbering	16
3.6 Conclusion of the Chapter	17
4 CELL DIVISION IN TIME AND SPACE	18
4.1 Cell Division as Inverse Wavelet Transformation	19
4.2 Cell Divisions in 2D (layered tissue of cells).....	21
4.3 Isometric Pixels	22
4.4 3D Image Transformation (Cell Division in Time and Space)	24
4.5 Conclusion of the Chapter	26
5 WAVELET REPRESENTATION OF CELL DIVISIONS AND DIFFERENTIATION.....	27
5.1 Classification of Cell Divisions.....	27

5.2	Symmetric Cell Divisions	27
5.3	Asymmetric Cell Divisions	27
5.4	Cell Division Tree Relation to Image Transformation.....	29
5.5	Cell Division Tree Relation to Image Pixel Numbering	31
5.6	Conclusion of the Chapter	31
6	ALGORITHM FOR DISCRETE DATA AND SIGNAL PROCESSING	33
6.1	Cell Division Tree Described as Inverse Wavelet Transformation.....	33
6.2	Wavelet Transformation (Haar Wavelet)	33
6.3	Cell Division Tree, Illustration of Cell Differentiation on Binary Tree.....	34
6.4	Predicting the Value of a Discrete Variable	35
6.5	Transformation	36
6.6	Inverse-Transformation	38
6.7	Practical Implementation of the Transformation.....	40
6.8	Comparing New Method to Existing (Haar) Transformation	43
6.1	Conclusion of the Chapter	46
7	ALGORITHM FOR DISCRETE DATA AND SIGNAL PROCESSING ENTROPY BASED HIERARCHICAL IMAGE COMPRESSION METHOD ...	47
7.1	Image Transformation before Applying Compression Method	47
7.2	Preparation: Image Transformation.....	50
7.3	Compression.....	51
7.4	Decompression (restoration)	54
7.5	Conclusion of the Chapter	55
8	CELL DIVISION AND DIFFERENTIATION ON THE GENE LEVEL AS ALGORITHM FOR LOGICAL DATA PROCESSING	57
8.1	Transformation (Coding and Decoding)	57
8.2	Biologically Inspired Algorithm for Bitmap Transformation	57
8.3	Analogy between Cell Differentiation and Image Transformation.....	59
8.4	Decoding an Image from Detail Coefficients.....	60
8.5	Coding	60
8.6	Compression.....	61
8.7	Predicting the Value of a Logical Variable	62
8.8	Conclusion of the Chapter	64
9	HOW TO COMPRESS A BINARY TREE LIKE BITMAP COMPRESSION METHOD FOR BINARY TREE LIKE BITMAPS	65
9.1	Introduction of an Internal Variable	65
9.2	Practical Realization of the Method	67

9.3	Multi-Valued Logical Variable	67
9.4	Coding, Decoding an Image	67
9.4.1	Option 1	68
9.4.2	Option 2	68
9.4.3	Option 3	69
9.4.4	Option 4	69
9.5	Introducing a New Type of Logical Variable	70
9.6	Conclusion of the Chapter	70
10	PRELIMINARY MODELING SPATIAL MOVEMENT OF aminoacyl-tRNA (aa-tRNA) MOLECULES IN THE CYTOPLASM.....	71
10.1	The use of <i>Escherichia coli</i> (<i>E. coli</i>) as a Model Organism.....	71
10.2	Method	72
10.3	Elastic Collision of Particles	73
10.3.1	Elastic Collision in 1 Dimension (1D)	74
10.3.2	Collision in 2 Dimensions (2D).....	75
10.3.3	Collision in 3 Dimensions (3D).....	77
10.4	Collision Detection.....	80
10.5	Flowchart	81
10.6	Simulation	84
10.7	Possible Physical Mechanisms of tRNA Preselection in the Cytoplasm of <i>Escherichia coli</i> Bacteria.....	88
10.7.1	Proposed Hypotheses on tRNA Preselection.	89
10.7.2	Transfer RNAs are Stored in or at the Ribosome.....	90
10.7.3	Signaling between Ribosome and tRNA.....	91
10.7.4	Transfer RNAs are Preselected Using a Different Mechanism.....	92
10.8	Conclusion of the Chapter	94
	Summary of the PhD Thesis.....	95
	Hungarian Summary of the PhD Thesis - Az értekezés összefoglalása.....	96
	CONTRIBUTIONS	97
	Theses in Hungarian - Tézisek.....	100
	EPILOGUE	104
	ACKNOWLEDGEMENTS	105
	BIBLIOGRAPHY	106

List of Figures

Figure 1.1 The route from DNA to protein	5
Figure 2.1 Development by cell divisions of a theoretical 2 dimensional sample organism	8
Figure 2.2 General DWT representation using cascading filter bank	9
Figure 2.3 Filtering progression in a 2 Dimensional system.....	10
Figure 2.4 An example of a hierarchical image transformation in 2D system.....	10
Figure 3.1 Embryonic development	12
Figure 3.2 <i>Caenorhabditis elegans</i> with 959 somatic cells	13
Figure 3.3 Cell lineage of <i>Caenorhabditis elegans</i> with 959 somatic cells	13
Figure 3.4 P0 was the zygote cell, yielding 8 descendant cells after the third cell division cycles	14
Figure 3.5 DNA strand identification.....	15
Figure 3.6 Replacing <i>A</i> and <i>T</i> letters by numeral <i>0</i> and <i>1</i>	16
Figure 3.7 Embryonic development by consecutive cell divisions	16
Figure 4.1 E. G. Conklin’s original caption, a remarkable fate map of the ascidian, published in 1905	18
Figure 4.2 Cleavage orientation planes (radial cleavages).....	19
Figure 4.3 Development of an organism from a single cell (spiral cleavages).....	20
Figure 4.4 Cleavage orientation planes (rotational)	20
Figure 4.5 Image transformation in 2D	21
Figure 4.6 Dividing cells	21
Figure 4.7 Example image development <i>x</i> and <i>y</i> direction consecutively	22
Figure 4.8 Isometric shape pixels (diamond shape arrangement)	23
Figure 4.9 Example image with isometric pixels	23
Figure 4.10 Filter design in 2D	24
Figure 4.11 Pixels are congruent to itself after rotated by 45°	24
Figure 4.12 Hierarchical transformation of a 3D signal.....	25
Figure 4.13 Possible downsampling method to maintain equal resolution in all directions	25
Figure 5.1 Symmetrical cell division	27
Figure 5.2 Asymmetric cell division	27
Figure 5.3 Nonconventional (abnormal) cell differentiation.....	28
Figure 5.4 Cell divisions follow a perfect binary tree structure.....	28
Figure 5.5 Cell divisions tree, considering cell differentiation	29
Figure 5.6 “Reverse cell division” representation with cascading filter bank	30

Figure 5.7 An example image of 4x4 pixels	31
Figure 5.8 Pixel numbers and the corresponding binary tree identification numbers.....	31
Figure 6.1 Matchup of wavelet transformation coefficients with dividing cells.....	33
Figure 6.2 Cell division binary tree.....	34
Figure 6.3 Cell differentiation	35
Figure 6.4 Example picture 256x256 pixels.....	40
Figure 6.5 Cell division type wavelet transform of Figure 6.4.	41
Figure 6.6 Inverse cell division based wavelet transform	41
Figure 6.7 Haar wavelet transform of the same image.....	42
Figure 6.8 Inverse Haar wavelet transform of the image	42
Figure 6.9 Histogram of the cell division and Haar wavelet.....	43
Figure 6.10 Sample color picture	44
Figure 6.11 Cell division wavelet transform of the sample image.....	44
Figure 6.12 Inverse cell division wavelet transform of the color image	45
Figure 6.13 Inverse Haar wavelet transform of the color picture	45
Figure 7.1 Entropy of gray scale images	47
Figure 7.2 Example for second generation wavelet transformation.....	48
Figure 7.3 Second generation wavelet transformation.	49
Figure 7.4 Detail coefficients of this bio-inspired wavelet transformation.....	49
Figure 7.5 Image transformation in 2D	50
Figure 7.6 Original image (Lena).....	50
Figure 7.7 Image transformation in 2D	50
Figure 7.8 Entropy values at the top of the tree	51
Figure 7.9 The end of a branch of the H entropy level binary tree	52
Figure 7.10 H entropy coefficients.....	52
Figure 7.11 H coefficients of the scaled image.....	53
Figure 7.12 D and E variables.....	53
Figure 7.13 D and E entropy difference coefficients	54
Figure 7.14 Scaled image of D and E coefficients	55
Figure 8.1 Expression of a particular gene after consecutive cell divisions and differentiations.....	58
Figure 8.2 Binary tree shape representation of cell differentiation	59
Figure 8.3 Detail coefficients of image transformation	59
Figure 8.4 Image to be compressed; size 256x256 pixels.....	61
Figure 8.5 Inverse transformation, reproducing the image of the zebra	62
Figure 8.6 Detail coefficients of the transformed image.....	63

Figure 8.7 Combined transformed image.....	63
Figure 9.1 Example: image of a zebra and its transform.....	65
Figure 9.2 H variable suggest the entropy level in the given branch of the binary tree	66
Figure 9.3 H entropy values of the image	66
Figure 9.4 d detail coefficient and H entropy coefficient is replaced by D multi-valued variable	67
Figure 10.1 The route from DNA to protein	71
Figure 10.2 shows two particles of masses, m_1 and m_2 , and velocities, v_1 and v_2 , before they collide (1-D collision).....	74
Figure 10.3 shows two particles of masses, m_1 and m_2 , and velocities, v_1' and v_2' , after they have collided (1-D collision)	74
Figure 10.4 shows two particles of masses, m_1 and m_2 , and velocities, v_1 and v_2 , before they collide (2-D collision).....	75
Figure 10.5 shows two particles of masses, m_1 and m_2 , and initial velocities, v_1 and v_2 , at the moment of collision (2-D collision)	75
Figure 10.6 shows the vector components of the velocities before collision.....	76
Figure 10.7 shows the vector components of the velocities after the collision.....	77
Figure 10.8 shows the schematic of two molecules before collision.	79
Figure 10.9 shows the schematic of two molecules at the moment of collision.	80
Figure 10.10 shows the schematic of two molecules virtually overlapping at the moment of collision.....	80
Figure 10.11 Simulation space contain a piece of the mRNA, 144 ribosome, 1600 tRNAs.....	84
Figure 10.12 Result of Matlab simulation.....	85
Figure 10.13 Ribosomal 30S subunit protein S7.....	89
Figure 10.14 Reading frame	90
Figure 10.15 Three sites of the ribosome	90
Figure 10.16 First hypothesis	91
Figure 10.17 Second hypothesis.....	91
Figure 10.18 Reading brackets.....	92
Figure 10.19 Third hypothesis; initiation.....	92
Figure 10.20 Third hypothesis; beginning of elongation	93
Figure 10.21 Third hypothesis; elongation in progress	93

List of Tables

Table 1 Information content of the genomes of different organisms	4
Table 1-1 The genetic code	7
Table 10-1 <i>E.coli</i> cell dimensions	73
Table 10-2 Flowchart	83
Table 10-3 Results of the simulations	86
Table 10-4 Comparison chart of possible explanations	87

GLOSSARY

Amino acid – an organic compound containing at least one amino group and one carboxyl group. In the 20 different amino acids that compose proteins, an amino group and carboxyl group are linked to a central carbon atom, to which a variable side chain is bound.

aa-tRNA – aminoacyl-tRNA, activated form of an amino acid, used in protein synthesis, consisting of an amino acid linked via a high energy ester bond to the 3'-hydroxyl group of a tRNA molecule.

Anticodon – sequence of three nucleotides in a tRNA that is complementary to a codon in an mRNA. During protein synthesis, base pairing between a codon and anticodon aligns the tRNA carrying the corresponding amino acid for addition to the growing peptide chain.

Apoptosis – programmed cell death in multicellular organisms.

Base pair – association of two complementary nucleotides in a DNA or RNA molecule stabilized by hydrogen bonding between their base components. Adenine pairs with thymine or uracil, and guanine pairs with cytosine.

Chromosome – in eukaryotes, the structural unit of the genetic material consisting of a single, linear double-stranded DNA molecule and associated proteins. In prokaryotes, a single, circular double-stranded DNA molecule constitutes the bulk of the genetic material.

Codon – sequence of three nucleotides in DNA or mRNA that specifies a particular amino acid during protein synthesis; also called triplet. Of the 64 possible codons, three are stop codons, which do not specify amino acids.

DNA – Deoxyribonucleic acid, long linear polymer, composed of four kinds of deoxyribose nucleotides, that is the carrier of genetic information. In its native state, DNA is a double helix of two antiparallel strands held together by hydrogen bonds between complementary purine and pyrimidine bases.

DNA methylation – is the addition of a methyl group to DNA nucleotides to alter gene expression in cells when they divide and differentiate from one cell type into another specific type.

DSP – Digital Signal Processing

DWT – Discrete Wavelet Transform

Elongation factor – one of a group of nonribosomal proteins required for continued translation of mRNA following initiation.

Enzyme – a biological macromolecule that acts as a catalyst. Most enzymes are proteins.

Eukaryotes – class of organisms, composed of one or more cells containing a membrane-enclosed nucleus and organelles, that constitutes one of the three distinct evolutionary lineages of modern-day organisms. Includes all organisms except viruses and prokaryotes.

Gene – physical and functional unit of heredity, which carries information from one generation to the next. In molecular terms, it is the entire DNA sequence – including exons, introns, and noncoding transcription-control regions – necessary for production of a functional protein or RNA.

Genome – total genetic information carried by a cell or organism.

Hydrogen bond – a noncovalent bond between an electronegative atom and a hydrogen atom covalently bonded to another electronegative atom. Particularly important in stabilizing the three-dimensional structure of proteins and formation of base pairs in nucleic acids.

Macromolecule – any large, usually polymeric molecule (e.g. a protein, nucleic acid, polysaccharide) with a molecular mass greater than a few thousand daltons.

mRNA – messenger RNA, the RNA that specifies the order of amino acids in a protein.

Nucleotide – a nucleoside with one or more phosphate groups linked via an ester bond to the sugar moiety. DNA and RNA are polymers of nucleotides.

Nucleus – large membrane-bounded organelle in eukaryotic cells that contains DNA organized into chromosomes; synthesis and processing of RNA and ribosome assembly occur in the nucleus.

Peptide bond – covalent bond that links adjacent amino acid residues in proteins; formed by a condensation reaction between the amino group of one amino acid and the carboxyl group of another with release of a water molecule.

Primary structure – in proteins, the linear arrangement (sequence) of amino acids and the location of covalent bonds within a polypeptide chain.

Prokaryotes – class of organisms, including the eubacteria and archaea, that lack a true membrane-limited nucleus and other organelles.

Protein – a linear polymer of amino acids linked together in a specific sequence and usually containing more than 50 residues. Proteins form the key structural elements in cells and participate in nearly all cellular functions.

Reading frame – the sequence of nucleotide triplets that runs from a specific translation start codon in a mRNA to a stop codon. Some mRNAs can be translated into different polypeptides by reading them in two different reading frames.

Ribosome – a large complex comprising several different rRNA molecules and more than 50 proteins, organized into a large subunit and small subunits; the site of protein synthesis.

RNA – Ribonucleic acid, linear, single-stranded polymer, composed of ribose nucleotides, that is synthesized by transcription of DNA or by copying of RNA. Three types of cellular RNA play roles in protein synthesis.

rRNA – ribosomal RNA, any one of several large RNA molecules that are structural and functional components of ribosome.

Secondary structure – in proteins, local folding of a polypeptide chain into regular structures including the α helix, β sheet, and U-shaped turns and loops. tRNA molecules also have secondary structure by local folding of nucleotides forming a double-stranded chain.

Tertiary structure – in proteins, overall three-dimensional form of a polypeptide chain, which is stabilized by multiple noncovalent interactions between side chains. tRNA molecules also have tertiary structure.

Transcription – process whereby one strand of a DNA molecule is used as a template for synthesis of a complementary RNA by RNA polymerase.

Translation – the ribosome-mediated production of a polypeptide whose amino acid sequence is specified by the nucleotide sequence in an mRNA.

tRNA – transfer RNA, a group of small RNA molecules that functions as amino acid donors during protein synthesis. Each tRNA becomes covalently linked to a particular amino acid, forming an aa-tRNA.

Wavelet transformation – The wavelet transformation is similar to the windowed Fourier transformation, but instead of sine and cosine functions its kernel functions are wavelets. The kernel functions used in wavelet transformation are all obtained from one prototype function, by scaling and translating the prototype function.

INTRODUCTION

This study is the culmination of author's investigations and findings in the field of biology as a mechatronics engineer. The practice in this subject, which applies the methods and approaches of several disciplines, is a real interdisciplinary challenge. For-instance biology, mathematics, engineering, informatics, computer sciences are separate disciplines, but they can be combined together. Each discipline yields discipline specific results. Since we are not biologists, we are not able to conduct biological experiments directly, only to give them inspirations through theoretical articles or to form together an interdisciplinary research team.

Life sciences are very important in science and medicine, because their topics are related to the existence of humankind and their results directly affect our well-being. Biology as an independent science was developed in the 19th century, when scientists discovered that organisms shared basic characteristics. Molecular biologists made important findings and learned to isolate, characterize, and manipulate the molecular components of cells and organisms in the last six decades. The structures of macromolecules such as deoxyribonucleic acid (DNA), ribonucleic acid (RNA), and proteins are already well known. Molecular biology also studies the complex interactions between the systems of the biological molecules, including the interrelationship of DNA, RNA, and protein biosynthesis, in addition to learning how these interactions are regulated.

The study of chromosome scaffold and its relation to the DNA strand contains white spaces to be filled with knowledge. By giving an engineering explanation of a biological process, like cell differentiation during cell divisions, compression of genetic information, we could be able to utilize it. These results of investigations or inspirations from nature could be used in different applications. For example it could be utilized in image or database transformation and compression methods (2, 3 or N dimensional databases) or this understanding could help us to create artificial chromosome (*in vivo* or in purely technical form) to store any kind of database. Nowadays we could easily generate terapixel pictures (e.g. in nuclear imaging). How useful it could be to find a handy method to compress it easily for transmission, and unfold it or portion of it later with the same easiness.

Biology has two main areas of interest to find the required information:

- Molecular Biology,
- Developmental Biology.

Molecular Biology

The fundamental part of the research is based on Molecular Biology and its findings. The structure of chromosome scaffold and especially its relation to the DNA strand could be best described by topological graph theory. The most important aspect

is to determine whether its structure is planar or non-planar, or maybe during replication it is planar, but in the compressed form, in condensed chromosomes the scaffold is in non-planar form. The scaffold is built up of different types of histone, each type of histone could be digested by different enzymes. Most probably this is the way to make temporal and spatial structural changes (fold or unfold chromosomes) in nanoscale.

These nanoscale structural changes of the chromosome scaffold yield changes in gene expression, thus resulting in switching in cell types after cell division. Examining the latest results and findings of nanoscale biology to yield inspirations and provide explanations how genetic data were stored, was the main purpose and aspiration to conduct Ph.D. studies.

Developmental Biology

Developmental Biology describes the process how a multicellular organism develops from a zygote (the cell produced by the union of two gametes, before it undergoes cleavage).

To be able to give an engineering description and provide a suitable working model of the development of a specimen, we have to categorize the different type of cells, cell divisions and to create more accurate cell lineages than presently available, to be able to follow each and every strand of DNA and its replicated copy.

Digital Signal Processing (DSP)

The technical scientific field in the third quarter of the last century was shaped by the achievements of electronics, the first quarter of the twenty-first century will be shaped by the applications of digital signal processing. As there was no any engineer without basic electronics knowledge, there will be no any engineer or scientist without basic DSP knowledge.

Signals usually originate from the world surrounding us in various forms, like sound, visual images, vibrations (mechanical, electromagnetic etc.) and most of the time they carry some kind of information. The number of areas where we use DSP day by day is countless, image and sound processing, telecommunication, medical, scientific, industrial and military applications etc. This thesis mainly is concerned about image transformation and compression methods within the wider area of digital signal processing.

How Mechatronics is Related to Biology and Vice Versa?

Mechatronics is a real multidisciplinary science combining almost all kinds of engineering principles. The main feature of interdisciplinary researches is that the area

under investigation is examined by the methods of a variety of fields together – for example biology, mathematics, engineering and information technology - and at the end each discipline yields discipline specific results. So why not to extend the areas toward molecular biology and try to find common grounds in exploration.

A few years ago I have started to study different areas of biology, to obtain a deeper knowledge mainly in the fields of molecular biology, cell biology and developmental biology. I try to comprehend biological processes by an engineering approach, to explain certain phenomena, on the other hand to learn ideas through close observation of nature, to be utilized in the field of engineering. In the meantime the knowledge gained particularly in the field of modeling and simulation of dynamic systems, digital image processing and experience in MATLAB programming was helpful in this. The examination of the mysteries of nature is a sort of "reverse engineering".

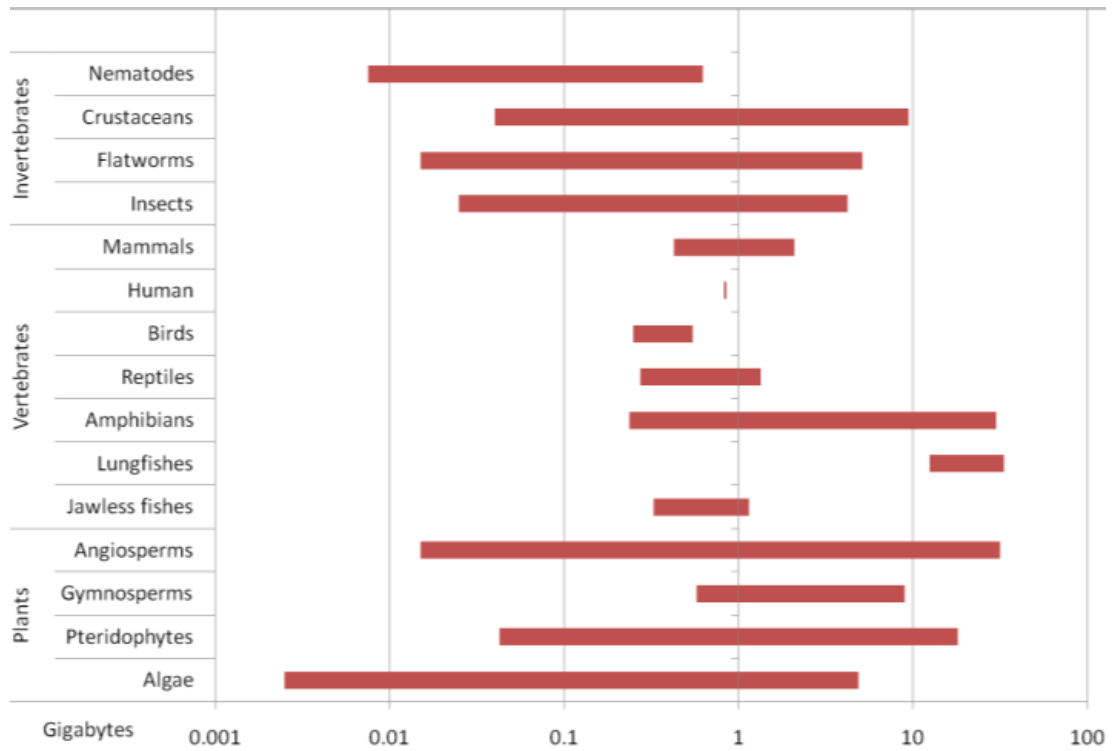
The target objective I have set up is, to propose hypotheses about the possible structure how the information is stored and compressed in the chromosomes, which refers to the three-dimensional structure of an organism. Our body consists of more than 10^{14} cells, they belong to one of the (about) 100-200 different cell types. Our genome is 3.3×10^9 base pairs long only (four different DNA bases exist in nature), Table 1 shows the information content of the genome of different species. In the course of my research work I investigated: how this 'assembly' information (body plan) is programmed and how an organism develops through a series of divisions based on that. To find the answer or possible answer to this question is the essence of my research program.

To achieve this goal, I have studied biology from the existing literature, especially about chromosome structure and topology-related ones. It is likely that this type of information is included not only in the genome, i.e. the DNA sequence, but also as how the DNA chain is bound to the chromosome scaffold proteins, or how the methylation pattern of the DNA chain changes by cell division cycles (epigenetic modifications).

In this particular case, we can expect not only to find an efficient method for compressing 2, 3 or multidimensional datasets, but it can be useful for the benefit of biology, serving with explanation of certain phenomena (if we manage to create an expressive model, that may be able to explain anomalies like certain type of cancerous alterations during cell divisions).

Every multicellular organism develops from a single cell, the zygote. During cell division, a cell always produces two descendent daughter cells. There are various descriptions in literature review of biological experiments, which suggest that there is a causal relationship between the replicated DNA strand and the type of the individual daughter cells. Based on that, it was possible to introduce a cell-numbering system.

Table 1 Information content of the genomes of different organisms



By studying the progression of cell division and cell differentiation we can conclude that this process in fact looks like an inverse discrete wavelet transformation (DWT). This transformation, derived from nature was worked out for two different cases:

- for the case of logical variables,
- for the case of discrete variables.

We also managed to develop new types of compression methods for these 2, 3 or multi-dimensional datasets cases (e.g. for images, moving pictures or for vector fields).

1 CHAPTER BIOLOGICAL BACKGROUND

In the field of biology we can find so-called information systems at multiple levels. The organisms are grouped in various ways, for example like unicellular and multicellular. Multicellular organisms must possess a bi-level information system, or dataset, first of all individual cell-type specific information about its cell components and secondly, for the set-up information of all the cells and tissues that make up the whole organism.

Since the DNA was discovered, it is known to be the material of inheritance. These DNA molecules are contained in the form of chromosomes in the cells. Half amount of the material of the chromosome is the DNA molecule itself and the other half is made up of different protein molecules (histone and non-histone). This protein backbone is called as chromosome scaffold. It is an emerging field of science to study the nanoscale structure and especially structural changes of chromosome scaffold during or between cell divisions.

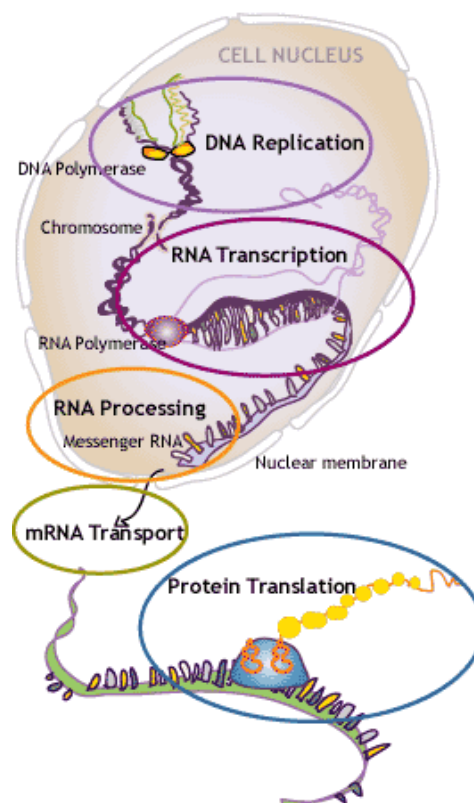


Figure 1.1 The route from DNA to protein [1]

1.1 The Route from DNA to Protein (DNA-RNA-Protein)

Cells consist of different components, including DNA, RNA, and proteins. These macromolecules are the most interesting and characteristic molecules of living

systems. DNA stores the genetic information of a cell which consists of thousands of genes. Each gene serves as a template on how to build a protein molecule. Proteins are among the most essential macromolecules, which perform vital tasks for the cell functions and serve as building blocks in human tissues. The orientation of the genetic information defines the protein composition and their functions for each cell.

1.2 DNA Replication

The principal role of DNA molecules is the long-term storage of genetic information. The unit of DNA that carries this genetic information is called a gene. DNA segments have functional purposes. There are other DNA sequences with structural tasks. They are also involved in the regulation and utilization of genetic information. DNA is arranged into structures called chromosomes and the complete set of chromosomes make up the genome within a cell. Eukaryotic organisms such as fungi, plants, and animals contain most of their DNA (except mitochondrial DNA) inside the cell nucleus, while in prokaryotes such as bacteria (e.g. *E. coli*), DNA is found in the cytoplasm [1] [2].

DNA is a long polymer molecule, made of monomers called nucleotides. Nucleotides consist of a pentose carbon sugar, a phosphate group and one of four different nitrogenous bases: guanine (G), adenine (A), cytosine (C), and thymine (T). Sugars and phosphate groups are joined by ester bonds.

1.3 RNA Transcription

Proteins are the building-block macromolecules of cells. Proteins are assembled from 20 different amino acids; usually a few hundred of them are connected into a chain by peptide bonds. The amino acid sequence of the protein is coded by the messenger RNA (mRNA) and the “factory” where the “assembling” takes place is the ribosome. The ribosome is a conglomerate of RNA molecules and proteins. The template of these RNA molecules is stored inside the chromosome in the form of double helix DNA chains containing the genetic information used in the development of all living organisms [3], [4].

Table 1-1 The genetic code

Codon	tRNA anticodon	Amino acid	Codon	tRNA anticodon	Amino acid	Codon	tRNA anticodon	Amino acid	Codon	tRNA anticodon	Amino Acid
UUU		Phe	CUU	GAAGu	Leu	AUU	UAAGu	Ile	GUU	CAAgu	Val
UUC	AAGa	Phe	CUC	GAGa	Leu	AUC	UAGau	Ile	GUC	CAGa	Val
UUA	AAUc	Leu	CUA	GAUc	Leu	AUA	UAU	Ile	GUA	CAUc	Val
UUG	AAC	Leu	CUG	GAC	Leu	AUG	UAC	Met	GUG	CAC	Val
UCU	AGAgU	Ser	CCU	GGAgU	Pro	ACU	UGAgU	Thr	GCU	CGAgU	Ala
UCC	AGGa	Ser	CCC	GGGa	Pro	ACC	UGGa	Thr	GCC	CGGa	Ala
UCA	AGUc	Ser	CCA	GGUc	Pro	ACA	UGUc	Thr	GCA	CGUc	Ala
UCG	AGC	Ser	CCG	GGC	Pro	ACG	UGC	Thr	GCG	CGC	Ala
UAU		Tyr	CAU		His	AAU		Asn	GAU		Asp
UAC	AUGa	Tyr	CAC	GUGa	His	AAC	UUGa	Asn	GAC	CUGa	Asp
UAA		Stop	CAA	GUUc	Gln	AAA	UUUc	Lys	GAA	CUUc	Glu
UAG		Stop	CAG	GUC	Gln	AAG	UUC	Lys	GAG	CUC	Glu
UGU		Cys	CGU	GCAgu	Arg	AGU		Ser	GGU	CCAgu	Gly
UGC	AGCa	Cys	CGC	GCGa	Arg	AGC	UCGa	Ser	GGC	CCGa	Gly
UGA		Stop	CGA	GCUc	Arg	AGA	UCUc	Arg	GGA	CCUc	Gly
UGG	ACC	Trp	CGG	GCC	Arg	AGG	UCC	Arg	GGG	CCC	Gly

*Data was compiled from different sources, based on [2].

1.4 Protein Translation

The manufacturing of proteins in the ribosome is carried out by the help of tiny “robots”: the transfer RNA (tRNA) molecules. They are possibly the smallest “autonomous robots” on earth. They can recognize a specific amino acid from the possible pool of 20 different amino acids. They are able to transport these blocks to the ribosome—the site where amino acids are assembled into a protein chain. A tRNA molecule acts similar to a mobile robot, as it delivers the components (amino acids) to the assembly-line (the ribosome). Extensive research on tRNA interaction with the ribosome exists, but these studies focus on their biochemical interactions. There are other factors of great significance that play an integral part in this assembly. The ribosome is capable of distinguishing the cognate tRNA (ctRNA) from all other tRNA molecules, but it accomplishes this task by trial and error, which is extremely time-consuming. It is highly probable that a mechanism exists in nature whereby the correct ctRNA is chosen before it enters the ribosome “A site.” It is important to note here that no description on the tRNA movement in the cytoplasm is found in literature, but extensive research is available on tRNA recognition, selection and relative movements into or inside the ribosome [5], [6].

2 CHAPTER COMPUTATIONAL METHODS

2.1 General Discrete Wavelet Transformation (DWT)

In our discussion for simplicity and for the sake of better understanding we will represent this sample organism as a 2D (2-dimensional) object (Figure 2.1), which elements are the daughter cells. Consecutive cleavages are perpendicular to each other, same way we have halved the pixels representing the dividing cells. After each cycle the number of elements is doubled, one time in x direction, next to the y direction. When we look at this resulting arrangement, it resembles to a two dimensional image with 2^n pixels after the n^{th} cycle (in our example this image has 4x4 pixels after the 4th cycle).

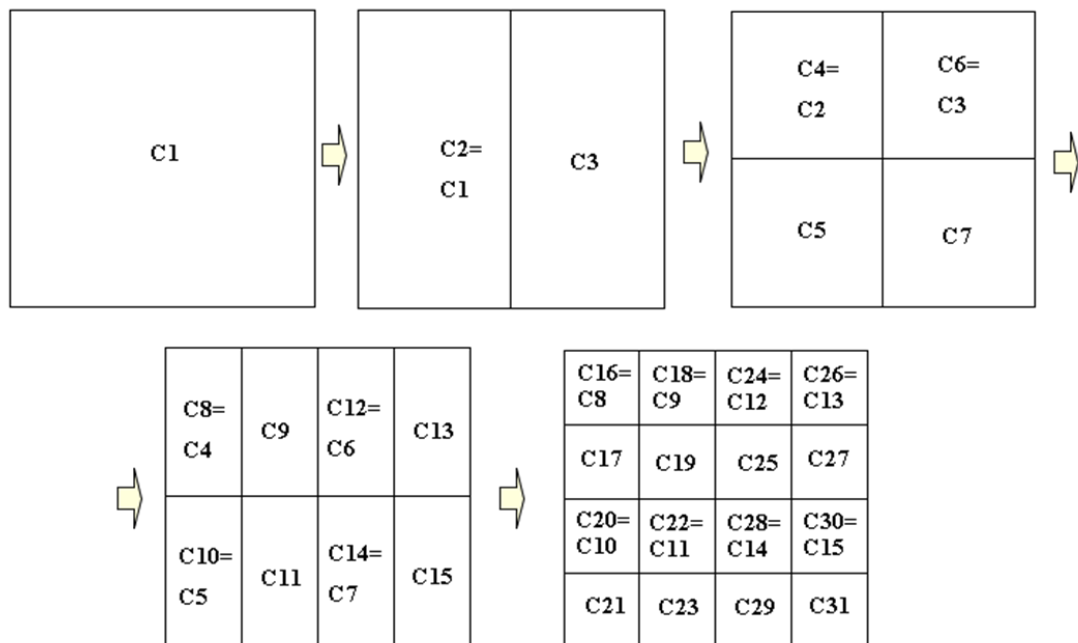


Figure 2.1 Development by cell divisions of a theoretical 2 dimensional sample organism

Let's look at this image by other way around, an image is given of a size of $M \times M$ pixels, where $M = 2^m$. Following the procedure opposite direction that we have seen in nature, halving the number of elements by each cycle, we create an image transformation algorithm. Our aim is, to find out to which known algorithm it resembles. Because in each cycle we halve the number of pixels, it means that we downscale the image by a factor of 2. Discrete wavelet transformation does the same (Figure 2.2), decompose the image with a high pass and a low pass filter; $h(n)$ high-pass filter provides the detail coefficients, $g(n)$ filter provides the approximation coefficients. After downsampling the image high frequency coefficients we get the

first level coefficients and downsampling the low frequency coefficients we could get the further (second and so on) level coefficients [7].

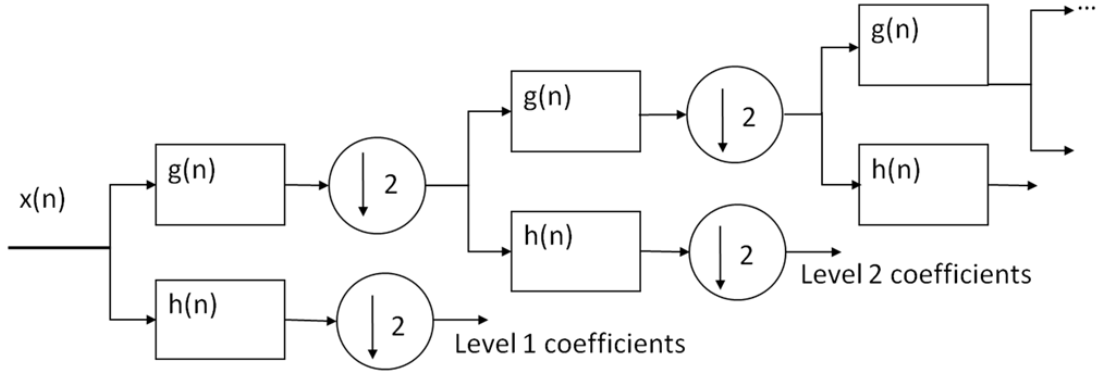


Figure 2.2 General DWT representation using cascading filter bank

Decomposition into low and high frequencies by convolution, followed by downsampling ($g(n)$ low pass, $h(n)$ high pass filters). Restriction on h and g filters: they must complement each other so, that we should be able to reconstruct the original signal.

2.2 Second Generation Wavelet Transformation of 1D (one-dimensional) Signal

Traditional signal processing transformations like Fourier transformation and even first generation wavelet transformations work well for infinite or periodical signals. Images and most of the datasets are bounded in size and typically are not periodic, but usually they have a positive property, that neighboring elements are highly correlated.

General wavelet transformation comprises mainly two steps, filtering (to compare the signal to the kernel) followed by a downsampling process. In the case of second generation wavelet transformation there are three steps, first is splitting the data into two parts, for example to even and odd elements, predict odds using even part, thus generating the detail coefficients and finally update even part using detail coefficients [8]. Inverse transformation has similar steps in opposite order. The simplest use of this wavelet transformation is for 1D signal.

Filtering:

In a 1D system we have an input signal f , and the transformed signal is g , than the linear constant coefficient difference transformation of the signal looks like:

$$\sum_{(k) \in \mathcal{R}_a} a_k g(m - k) = \sum_{(k) \in \mathcal{R}_b} b_k f(m - k) \quad (2-1)$$

The region of the filters are \mathcal{R}_a and \mathcal{R}_b , a_k and b_k are real numbers. By rearranging this difference equation we get, that if $a_0 \neq 0$ and $(0) \in \mathcal{R}_a$ the transformed signal would be:

$$g(m) = -\sum_{(k) \in \mathcal{R}_a} a_k' g(m - k) + \sum_{(k) \in \mathcal{R}_b} b_k' f(m - k) \quad (2-2)$$

for example let the direction of recursion be left to right.

2.3 Spatial 2D System

How to carry out wavelet transformation in higher dimensional systems? In a general 2D spatial system the two variables are m and n , the linear constant coefficient difference equations could be written as:

$$\sum_{(k,l) \in \mathcal{R}_a} a_{k,l} g(m-k, n-l) = \sum_{(k,l) \in \mathcal{R}_b} b_{k,l} f(m-k, n-l) \quad (2-3)$$

Whereas f is the input image, g is the transformed image, $b_{k,l}$ are the feedforward coefficients, $a_{k,l}$ are the feedback coefficients [9]. The solution for y :

$$g(m, n) = \sum_{(k,l) \in \mathcal{R}_a} a'_{k,l} g(m-k, n-l) + \sum_{(k,l) \in \mathcal{R}_b} b'_{k,l} f(m-k, n-l) \quad (2-4)$$

The progression could be e.g. horizontally left to right, vertically up to down by each row, Figure 2.3 gives an example.

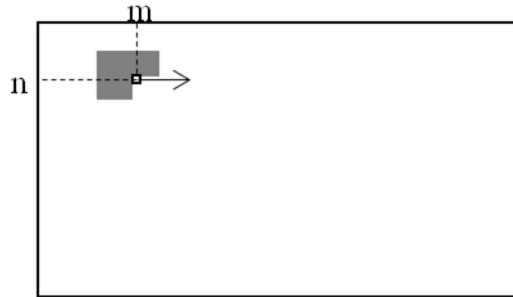


Figure 2.3 Filtering progression in a 2 Dimensional system

Wavelet transformations and the progression of cell division and differentiation chain, all are hierarchical transformations. Figure 2.4 example shows the transformation of a square shape image into fragments of similar shapes [10].

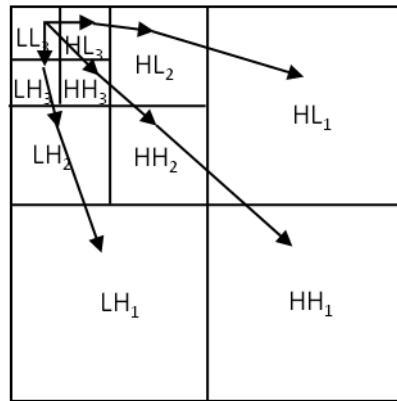


Figure 2.4 An example of a hierarchical image transformation in 2D system
 L is low frequency component, H is the high frequency component of the signal

2.4 Matlab Modeling and Simulation

All the image processing, modeling and simulation work mentioned in this thesis were performed by Matlab. Matlab is a matrix based numerical computational program with graphical interface to visualize the results. There are several toolboxes available to help the user to accomplish different tasks. Image processing toolbox was used to acquire images for further manipulations – the presented image transformation and compression jobs and visualization of tRNA movement. All computations and simulations for the publications in this theme were done by Matlab.

Matlab strength however lies in matrix calculations, direct matrix algebraic operations are much faster with it, compared to cycles. Filtering shown in Figure 2.3, for example suggests, that the structure of the transformation program could be arranged as a double cycle along x and y axes. Due to the hierarchical structure of processes developed here, they can be solved by direct matrix operations, avoiding organizing cycles, which increases the magnitude of the processing speed.

Of course, there are other available alternatives, like Scilab or LabVIEW for instance. Scilab is very similar to Matlab, main distinction that Scilab is freely available, but unfortunately there is no any reliable image processing toolbox on hand for it.

LabVIEW is a graphical programming language. All the above mentioned assignments could be solved by LabVIEW as well. (This is the future plan, image transformation method is already worked out by LabVIEW.)

3 CHAPTER

UNIVERSAL CHROMOSOME NUMBERING SYSTEM FOR CELL LINEAGES, BACTERIA COLONIES ETC.

3.1 Cell Numbering System

What is the benefit to establish a cell numbering system? First of all we could be able to name and identify each and every individual cell of the body of a given organism, on the other hand to find out the structure or method of body plan coding. We can presume that cell division (asexual) is well choreographed (Figure 3.1), so we can suppose that exactly the same half of the DNA double-chain should remain always in the 'old' cell and always the other goes to the 'new' cell (immortal strand hypothesis [11]). Linking cell lineages and this numbering system could help our understanding about how body plan information could be stored in our genome. Considering the pattern of cell differentiations after divisions, this process best could be interpreted as a kind of wavelet transformation. Describing cell divisions and differentiations, brings us closer to the understanding how body plan information is compressed, what is the possible structure of this information.

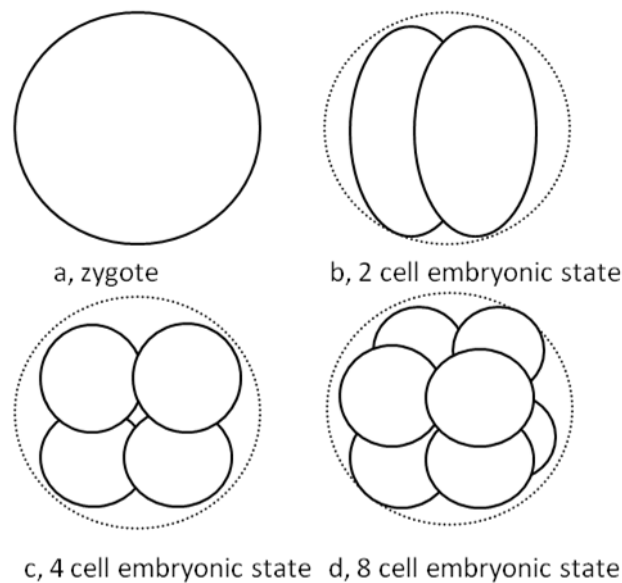


Figure 3.1 Embryonic development

Cell lineage patterns follow perfect binary tree structure; DNA strand replications also follow binary tree structure. Recent experiments hint, that there could be high correlation between these tree structures [12], [13], [14]. Considering this fact and the nature of cell divisions (symmetric or asymmetric), embryonic development could be described as a kind of wavelet transformation.

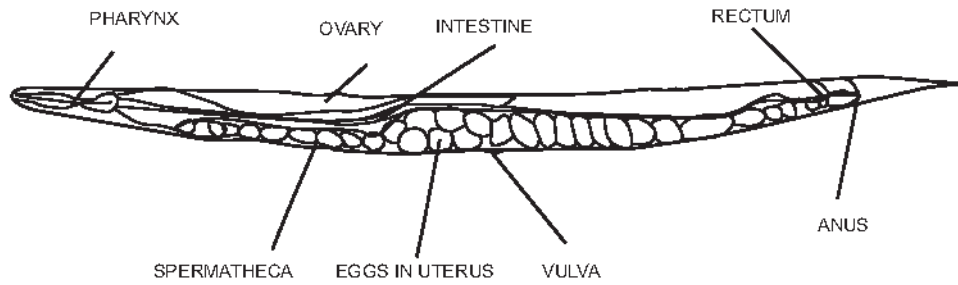


Figure 3.2 *Caenorhabditis elegans* with 959 somatic cells [15]

By proposing a cell numbering system we could be able to name and identify each and every individual descendant of a predecessor chromosome or prokaryotic cell (a zygote, practically the first cell after sexual cell division or crossing over).

3.2 Cell Lineage

The history to draw cell lineages is dating back to the discovery of microscopy, when cells and tissues were first discovered. The most studied organism is a worm *Caenorhabditis elegans* with 959 somatic cells, shown in Figure 3.2 and Figure 3.3 [16].

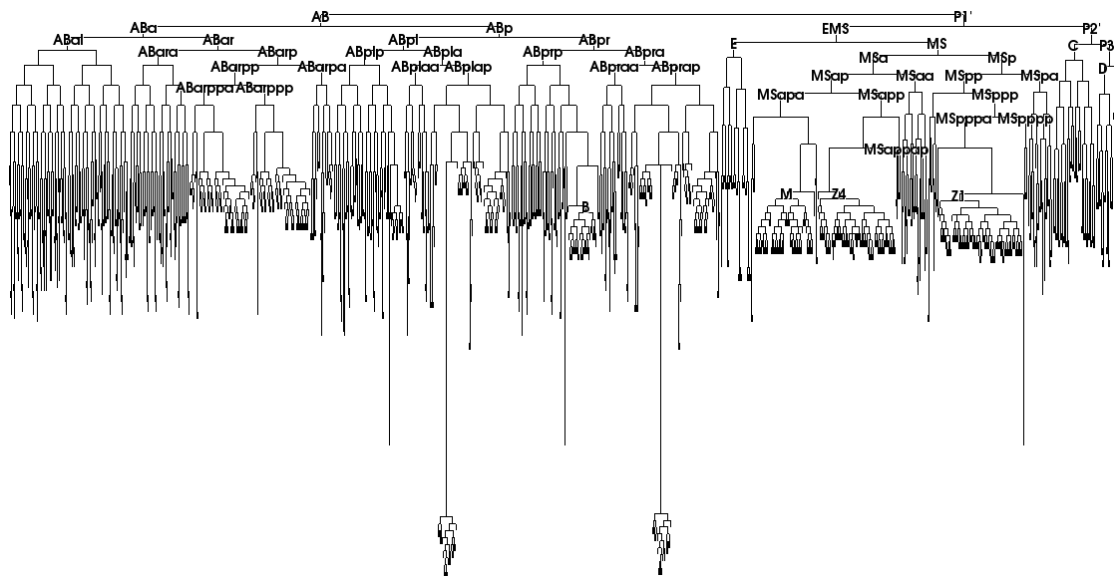


Figure 3.3 Cell lineage of *Caenorhabditis elegans* with 959 somatic cells [16]

Figure 3.4 shows the cell lineage of *C. elegans* after the third cell division cycle, the present number of cells in this stage is 8 and the total number of previously existed cells is 7. Cell divisions follow a well choreographed and documented pattern and it can be described as a binary tree structure.

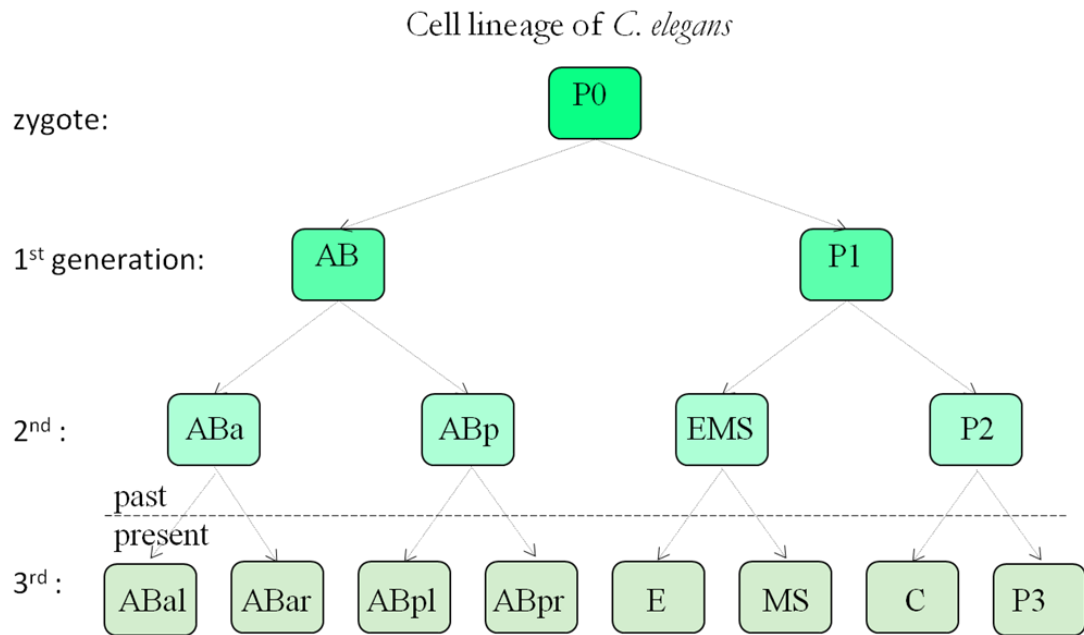


Figure 3.4 P0 was the zygote cell, yielding 8 descendant cells after the third cell division cycles

3.3 Identification of DNA Strands

DNA molecules are double helices. How to tell apart each other if they are all identical? There is a relative DNA strand naming method to call one strand Watson strand and the complement as Crick strand [17]. It would be better to generate a consensus in naming the individual strands. DNA replication starts at the origin of replication initiation regions. They contain a unique DNA sequence pattern. One strand is rich in thymine (T) so consequently the complement strand is rich in adenine (A).

ARS consensus sequence strand [18]:

5'~**ATTTATGTTT**~3'

so the corresponding complement strand should contain the sequence:

3'~**TAAATCAAAA**~5'

This way we can call the strand containing the first ARS sequence as **T** strand and the complement strand containing the secondly mentioned sequence as strand **A**.

3.4 Binary Tree; Cell Numbering System

Development of a multicellular organism originates from the zygote and cell divisions follow a binary tree structure. For the sake of simplicity take an example of an organism with a single chromosome only (like the Australian jack jumper ant, *Myrmecia pilosula*, their genome is squeezed into a single pair of chromosomes, moreover males are haploid, so they have a single chromosome only). Meselson, M. and Stahl, F.W. (1958) proved, that during cell division DNA strand of each chromosome duplicates semi-conservatively.

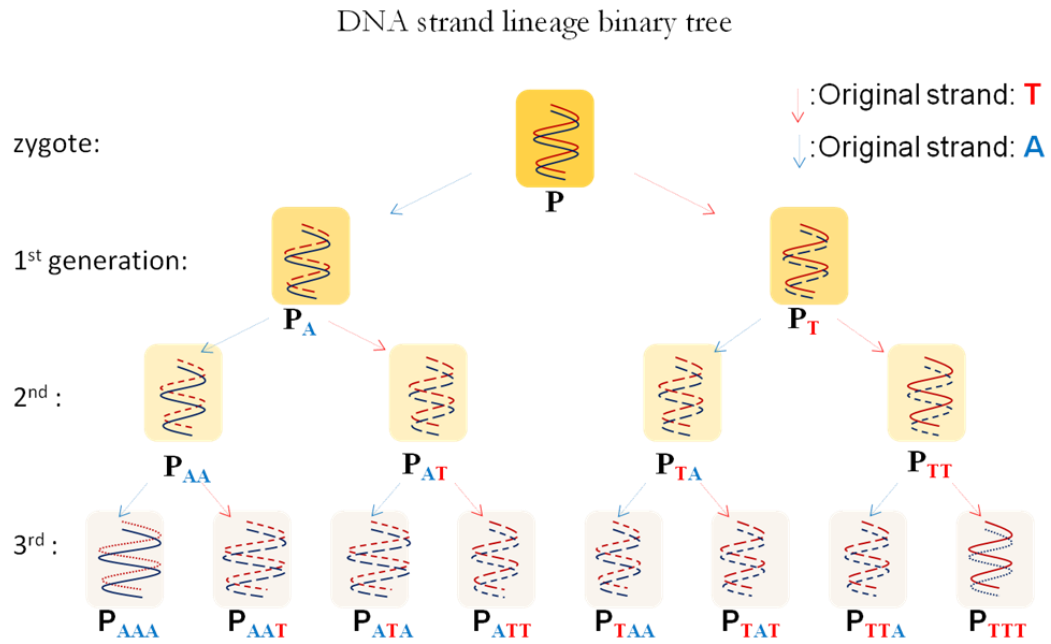


Figure 3.5 DNA strand identification

We can draw a binary tree, like the one shown in Figure 3.5, to follow each and every old and newly synthesized strand. Cell **P**, the ancestor contains the pair of originator DNA strands and both of its descendants carry one old and one new strand. Letter **A** or **T** indicates the original one.

To create a cell numbering system, letters are replaced by numbers, the last digit of the ID number is set to **0** when strand **A** was the original strand and number **1** in the case of letter **T** (Figure 3.6). For example cell number **C₁₀₀₀** is a cell in the third generation after the third cell division cycle, containing the original **A** strand of the zygote (**C₁**), cell **C₁₁₁₁** of the 3rd generation holds the original **T** strand, and so on.

Replacing **A** and **T** letters by numeral 0 and 1

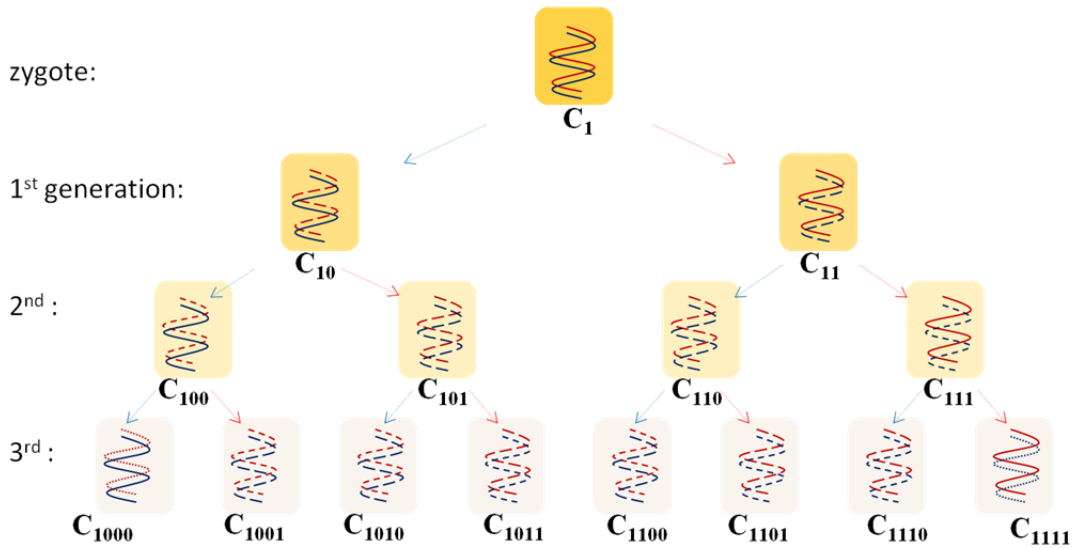


Figure 3.6 Replacing *A* and *T* letters by numeral 0 and 1

3.5 Correlation between Cell Lineage and Numbering

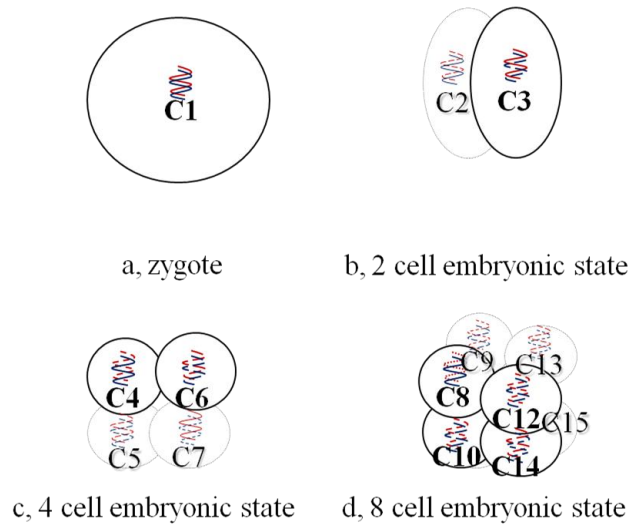


Figure 3.7 Embryonic development by consecutive cell divisions
different orientation planes are marked

There is an ongoing debate about immortal strand hypothesis, but there is no argument about the existence of high degree correlation level between cell type and DNA identity of the descendant's chromosomes [19]. If we can say that correlation exists between cell lineage and replicated chromosomes → replicated DNA strands, we can rightly make logical connection between these two binary tree structures (Figure 3.4 and Figure 3.6).

Figure 3.7 combines this double entity by illustrating a developing embryo and identifying its cells by these corresponding cell numbers. Of course it is a fictitious numbering now, since there is no any experiment carried out yet for correct identification, however there are experiments pointing out its existence [20], [21]. Cell numbers are shown here as decimal numbers (ex. C_{III}^2 binary cell number is the same as C_{15}^{10} decimal cell number).

3.6 Conclusion of the Chapter

Our body contains at least 10^{14} cells, all of them are originated from only one by consecutive cell divisions. After every cell division cycle the number of cells is doubled. The number of cells after the n^{th} cycle is 2^n and the number of all previous stages or the number of cell divisions is $2^n - 1$. The number of cycles to produce this quantity (10^{14}) of cells should be more than $(\log_2(10^{14})) = 47$, or even more taking into account of apoptosis (programmed cell death) or other cell losses. What is the amount of information to be inherited if it were not compressed in the genome? The number of different cell types is in the range of 100-200. Similarly to the genetic code, DNA base triplets are coding for the possible 20 amino acids ($4^3 = 64$), on this analogy it needs base quadruplets (at least 4 nucleotide bases long words) to code for the different cell types ($4^4 = 256$). To describe each and every cell of the body, this information alone requires $10^{14} \times 4$ bases = 4×10^{14} nucleotide base pairs from our genome. Contrarily the length of our genome is 3.3×10^9 base pairs only, so this kind of information, related to body plan, seems to be highly compressed.

Cell lineage patterns follow perfect binary tree structure; DNA strand replications also follow binary tree structure. Recent experiments hint, that there could be high correlation between them (immortal strand hypothesis). If we can say, that there exists correlation between cell lineage and replicated chromosomes \rightarrow replicated DNA strands, we can rightly make logical connection between these two binary tree structures.

By introducing this cell numbering system we are able to name and identify each and every individual descendant of a predecessor chromosome or prokaryotic cell.

4 CHAPTER CELL DIVISION IN TIME AND SPACE

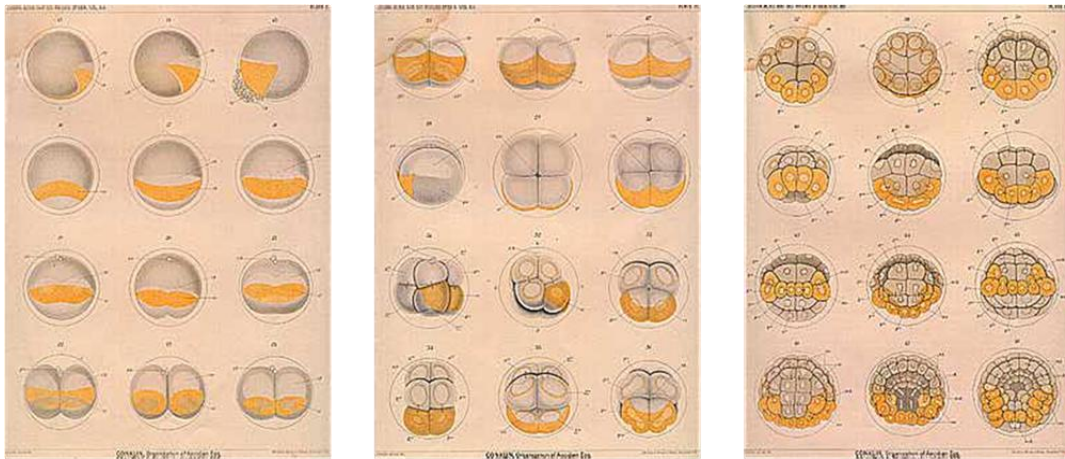


Figure 4.1 E. G. Conklin's original caption, a remarkable fate map of the *ascidian*, published in 1905

Multicellular organisms like animals, plants and fungi, all are eukaryotes. Eukaryotes, unlike prokaryotes, possess nucleus, which contains the genetic information in the form of DNA molecules by base sequence coding. (Prokaryotes are much more simple organisms, smaller in size, there is no any multicellular prokaryote, they lack a nucleus with condensed chromosome and they usually have a single circular DNA chain to inherit their genetic information. Bacteria are prokaryotes.) These DNA chains are coiled up by the help of some protein molecules (histones) and they form the chromosomes. After duplicating this genetic information (replication of DNA), these cells are able to divide into two, thus forming two new cells, this is the mitotic cell division. These daughter cells could be the same, or could be different than their predecessor. (Meiosis is the other type of cell division which is required for sexual reproduction, but it is out of the scope of our investigation.) [3], [15]

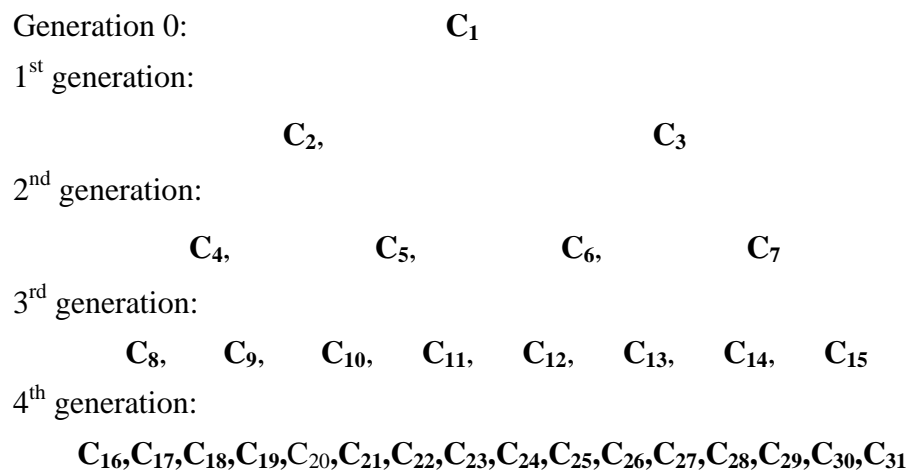
After sexual reproduction a new individual multicellular organism can develop from the fertilized egg cell (ovum). So every multicellular organism develops from one cell, the zygote by consecutive cell divisions to form its body finally, this is embryogenesis about. After each cell division or cell division cycle the number of cells doubles, this process follows a perfect binary tree structure. The information about the cell divisions is stored somehow in the genome, which governs the development of different cell types, tissues (morphogenesis) [22]. Each cell belongs to one of the few hundred cell types. The resulting new cells occupy an available new location, position in the 3 dimensional space, it resembles a kind of image. Here comes the idea to compare this biological process to image processing, particularly to digital image processing, if we consider cells as pixels.

Cell division takes place in the 3 dimensional space, so the cell division should have spatial orientation. Right from the time of the discovery of microscopy,

biologists started to investigate embryonic development and followed cell fate. For example Edwin G. Conklin [23] in 1905 has published a cell fate map of an emerging ascidians embryo (Figure 4.1).

4.1 Cell Division as Inverse Wavelet Transformation

The sequence of consecutive cell divisions forms a perfect binary tree structure, because each cell division yields two new daughter cells. When we look at this tree upside down, i.e. in opposite direction in time, we can recognize it as an image transformation. So we can consider the development of a multicellular organism as an inverse image transformation.



Let start to investigate this process from the very beginning, from the first cell divisions (Figure 4.2) [15]. According the spatial direction of the cleavage orientation planes we can distinguish different cleavage patterns, like radial, spiral and rotational.

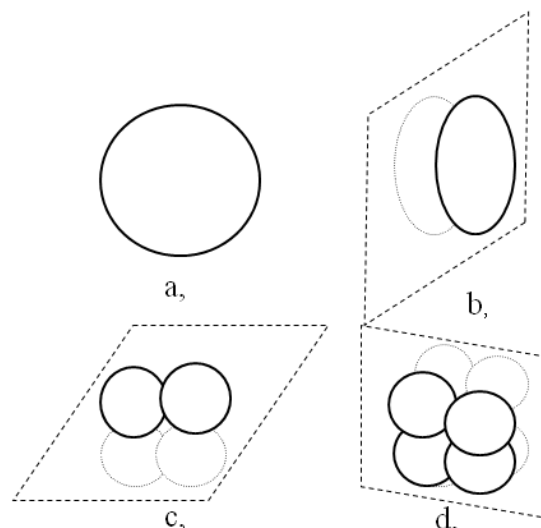


Figure 4.2 Cleavage orientation planes (radial cleavages)

For example Figure 4.2 shows radial cleavages. The plane of the first cleavage (bisection of the zygote) is vertical, the second is horizontal and the third is vertical again, but perpendicular to the first and to the second plane of cleavages too. Some of the vertebrates and echinoderms (marine animals) follow this pattern.

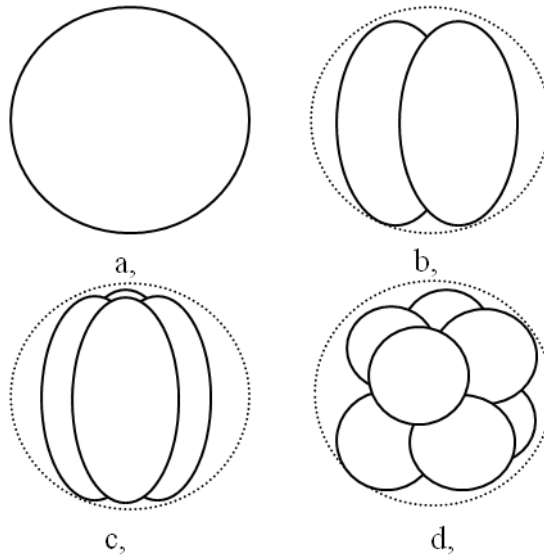


Figure 4.3 Development of an organism from a single cell (spiral cleavages)

Figure 4.3 shows spiral cleavages, it is typical for Spiralia and Figure 4.4 shows the cleavage orientation planes of rotational cleavages, it is the characteristic of mammalian development.

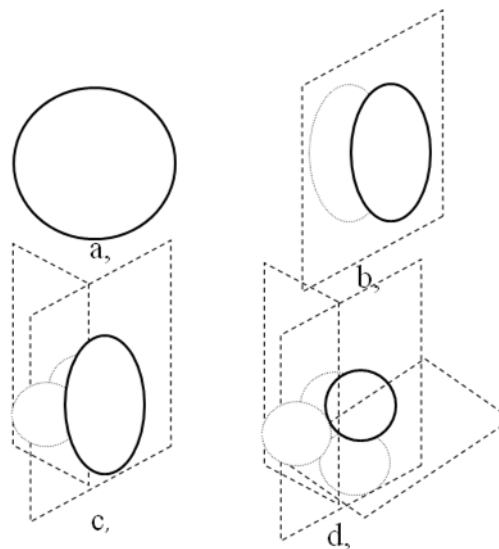


Figure 4.4 Cleavage orientation planes (rotational)

4.2 Cell Divisions in 2D (layered tissue of cells)

One important step of wavelet transformation is downsampling. When the signal is 1D, it is very simple to take every other element apart, select even and odd elements. To answer this question is not so self-explanatory. Cells in developing tissue have the tendency to form 2 dimensional layers. Because in each cell division cycle the number of cells are doubled, in the term of image processing we can imagine it as to transform the picture first along x axis, then y , x , y and so on (Figure 4.5 and Figure 4.6).

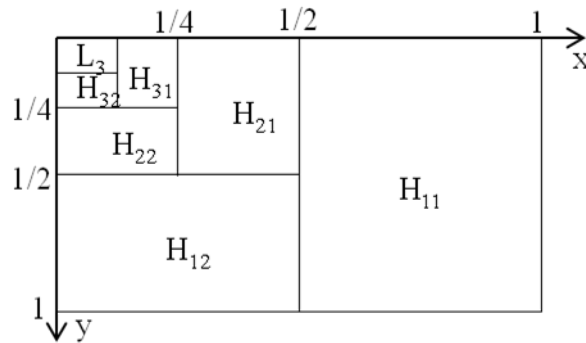


Figure 4.5 Image transformation in 2D

The most important observation is, that the shape of the pixels, so the shape of the transformed image becomes distorted after every other transformation step (H_{11} , H_{21} , H_{31} in Figure 4.5). It is the most revealing on low resolution images of Figure 4.7. There are some types of tissues, where the cells are layered and their shape looks like rectangles. In nature distorted shape of a cell means reduced viability, avoiding this cell divisions occur in pairs, in fast successions (Figure 4.6. a).

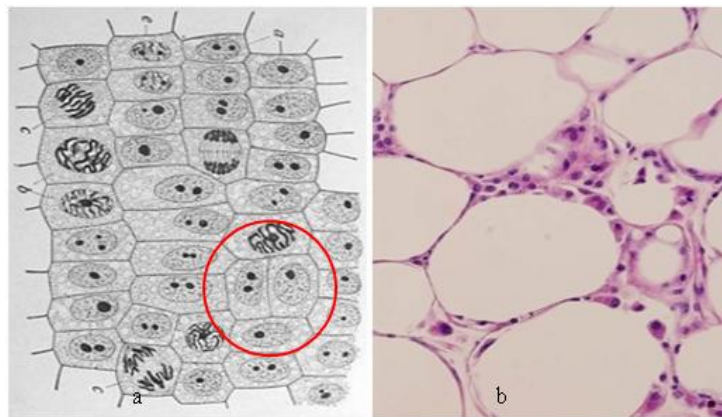


Figure 4.6 Dividing cells

a, some of the newly divided cells have distorted shape, waiting for a second consecutive division (marked with red circle), **b**, cells tend to occupy the available space [24], [25]

If we use this transformation as a 2D image transformation we will get the discomfort of uneven resolution in different directions. For example every other sub-

image of Figure 4.7 (images in the second and fourth row) contains rectangular pixels and their resolution is half horizontally than vertically.

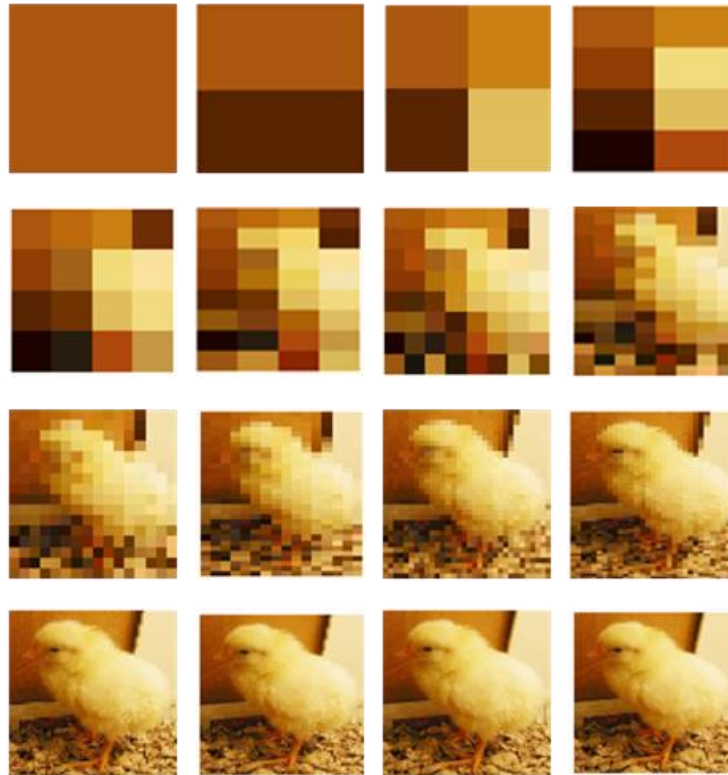


Figure 4.7 Example image development x and y direction consecutively (sequence of images from left to right and up to down)

4.3 Isometric Pixels

How to avoid odd shape pixels, why do we not take inspiration from nature? If the ordinary shape of the pixel is square and the length of its side is a , to reduce the size to half, the length of the pixel side should be equal to $a/\sqrt{2}$, but how to arrange them? Taking the example of spiral cleavages depicted in Figure 4.3, the direction of the downsampling could be diagonal, not necessarily vertical or horizontal.

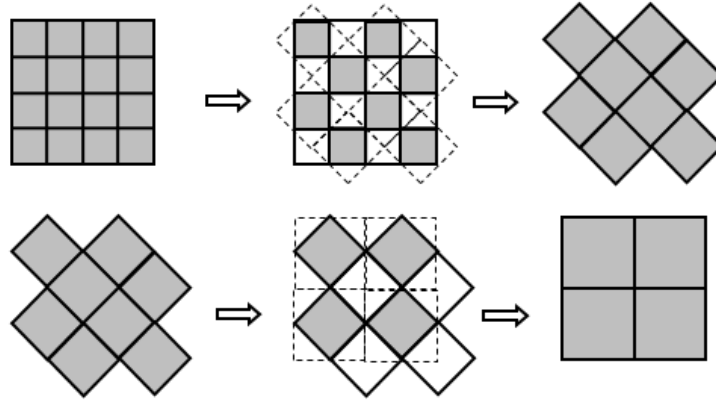


Figure 4.8 Isometric shape pixels (diamond shape arrangement)

Squares have the property that rotating them by 45° then their height will be $a/\sqrt{2}$ (Figure 4.8). Downsampling the image by a factor of 2 diagonally, we get this diamond shape arrangement. It has two favorable properties, the pixels maintain their isometric shape (square) and same time the vertical and horizontal resolution of the transformed image would be the same. Figure 4.9 shows the diagonal transformation series of the same picture as Figure 4.7.

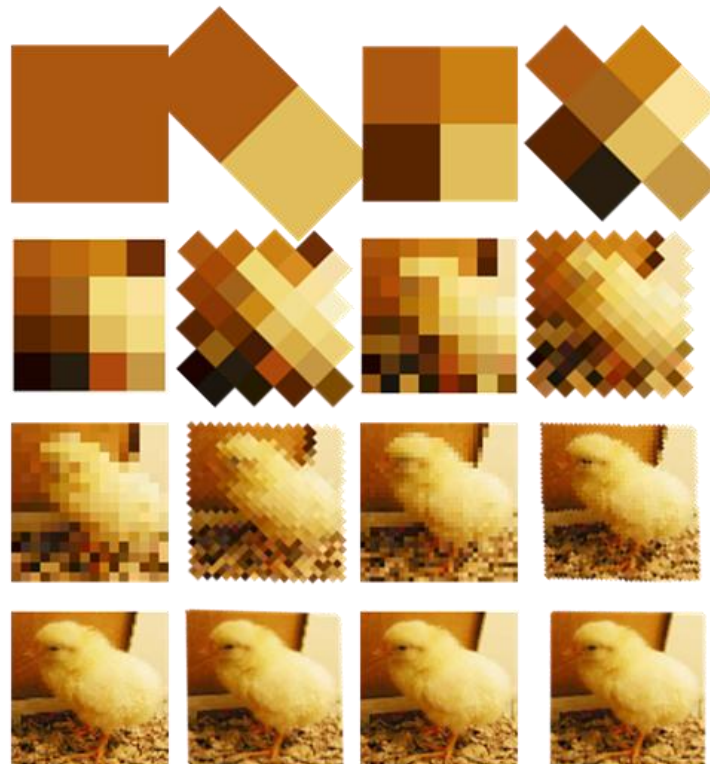


Figure 4.9 Example image with isometric pixels
(sequence of images from left to right and up to down)

How to design a suitable filter? Equation (2-4) describes the transformation, but how to choose the candidate pixels to determine their coefficients. Figure 4.10 shows examples of the approximation of P_{ij} pixel by the values of pixels in the

surrounding area for both cases (the nearest 16 are indicated by gray color). In case **a**, when the direction of downsampling is x direction, there are two direct neighbors, but in case **b** in Figure 4.10, when the downsampling is diagonal, there are 4 direct neighbors, they are the most suitable subject for approximation. The nearest is the pixel used for approximation the higher is the correlation between them.

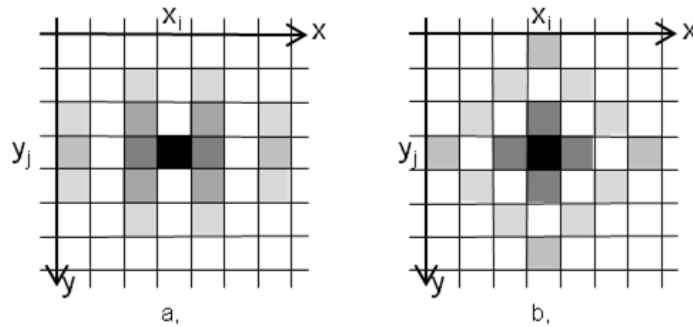


Figure 4.10 Filter design in 2D

Left side (**a**,) when during downsampling we discard every other row or column (here row) and right side figure (**b**,) shows when we discard every other element diagonally (diamond shape arrangement)

Sometimes cells do not fill the available space entirely, there could be some gaps between them (Figure 4.6b). Good number of the practical implementations also deploys pixels different than square shape and usually they do not occupy the available surface completely, for example the pixels of a monitor. Figure 4.11 shows a possible example, when the pixels are congruent to itself after rotated by 45° , in this case it is not noticeable, that they are rotated after each downsampling cycle.

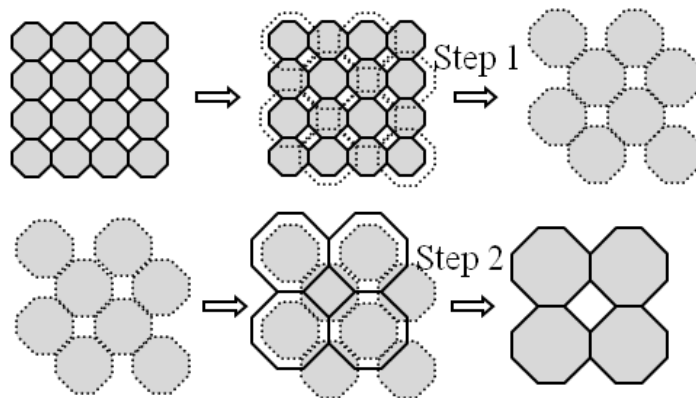


Figure 4.11 Pixels are congruent to itself after rotated by 45°

4.4 3D Image Transformation (Cell Division in Time and Space)

How to transform a 3-dimensional image or 3D spatial database [26], or in the case of a 2D movie (motion pictures) to transform a time sequence of 2D images/frames? The easiest way for hierarchical transformation of a 3D signal is, as we have seen at 2D cases, to downsample along each axis consecutively (Figure 4.12), downsample first along x axis, then y , z and so on again. We will face the same

disadvantages as it was at 2D case, pixels shape will be rectangular solid instead of the original cube form and their resolution will be half or double in one direction than the other two directions. The shape of the pixels of the transformed image would be cubical once more only after every three transformations (Figure 4.12).

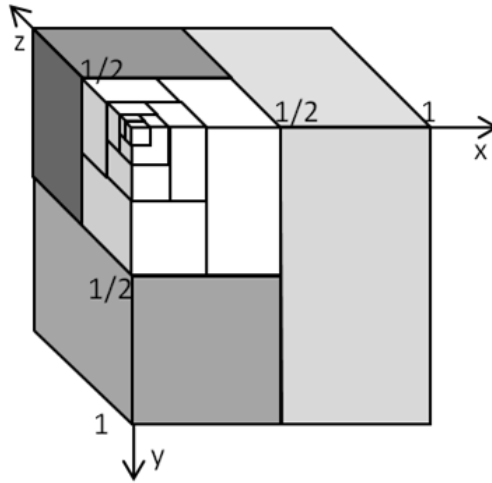


Figure 4.12 Hierarchical transformation of a 3D signal downsampling first along x axis, then y, z and so on again

In nature cells tend to occupy the available space they have and to maintain their desired shape, Figure 4.6,b and Figure 4.13 demonstrate possible downsampling method to maintain equal resolution in all directions. Unfortunately because of geometrical constraints it is not achievable to preserve the pixel's cubical shape, only in every three steps. The example of Figure 4.13 is most appropriate for movie (motion picture) transformation. One complete cycle of the hierarchical transformation is completed in three steps. The first step is to reduce the number of pixels diagonally by two in each frame, the second step is to reduce the number of frames by two, the third step diagonal downsampling again.

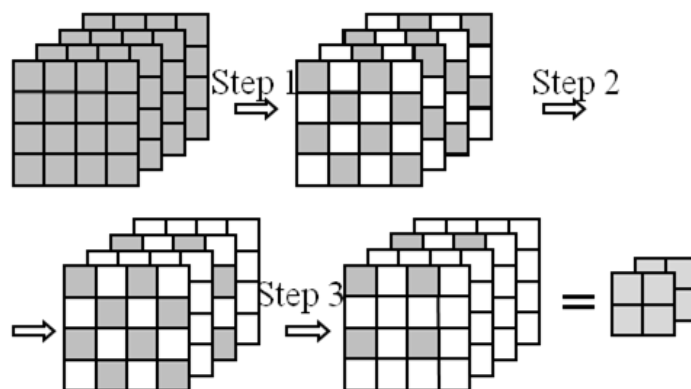


Figure 4.13 Possible downsampling method to maintain equal resolution in all directions

4.5 Conclusion of the Chapter

Each cell belongs to one of the few hundred cell types. The resulting new cells occupy an available new location, position in the 3 dimensional space, it resembles a kind of image.

Regardless that the original signal was a 1D, 2D, 3D or even higher dimensional signal or correlated dataset, it is converted to a binary tree structure. Seemingly the transformed image preserves its original structure, but actually they are binary tree structures. Data manipulating software usually use matrix format, but even that is stored some kind of linear series. The information content necessary for embryogenesis is stored somehow in the chromosomes as DNA base sequence. Since DNA is a chain, we can consider it as a 1D dataset. Further transformation and compression are potentially achievable on this binary tree shape transform.

The method of downsampling, i.e. spatial orientation, has great influence on filter design. The nearest is the pixel used for approximation the higher is the correlation between them.

5 CHAPTER WAVELET REPRESENTATION OF CELL DIVISIONS AND DIFFERENTIATION

5.1 Classification of Cell Divisions

There are two types of cell divisions: symmetric and asymmetric cell divisions, but in either case at least one of the daughter cells should look like as their predecessor.

5.2 Symmetric Cell Divisions

After completion of a mitotic cell division cycle of a eukaryotic organism it produces two identical daughter cells.

Type “*a*” cell yields two “*a*” type cells, Figure 5.1.

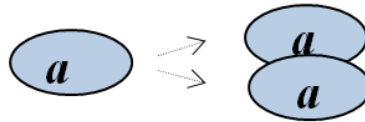


Figure 5.1 Symmetrical cell division

5.3 Asymmetric Cell Divisions

Contrary to symmetric cell division the resulting daughter cells are of different types. One would be the same like their precursor and the other one would be different type [14].

In case of **intrinsic asymmetry** type “*a*” cell would develop and divide into one “*a*” and one different “*b*” type immediately, Figure 5.2.

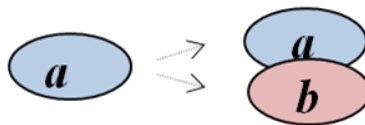


Figure 5.2 Asymmetric cell division

In the case of **extrinsic** (induced) **asymmetry**, daughter cells are similar directly after cell division but later their environment could induce differentiation, one of them could become different by interaction or by signaling with its neighbours [27]. An example for induced asymmetry is the way how ciliated epithelium cells emerge in *Xenopus laevis* embryos. Cells segregate to adopt a predicted particular fate by following a special patterning rule [28], [29].

Third possible scenario is, when “*a*” gives “*b*” and “*c*”, Figure 5.3:

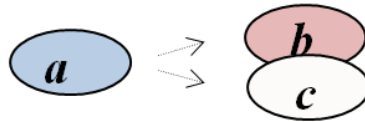


Figure 5.3 Nonconventional (abnormal) cell differentiation

This is stem cell behaviour, feature for repair mechanism and possibly for cancer development, but not characteristic attribute in developmental biology, therefore this scenario is not part of the further considerations.

Cell divisions are categorized as symmetrical or asymmetrical cell divisions. Symmetrical cell divisions always yield two identical daughter cells, contrarily to asymmetric cell division, by which one of the daughter cells would be different to its predecessor [27]. Cell differentiation has two types, intrinsic and environmentally induced differentiation [30]. The number of different cell types we could categorize cells are between 100 and 200. After any mitotic cell division one of the siblings is always belongs to the same type as the original and the other could be different or the same as their common predecessor.

For example if we have a cell, let call it type “A”, in case of symmetrical cell division it produces two type “A” new cells, or in case of an asymmetric cell division, it would divide into a type “A” and a different type, let us call it type “B” cells.

0: Symmetrical cell division: ‘A’ → ‘A’ + ‘A’

1: Asymmetrical cell division: ‘A’ → ‘A’ + ‘B’

Figure 5.4 shows, that cell divisions follow a perfect binary tree structure. Starting from a single cell (after fertilization), it is marked as C1. After the first cycle there will be 2, after the next 4 ... and after the nth cycle the number of cells would be 2ⁿ.

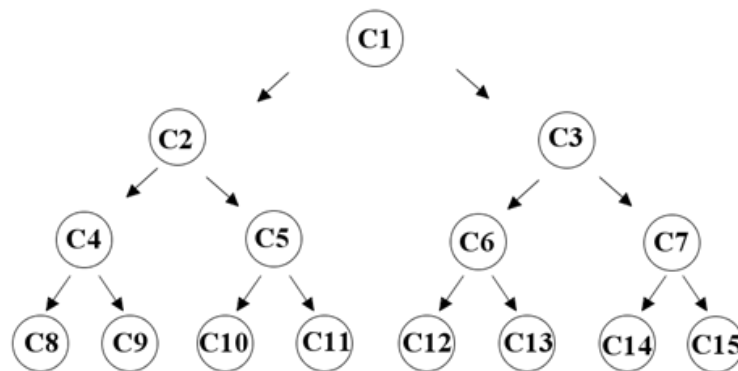


Figure 5.4 Cell divisions follow a perfect binary tree structure

What is the essence of this biological process from informatics point of view? Any cell division bears two daughter cells, for example the first cycle yields ‘C2’ and ‘C3’ from ‘C1’.

– 1st division: ‘C1’ → ‘C2’ + ‘C3’

Cell divisions could be either symmetrical or asymmetrical:

- if it is symmetrical, then 'C2'='C3'='C1',
- if it is asymmetrical, then 'C2'='C1' but 'C3'≠'C1'.

So we can notice that regardless of a cell division is symmetrical or asymmetrical, 'C2' always should be the same type of cell as its ancestor 'C1' was. 'C3' should be the same as 'C1' was in case of a symmetrical cell division or should be different if the cell division was an asymmetric one.

In this short example after the 3rd cycle we will have 8 new cells: 'C8'...'C15'. Figure 5.5 shows the same speculative binary tree cell fate map what we have seen in Figure 5.4 previously, but considering the possible similarity of the daughter cells to their predecessors. Following the previously described process, 'C8' cell should be the same type as 'C1' was, 'C10' should be the same as 'C5' was, etc. (We could conclude that 'C1' → 'C2' → 'C4' → 'C8' are stem cells and it is most probably the germ line.)

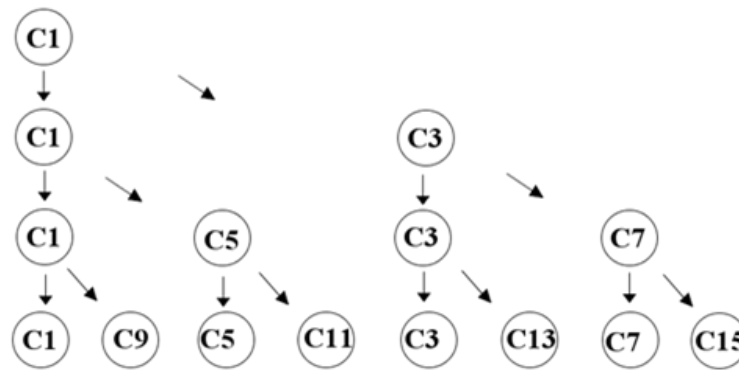


Figure 5.5 Cell divisions tree, considering cell differentiation

'C1' contains the compressed form of the information (database) about the structural feature of the entire individual organism. So ontogenesis and morphogenesis could be considered as a type of decompression and inverse transformation method of a highly compressed database or 'image'. Our task is to find out the method how this information was compressed. Of course in nature it took millions of years to develop this compression gradually.

5.4 Cell Division Tree Relation to Image Transformation

When we look at the cell division-cell differentiation biological process from the output end, this "reverse cell division" could be analyzed and considered as an image transformation. Figure 5.6 shows what is the main difference between this reverse biological progression and the general discrete wavelet transformation (compare to Figure 2.2). It means that low-pass filtering degenerates to a simple downsampling, because after cell division anyhow one of the daughter cells should be

identical to their common predecessor regardless of the type of cell division was symmetrical or asymmetrical.

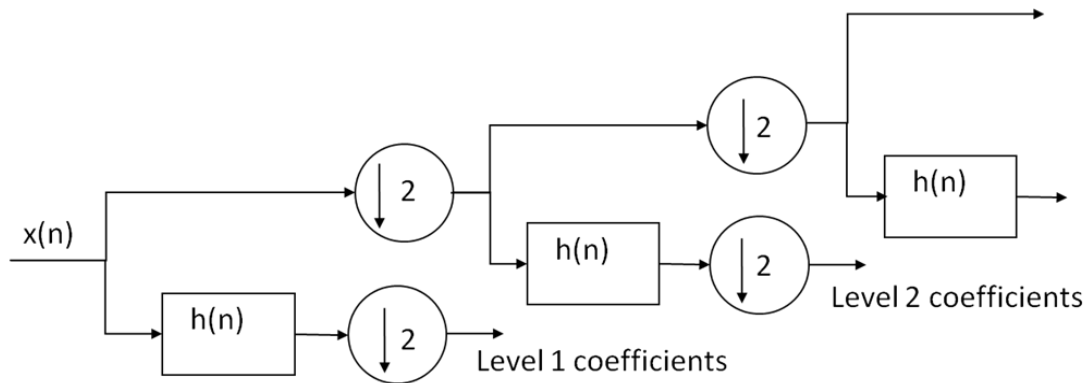


Figure 5.6 “Reverse cell division” representation with cascading filter bank

The author’s aim is, to find suitable matching $h(n)$ functions, fulfilling certain criteria. The most similar wavelet transform is the Haar wavelet [31], [32]. The main difference between these two is, that Haar wavelet transform’s $g(n)$ function calculates the average of two consecutive elements, however this transformation takes simply one of the two elements:

$$g(n) = x(n) \tag{5-1}$$

Haar transform is called as the simplest of the wavelet transforms, but the transformation introduced by nature through billions of years seems to be even simpler.

2x2 Haar matrix:

$$\begin{bmatrix} \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{-1} \end{bmatrix} \tag{5-2}$$

2x2 matrix of reverse cell division transform:

$$\begin{bmatrix} \mathbf{1} & \mathbf{1} \\ \mathbf{0} & \mathbf{-1} \end{bmatrix} \tag{5-3}$$

or even it could be simple downsampling, if it is accompanied with a fitting compression method:

$$\begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \tag{5-4}$$

5.5 Cell Division Tree Relation to Image Pixel Numbering

Example dataset is shown in Figure 5.7, predicted value of $a_{1,0}$ is $a_{0,0}$ and so on:

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$
$a_{2,0}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$
$a_{3,0}$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$

Figure 5.7 An example image of 4x4 pixels

By renaming $a_{0,0}$ as C_{16} and so on, we get values from C_{16} to C_{31} as the following (Figure 5.8):

	$y_1=0; y_0=0$	$y_1=0; y_0=1$	$y_1=1; y_0=0$	$y_1=1; y_0=1$
$x_1=0; x_0=0$	$C_{16}=a_{00,00}$	$C_{18}=a_{00,01}$	$C_{24}=a_{00,10}$	$C_{26}=a_{00,11}$
$x_1=0; x_0=1$	$C_{17}=a_{01,00}$	$C_{19}=a_{01,01}$	$C_{25}=a_{01,10}$	$C_{27}=a_{01,11}$
$x_1=1; x_0=0$	$C_{20}=a_{10,00}$	$C_{22}=a_{10,01}$	$C_{28}=a_{10,10}$	$C_{30}=a_{10,11}$
$x_1=1; x_0=1$	$C_{21}=a_{11,00}$	$C_{23}=a_{11,01}$	$C_{29}=a_{11,10}$	$C_{31}=a_{11,11}$

Figure 5.8 Pixel numbers and the corresponding binary tree identification numbers

By carefully investigating Figure 5.8, we can notice a regularity, how to generate a 1D dataset from a 2, 3 or higher dimensional dataset. On this figure purposely the matrix coordinates are written in binary code. Let us write the C pixels index numbers in binary code too. For example C_{17} 's index written in binary is C_{10001} , it is the same as $a_{01,00}$ pixel of a two dimensional image. $C_{1y_1x_1y_0x_0}$ (Usually the matrix indexing or in image processing x axis of a picture runs vertically, y axis runs horizontally.) After the first downsampling C_{16} and C_{17} would be replaced by C_8 see Figure 2.1 (in indexing it will become $C_{1y_1x_1y_0}$). In the case if we have more than two dimensions, example x , y and t , the indexing to convert the dataset into a one-dimensional set: $C_{1t_ny_nx_n \dots t_1y_1x_1t_0y_0x_0}$.

5.6 Conclusion of the Chapter

Considering cell divisions and differentiations like a kind of wavelet transformation seems to help us to describe this process. This kind of wavelet transformation or more precisely inverse transformation is the simplest form of all wavelets.

Cell division binary tree comprises to wavelet transformation, if we consider it as an inverse wavelet transformation. The most similar known wavelet is the Haar wavelet. The main distinction comparing to Haar wavelet: instead of averaging, this one uses purely downsampling only. Averaging could introduce strange values, which did not exist in the original dataset and does not support comparison to predicted values. Any organism we can observe is functioning in all its form during their embryonic development from the zygote unto its fully developed form. The emerging image also resembles to its highest resolution form.

6 CHAPTER ALGORITHM FOR DISCRETE DATA AND SIGNAL PROCESSING

6.1 Cell Division Tree Described as Inverse Wavelet Transformation

Dataset or signal transformations like Fourier, FFT, wavelet etc. transformations transform an image, any set of data into some kind of components and represent each component by its coefficients. Figure 6.1 shows **C8...C15** cells as an image, its coefficients are **D1...D7** and initial value is **C1** cell (in the original context it is the zygote cell itself).

Wavelet transformation values and coefficients

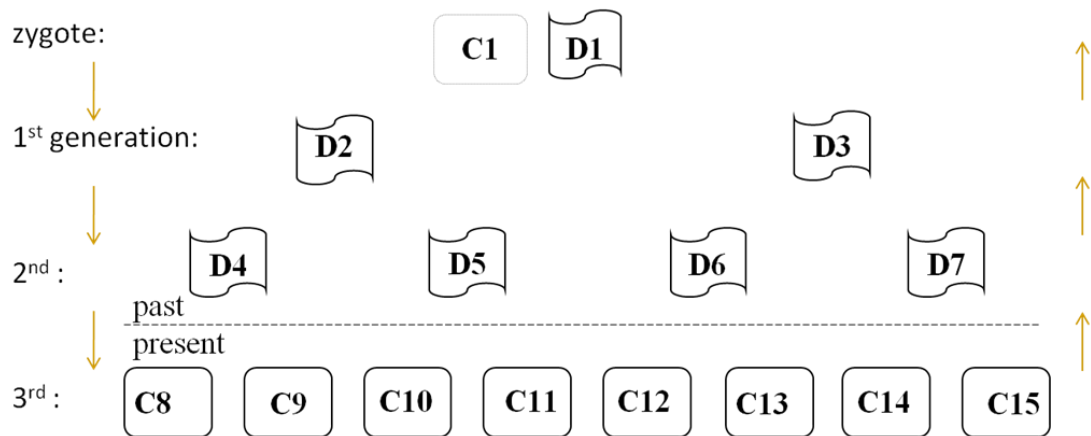


Figure 6.1 Matchup of wavelet transformation coefficients with dividing cells

Why is it beneficial to nature to use any kind of transformation? In the case of symmetric cell division, the corresponding coefficient would be equal to zero, in case of induced asymmetry it become also 0, because the new cell will take form of a predicted model. In the Introduction chapter we highlighted, that body plan should be a well compressed data pack. The aim of any transformation method is, to produce low value or almost zero coefficients. The transformation described here also produces mostly zero value coefficients, making the compression simple and efficient.

6.2 Wavelet Transformation (Haar Wavelet)

The cell differentiation process is a kind of inverse wavelet transformation, as there is correlation between DNA replication binary tree and cell lineages binary tree (spatial arrangement of differentiated cells) [33].

Haar wavelet transformation is considered to be the simplest one amongst wavelet transformations, but this one, invented by Mother Nature is even simpler.

Haar wavelet takes the average and the difference of two adjacent element of a dataset. This wavelet transformation takes one element of a pair (simple downsampling) and their differences. Furthermore in the case of induced asymmetrical cell division, the cell identity is determined by signaling and communication with its neighbors, so this means it obtains a predicted value. This is the most remarkable difference between these two transformations. Haar wavelet cannot make comparison to a predicted value. Example: two elements are: a and b , their average is $(a+b)/2$ and their difference is $a-b$, knowing these two values we are able to reconstruct the original a and b elements.

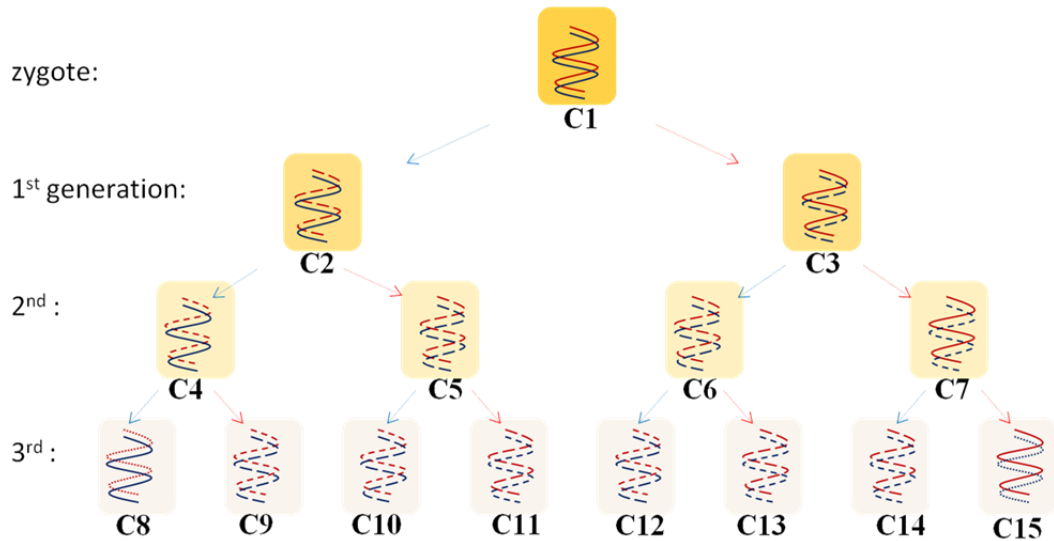


Figure 6.2 Cell division binary tree

This transformation takes one of them, say a and their differences $a-b$, or the difference to a p predicted value $p-b$. When this prediction is quite accurate, this difference could be negligible or zero. There is another factor in nature that could further simplify or reduce the amount of information to be stored in the genome. Like a “state machine”, a certain cell type can differentiate into a small number (or only to another) of other cell types. Only stem cells are able to differentiate to a variety of different cell types.

6.3 Cell Division Tree, Illustration of Cell Differentiation on Binary Tree

In paragraph 3.4 we had seen, that the cell division’s binary tree has two major branches: when the main branch is A or when the main branch is in T direction (i.e. whether the cell which bears the trait of its ‘parent’ contains the original A or T DNA strand). This is not known yet, if nature has any preferred direction, or even possible, that some part of an organism development follows one particular direction and some part of the same organism follows the other. In Figure 3.7.d for example is marked $C8$, $C10$, $C12$ and $C14$ as the cells of the vegetal pole and $C9$, $C11$, $C13$ and $C15$ cells as

animal pole cells, they might follow different traits, or this could be the case with insects, butterflies during metamorphism, they might change direction in their development after pupa stage.

Take an example when the main trait is the stream of the original A-s (Figure 6.3).

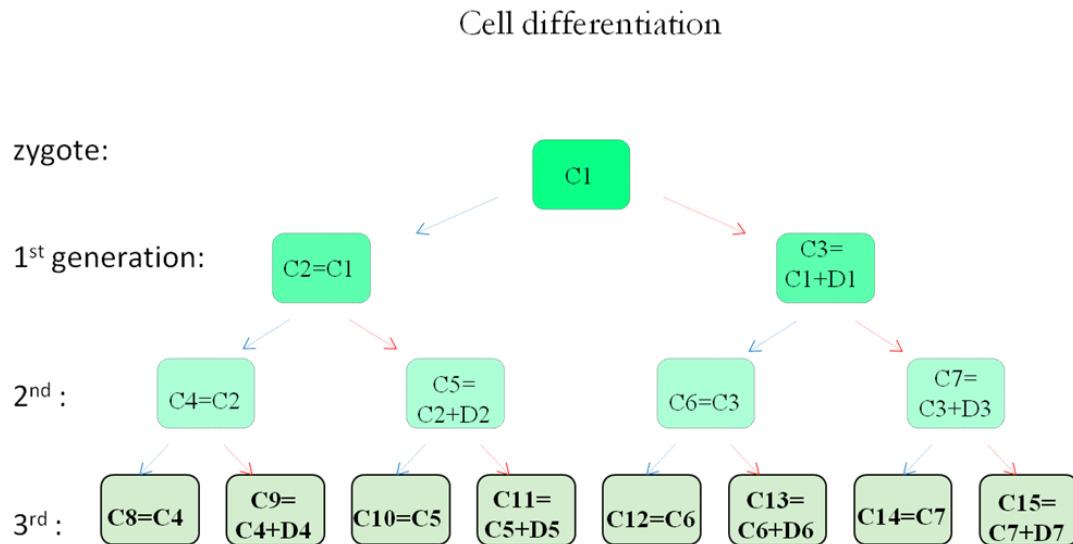


Figure 6.3 Cell differentiation

Let us examine the first cell division cycle, regardless that it is symmetric or asymmetric cell division, anyhow one of the progenies would be similar to the original. $C2$ is similar to $C1$, $C3$ might be the same or might be different, depending its fate coded in this organism's body plan. $C3=C1+D1$

In case of symmetric cell division $C3=C1$, so $D1=0$ or in case of asymmetric cell division $C3 \neq C1$ and this makes $D1 \neq 0$. And so on... in the moment illustrated in Figure 6.3, the actual cells are $C8...C15$, $C1...C7$ cells were the previously existed stages. In this example $C8=C4=C2$ $C10=C5$, $C12=C6=C3$ and $C14$ is definitely similar to $C7$ ($C14=C7$). Values (cell type) of $C9$, $C11$, $C13$ and $C15$ cells depend on $D4$, $D5$, $D6$ and $D7$.

6.4 Predicting the Value of a Discrete Variable

We have a finite set of variables, in case when the elements of this dataset are correlated to each other, then we are able to predict any element of this set. The elements of any dataset are usually arranged into a structure by some parameter, as time, physical position or any possible criteria. The most common way of the prediction is usually the comparison by its neighbor or neighbors.

We can get better estimation of an element, by fine-tuning this prediction, as a result the value of D difference could be less or increases the possibility of zeros. The

easiest way to find a good prediction is spatial filtering of existing elements. This is the technique that makes this kind of transformation more useful than the well known Haar wavelet.

6.5 Transformation

The purpose of the transformation is to replace dataset elements by the extent of deviation from their predicted values. We have assumed that the elements are correlated to each other. To achieve this, we created a tree structure by downsampling.

We have A , a set of data to be compressed. To understand this algorithm better, let the size of the set be the power of 2. $\{A\}=2^n$

$$a_1, a_2, \dots, a_{2k-1}, a_{2k}, \dots, a_{2^n-1}, a_{2^n}$$

For demonstration purposes we used the easiest prediction, the comparison to its neighbors. For example we can compare any value of the dataset to the average of its two neighbors, predicted value of a_2 : $p_2=(a_1+a_3)/2$, for a_4 : $p_4=(a_3+a_5)/2$ and so on, a_{2k} : $p_{2k}=(a_{2k-1}+a_{2k+1})/2$

Deviations from the predicted value:

$$d_2 = a_2 - p_2, d_4 = a_4 - p_4, \dots, d_{2k} = a_{2k} - p_{2k}, \dots, d_{2^n} = a_{2^n} - p_{2^n}$$

These predicted values could be generated by digital filtering methods in the case of a digital wavelet transformation. The uniqueness of this cell division inspired method is, that besides filtering, predicted values could be generated by any suitable algorithm.

We can proceed the same algorithm with the remaining elements in n steps, finally retaining only a_1 . Let see an example:

Initial set of data:

$$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, a_{14}, a_{15}, a_{16}$$

After 1st step:

$$a_1, a_3, a_5, a_7, a_9, a_{11}, a_{13}, a_{15} \qquad d_2, d_4, d_6, d_8, d_{10}, d_{12}, d_{14}, d_{16}$$

2nd step:

$$a_1, a_5, a_9, a_{13} \qquad d_2, d_4, d_6, d_8, d_{10}, d_{12}, d_{14}, d_{16}$$

$$d_3, d_7, d_{11}, d_{15}$$

3rd step:

$$a_1, a_9 \qquad d_2, d_4, d_6, d_8, d_{10}, d_{12}, d_{14}, d_{16}$$

$$d_3, d_7, d_{11}, d_{15}$$

$$d_5, d_{13}$$

4th step:

$$\begin{array}{cccccccc}
 d_2, d_4, d_6, d_8, d_{10}, d_{12}, d_{14}, d_{16} & & & & & & & \\
 d_3, & d_7, & d_{11}, & d_{15} & & & & \\
 d_5, & & & d_{13} & & & & \\
 & & & & & & d_9 &
 \end{array}$$

a_1

After the last step we got a_1 initial value and a set of coefficients from d_2 to d_{16} .

The practical implementation of discrete wavelet transformation was described earlier in paragraph 2.3. The undesirable or objectionable feature of that method is to perform instruction cycles. Numerical programs like Matlab execute matrix operations and manipulations faster than cycles. We can create transformation matrix T , which is capable to perform the downsampling and approximation in one run. T transformation matrix itself works with vertical downsampling, T^T transpose in horizontal spatial direction.

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \dots & 0 & 0 \\ \vdots & & & & & \dots & 0 & 0 \\ \dots & \dots & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & -1 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & \dots & 0 & 0 \\ \vdots & & & & & \dots & 0 & 0 \\ \dots & \dots & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

6-1

$$\mathbf{T}^T = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

6-2

P is a picture or dataset to be transformed, P' would contain the approximation coefficients and D submatrix for the detail coefficients.

$$T \cdot P = \begin{bmatrix} P' \\ D \end{bmatrix} \tag{6-3}$$

$$P \cdot T^T = [P', D] \tag{6-4}$$

6-1 equation works in x direction and 6-2 in y direction. One-one example is shown for each.

Transformation in x direction:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} \end{bmatrix} = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \\ a_{1,1}-a_{2,1} & a_{1,2}-a_{2,2} & a_{1,3}-a_{2,3} & a_{1,4}-a_{2,4} \\ a_{3,1}-a_{4,1} & a_{3,2}-a_{4,2} & a_{3,3}-a_{4,3} & a_{3,4}-a_{4,4} \end{bmatrix}$$

6-5

and an example in y direction:

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & -1 \end{bmatrix} = \begin{bmatrix} a_{1,1} & a_{1,3} & a_{1,1}-a_{1,2} & a_{1,3}-a_{1,4} \\ a_{2,1} & a_{2,3} & a_{2,1}-a_{2,2} & a_{2,3}-a_{2,4} \\ a_{3,1} & a_{3,3} & a_{3,1}-a_{3,2} & a_{3,3}-a_{3,4} \\ a_{4,1} & a_{4,3} & a_{4,1}-a_{4,2} & a_{4,3}-a_{4,4} \end{bmatrix}$$

6-6

6.6 Inverse-Transformation

Following this process reversely we can get back the original information, database or image, which is the inverse-transformation, similar to the example we could observe in nature. The previously described transformation is completely lossless, but it could be modified for lossy transformation, if compression is more important than fidelity [34].

During restoration of the original image we have to use the inverse of the transformation matrix, i.e.:

$$\mathbf{P} = \mathbf{T}^{-1} \cdot \begin{bmatrix} \mathbf{P}' \\ \mathbf{D} \end{bmatrix} \quad 6-7$$

$$\mathbf{P} = [\mathbf{P}', \mathbf{D}] \cdot (\mathbf{T}^T)^{-1} \quad 6-8$$

For example in x direction:

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \\ a_{1,1}-a_{2,1} & a_{1,2}-a_{2,2} & a_{1,3}-a_{2,3} & a_{1,4}-a_{2,4} \\ a_{3,1}-a_{4,1} & a_{3,2}-a_{4,2} & a_{3,3}-a_{4,3} & a_{3,4}-a_{4,4} \end{bmatrix}$$

6-9

and in y direction:

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} \end{bmatrix} = \begin{bmatrix} a_{1,1} & a_{1,3} & a_{1,1}-a_{1,2} & a_{1,3}-a_{1,4} \\ a_{2,1} & a_{2,3} & a_{2,1}-a_{2,2} & a_{2,3}-a_{2,4} \\ a_{3,1} & a_{3,3} & a_{3,1}-a_{3,2} & a_{3,3}-a_{3,4} \\ a_{4,1} & a_{4,3} & a_{4,1}-a_{4,2} & a_{4,3}-a_{4,4} \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

6-10

In the case when we want to filter the signal, the signal filtering can be addressed also with matrix operations. The simplest case is, when a pixel value is estimated or predicted by the average of its two neighbors ($\frac{1}{2} \ 0 \ \frac{1}{2}$) like in paragraph 6.5. The filtration is carried out by using the matrix F in x direction (equation 6-12) and its transpose in the y direction (equation 6-13).

$$F = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

6-11

$$F \cdot P = P_F$$

6-12

$$P \cdot F^T = P_F$$

6-13

where P_F is the filtered image. We can notice that the pixels of the last row or column have neighbor only at one side vertically or horizontally, obviously it is not possible to figure out average value.

For example in x direction:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} \end{bmatrix} = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ \frac{1}{2}(a_{1,1}+a_{3,1}) & \frac{1}{2}(a_{1,2}+a_{3,2}) & \frac{1}{2}(a_{1,3}+a_{3,3}) & \frac{1}{2}(a_{1,4}+a_{3,4}) \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \end{bmatrix}$$

6-14

and in y direction:

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} \end{bmatrix} \cdot \begin{bmatrix} 1 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} a_{1,1} & \frac{1}{2}(a_{1,1}-a_{1,3}) & a_{1,3} & a_{1,3} \\ a_{2,1} & \frac{1}{2}(a_{2,1}-a_{2,3}) & a_{2,3} & a_{2,3} \\ a_{3,1} & \frac{1}{2}(a_{3,1}-a_{3,3}) & a_{3,3} & a_{3,3} \\ a_{4,1} & \frac{1}{2}(a_{4,1}-a_{4,3}) & a_{4,3} & a_{4,3} \end{bmatrix}$$

6-15

Equation 6-16 shows an example of a higher order filter using quadratic spline interpolation $(-1/16 \ 0 \ 9/16 \ 0 \ 9/16 \ 0 \ -1/16)$. Because of obvious reasons the filter is always distorted near the edges. The size of this example image (the image is 8x8 pixel size only) is too small for this filter, only the fourth row (highlighted) contains the complete kernel of the filter.

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 9/16 & 0 & -1/16 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1/16 & 0 & 9/16 & 0 & 9/16 & 0 & -1/16 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1/16 & 0 & 9/16 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1/8 & 0 & 9/8 & 0 \end{bmatrix}$$

6-16

There are different methods to solve the problem when the filter is overlapping the edge of the picture. For example we can adjoin black or white pixels outside the boundary or mirroring the last rows or columns like it was done above.

6.7 Practical Implementation of the Transformation

The best way to demonstrate this algorithm is through the example of an image transformation. Figure 6.4 is a 256 by 256 pixels size 8 bit gray scale image, we can get its transform Figure 6.5 after “Cell division” type transformation. For generating a predicted value we have used the average value of two neighbors of a_{2k} : $p_{2k} = a_{2k-1} + a_{2k+1}$.



Figure 6.4 Example picture 256x256 pixels



Figure 6.5 Cell division type wavelet transform of Figure 6.4.

By the help of the transformed image we can restore the original image by inverse transformation as it is shown step by step in Figure 6.6, starting from the one pixel stage unto the next to the last stage (from 1st to (n-1)th step 256x128 pixels image).

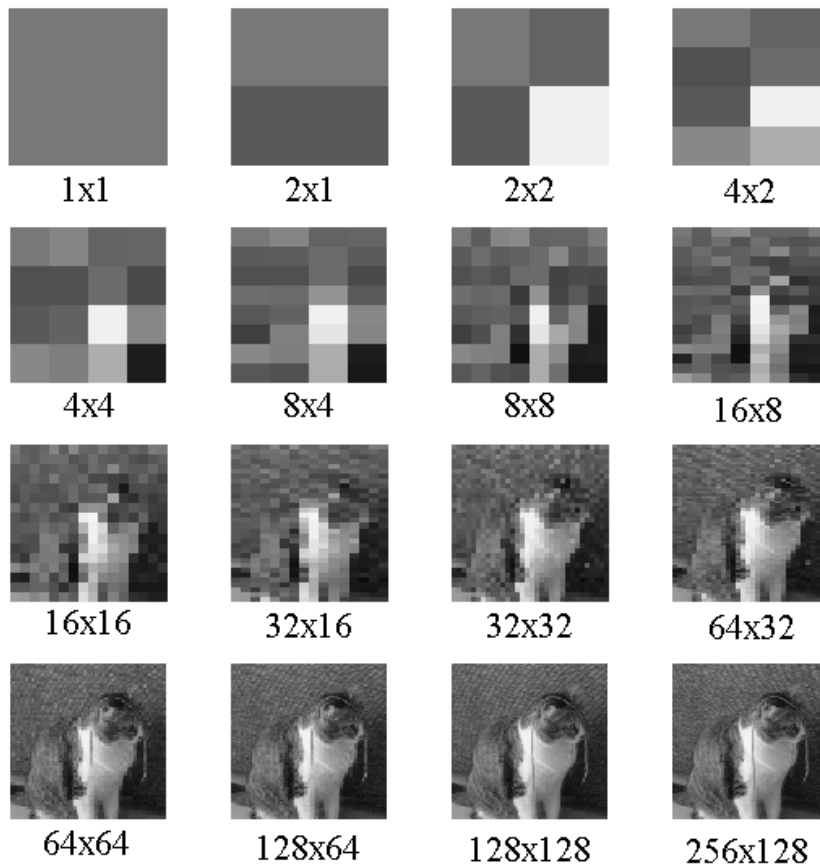


Figure 6.6 Inverse cell division based wavelet transform

Figure 6.7 presents the Haar wavelet transform of the same image. What are the differences between this cell division based transformation and the well known Haar wavelet transformation?



Figure 6.7 Haar wavelet transform of the same image

The step by step inverse Haar transform of the picture is shown in Figure 6.8. Comparison of the two transformed images does not reveal significant differences, but comparing the inverse transformation progression sequences, the difference is obvious.

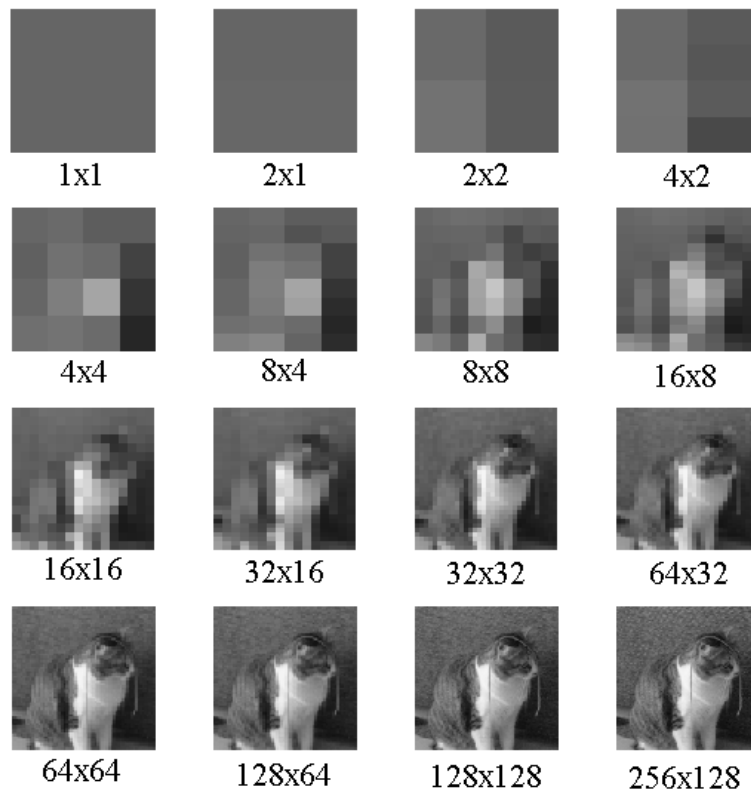


Figure 6.8 Inverse Haar wavelet transform of the image

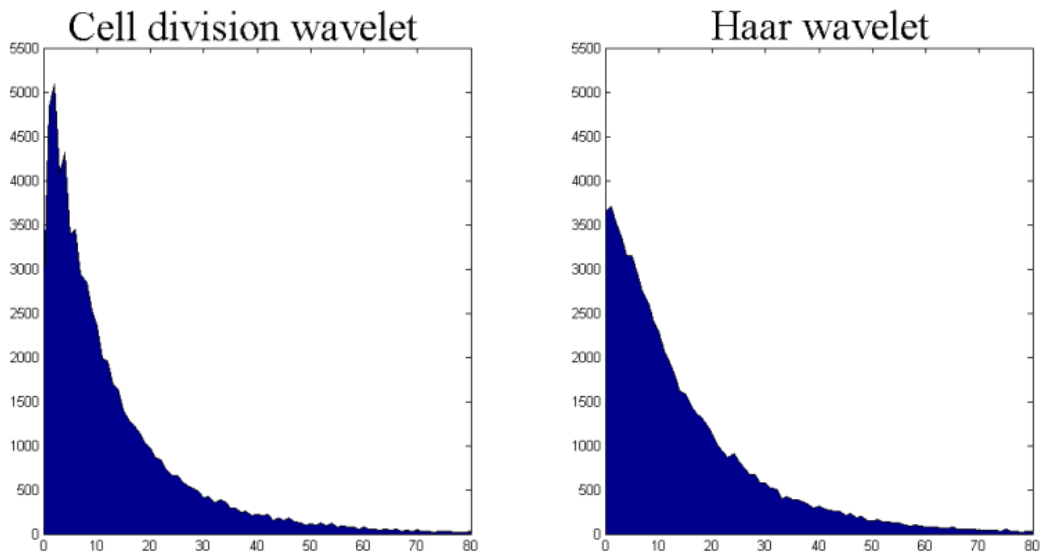


Figure 6.9 Histogram of the cell division and Haar wavelet

Efficiency of transformation could be demonstrated by comparing the histogram of the two transformed images. Cell division wavelet transform coefficient distribution shows, that its values are smaller than the Haar wavelet coefficients (Figure 6.9). (To be fair, to compare similar to similar, Haar wavelet was compared to the simple two neighbor averaging method.) The average absolute deviation of the cell division wavelet in this case was 6.44 and for the Haar wavelet transform it was 7.59. Using advanced prediction method (example spline interpolation or some type of pattern recognition [35]) the transformation could be more effective.

6.8 Comparing New Method to Existing (Haar) Transformation

There is desirable to have methods for datasets compressions for storage and transmission purposes. When we have a structured or correlated discrete dataset, we have a better chance to find a suitable algorithm. Figure 6.11 represents the cell division wavelet transform of Figure 6.10 color picture.

In Figure 6.13 and Figure 6.12 we can compare the steps of the known Haar wavelet inverse transformation and the way how a multicellular organism develops. When we start the comparison from the last subplot (256x128 pixel phase), we can notice, that by Haar wavelet transform we always get the average of two adjacent pixels, by cell division wavelet – because of the downsampling effect – we always get a previously existed value. In Figure 6.12 color picture series starting from the end we always find previously “known” colors, contrarily in Figure 6.13 going through from the last to the first subplot every pixel of the new image is a newly introduced color, false color.

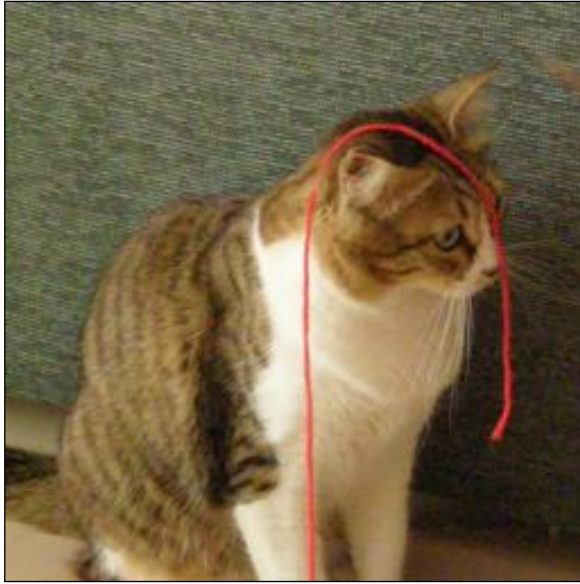


Figure 6.10 Sample color picture



Figure 6.11 Cell division wavelet transform of the sample image

Developing an image or dataset transformation method, which is able to provide low coefficient values, is a definite advantage in the case when we would like to apply any further dataset compression method.

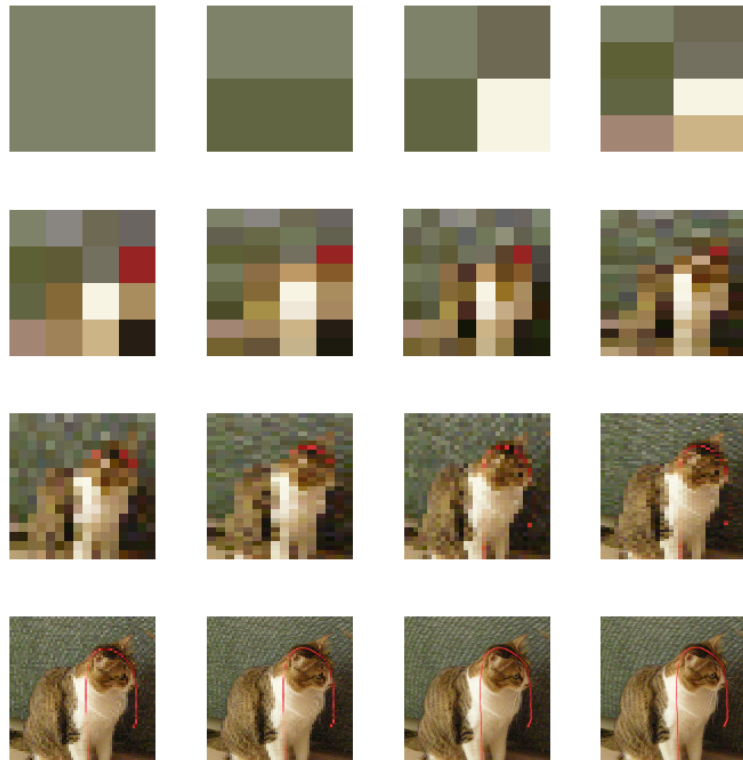


Figure 6.12 Inverse cell division wavelet transform of the color image

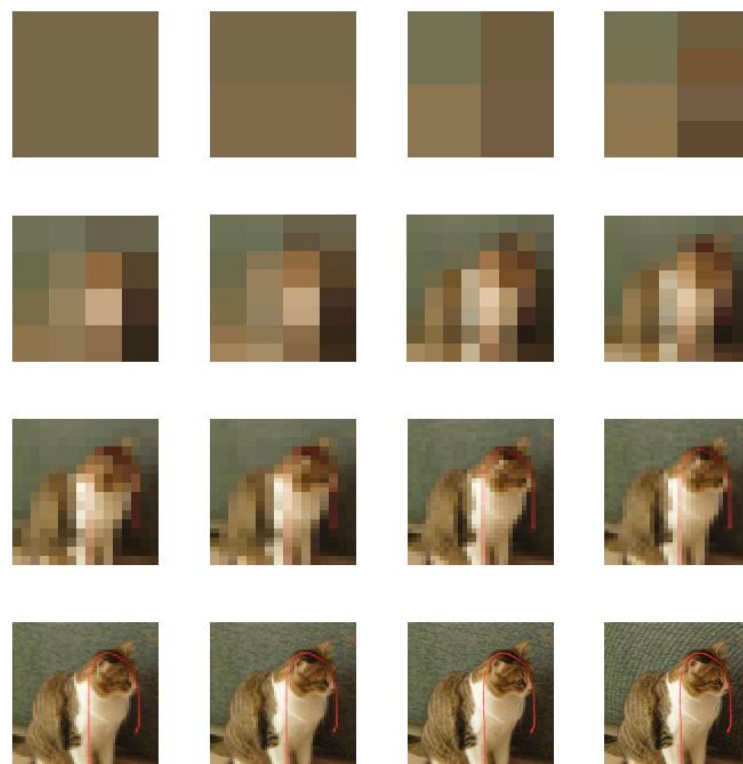


Figure 6.13 Inverse Haar wavelet transform of the color picture

6.9 Conclusion of the Chapter

This chapter is about an algorithm, how to transform and compress discrete dataset for storage and transmission purposes. The transformation is based on the same algorithm how a multicellular organism develops from a single cell by consecutive cell divisions and differentiations. This algorithm could provide an economical way for handling a correlated or highly correlated discrete or quantized analog signal, image or any multidimensional data set. The means of this algorithm is a special binary tree structure, inspired by developmental biology, the way, how a multicellular organism develops from a single cell. This observed natural process is an inverse transformation so on this base is possible to prepare a corresponding wavelet transformation method.

7 CHAPTER ALGORITHM FOR DISCRETE DATA AND SIGNAL PROCESSING ENTROPY BASED HIERARCHICAL IMAGE COMPRESSION METHOD

7.1 Image Transformation before Applying Compression Method

Compression methods are very useful to reduce data volume for archiving or transmitting images, datasets or packages, to reduce memory space, or to reduce required transmission time duration. The extent of compression depends on the rate of correlation between nearby elements of the dataset to be compressed. There are significant differences between compression methods. The higher is the correlation (similarity) between elements of a neighborhood, the higher is the achievable compression ratio of the compressed signal compared to the original signal.

Entropy is a measure of information content. For example in Figure 7.1 *a* and *b* both are 16 grades gray scale images, each grade with equal probability, so the two images have the same entropy value, in our example this value is equal to 4. To achieve high compression ratio, entropy should be low. If entropy really relates to the measure of disorder, then entropy values should differ from case *a* and *b* images. Common sense dictate, that image *a* could be compressed more efficiently than image *b*, although both of them contain exactly the same number of each pixel values.

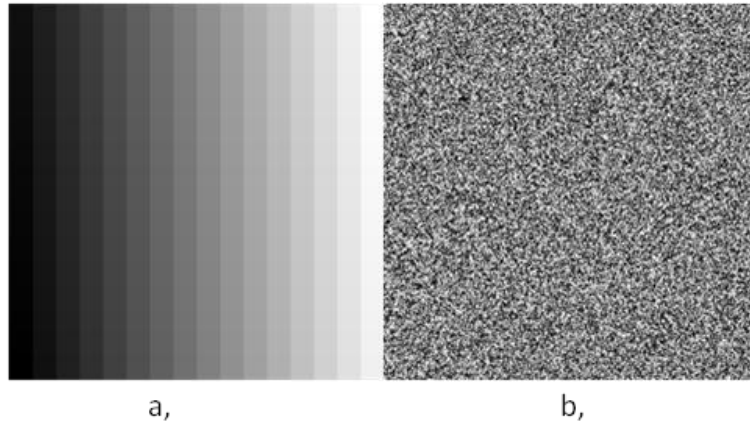


Figure 7.1 Entropy of gray scale images
a and **b** both are 16 grades gray scale images, each grade with equal probability, so the two images have the same entropy value (John Loomis)

Typically image entropy is defined as:

$$H(x) = - \sum_{i=1}^n p(x_i) \log_b(p(x_i)) \quad (7-1)$$

whereas $p(x_i)$ is the probability mass function of outcome x_i . (H is the entropy of gray scale image, p is the probability of a given intensity value in the image, if $b=2$, then we get the unit of entropy in bits.) [36]

Logically defining entropy, i.e. measure of disorder, a neighborhood should be taken into account, called local entropy [37]. For example in MATLAB command *entropy(image)* computes the entropy value (a single value) of a logical array or a gray scale image. However *entropyfilt(image)* command presents an array of entropy values, considering each pixel's 9-by-9 surrounding neighborhood, this output image is the same size as the input image was.

Compression methods are lossless or lossy. This bio inspired compression method is fully lossless, but could be modified to increase compression ratio further by modifying it to a lossy compression method. The requisites of this compression method are: the signal should be discrete and its region should be bounded.

To perform this kind of compression, the image or the digital (sampled, quantized and digitized) dataset should be transformed to a tree like structure [38]. Some of the hierarchical transformations, for example second generation wavelet transformations could generate binary tree structure transform of the signal or image.

An example for second generation wavelet transformation is shown in Figure 7.2, the signal is split into even and odd components.

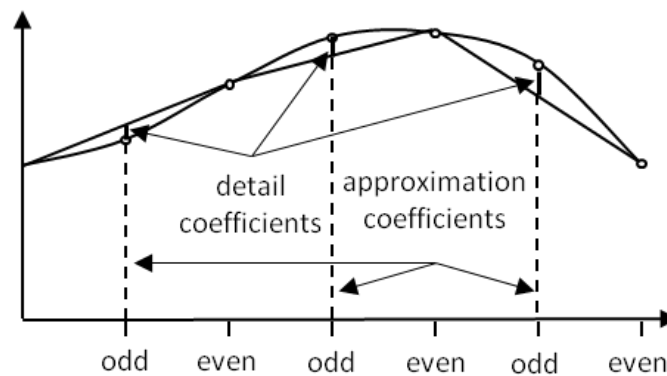


Figure 7.2 Example for second generation wavelet transformation

The odd components are predicted by approximated values, calculated by the help of neighboring even elements. These values are known as detail coefficients, even samples can be updated and replaced, that is called approximation coefficients. If the correlation is high between neighboring elements, than detail coefficient absolute values will be much smaller than the original signal values and preferably even zero. The aim is to attain as much as possible number of zeros in the binary tree. There are two means or bases for developing suitable compression methods, first of all to find areas where detail coefficients are zero and secondly to find a way to denote small values with less bits than their normal representation.

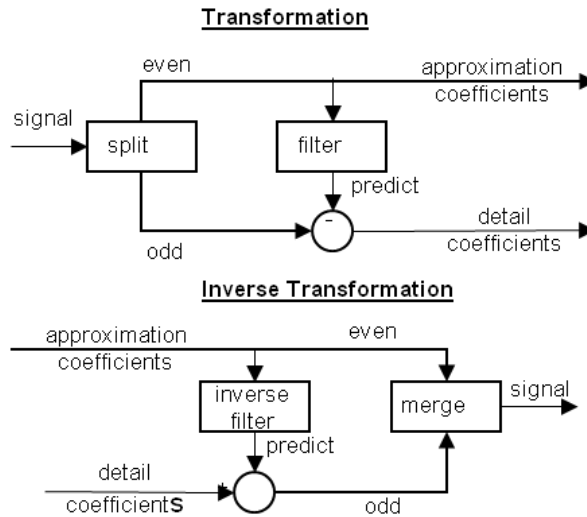


Figure 7.3 Second generation wavelet transformation.

Downsampling (splitting), filtering, predicting and producing detail coefficients.

Take the remaining approximation coefficients, divide these elements again into two groups: odds and evens and repeat previous procedure (Figure 7.3). This process will result a binary tree structure and the detail coefficients will be the element of this tree. Regardless the dimensionality of the original signal, image or dataset, the result would be a binary tree shape detail coefficients heap and a single remaining approximation coefficient (Figure 7.4). If the number of elements in the original dataset was: 2^N , then the number of steps to establish the complete binary tree is N , the number of detail coefficients are $2^N - 1$ (extreme case: $N=1 \rightarrow$ gives one detail coefficient and one approximation coefficient). In case the number of elements is not equal to the power of 2, then better to insert padding elements.

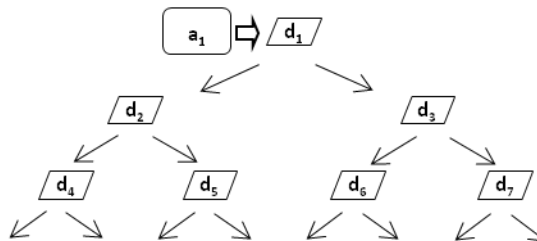


Figure 7.4 Detail coefficients of this bio-inspired wavelet transformation resemble to a perfect binary tree shape (“a” stands for approximation coefficient, “d” for detail coefficients)

If the original signal is not a 1D signal, it is better to maintain the shape of that database’s structure and transform it so, to attain a binary tree like structure [39]. For example a previously mentioned 2D transformation was shown in Figure 4.5 (the down sampled image) and Figure 7.5 shows the corresponding transform a stands for approximation coefficient, d for detail coefficients. Transformation is carried out consecutively along x then y , again x axis and so on. The actual representation of the transform is a perfect binary tree. (Assertion: the dimension of the dataset each direction should be a whole multiple of power 2.)

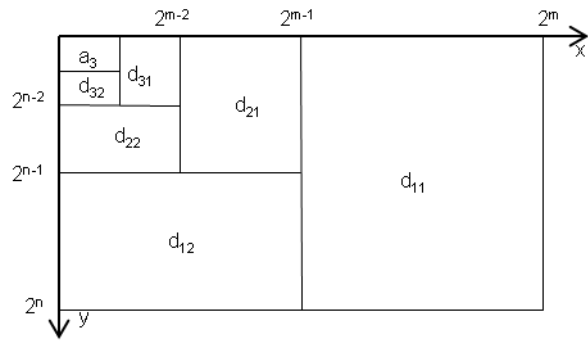


Figure 7.5 Image transformation in 2D,

7.2 Preparation: Image Transformation



Figure 7.6 Original image (Lena [40])

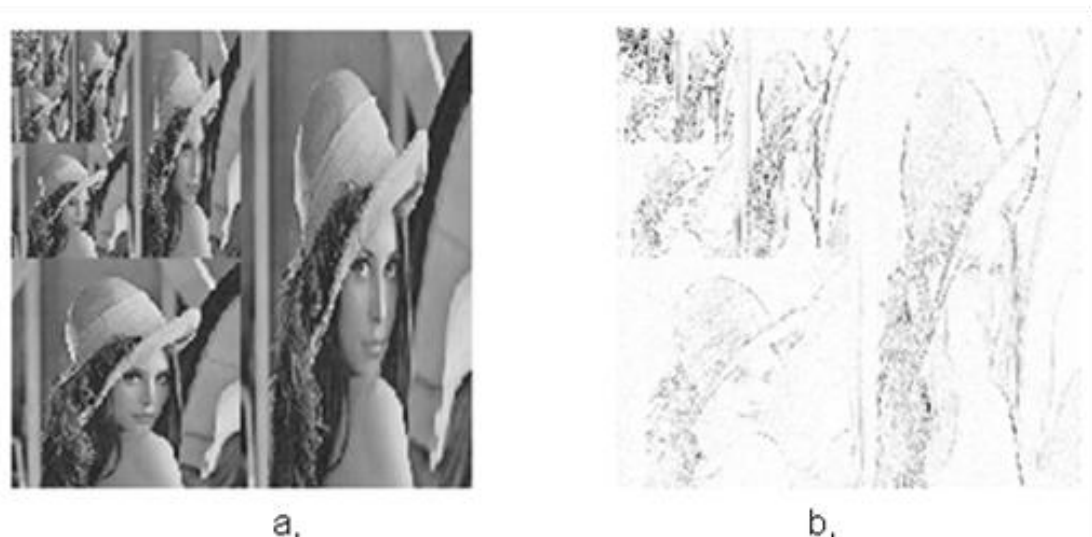


Figure 7.7 Image transformation in 2D

Fig. a contains the approximation coefficients and **Fig. b** the corresponding detail coefficients

Figure 7.7/b shows the transformed image of the original image (Figure 7.6) and Figure 7.7/a contains the approximation coefficients. Since the values of the detail coefficient matrix are close to zero or zero, the resulting image would be almost black, that is why the transformed image shown in negative in Figure 7.7/b.

A number of the detail coefficients i.e. several elements of the binary tree are 0 or close to 0 values if the pixels are correlated to their neighbors. When some part of an image is undisturbed or the pattern is predictable, then the corresponding branch of the detail coefficient binary tree contains only 0 values.

7.3 Compression

If we are certain that a branch of this tree contains only 0 values, then we can omit that branch of the tree. Let us introduce an internal variable H , this shows us whether a branch is disturbed or undisturbed. H is a kind of entropy value; let the entropy be $H=0$ when that branch of this tree contains only $d=0$ values and $H \geq 1$ when any of the detail coefficients is/are higher than 0 in that branch. Figure 7.8 depicts these entropy values besides the previous detail coefficients of the binary tree.

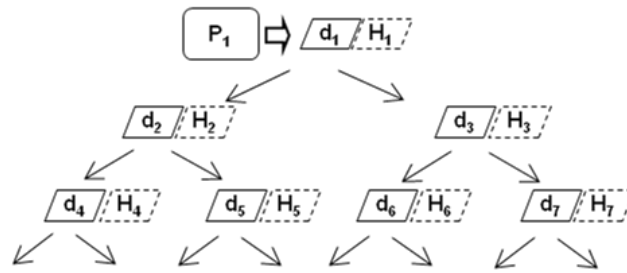


Figure 7.8 Entropy values at the top of the tree

P_1 is the first approximation element and d_1 to d_{2N-1} are the detail coefficients of the transformed image. " H " variable suggest the entropy level in the given branch of the binary tree

How to generate these entropy-like coefficients? Its value should be equal to 0, when all of the detail coefficients of that branch of the binary tree were zero and moreover it should have similar value like the detail coefficients from which it is generated.

$$H_n = \frac{d_n + \frac{H_{2n} + H_{2n+1}}{2}}{2} \quad (7-2)$$

$$D_n = d_n - \frac{H_{2n} + H_{2n+1}}{2} \quad (7-3)$$

$$E_n = H_{2n+1} - H_{2n} \quad (7-4)$$

There are two types of extreme cases: the top of the tree, d_1 and H_1 and the other end, at the lowest level, at the very beginning of the compression.

The H variable implies the entropy level in the given branch of the binary tree. At the lowest level let the H entropy value be equal to the d detail coefficients (Figure 7.9),

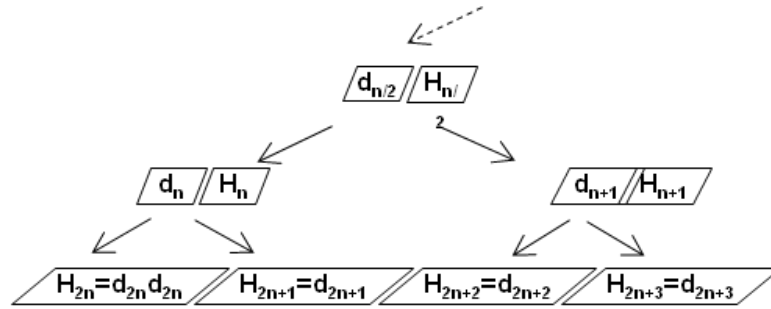


Figure 7.9 The end of a branch of the “ H ” entropy level binary tree and at the top of the tree is H_I :

$$H_1 = \frac{d_1 + \frac{H_2 + H_3}{2}}{2} \quad (7-5)$$

$$D_1 = d_1 - \frac{H_2 + H_3}{2} \quad (7-6)$$

$$E_1 = H_3 - H_2 \quad (7-7)$$

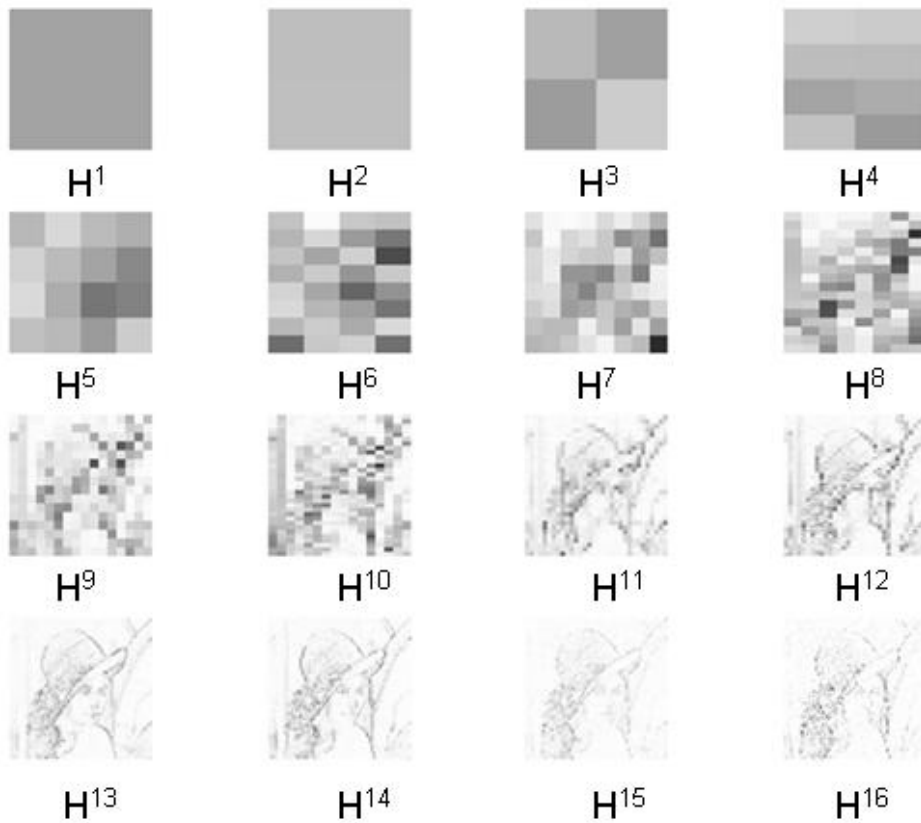


Figure 7.10 H entropy coefficients
 H^1 is the highest level, H^{16} entropy coefficients belong to the lowest level

Figure 7.10 shows the progression of the H entropy coefficients, while Figure 7.11 is a scaled visualization of it.

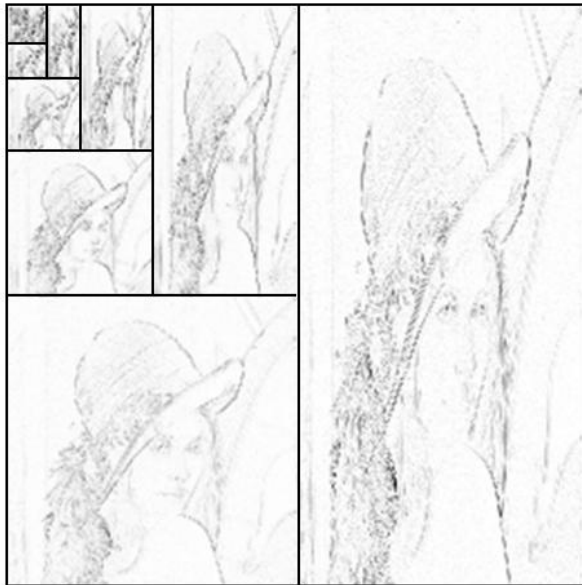


Figure 7.11 H coefficients of the scaled image

The progression of this compression starts from the lowest level. As it was mentioned previously, the H entropy coefficients are equal to the d detail coefficients. In order to get the higher level H entropy coefficients, we have to use Equation 7-2, and using Equations 7-3 and 7-4 successively we can generate E and D set of coefficients: $D_n = d_n - \frac{H_{2n} + H_{2n+1}}{2}$ and $E_n = H_{2n+1} - H_{2n}$. We should get these terms, because later on during image reconstruction we have to use these coefficients. The overall size of E and D sets should be as same as the corresponding d set was. These steps should be followed on and on... and finally we get a similar size new tree structure of these E and D entropy differences, shown in Figure 7.12. Lastly we get H_I , E_I and D_I . If the total number of pixels was 2^N , then number of elements of D and E sets both will be $2^{N-1} - 1$, together with H_I and P_I add together to 2^N exactly.

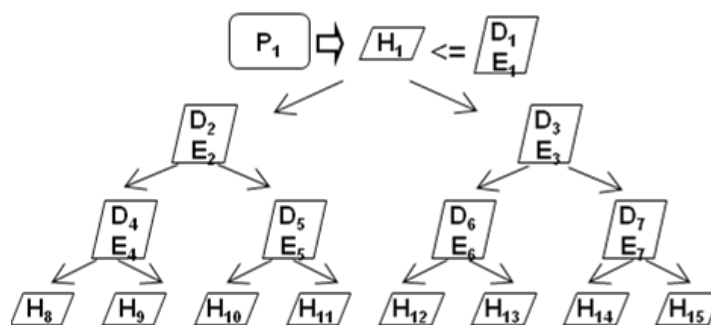


Figure 7.12 'D' and 'E' variables

'd' detail coefficient and 'H' entropy coefficients are replaced by 'D' and 'E' variables.

Applying this method on the Lena image we would get D and E entropy difference coefficients as it is in Figure 7.13 and in turn Figure 7.14 shows the scaled image of it.

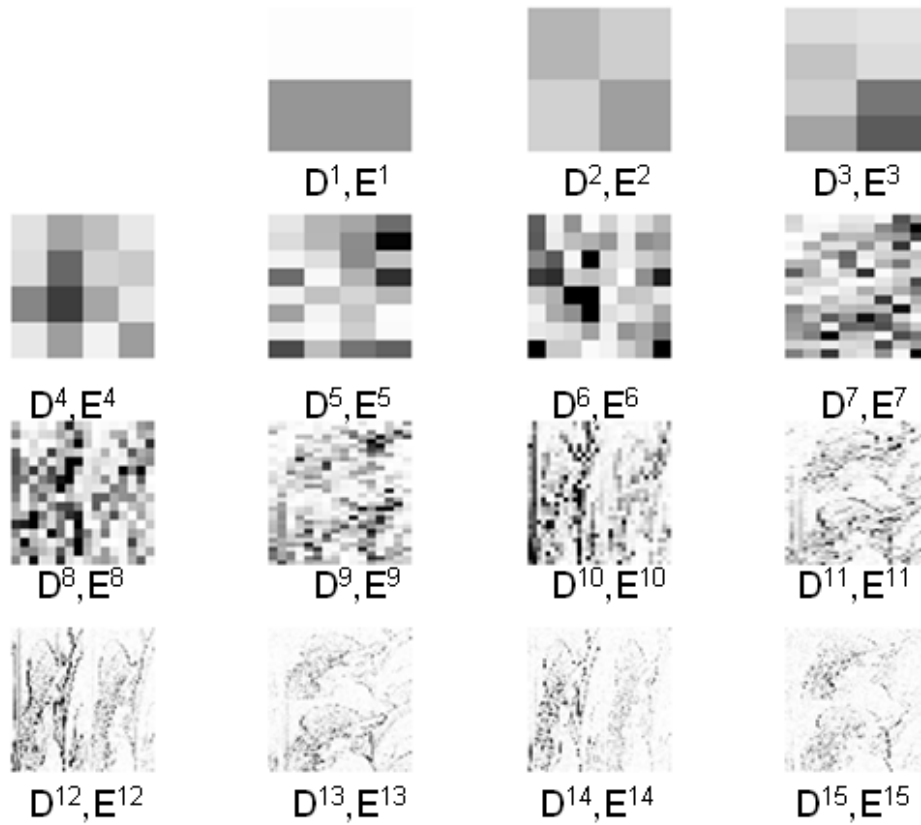


Figure 7.13 D and E entropy difference coefficients
 $E1$ and $D1$ are the highest level, $E15$ and $D15$ entropy coefficients belong to the lowest level

Why can we call it compression? Wherever entropy H is zero, this means that branch definitely does not contain any d detail coefficient, thus consequently all D and E entropy difference coefficients are also equal to zero and no need to save them. (Of course the opposite is not true, D value can be equal to zero even when H is not zero.) Altogether the maximum number of D and E elements plus the first approximation coefficient P_1 adds up to 2^N number of values, equivalent exactly to the number of pixels of the original image. The higher is the correlation between neighboring pixels or elements of a dataset, the more of the H values, therefore the more D coefficients could be neglected.

7.4 Decompression (restoration)

Decompression means inverting the same process, three steps, first reinstate H values by the help of D and E coefficients, starting with:

$$d_1 = \frac{2H_1 + D_1}{2} \quad (7-8)$$

$$H_2 = \frac{2H_1 - D_1 - E_1}{2} \quad (7-9)$$

$$H_3 = \frac{2H_1 - D_1 + E_1}{2} \quad (7-10)$$

and in general case:

$$d_n = \frac{2H_n + D_n}{2} \quad (7-11)$$

$$H_{2n} = \frac{2H_n - D_n - E_n}{2} \quad (7-12)$$

$$H_{2n+1} = \frac{2H_n - D_n + E_n}{2} \quad (7-13)$$

from that to restore d values and finally to get back the original dataset elements or pixel values of an image. Wherever H is zero, there is no associated D value and reinstated d values of that branch should be zero.

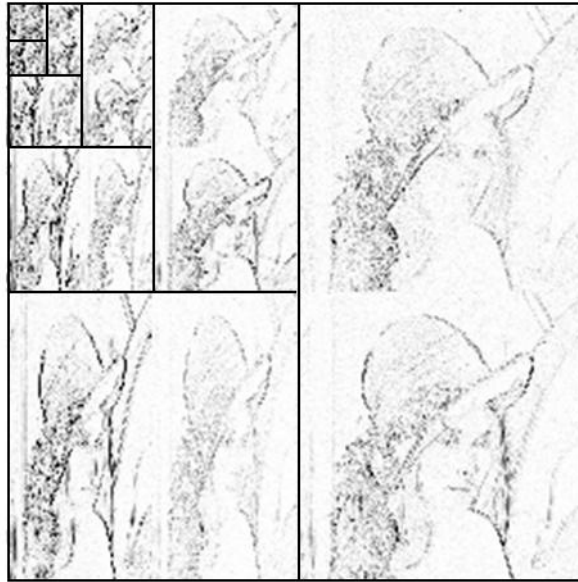


Figure 7.14 Scaled image of D and E coefficients
(In each segment E coefficients are located above or at the right side,
 D coefficients are down or at the left.)

There are some remarks for the practical implementation:

-Detail coefficients d should be unsigned integers (not negative whole number) to produce not negative average ($\frac{H_{2n} + H_{2n+1}}{2}$);

-In the case when d values are signed integers, calculating average value could be zero, but by shifting values on the number line, or by converting d values to unsigned integers, this problem could be solved.

7.5 Conclusion of the Chapter

The above described transformation or compression could be used as a totally lossless process, or could be altered to a lossy process. The best way to make it lossy

if we take advantage of the attribution of H entropy coefficients, that way to select the least significant branches (branches with the lowest H value) and omitting them.

There are different methods to compress discrete signals, datasets or images, each of them has some special characteristics. The procedure introduced here is entropy based adaptive compression method, its main feature is the hierarchical structure and entropy based compression. Wavelet transform of a dataset could be arranged to a binary tree like structure, which is the starting point of this compression. Degree of complexity of an image varies, different parts have different information content. Detailed regions require more thorough description than relaxed, predictable parts. To accomplish this task a new type of variable was introduced, resembling the entropy of the particular area of the image. The higher is the value of this variable indicating entropy, the higher is the importance of the area's detail coefficients.

Applicability of the method is not restricted to 1D or 2D signals only, it is equally applicable to images, motion pictures, 4D ultrasound recordings, any multidimensional signals, since each can be transformed into a binary tree structure.

8 CHAPTER

CELL DIVISION AND DIFFERENTIATION ON THE GENE LEVEL AS ALGORITHM FOR LOGICAL DATA PROCESSING

8.1 Transformation (Coding and Decoding)

In previous chapters it was shown that a biological process, like cell division and differentiation could be described as a kind of wavelet transformation. By mimicking this process, we were able to transform a 2-3 or more dimensional image or **discrete** dataset into a perfect binary tree structure. Here we show potentially possible approach how to compress similar way bitmaps or binary datasets containing **logical** variables. The main advantage of this lossless method is that the resolution of the reconstructed image is doubling by each step, i.e. hierarchical.

When we have a set of logical variables and the elements of this dataset are correlated to each others, then we are able to predict any element of it higher than 50% accuracy. To predict the value of an element usually we compare it to its direct neighbor or neighbors. The principle of wavelet transformation is the comparison between elements. The result of this transformation is usually a perfect binary tree. The higher is the predictability of the elements, the higher is the percentage of logical zeros than the logical ones in the transformed image. The aim of any image transformation to binary images is, to reduce the number of logical ones compared to zeros. This could increase the efficiency of any compression method. In previous chapters it was shown, that cell division and differentiation resembles to a kind of wavelet transformation [33]. In this study we show how binary dataset transformation and compression is used by nature, how this method could be utilized by us for image or dataset compression.

8.2 Biologically Inspired Algorithm for Bitmap Transformation

Any multicellular living organism develops from a single cell through consecutive cell divisions. After cell division one of the daughter cells retains the properties of the initial cell, the other might be the same or different. Any cell of an organism contains the same genetic information. The type of a cell is determined by its gene expression profile, some of the genes are enabled for transcription and the others are repressed [41]. In Figure 8.1 we represent the status of a particular gene. In the first generation we have only one cell, the status of this gene is G_1 (either 0 when this gene is repressed or 1 when the status is enabled). In the second generation one daughter cell retains this status,

$$G_{10}=G_1$$

the other sister cell's similar gene could have the same status or the opposite.

$$G_{II}=G_I \text{ or } G_{II}=\sim G_I$$

(\sim logical NOT)

How to describe this cell differentiation for this particular gene? Variable d represents whether the expression of the gene is the same or altered after cell division.

$d=0$ means same status, $d=1$ stands for altered status.

$$d_I = G_{II} \oplus G_I$$

(\oplus logical operation XOR)

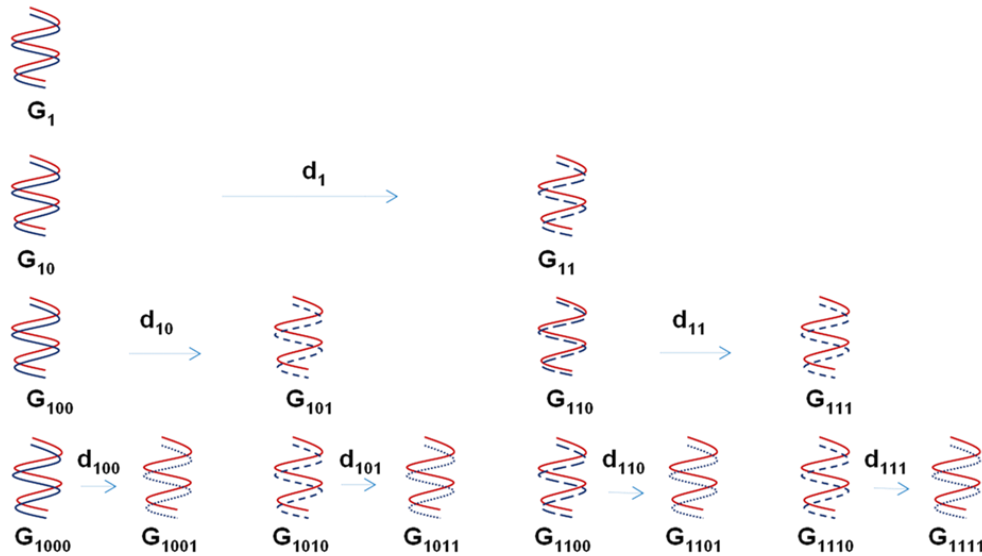


Figure 8.1 Expression of a particular gene after consecutive cell divisions and differentiations

Development of an organism is well predictable; these factors are stored somewhere in the genetic information. Since always one cell divides into two new cells this process looks like a binary tree. These d values could be also represented by a binary tree (shown in Figure 8.3). Indexing is in binary system. d is a logical variable, its value either 0 or 1 , true or false.

8.3 Analogy between Cell Differentiation and Image Transformation

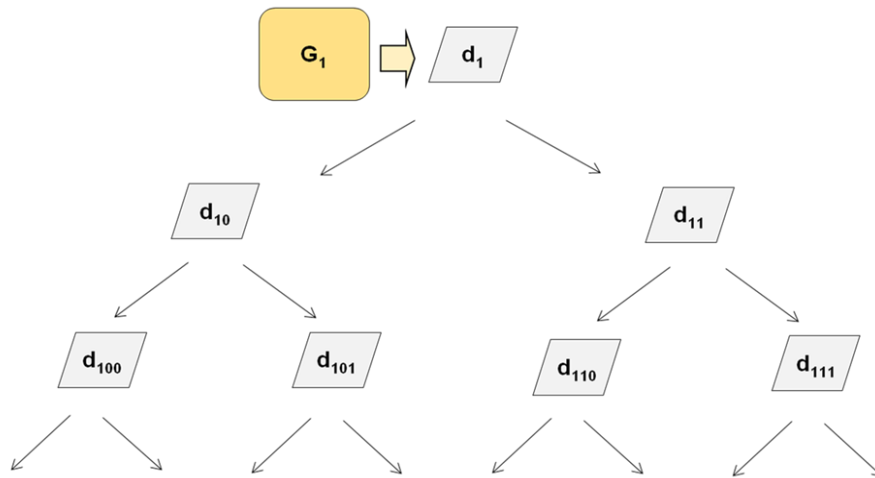


Figure 8.2 Binary tree shape representation of cell differentiation

In Figure 8.3 P_1 could be the status of a gene in a biological system (G_1 is the initial state of G gene) or it could be the value of a pixel. By consecutive divisions we reach to the fully developed organism. If it is about image processing, this is actually an inverse transformation [42].

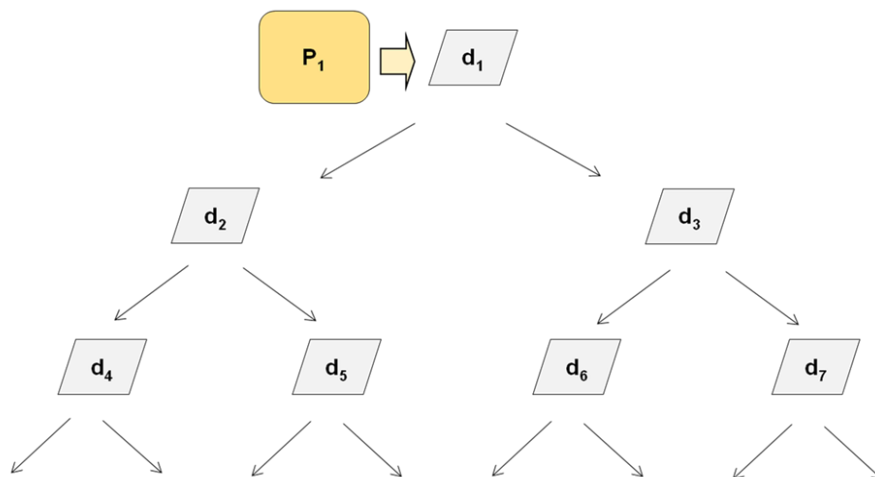


Figure 8.3 Detail coefficients of image transformation

The number of cells of a multicellular organism is order of magnitude greater than the number of base pairs of the DNA chains in the nucleus. (There are 4 different nucleotides; the sequence of these nucleotides in the DNA chain of the chromosomes is called genetic information.) Moreover the number of genes in the genome is in the order of tens of thousands [43]. That means, the genetic information concerning cell differentiation must be highly compressed.

8.7 Predicting the Value of a Logical Variable

We have a finite set of logical variables. When the elements of this dataset are correlated to each others, then we are able to predict any element of this set higher than 50% accuracy. The elements of any dataset are usually arranged into a structure by some parameter, for example time, physical position or any possible criteria. The most common way of the prediction is usually the comparison by its neighbor or neighbors.

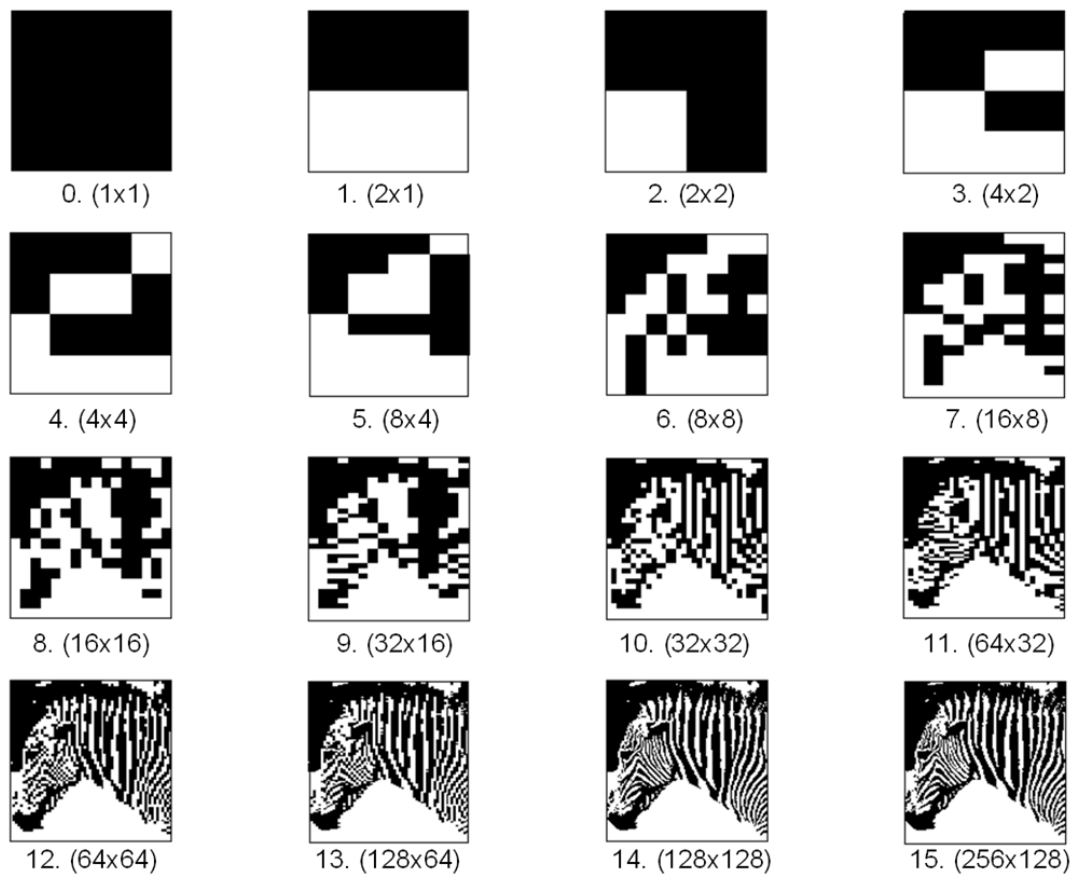


Figure 8.5 Inverse transformation, reproducing the image of the zebra

An example is shown in Figure 8.5, (Figure 8.4 is the original image) subplot 15 is the first downsampled level 256x128 pixels and backwards 14...0 are the approximation coefficients (simple downsampling). Figure 8.6 shows the corresponding detail coefficients, most of the detail coefficients have 0 values (black).

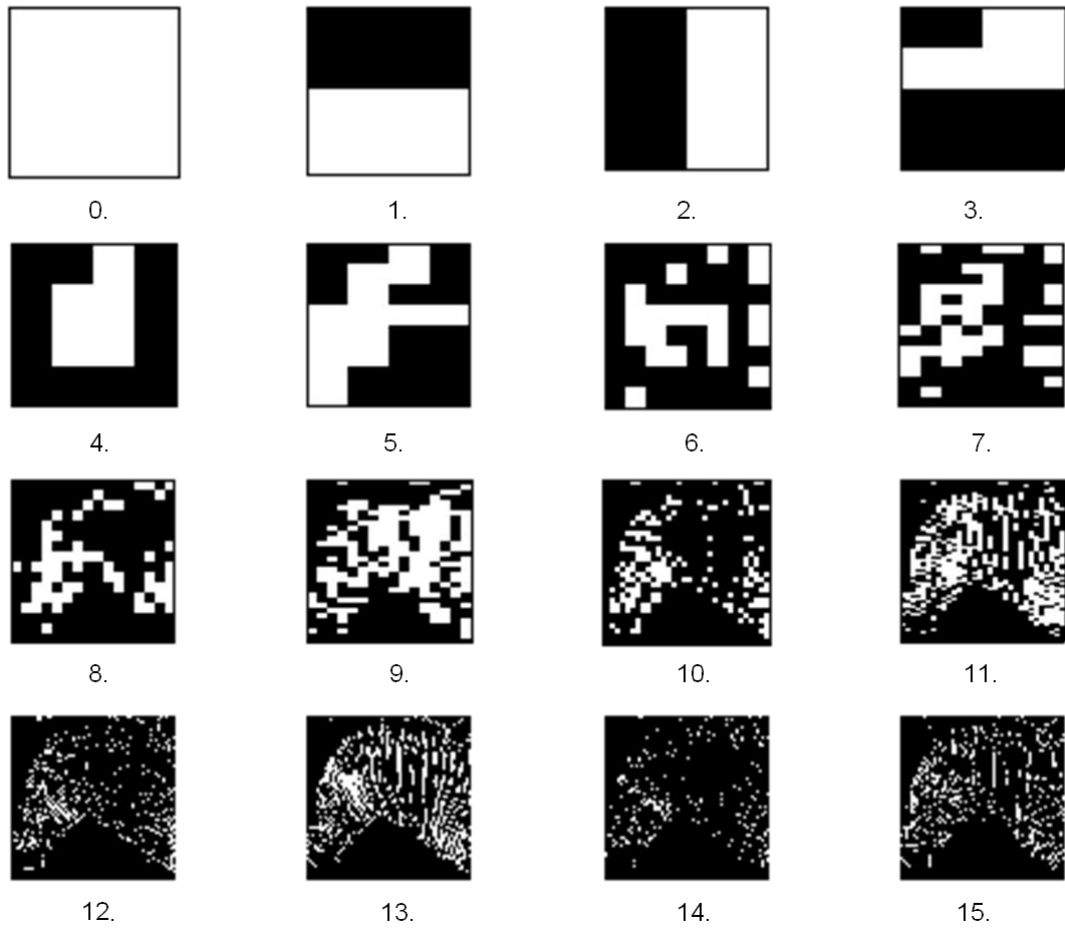


Figure 8.6 Detail coefficients of the transformed image

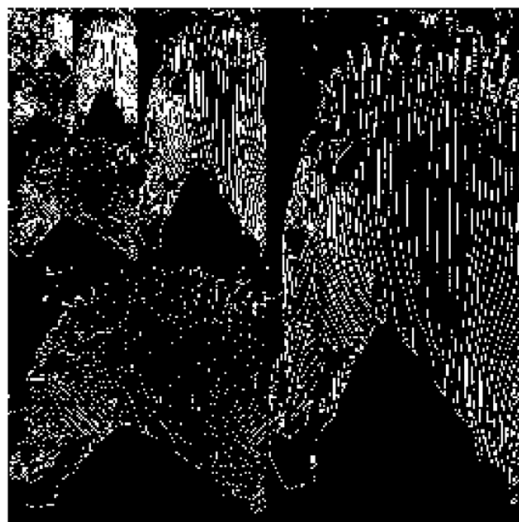


Figure 8.7 Combined transformed image

8.8 Conclusion of the Chapter

When we have a set of logical variables and the elements of this dataset are correlated to each other, then we are able to predict any element of it higher than 50% accuracy. To predict the value of an element usually we compare it to its direct neighbor or neighbors. The principal of wavelet transformation is making comparison between elements. The result of this kind of transformations is usually a perfect binary tree. The higher is the predictability of the elements, the higher is the percentage of zeros than the ones in the transformed image. The aim of any image transformation to binary images is, to reduce the number of logical ones compared to zeros. This could increase the efficiency of any compression method.

9 CHAPTER
HOW TO COMPRESS A BINARY TREE LIKE BITMAP
COMPRESSION METHOD FOR BINARY TREE LIKE BITMAPS

9.1 Introduction of an Internal Variable

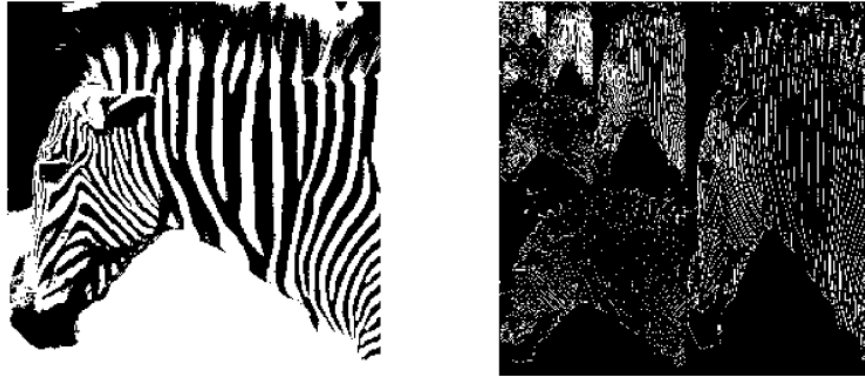


Figure 9.1 Example: image of a zebra and its transform

As it was seen in Figure 9.1, most of the detail coefficients i.e. most of the element of the binary tree are logical 0 if the cells/pixels are correlated to their neighbors. When some part of an image is undisturbed or the pattern is predictable, then the corresponding branch of the detail coefficient binary tree contains only 0 values.

If we are certain that a branch of this tree contains only 0 values, then we can omit that branch of the tree. Let us introduce an internal logical (Boolean) variable H , this shows whether a branch is disturbed or undisturbed.

$H=0$

when the entropy of that branch is 0 and

$H=1$

when the entropy is higher than 0 .

To reconstruct an image after this wavelet transformation we should know the highest level approximation coefficients and all of the detail coefficients (lossless transformation). In our case we should know the value of P_1 and all d_i values where

$$1 \leq i \leq 2^n - 1$$

n is the number of transformation levels (in our example $n=16$).

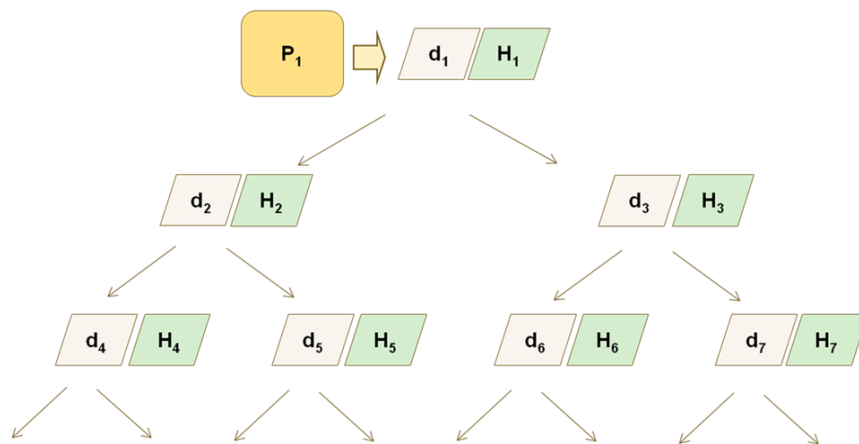


Figure 9.2 H variable suggest the entropy level in the given branch of the binary tree

Depending on the value of H , some of the branches could be truncated. $H=0$ indicates that all of the d detail coefficients of that branch downstream are equal to 0 .

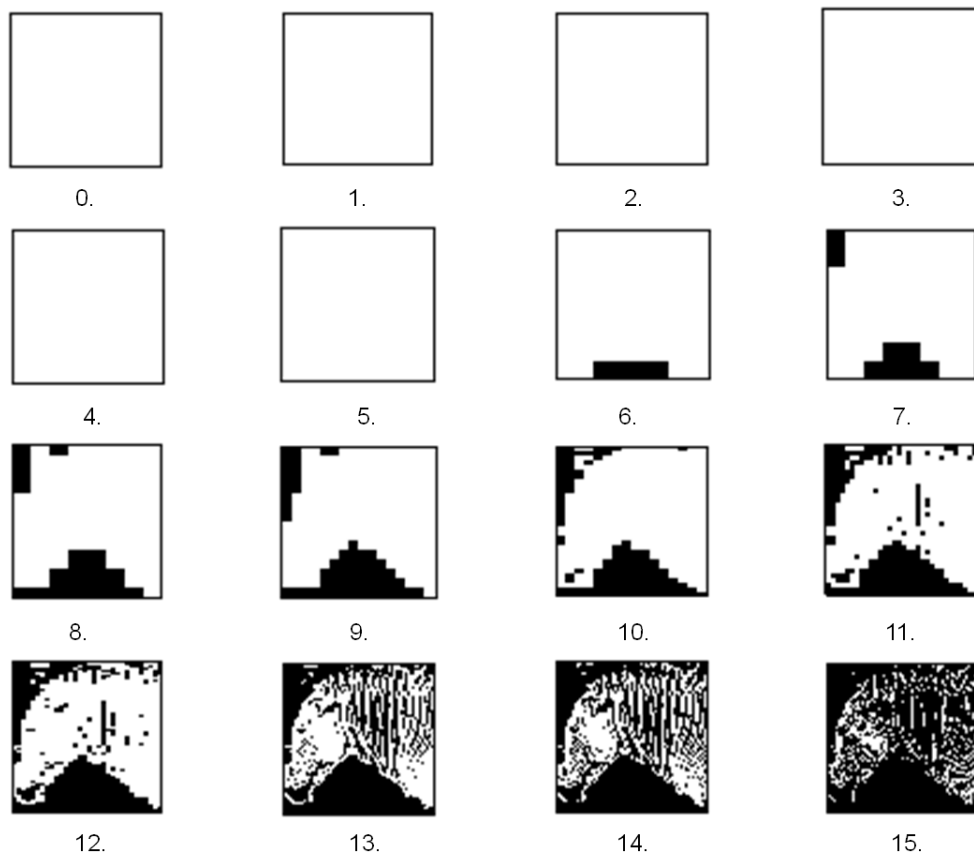


Figure 9.3 H entropy values of the image

Legend: white represent logical 1 , black pixel means logical 0 value

Looking at Figure 9.2 one might think that 2^n numbers of pixels are represented by 2×2^n bits, why is it called compression? In case of high correlation

between neighboring elements, many of the branches contain mostly logical 0 values. If we are able to omit more than half of the detail coefficients, then the total amount of remaining d and H values should be less than 2^n .

In the case of our example image, Figure 9.3 shows the entropy values in all 16 levels.

9.2 Practical Realization of the Method

The result of the bio-inspired wavelet transformation is a binary tree, the newly introduced internal variable entropy coefficients are also structured into a similar size tree shape (Figure 9.4).

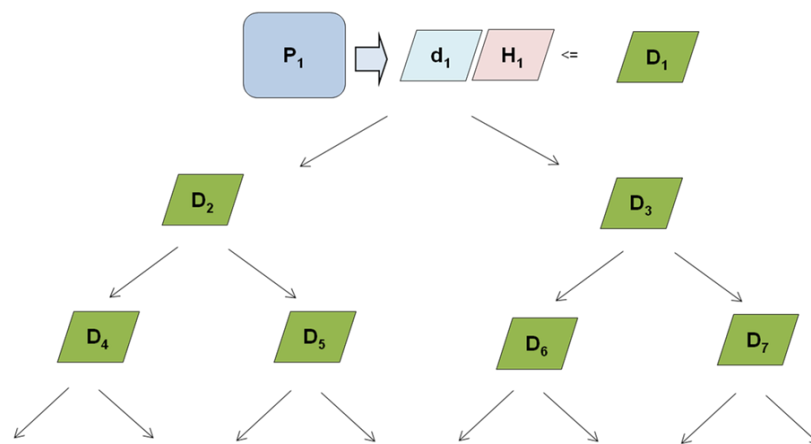


Figure 9.4 d detail coefficient and H entropy coefficient is replaced by D multi-valued logical variable

Instead of d detail coefficients we have to introduce a different coefficient, let's call it D . D values obviously should contain information about the relevant value of the given d coefficient and should update the value of the internal entropy variable H .

9.3 Multi-Valued Logical Variable

D variable has to have more than two values, not only “true” or “false”. Besides coding for the detail coefficient d , it should give additional information for updating H entropy internal variable too. D is some kind of fuzzy variable [45].

9.4 Coding, Decoding an Image

How to describe this D variable? We have numerous options. There are four different possible methods explained below.

9.4.1 Option 1

For example we can define H entropy variable of Figure 9.2 as:

$$H_1 = d_1 + H_2 + H_3;$$

$$H_2 = d_2 + H_4 + H_5; \quad H_3 = d_3 + H_4 + H_5;$$

etc.

(+ : logical operation OR)

Where D_1 is coding for d_1, H_2 and H_3

- If $H_1 = 0$, means that there is no any disturbance on its downstream branch, D_1 has no any value.
- If $H_1 = 1$, means that there is/are disturbance/s at any of d_1, H_2 or H_3 according the following truth table

This truth table define the possible values of the multi-valued D variable:

d_1	H_2	H_3	D_1
0	0	0	_____
		*	
0	0	1	a
0	1	0	b
0	1	1	c
1	0	0	d
1	0	1	e
1	1	0	f
1	1	1	g

*(H_1 should be equal 1)

In this option D should be a 7 valued variable. The lowest level H logical values are equal to the original d detail coefficients. The sum of a geometric progression gives the total number of D variables. When the size of the dataset to be transformed is 2^N , then the number of lowest level d detail coefficients is 2^{N-1} . The total number of D variables could be maximum $2^{N-1} - 1$.

9.4.2 Option 2

In the second option we defined H entropy variable as:

$$H_1 = d_2 + d_3 + H_2 + H_3;$$

$$H_2 = d_4 + d_5 + H_4 + H_5; \quad H_3 = d_6 + d_7 + H_6 + H_7;$$

etc.

To achieve this D could be:

D_2 is coding for d_2 and H_2

D_3 is coding for d_3 and H_3

If $H_1=1$ then:

d_2	H_2	D_2	d_3	H_3	D_3
0	0	a	0	0	a
0	1	b	0	1	b
1	0	c	1	0	c
1	1	d	1	1	d

D is a 4 valued variable.

9.4.3 Option 3

$H_2=d_1+H_3; H_3=H_4+H_6;$

$H_4=d_2+H_5; H_5=H_8+H_{10}; H_6=d_3+H_7; H_7=H_{12}+H_{14};$

etc.

D_2 is coding for d_1 and H_3

D_3 is coding for H_4 and H_6

If $H_2=1$ then:

If $H_3=1$ then:

d_2	H_3	D_2	H_4	H_6	D_3
0	0	-	0	0	-
		*			**
0	1	a	0	1	a
1	0	b	1	0	b
1	1	c	1	1	c

*(H_2 should be equal 1)

** (H_3 should be equal 1)

D is a 3 valued variable.

9.4.4 Option 4

$H_1=d_1+H_2+H_3;$

$H_2=d_2+H_4+H_5; H_3=d_3+H_6+H_7;$

etc.

D_2 is coding for d_2 and H_2

D_3 is coding for d_3 and H_3

If $H_1=1$ then:

d_2	H_2	D_2	d_3	H_3	D_3
0	0	a	0	0	a
0	1	b	0	1	b
1	0	-*	1	0	- **
1	1	c	1	1	c

* H_2 could not be 0 if $d_2=1$

** H_3 could not be 0 if $d_3=1$

D is a 3 valued variable. In options 2, 3 and 4 the maximum number of D variables is as same as the number of d detail coefficients, 2^N-1 . Of course when any part of the image is undisturbed, then H values became logical 0 and consequently the number of D coefficients will be less than the theoretical maximum.

9.5 Introducing a New Type of Logical Variable

Logical algebraic expressions have two values, true or false. The result of any logical function is also a logical value: true or false, but what we are talking about is some kind of multi-valued logic with 4 different logical values [46]. It would be our next task, to find out how nature is able to perform this kind of compression.

Option 3 and 4 are the most feasible versions. To give three values to “D” we could use Huffman coding [47] for example:

a=00; b=01; c=1

9.6 Conclusion of the Chapter

The most common and very simple compression method is the run-length encoding. This cell division inspired transformation coupled with this compression method grant a way by which we are getting closer and closer to the original image through a series of higher and higher resolution images. That is why we can call it hierarchical transformation.

Pros and cons of this method: this method provides similar compression ratios than other methods, but it takes us step by step closer to the reconstructed image, in each step we get a realistic image only in lower resolution. One useful utilization could be to code for object’s location in intelligent space for robotics or among in others for compression of monochromatic images [48].

10 CHAPTER

PRELIMINARY MODELING SPATIAL MOVEMENT OF aminoacyl-tRNA (aa-tRNA) MOLECULES IN THE CYTOPLASM

The aim is to develop a model of tRNA molecular movement in the bacterial cytoplasm and run simulations according to different tRNA concentrations and velocity conditions. The main criterion required in protein synthesis is the availability of the necessary amino acid in the vicinity of the ribosome. Therefore, we examine the spatial movement or placement of aminoacyl-tRNA (aa-tRNA) molecules in the cytoplasm—viewed from the perspective of that particular aa-tRNA (charged with a specific amino acid). The aim is to set up a kinetic model of the mRNA – ribosome – tRNA system and construct a simulation. The purpose of the simulation is to examine the conditions necessary for the tRNA to deliver the correct amino acid to the ribosome within a biological timeframe.

The significance of this research is that it attempts to present a basic dynamic model to understanding tRNA's "modus operandi" in assembling amino acids into a protein chain. The more we know about such a process, for example the more efficient antibiotics we are capable to synthesize.

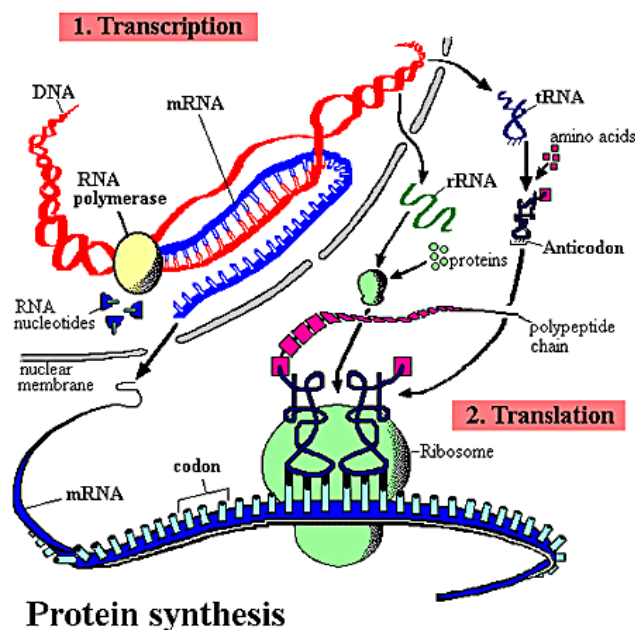


Figure 10.1 The route from DNA to protein [1]

10.1 The use of *Escherichia coli* (*E. coli*) as a Model Organism

Since protein biosynthesis (Figure 10.1) is similar in prokaryotes and eukaryotes, *Escherichia coli* (*E. coli*), a prokaryotic microorganism, has been chosen as a model organism in this study.

E. coli is frequently studied in micro- and molecular biology. It is a prokaryotic organism with a simple structure that renders it an excellent target for investigation and experimentation. *E. coli* can also be grown easily and its genetic components are comparatively simple and easy to manipulate, making it one of the most studied prokaryotic model organisms in biotechnology.

The DNA sequence of *E. coli* reveals 4441 open reading frames corresponding to 4322 proteins and 122 rRNAs and tRNAs. The entire sequence of the *E. coli* genome has already been identified. Data and statistics are readily available for the calculations conducted in this research [49], [50]. The model organism *E. coli* is 2-4 μm long with an average diameter of 500-800 nm. The cytoplasm of the whole cell thus contains about 40 million molecules in addition to water molecules or, if we omit ions and small organic molecules, about half a million macromolecules [49].

Little is known about the movement of macromolecules in bacterial cells. In early experiments, it has been shown that the motion of macromolecules was consistent with simple diffusion on a time scale of < 1 sec and bacterium length of ~ 1 μm [51]. Deich et al. [52] investigated individual fluorescently labeled proteins in the membrane of *C. crescentus* and characterized their two-dimensional motion as diffusive. Golding et al. have found that cytoplasmic motion is subdiffusive on a time scale of seconds to minutes [53].

10.2 Method

Many approaches exist for the simulation of biochemical cellular processes using deterministic and stochastic modeling approaches. Three types of cell models are generally discussed: macroscopic, mesoscopic and microscopic models [49]. Macroscopic models deal with molecular concentrations determined by stochastic differential equations. Forces between or inside the molecules are ignored. Usually an infinite reaction volume is assumed. Mesoscopic models deal with individual molecular dynamics of biochemical reactions. Generally physical forces between or inside the molecules are not considered. Microscopic models are the only models that deal with physical forces within or between molecules. This type of modeling is fine-grained and is not suitable for whole cell simulation because of computational restrictions, and the nature of the interactions considered. A microscopic model was used for describing molecular folding - e.g. secondary structure of proteins [54], [55].

The relatively small number of tRNAs compared to the number of ribosome per bacterial cells is another important fact. According to literature there are approximately ten times more tRNAs present in *E. coli* than the number of ribosome [50]. If the quantity of each amino acid specific tRNA is about 2%, then there is only one tRNA molecule for five ribosome. Therefore tRNAs are quite busy and well utilized in providing amino acid molecules for the ribosome.

Most cell simulations concentrate on biochemical interactions, but in this case we extend our research into the physical domain (physical interactions). The model sought does not fit into any of the above mentioned model categories. Instead of two dimensions, a 3-dimensional model can be used even if it is not a whole cell model. Particles are considered to follow the rules of Brownian motion. The speed of differently sized particles can be approximated using literature data. The statistical average time interval (i.e. the delivery time of a particular amino acid to the ribosome) when the same tRNA interacts with the same ribosome, is included [5], [56], [57].

Most of the data used in our model were taken from Project CyberCell *E.coli* Statistics [50]. The average rate of amino acid assembly is around 20 ms per base, so if our simulation results show a rate of assembly that fits into this time frame, it is possible to conclude that the selection process is purely statistical or random [51].

Table 10-1 *E.coli* cell dimensions

Cell length	2 μm or 2×10^{-6} m	Mean Velocity of small molecules (cytoplasm)	50 nm/ms = 5×10^{-5} m/s
Cell diameter	0.8 μm or 0.8×10^{-6} m	Volume occupied by water	70%
Cell total volume	1×10^{-15} L or 1×10^{-18} m ³ (other est. at 0.88×10^{-15} L)	Volume occupied by protein	17%
Average size of protein	360 residues	Volume occupied by all RNA	6%
Average diameter of ave. protein	5 nm	Volume occupied by rRNA	5%
Average MW of protein	40 kD	Volume occupied by tRNA	0.8%
Average size of mRNA	1100 bases	Volume occupied by mRNA	0.2%
Average length of mRNA	370 nm	Volume occupied by DNA	1%
Mean Velocity of 70 kD protein (cytoplasm)	3 nm/ms = 3×10^{-6} m/s	Volume occupied by ribosomes	8%
Mean Velocity of 40 kD protein (cytoplasm)	5 nm/ms = 5×10^{-6} m/s	Translation rate	40 aa/sec
Mean Velocity of 30 kD protein (cytoplasm)	7 nm/ms = 7×10^{-6} m/s	Number of mRNA/cell	4000
Mean Velocity of 14 kD protein (cytoplasm)	10 nm/ms = 10×10^{-6} m/s	Number of tRNA/cell	200,000
Number of ribosomes/cell	18,000	MW of ribosome	2700 kD
Diameter of ribosome	20 nm	Volume of ribosome	4.2×10^{-24} m ³

*Data was based on [50].

In our proposed model, molecular particles move in a 3 dimensional space and collide with each other. The most important part of the simulation program is to detect which particle collides with which object, when and where they collide, as well as be able to calculate their velocity and direction of movement after the collision.

10.3 Elastic Collision of Particles

In elastic collisions, kinetic energy and momentum are conserved, i.e. there is no energy loss in the form of heat, etc.

Kinetic energy (E_k) of a moving object is:

$$E_k = \frac{1}{2}mv^2 \quad (10-1)$$

And its momentum (P) is:

$$\vec{P} = m\vec{v} \quad (10-2)$$

Where m is the mass and v is the velocity. Velocity and momentum are vectorial quantities.

If we have two objects colliding centrally, then

$$\sum_i E_{k_i} = \sum_i E'_{k_i} \quad (10-3)$$

(apostrophe indicates the value after collision), and

$$\sum_i P_i = \sum_i P'_i \quad (10-4)$$

Conservation of kinetic energy and momentum give equations 5 and 6:

$$\frac{1}{2}m_1v_1^2 + \frac{1}{2}m_2v_2^2 = \frac{1}{2}m_1v_1'^2 + \frac{1}{2}m_2v_2'^2 \quad (10-5)$$

$$m_1\vec{v}_1 + m_2\vec{v}_2 = m_1\vec{v}_1' + m_2\vec{v}_2' \quad (10-6)$$

10.3.1 Elastic Collision in 1 Dimension (1D)

Object₁ and **Object₂** are moving at v_1 and v_2 velocities respectively, as shown in Figure 10.1.



Figure 10.2 shows two particles of masses, m_1 and m_2 , and velocities, v_1 and v_2 , before they collide (1-D collision)

From eq.10-5 and 6 we are able to calculate the objects velocities after the collision (shown in Figure 10.2):



Figure 10.3 shows two particles of masses, m_1 and m_2 , and velocities, v_1' and v_2' , after they have collided (1-D collision)

$$\mathbf{v}'_1 = \frac{(m_1 - m_2)v_1 + 2m_2v_2}{m_1 + m_2} \quad (10-7)$$

and

$$\mathbf{v}'_2 = \frac{(m_2 - m_1)v_2 + 2m_1v_1}{m_1 + m_2} \quad (10-8)$$

respectively.

10.3.2 Collision in 2 Dimensions (2D)

The unit normal vector of collision is:

$$\vec{\mathbf{u}}_n = \frac{\vec{\mathbf{n}}}{|\vec{\mathbf{n}}|} \quad (10-9)$$

where, $\vec{\mathbf{n}} = ((x_2 - x_1), (y_2 - y_1))$

and the unit tangential vector is:

$$\vec{\mathbf{u}}_t = (-u_{ny}, u_{nx}) \quad (10-10)$$

(x_1, y_1) and (x_2, y_2) are the coordinates of the objects centers at the moment of collision (\mathbf{u}_n and \mathbf{u}_t are shown in Figure 10.3, 4 and 5).

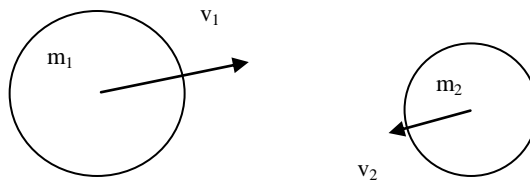


Figure 10.4 shows two particles of masses, m_1 and m_2 , and velocities, v_1 and v_2 , before they collide (2-D collision)

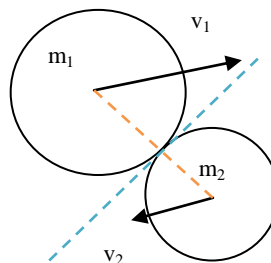


Figure 10.5 shows two particles of masses, m_1 and m_2 , and initial velocities, v_1 and v_2 , at the moment of collision (2-D collision)

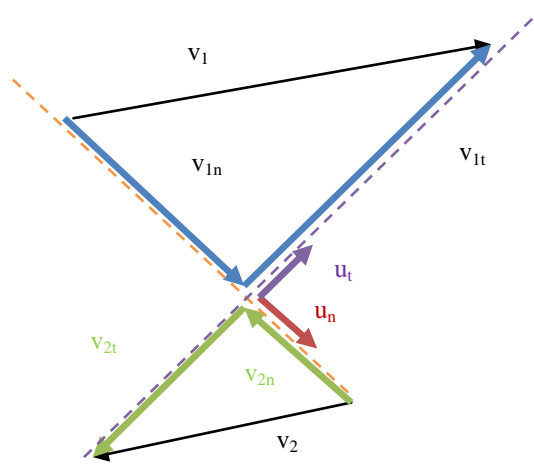


Figure 10.6 shows the vector components of the velocities before collision

After the collision, tangential velocity components are unchanged. v_{1n} is the scalar velocity in the normal direction, and v_{1t} is the tangential component.

Using dot products, we calculate the velocity components using:

$$v_{1n} = \vec{u}_n \cdot \vec{v}_1 \quad (10-11)$$

$$v_{1t} = \vec{u}_t \cdot \vec{v}_1 \quad (10-12)$$

and

$$v_{2n} = \vec{u}_n \cdot \vec{v}_2 \quad (10-13)$$

$$v_{2t} = \vec{u}_t \cdot \vec{v}_2 \quad (10-14)$$

Because tangential components do not change during the collision, then

$$v'_{1t} = v_{1t} \text{ and } v'_{2t} = v_{2t}.$$

To get the normal components of the velocity vector after the collision, a formula similar to the 1D case could be used:

$$v'_{1n} = \frac{(m_1 - m_2)v_{1n} + 2m_2v_{2n}}{m_1 + m_2} \quad (10-15)$$

and

$$v'_{2n} = \frac{(m_2 - m_1)v_{2n} + 2m_1v_{1n}}{m_1 + m_2} \quad (10-16)$$

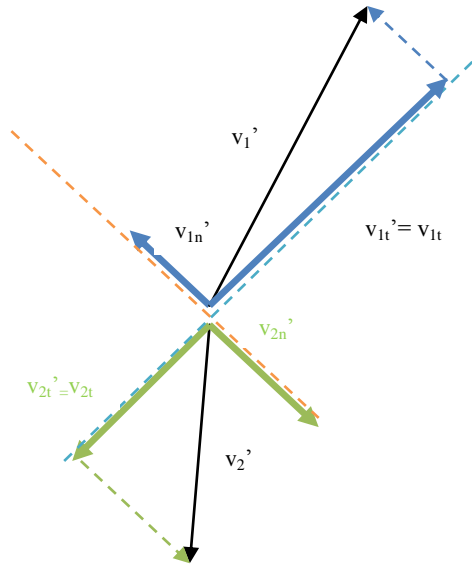


Figure 10.7 shows the vector components of the velocities after the collision

In Figure 10.6, to calculate the velocity vectors after the collision, normal vectors should be multiplied by the corresponding scalar velocity components:

$$\vec{v}'_{1n} = v'_{1n} \vec{u}_n \quad (10-17)$$

$$\vec{v}'_{1t} = v_{1t} \vec{u}_t \quad (10-18)$$

and

$$\vec{v}'_{2n} = v'_{2n} \vec{u}_n \quad (10-19)$$

$$\vec{v}'_{2t} = v_{2t} \vec{u}_t \quad (10-20)$$

Finally to get each object's velocity after the collision, normal and tangential vector components are added as shown in equations 10-21 and 22:

$$\vec{v}'_1 = \vec{v}'_{1n} + \vec{v}'_{1t} \quad (10-21)$$

$$\vec{v}'_2 = \vec{v}'_{2n} + \vec{v}'_{2t} \quad (10-22)$$

10.3.3 Collision in 3 Dimensions (3D)

Similarly to the 2D case, the normal unit vector of the collision was found, but instead of having one tangential component, there are 2 orthogonal components because now we have a plane rather than a line of collision.

To find the normal vector of the plane of collision:

$$\vec{u}_n = \frac{\vec{n}}{|\vec{n}|} \quad (10-23)$$

where,

$$\vec{n} = ((x_2 - x_1), (y_2 - y_1), (z_2 - z_1)) \quad (10-24)$$

An example of a pair of tangential unit vectors is given below:

$$\vec{t}_1 = ((u_{n_y} - u_{n_z}), (u_{n_z} - u_{n_x}), (u_{n_x} - u_{n_y})) \quad (10-25)$$

$$\vec{t}_2 = \begin{pmatrix} u_{n_y}(u_{n_x} - u_{n_y}) - u_{n_z}(u_{n_z} - u_{n_x}), \\ u_{n_z}(u_{n_y} - u_{n_z}) - u_{n_x}(u_{n_x} - u_{n_y}), \\ u_{n_x}(u_{n_z} - u_{n_x}) - u_{n_y}(u_{n_y} - u_{n_z}) \end{pmatrix} \quad (10-26)$$

$u_{n_x}, u_{n_y}, u_{n_z}$ are the x, y and z coordinates of the collision normal vector, t_1 and t_2 are the tangential vectors. Their unit vectors with x, y and z components are given below:

$$\vec{u}_{t_1} = \frac{\vec{t}_1}{|\vec{t}_1|} \quad (10-27)$$

$$\vec{u}_{t_2} = \frac{\vec{t}_2}{|\vec{t}_2|} \quad (10-28)$$

Again, similar to the 2D case, we can calculate velocity components using the dot product. The normal component of v_1 velocity is:

$$v_{1_n} = \vec{u}_n \cdot \vec{v}_1 \quad (10-29)$$

and $v_{1_{t_1}}$ and $v_{1_{t_2}}$ are its tangential components:

$$v_{1_{t_1}} = \vec{u}_{t_1} \cdot \vec{v}_1 \quad (10-30)$$

$$v_{1_{t_2}} = \vec{u}_{t_2} \cdot \vec{v}_1 \quad (10-31)$$

To obtain the second velocity components $v_{2_n}, v_{2_{t_1}}$ and $v_{2_{t_2}}$, we use:

$$v_{2_n} = \vec{u}_n \cdot \vec{v}_2 \quad (10-32)$$

$$v_{2_{t_1}} = \vec{u}_{t_1} \cdot \vec{v}_2 \quad (10-33)$$

$$v_{2_{t_2}} = \vec{u}_{t_2} \cdot \vec{v}_2 \quad (10-34)$$

Similar to the 2D case the tangential components are the same after the collision as they were before. The normal velocity components are obtained as in the 1D and 2D cases (equations 10-15 and 16).

$$v'_{1n} = \frac{(m_1 - m_2)v_{1n} + 2m_2v_{2n}}{m_1 + m_2} \quad (10-35)$$

$$v'_{2n} = \frac{(m_2 - m_1)v_{2n} + 2m_1v_{1n}}{m_1 + m_2} \quad (10-36)$$

So the velocity vectors after collision become:

$$\vec{v}'_1 = \vec{v}'_{1n} + \vec{v}'_{1t_1} + \vec{v}'_{1t_2} \quad (10-37)$$

$$\vec{v}'_2 = \vec{v}'_{2n} + \vec{v}'_{2t_1} + \vec{v}'_{2t_2} \quad (10-38)$$

More simplification can be made to equation 10-38. Note that $v'_{1t_1} = v_{1t_1}$, $v'_{1t_2} = v_{1t_2}$ and $v'_{2t_1} = v_{2t_1}$, $v'_{2t_2} = v_{2t_2}$, so

$$\vec{v}'_1 = \vec{v}'_{1n} + \vec{v}_{1t_1} + \vec{v}_{1t_2} \quad (10-39)$$

furthermore,

$$\vec{v}_{1t_1} + \vec{v}_{1t_2} = \vec{v}_1 - \vec{v}_{1n} \quad (10-40)$$

Equation 10-37 becomes:

$$\vec{v}'_1 = \vec{v}'_{1n} + \vec{v}_1 - \vec{v}_{1n} \quad (10-41)$$

Similarly, equation 10-38 becomes:

$$\vec{v}'_2 = \vec{v}'_{2n} + \vec{v}_2 - \vec{v}_{2n} \quad (10-42)$$

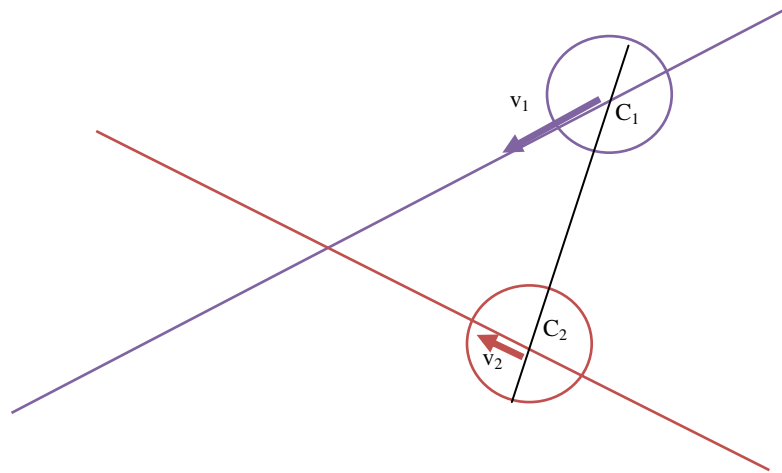


Figure 10.8 shows the schematic of two molecules before collision.

10.4 Collision Detection

If two objects are approaching each other as shown in Figure 10.7, **Object₁** is moving with a velocity of v_1 , **Object₂** with a velocity of v_2 . C_1 and C_2 are the objects' centers of mass. The line segment connecting C_1 and C_2 should be parallel to itself until the objects collide. Because the model is based on discrete time intervals and not continuous ones, the exact moment of the collision cannot be caught or determined (Figure 10.8).

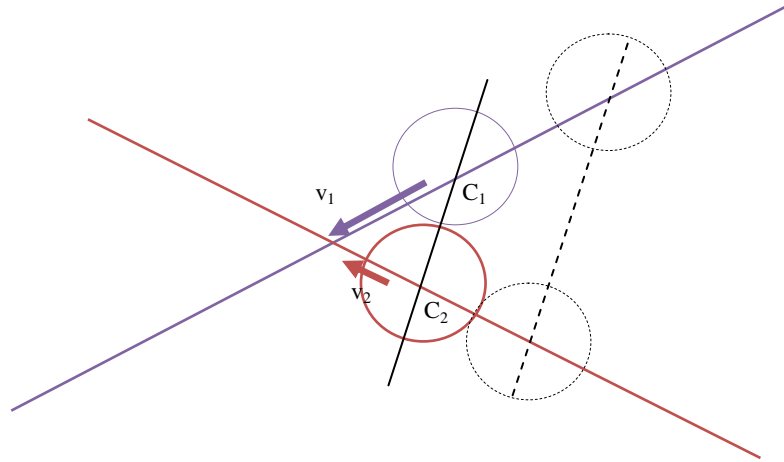


Figure 10.9 shows the schematic of two molecules at the moment of collision.

Consider that any two particles have collided when the distance between C_1 and C_2 is smaller than the sum of the two radii, Figure 10.10. The Δt sampling time interval should be small enough to be able to detect any collisions taking place. Since the line segment between C_1 and C_2 is always heading in the same direction, i.e. parallel to its previous state, it is not necessary to determine the exact moment of collision to get an approximation of the normal vector of the collision. The direction shown by the C_1 and C_2 line segment is:

$$\vec{n} = ((x_2 - x_1), (y_2 - y_1), (z_2 - z_1)) \quad (10-43)$$

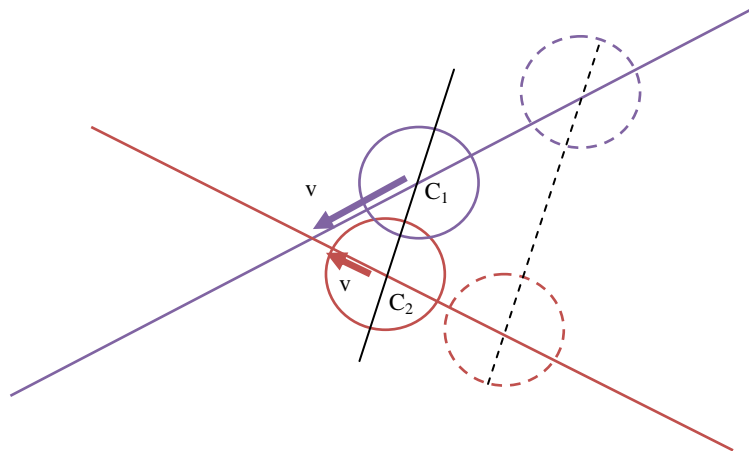


Figure 10.10 shows the schematic of two molecules virtually overlapping at the moment of collision

10.5 Flowchart

The MATLAB software was used for our computations. A flowchart was developed to organize the different subroutines (

Table 10-2). Computational input data, representing the initial conditions, are based on biological measurements and are summarized in Table 10-1. During initialization, we create a simulation space, place particles inside that space, randomly assign each a mass, radius, initial velocity and location. Placing objects one by one into the simulation space guarantees that no object overlaps with another. Ribosome's velocities are assumed to be zero and simulated as fixed objects in the cytoplasm.

To calculate the velocity of particles after collision, the following equations from the previous section were used in the MATLAB code:

Collision normal vector:

$$\vec{n} = ((x_2 - x_1), (y_2 - y_1), (z_2 - z_1)) \quad (10-44)$$

Normal unit vector:

$$\vec{u}_n = \frac{\vec{n}}{|\vec{n}|} \quad (10-45)$$

Object₁ velocity normal component before collision:

$$v_{1_n} = \vec{u}_n \cdot \vec{v}_1 \quad (10-46)$$

Object₂ velocity normal component before collision:

$$v_{2_n} = \vec{u}_n \cdot \vec{v}_2 \quad (10-47)$$

The velocity values after the collision are:

Object₁ normal component:

$$v'_{1_n} = \frac{(m_1 - m_2)v_{1_n} + 2m_2v_{2_n}}{m_1 + m_2} \quad (10-48)$$

$$\vec{v}'_{1_n} = v'_{1_n} \vec{u}_n \quad (10-49)$$

Object₂ normal component:

$$v'_{2_n} = \frac{(m_2 - m_1)v_{2_n} + 2m_1v_{1_n}}{m_1 + m_2} \quad (10-50)$$

$$\vec{v}'_{2_n} = v'_{2_n} \vec{u}_n \quad (10-51)$$

Object₁ velocity:

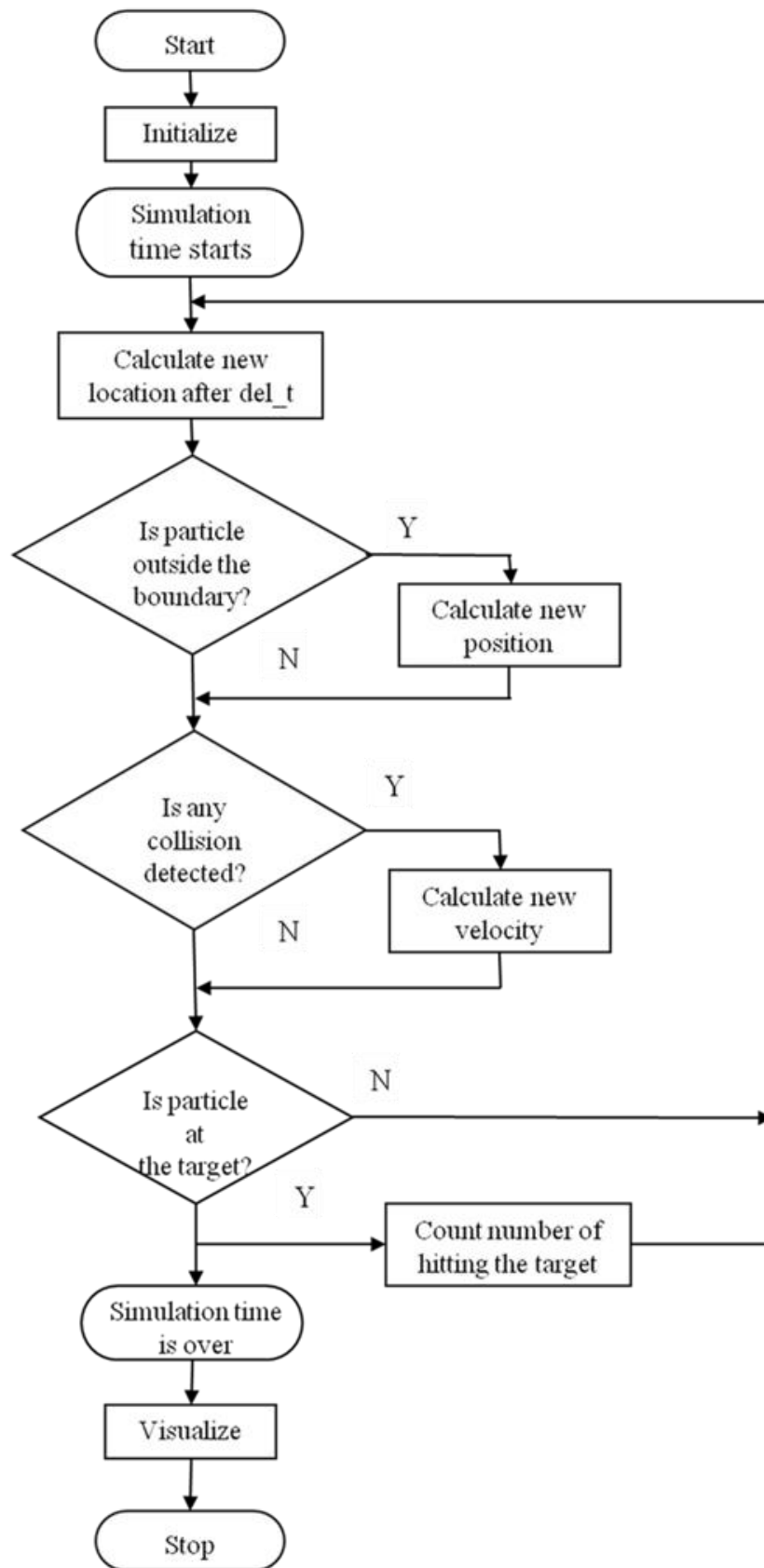
$$\vec{v}'_1 = \vec{v}'_{1_n} + \vec{v}_1 - v_{1_n} \vec{u}_n \quad (10-52)$$

Object₂ velocity:

$$\vec{v}'_2 = \vec{v}'_{2_n} + \vec{v}_2 - \vec{v}_{2_n} \quad (10-53)$$

The simulation calculates new locations for each particle every **delta_t** time interval and checks whether any collision occurs or any particle is located outside the simulation space. If there is a collision detected, then new velocities are calculated to both particles involved. When any particle crosses the boundary, it would change direction. After each step, the program checks whether any cognate tRNA reached the target, the ribosome, located in the centre of our simulation space.

Table 10-2 Flowchart



10.6 Simulation

After developing the computer program to simulate the tRNA movement, simulations were run using different settings.

For visualization purposes an 'avi' movie file was created using the particles' location matrix. Each frame comprises the 3D MATLAB graph taken every Δt time interval. In Figure 10.12, we can see the central ribosome with the mRNA and ctRNAs moving in simulation space. For simplicity, objects other than ctRNA are not shown; but are involved in the simulation, including other ribosome and other noncognate tRNAs, different proteins as well as other macromolecules capable of diverting the path of tRNAs in the cytoplasm.

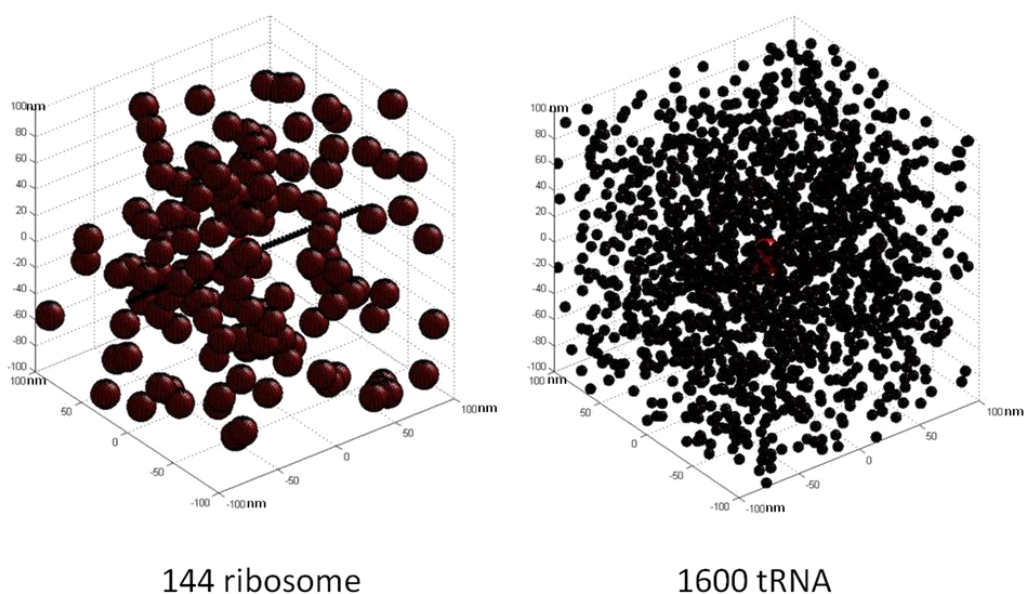


Figure 10.11 Simulation space contain a piece of the mRNA, 144 ribosome, 1600 tRNAs

The model was simulated in a $200 \text{ nm} \times 200 \text{ nm} \times 200 \text{ nm}$ cube. The particles are spaced randomly throughout the cube. The ribosome radius is set to 10 nm, tRNA radius to 3.5 nm, and the size of other macromolecules, capable of affecting the movement of tRNA under investigation, is randomly selected to be same number of tRNA, ctRNA, ribosome and other large macromolecules as reported in literature for *E. coli*). Multiple simulations were performed, by different density of the cytoplasm, by means of different numbers of other particles present in the simulation space. Example when 1,000 other objects are present, it fills about 36% of the simulation space whereas the presence of 3,000 other particles in addition to 144 ribosome, and more than 1600 tRNAs, these molecules occupy 72% of the simulation space.

Length of the simulation time is set to 20 ms which is the normal amino acid assembling rate. The simulation time step is selected to be 0.1 ms in order to detect all possible collisions among these particles. Each set of conditions was simulated 20

times except the normal case, what was simulated 100 times. Table 10-3 summarizes the results of the average number of hits on the ribosome by the ctRNA.

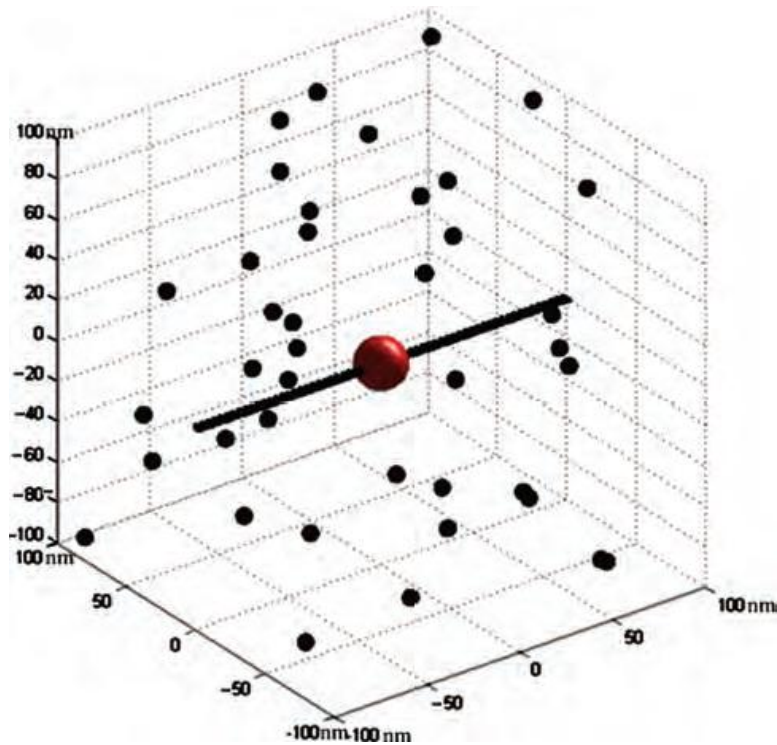


Figure 10.12 Result of Matlab simulation

First, we simulated the average hits in a normal cell (i.e. a cell that has the same number of tRNA, ctRNA, ribosome and other large macromolecules as reported in literature for *E Coli*). Our results show that, on average, 0.28 ctRNAs reach the ribosome's surface. Thus, the probability of the ctRNA biologically binding to the actual Aminoacyl-tRNA binding site is much smaller. These findings lead us to support the hypothesis that the ctRNA binding to the "A site" is not a random process in *E coli*. Next, we used different cellular conditions to simulate our dynamic system. We tripled the number of other particles in our simulation space. The results of 20 simulations show that the average ctRNAs reaching the surface decreased to 0.25 because these "other macromolecules" act as obstacles in the path of the ctRNA. We also reduced the number of other particles by 80% and observed that on average 0.3 ctRNAs reach the ribosome site. The average here is similar to that observed in a normal cell, a slight difference could be observed between the two cases. Next, we increased the length of simulation to 100 ms (instead of 20 ms) to simulate slower metabolism. The results show 0.85 ctRNA hits the surface of the ribosome. Therefore, even by reducing the rate of amino acid assembly from the normal 50 aa/second to 10 aa/ second, which would give the ctRNA a better chance to reach the ribosome, less than 1 ctRNA is able to reach any location on the ribosomal surface.

Table 10-3 Results of the simulations

Simulation sets		Results (Average of 20 runs)
Serial number	Special conditions	
1	Normal cell ¹	0.28
2	Cell is dry ²	0.25
3	Diluted cytoplasm ³	0.3
4	5 times longer time frame ⁴	0.85
5	ctRNA density doubled ⁵	0.6
6	ctRNA density tripled ⁶	0.7
7	Velocity doubled ⁷	0.4
8	ctRNAs velocity agitated ⁸	2.0
9	Sampling rate increased (x2) ⁹	0.25

¹Average of 100 simulations.

²Simulated by tripling the number of other particles in the cytoplasm.

³ Number of other particles in the cytoplasm is reduced by 80%.

⁴Simulation time is increased by 400%, (simulating slower metabolism).

⁵ Number of ctRNAs are increased, simulating tRNAs with concentration higher in the genome.

⁶ Similar to set#5.

⁷ The initial velocity of all moving objects in the simulation space is doubled.

⁸ There was a force applied to ctRNAs, comparable to Coulomb force exerted to electrically charged objects, to simulate a case when ctRNAs are able to move faster than other noncognate tRNAs.

⁹ These simulations are meant to check whether our estimation for sampling rate is correct and the number of overlooked collisions is insignificant.

We also used our code to study the effect of doubling and tripling the number of ctRNAs in our simulation space. Results show that even by significantly increasing the number of ctRNAs in the vicinity of the ribosome, the average hits recorded did not reach 1. Table 10-3 shows that the only simulation that succeeded in increasing the number of hits on the ribosome was achieved when a specific attractive force was applied to ctRNA only from the ribosome (the speed of all other molecules remained the same as used in our previous simulations). Our simulations averaged 2 ctRNA hits per run. It is important to reiterate that results shown in Table 10-3 report the average number of ctRNA reaching the surface of the ribosome in a given timeframe, it does not mean that the cognate transfer RNA is inserted in the “A site” of the ribosome, keeping in mind that the surface area of the ribosome is at least an order of magnitude larger than the “A site” itself. If we approximate the surface area of the ribosome as that of a sphere with a diameter of 20 nm, we get the total ribosomal

surface area of 1256 nm². On the other hand, the area of the “A site” could be approximated as that of a circle similar in size to the tRNA cross-section. In this case, we get an “A site” area of 38.5nm². Thus, the ratio of the two areas is approximately 1:33, which means that any ctRNA that reaches the ribosome has an even smaller chance of binding to the ribosomal “A site”.

Therefore it is obvious, that a prospective consecutive ctRNA molecule has no realistic chance of reaching a ribosome at a realistic rate in normal circumstances. The simulation shows that more than one ctRNA can hit the target—the ribosomal surface—only when the number of ctRNA-s or the time interval is exaggerated. Since the results of the simulations presented in this chapter proved that it is virtually impossible for the tRNA to reach the “A site” of the ribosome by random motion, other possible explanations for this biological phenomenon are given below in Table 10-4.

The ribosome could, although unlikely, store different types of tRNAs and preselect the cognate before entering the “A site”. Another hypothesis is the existence of a signaling mechanism between the ribosome and the tRNA which allows for the recognition of the cognate and accelerates its movement towards the “A site”. A third hypothesis is that tRNAs reach the ribosome in an orderly preselected manner.

Table 10-4 Comparison chart of possible explanations

Hypothesis	Pros	Cons	Opinion
1. The ribosome stores tRNAs and preselects them.	-	Timeframe is not sufficient.	Very unlikely.
2. Signaling exists between the ribosome and the cognate tRNA.	Logical explanation. No contradiction against it.	There is no observation on the existence of signaling or specific force between the tRNA and the ribosome. It cannot explain the wobbling effect.	Possible. Should be proved or disproved by conducting biological experiments or observations.
3. The tRNAs reach the ribosome in a preselected manner.	Logical explanation. RNA-RNA interactions are known phenomena. It can explain the wobbling effect.	There is no any observation.	Possible. Should be proved or disproved by conducting biological experiments or observations. New modeling and simulation is needed.

10.7 Possible Physical Mechanisms of tRNA Preselection in the Cytoplasm of *Escherichia coli* Bacteria

The functions of transfer RNA (tRNA) molecules are well known. Also well known and documented is how tRNAs are assembled and how they get charged by amino acids. Studies are scarce, however, on the existence of any signaling between tRNAs and ribosome. Possible physical forms of communication between these two entities could be mechanical vibrations through the liquid cytoplasm, and/or electromagnetic waves. If tRNAs are able to receive any signal emitted by the ribosome, they should have special receptors for this purpose. tRNAs tertiary structures contain four loops namely, D, TΨC, anticodon and “variable” loops. Their shape and diameter are specific to each type of tRNA. During activation, the tRNA type is recognized by the aminoacyl-tRNA synthetase enzyme according to the geometry of these loops[4]. In as far as the tRNA preselection in *Escherichia coli* (*E. coli*) bacteria is concerned, the question remains as to where exactly are tRNAs getting charged in the bacterial cytoplasm. Are they being charged in a specific place in the bacteria or does the charging happen throughout the cytoplasm?

The relatively small quantity of tRNAs compared to the number of ribosome per bacterial cells is another important fact. According to literature, there are approximately ten times more tRNAs present in *E. coli* than ribosome [50]. If the quantity of each amino acid specific tRNA is about 2%, then there is only one tRNA molecule for every five ribosome. Thus, tRNAs are quite “busy” and well utilized in providing amino acid molecules to ribosome.

In the previous section, we concluded that tRNAs are not able to approach the ribosome, by random Brownian motion, at a sufficient rate for protein synthesis [58]. To come to this conclusion, we used kinetic models to simulate the physical movement of the tRNA in the cytoplasm. In this section, I propose three mechanisms to explain tRNA preselection in prokaryotes [59]. First, we summarize the findings of available literature studies that deal with tRNA movements in the vicinity of the ribosome.

Studies on tRNA selection for protein synthesis emphasize the accuracy and speed with which the ribosome is able to carry out this function. These studies agree on the existence of two stages in this process. Thus, codon-anticodon recognition has two main stages, namely, initial selection and proofreading [51], [58], [60], [61]. These articles give a detailed description of the “double-trigger” mechanism (in tRNA approaching the ribosome) and the geometric constraints (of the tRNA movement) on a molecular and atomic level. Based on our kinematic simulations we believe that in addition to the two stages mentioned above, it is possible that the tRNA selection process could have a preliminary stage that we refer to in paper as “preselection.”

```

gagttttggacaatcctgaattaacaacggagatatttcc
1 - atg cca cgt cgt cgc gtc att ggt cag cgt
31 - aaa att ctg ccg gat ccg aag ttc gga tca
61 - gaa ctg ctg gct aaa ttt gta aat atc ctg
“
451 - ttc gca cac tac cgt tgg tta tcc ctt cgg
481 - agt ttt agt cac cag gcg ggc gct tcc agt
511 - aag cag ccc gct ttg ggc tac tta aat tga

```

Figure 10.13 Ribosomal 30S subunit protein S7
In our example we have used the highlighted portion of the gene.

10.7.1 Proposed Hypotheses on tRNA Preselection.

As we mentioned in the introduction, our kinetic model proved that it is virtually impossible for the tRNA to reach the “A site” (aminoacyl-tRNA binding site) of the ribosome by random motion. The purpose of this study is to provide possible explanations/mechanisms for tRNAs approaching the ribosome. The following three scenarios are envisioned:

1. The ribosome stores tRNA molecules and preselects them,
2. Signaling between the ribosome and the cognate tRNA exists,
3. tRNA molecules reach the ribosome in a preselected manner.

Before we discuss these three proposed mechanisms, we refer to the importance and the definition of the “reading frame” concept. To illustrate this concept we use [EG10906] rpsG: 540 bp - 30S ribosomal subunit’s S7 protein (a portion of the *E. coli* mRNA gene)[62], shown in Figure 10.13 and Figure 10.14. Messenger RNA (mRNA) contains the information (the amino-acid sequence) in the form of triplets of nucleotide bases. The difficulty in decoding the mRNA message arises because the beginning of the code is not obvious. Thus, depending on where deciphering starts, three possible “senses” could be interpreted. This is referred to as a “reading frame” [4]. Once the ribosome is attached to the mRNA, the position of the reading frame is fixed. During translation the ribosome moves along triplet by triplet until it reaches the “stop” codon. Next we discuss our three hypotheses.

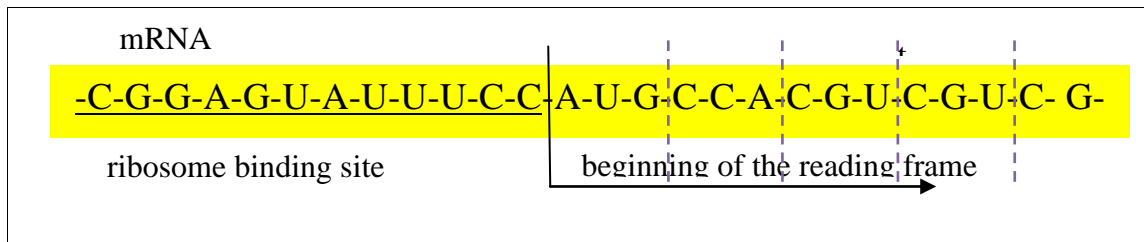


Figure 10.14 Reading frame
 (This is a portion of an mRNA:
 a gene of *E. coli* [EG10906]rpsG: 540 bp - 30S ribosomal subunit's S7 protein 8.)

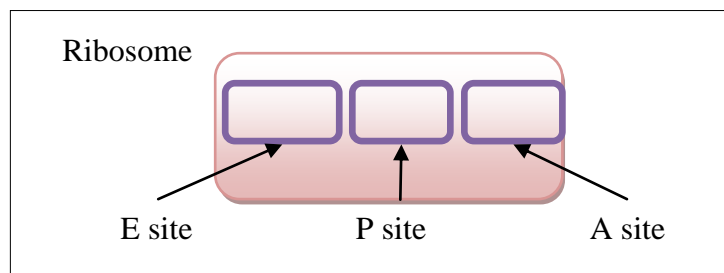


Figure 10.15 Three sites of the ribosome
 A site: aminoacyl-tRNA binding site
 P site: peptidyl-tRNA binding site
 E site: Exit site

10.7.2 Transfer RNAs are Stored in or at the Ribosome

The results of our previous simulations show that statistically, the number of consecutive amino acids delivered to the ribosome by random motion in a given time frame is slower than the rate at which proteins are synthesized. Suppose the ribosome collects available tRNA molecules in its neighborhood and stores them for future amino acid assembly. This could happen only, when all type of tRNAs are readily available to provide the proper amino-acid molecule for protein synthesis. The problem is that, there are no studies in literature that report the ability of ribosome to store tRNA molecules. Another fact that renders this hypothesis unpractical is the lack of time for a trial and error process for all types of tRNAs. The duration of one trial is about 3 ms, so the ribosome is capable of attempting to fit an average of 6 or 7 “probes” within the allowable 20 ms time interval. The number of possible tRNA molecules is 47 (the actual number of different tRNAs in *E. coli* bacteria is 86, but some are redundant) [63]. In order for this hypothesis to be viable, tRNAs should have, on average, enough time for 21 trials, or 63 ms, but nature allows 20 ms only. Therefore, this hypothesis seems to be less likely as illustrated in Figure 10.16.

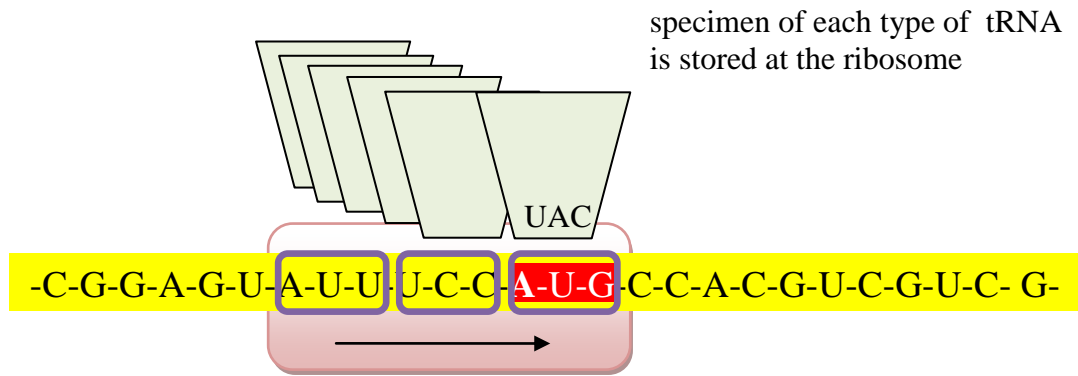


Figure 10.16 First hypothesis

In the first hypothesis we assume, that ribosome is capable to capture any tRNA from its proximity and select the cognate one from this stockpile time and time again.

10.7.3 Signaling between Ribosome and tRNA

In this hypothesis, we assume that cognate tRNAs (ctRNA) are capable of following a beacon signal emitted by the ribosome which propels them towards its “A site” (Figure 10.17). Simulation results reported earlier showed that when the velocities of ctRNA molecules are increased, more cognate transfer RNA molecules can reach the ribosome. In order to justify this hypothesis, we need to prove that ctRNAs are moving faster than other non-cognate tRNAs in the cytoplasm. There are no reports in literature that show evidence of differences in speed between ctRNA and other tRNA molecules. Additionally, no data is available on the existence of specific forces or signals acting between the ribosome and tRNAs when both are relatively far from each other. Cells are known to emit electromagnetic radiation, but there is no data that support the existence of such signaling between these two entities. Thus, based on the lack of evidence of any signaling between tRNA molecules and ribosome, we conclude that this hypothesis is also unlikely.

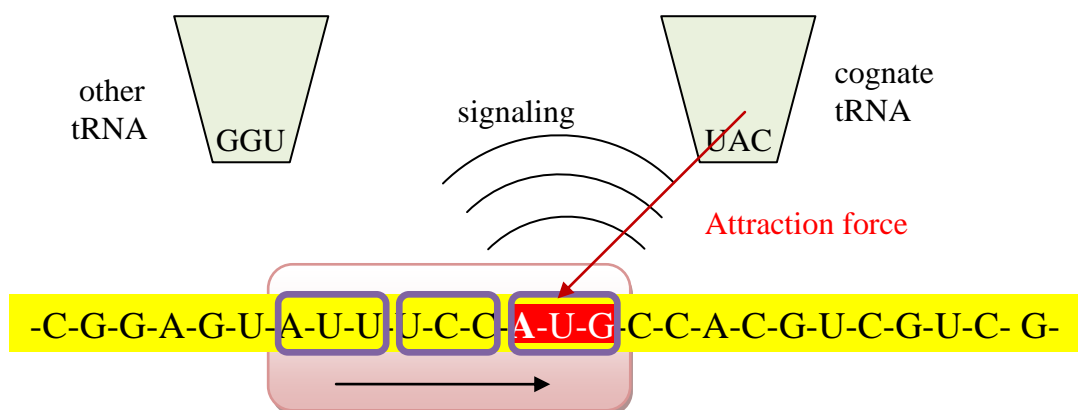


Figure 10.17 Second hypothesis

in the second hypothesis we assume, that the ribosome emits some kind of signal throughout the cytoplasm, to address any cognate tRNA nearby to agitate its movement towards this ribosome

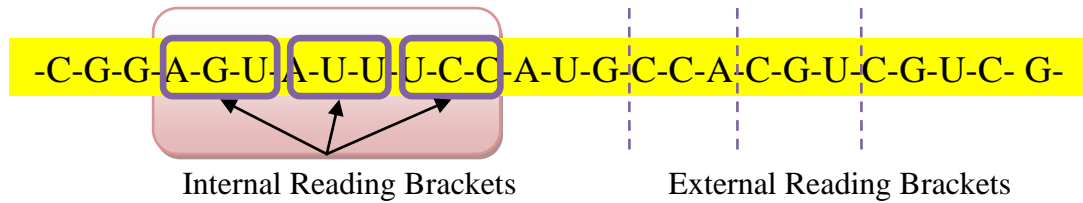


Figure 10.18 Reading brackets.

Ribosome's sites, A, P and E sites, are constraining tRNAs mechanically to follow the reading frame strictly. So these sites act as brackets, that's why we called them "reading brackets". Ribosome is stepping forward along the mRNA triplet by triplet, maintaining translation to follow the reading frame correctly, doing it so, the ribosome itself covers new portions of the mRNA triplet by triple. Thus the exterior of the ribosome also could act as a bracket.

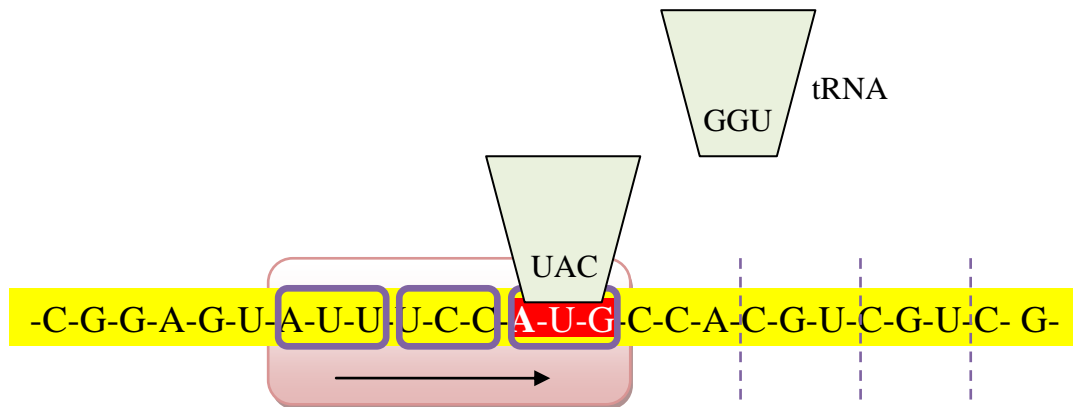


Figure 10.19 Third hypothesis; initiation.

Ribosome is assembled, the first tRNA to start protein synthesis reached by diffusion to ribosome's A site

10.7.4 Transfer RNAs are Preselected Using a Different Mechanism

RNA molecules and nucleotides tend to form pairs with their antisense counterpart. For example, the 16S ribosomal RNA (rRNA) component is able to bind to the Shine-Dalgarno sequence of the ribosomal binding site, which is its antisense. Once the ribosome subunits are fixed on the mRNA, the beginning of the actual reading frame is determined by its position. The ribosome has three sites to accommodate a tRNA molecule, namely the "A site", "P site" (peptidyl-tRNA binding site) and "E site" (Exit site). These sites act as mechanical constraints. In this paper we will refer to them as 'Reading Brackets' (see Figure 10.18). The ribosome translocates along the mRNA, triplet by triplet, maintaining the correct translation of the entire reading frame from the "start" codon (AUG) to the terminating "stop" codon (UAA, UAG or UGA). We posit that the ribosome is capable of determining "an external reading bracket" in addition to its normal "internal reading brackets" (Figure 10.19). The ribosome's body may be capable of mechanically constraining tRNA molecules by its internal sites and external shape (Figure 10.20). The free floating tRNAs have an equal chance of binding to any part of the mRNA where the

three consecutive bases are complementing its anticodon, regardless of the position of the reading frame. The ribosome is able to determine, or increase the possibility, that the three bases next to its location will be occupied by a suitable charged tRNA molecule. Our above hypothesis (illustrated in Figure 10.19, Figure 10.20 and Figure 10.21) is proposed to account for the results of the simulation reported earlier [6].

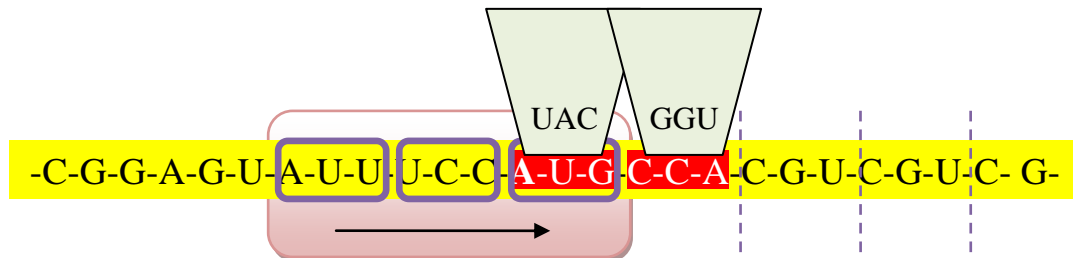


Figure 10.20 Third hypothesis; beginning of elongation

Other tRNAs are attached randomly to the naked portion of the mRNA, highly increasing the probability, that the suitable tRNA for the next step is the nearest tRNA to the ribosome, when it steps forward

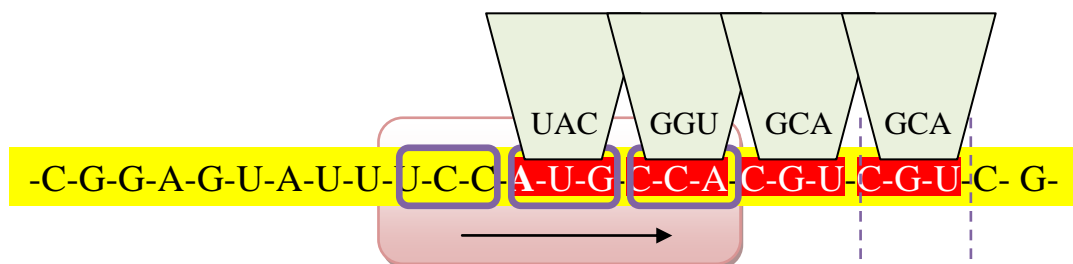


Figure 10.21 Third hypothesis; elongation in progress

More and more tRNAs are deposited on to the mRNA, similarly to sedimentation

Unlike eukaryotes, *E. coli* cells have no compartments. Therefore, the only mean of material transport viable is through diffusion [64], [65]. Despite this fact, in all the articles explaining the translation process, tRNA molecules are assumed to be floating directly to the “A site” of the ribosome [2]. Moreover, the tRNA cognate always reaches the required site. In reality, at any given moment, only one type of tRNA’s anticodon (out of more than 40 possible tRNA molecules) matches the actual codon. The ration of the number of mismatches to one tRNA successful proofreading [66] is estimated of 1 in 10^3 - 10^4 .

The essence of our theoretical proposal is that the mRNA itself is capable of attracting tRNA molecules and storing them. These two processes, attracting and attaching the tRNA molecules to the mRNA template itself, are proceeding in a parallel manner contrarily to what was hypothesized earlier (in the first and second hypotheses the imagined process is taken place in a serial manner at the ribosome). Thus, the timing is not critical in this case. Moreover, tRNAs are preselected

automatically if they are able to align themselves with each other without gaps before reaching the ribosome. In other words, the chance that three types of tRNAs present next to the ribosome is increased dramatically before the next tRNA molecule is preselected. tRNA molecules are moving randomly (by diffusion) in the cytoplasm and are interacting with the mRNA. Once the ribosome is assembled and attaches itself to the mRNA, the “reading brackets” are established. As the ribosome moves along, these external reading brackets move forward as well.

10.8 Conclusion of the Chapter

In this chapter we have concluded that tRNAs are not able to approach the ribosome, by random Brownian motion, at a sufficient rate for protein synthesis. To come to this conclusion, we used a kinetic model to simulate the physical movement of the tRNA in the cytoplasm of *Escherichia coli* bacteria.

Three hypotheses that attempt to explain tRNAs preselection leading to their binding at the ribosomal “A site” are discussed in this chapter. The first hypothesis is that the ribosome stores tRNA molecules inside its structure and aids in the pre-selection process. Because no previous articles report on such a pre-selection process, this hypothesis is most likely invalid. The second hypothesis suggests the existence of signaling between the ctRNA and the ribosome that permits the correct ctRNA to approach and bind to the ribosomal “A site”. As with the first hypothesis, there are no experimental proofs of a signaling mechanism between the ribosome and ctRNA which suggests that the second hypothesis is also unlikely. Our third hypothesis proposes that tRNA molecules reach the ribosome by diffusion. Although the functions of tRNAs, how they are assembled and how they get charged by amino acids are widely studied and well known, few if any studies report on the mechanism by which tRNAs are pre-selected to reach the ribosome. This chapter suggests several mechanisms to explain this preselection process.

Summary of the PhD Thesis

This PhD thesis examines current problems of biology in a novel approach, which explains the title "Engineering application of biological models". The details of the results presented in this thesis thematically can be divided into five parts.

The first is about a cell numbering system. The material of inheritance and information storage is the DNA in the cells, DNA molecules are replicated semiconservatively during cell division. Any multicellular living organism develops from a single cell through subsequent cell divisions. If we follow the fate of the DNA copies, we can formulate a binary tree based on that, which is the basis of this numbering. Certain biological experiments indicate that cell division (asexual) is well choreographed, it is then assumed that exactly the same half of the DNA double chain remains always in the old cell and the other will go always to the new cell. Linking cell lineages and this numbering system can help us to understand how body plan information might be stored in the genome. (Chapter 3, Thesis 1.a)

Our body consists of more than 10^{14} cells, a maximum of 100-200 different cells types are classified. Each cell is allocated into one of this few hundred cell types. The emerging new cells occupy a new place, position in the three-dimensional space, thus it could be considered as a 3D volumetric image. Following a natural algorithm, a multidimensional database can be converted into a binary tree shape structure. Considering differentiation pattern after cell division, this process can be determined the best as a kind of wavelet transformation. (Chapter 4-5, Thesis 1.b-c)

This observed natural process can be interpreted as an inverse image transformation, on this basis a wavelet transformation method was created. Since the means of this algorithm is a special, developmental biology inspired binary tree structure - alike as a multicellular organism develops from a single cell - further considering this, an entropy-based hierarchical compression method was developed. (Chapter 6-7, Thesis 2)

By following the states of a gene through a series of cell divisions an algorithm was set up, which is suitable for logical database transformation. A transformation was developed for logical variables, based on its properties it can be considered as a kind of wavelet transformation. This is also accompanied by an entropy-based hierarchical compression method, using multi-value logic. (Chapter 8-9, Thesis 3)

Finally a kinetic model was developed to simulate the physical movement of transfer RNA (tRNA) in the cytoplasm of *Escherichia coli* bacterium. The simulations carried out show that contrary to common believe tRNA molecules are not able to approach the ribosome by random Brownian motion at a sufficient rate for protein synthesis. There must be some other influence in the process to selectively influence cognate tRNAs movement or preselection. (Chapter 10, Thesis 4)

Hungarian Summary of the PhD Thesis - Az értekezés összefoglalása

Az értekezés a biológia aktuális problémáit vizsgálja újszerű megközelítésben, ez a magyarázata a „Biológiai modellek mérnöki alkalmazása” címnek. Az értekezésben bemutatott eredmények leírása tematikailag öt részre bontható.

Az első rész egy sejtszámozási rendszerről szól. A sejtek információtároló és örökítő anyaga a DNS, a DNS molekulák szemikonzervatíván másolódnak a sejtosztódások során. Bármely többsejtű élő organizmus egyetlen sejtből fejlődik ki egymást követő sejtosztódásokon keresztül. Ha a DNS másolatok sorsát követjük, akkor ez alapján egy bináris fát állíthatunk fel, ami a számozás alapját adja. Egyes biológiai kísérletek arra utalnak, hogy a sejtosztódás (aszexuális) jól koreografált, így feltehető, hogy a DNS kettős láncának pontosan ugyanaz a fele marad mindig a régi sejtben és mindig a másik megy az új sejtbe (halhatatlan szál hipotézis). A sejtvonalak és ezen számozási rendszer összekapcsolása segíthet abban, hogy megértsük, hogyan lehet tárolva a genomban a test felépítésére vonatkozó információ. (3. fejezet, 1.a tézis)

A testünk nagyságrendileg 10^{14} sejtből épül fel, a sejteket maximum 100-200 féle típusba lehet besorolni. Minden sejt a néhány száz sejtípus egyikébe tartozik. A keletkező új sejtek új helyet, pozíciót foglalnak el a 3-dimenziós térben, úgy tekinthető, mintha egy 3D volumetrikus kép lenne. Egy természetes algoritmust követve alakíthatunk át egy többdimenziós adatbázist bináris fa szerkezetté. Tekintve a sejtosztódást követő differenciálódás mintázatát, ezt a folyamatot legjobban egyfajta wavelet transzformációként lehet leírni. (4-5. fejezet, 1.b-c tézis)

Ez a megfigyelt természeti folyamat egy inverz képtranzformációként értelmezhető, ebből kiindulva egy wavelet transzformációs módszert készítettem. Mivel ennek az algoritmusnak egy speciális, fejlődésbiológia ihlette binárisfa struktúra az eszköze - hasonlóan ahogyan egy többsejtű szervezet kialakul egyetlen sejtből - ezt továbbgondolva fejlesztettem ki egy entrópia alapú hierarchikus tömörítési eljárást. (6-7. fejezet, 2. tézis)

Követve egy adott gén állapotait sejtosztódások sorozatán keresztül, egy olyan algoritmust állítottam fel, amivel logikai adatbázist lehet transzformálni. Létrehoztam egy transzformációt logikai változók esetére, amelyet tulajdonságai alapján egyfajta wavelet transzformációnak tekinthetünk. Ehhez társul egy szintén entrópia alapú hierarchikus tömörítési eljárás többértékű logika felhasználásával. (8-9. fejezet, 3. tézis)

Végül egy kinetikai modellt hoztam létre a transzfer RNS-ek fizikai mozgásának szimulációjára *Escherichia coli* baktérium citoplazmájában. Az elvégzett szimulációk arra utalnak, hogy az általános meggyőződéssel ellentétben a tRNS molekulák nem képesek csak a véletlenszerű Brown-mozgás révén olyan gyakorisággal megközelíteni a riboszómát, ami elegendő lenne a fehérjeszintézis fenntartásához. Lennie kell még valamilyen más hatásnak is a folyamatban, ami a kognáns tRNS-ek mozgását vagy előkiválasztását szelektíven befolyásolja. (10. fejezet, 4. tézis)

CONTRIBUTIONS

Thesis 1

As far a multicellular organism is considered a three-dimensional object, just like the progression of cell division and differentiation can be described by wavelet transformation. Based on this:

- 1.a. set up a cell numbering system to support the applicability of wavelet transformation,**
- 1.b. proved the existence of that wavelet transformation, which can be represented by the same binary tree, like cell division progression,**
- 1.c. showed that similarly to the spatial orientation of cell division, downsampling required for wavelet transformation may have different spatial orientations.**

By proposing a cell numbering system we could be able to name and identify each and every individual descendant of a predecessor chromosome or prokaryotic cell (a zygote, practically the first cell after sexual cell division or crossing over). We can assume that cell division (asexual) is well choreographed, so we can suppose that exactly the same half of the DNA double-chain should remain always in the old cell and always the other goes to the new cell (immortal strand hypothesis). Linking cell lineages and this numbering system could help our understanding about how body plan information could be stored in our genome. Each cell belongs to one of the few hundred cell types. The resulting new cells occupy an available new location, position in the 3 dimensional space, it resembles a kind of image. Considering the pattern of cell differentiations after divisions, this process best could be interpreted as a kind of wavelet transformation. Describing cell divisions and differentiations brings us closer to the understanding, how body plan information is compressed, what is the possible structure of this information. Based in this, author has developed an algorithm to transform and compress datasets for storage and transmission purposes.

Thesis 2

In the course of formatting the thesis about a biologically inspired wavelet transformation and compression method for multi-dimensional image, discrete data set:

- 2a. Exemplified analogy between cell division binary tree on chromosome-level and image or discrete database wavelet transformation in conformity with Thesis 1. The transformed dataset is suitable for the subsequent compression.**

- 2b. Defined an entropy like term resembling the orderliness of an image. Applying this, a hierarchical discrete dataset or image compression method was developed.**

This thesis is about an algorithm, to transform and compress discrete dataset for storage and transmission purposes. The transformation is based on the same algorithm how a multicellular organism develops from a single cell by consecutive cell divisions and differentiations. This algorithm could provide an economical way for handling a correlated or highly correlated discrete or quantized analog signal, image or any multidimensional dataset. The means of this algorithm is a special binary tree structure, inspired by developmental biology, the way, how a multicellular organism develops from a single cell. This observed natural process is an inverse transformation so on this base is possible to prepare a corresponding wavelet transformation method.

There are different methods to compress digital signals, datasets or images, each of them has some special characteristics. The procedure introduced here is entropy based adaptive compression method, its main feature is the hierarchical structure and entropy based compression. Wavelet transform of a dataset could be arranged to a binary tree like structure, which is the starting point of this compression. Degree of complexity of an image varies, different parts have different information content. Detailed regions require more thorough description than relaxed, predictable parts, which need less. To accomplish this task a new type of variable is introduced, resembling the entropy of the particular area of the image. The higher is the value of this variable indicating entropy, the higher is the importance of the area's detail coefficients.

Thesis 3

The status of a gene is actually considered as a logical variable, its changes in the course of cell divisions can be represented as a binary tree, likewise a logical binary data set, bitmap or a monochromatic image too, therefore:

- 3a. An algorithm was developed, which can describe the gene-level cell division and differentiation, and can transform a multi-dimensional bitmap into binary tree structure. This logical variable transformation shows structural similarity to the discrete wavelet transformation described in Thesis 2.**
- 3b. An entropy concept was created for this binary tree transformation that is capable to compress logical data sets, binary images. On this basis a hierarchical entropy based image compression method was worked out to compress bitmaps or binary images using Boolean algebra and multi-valued logic variables.**

Any multicellular living organism develops from a single cell, through consecutive cell divisions. After cell division one of the daughter cells retains the properties of the

initial cell, the other might be the same or different. Any cell of an organism contains the same genetic information. The type of a cell is determined by its gene expression profile, some of the genes are enabled for transcription and the others are repressed, through epigenetic modifications. This thesis explains an algorithm, which was developed to transform logical dataset, like a status of a particular gene, for storage and transmission purposes. This algorithm could provide an economical way for handling a correlated or highly correlated logical dataset. The means of this algorithm is a special tree structure to transform a multidimensional dataset into a binary tree structure, following a natural algorithm. Hereinafter is shown a potentially possible approach, how to compress this kind of bitmap. The main advantage of this lossless method is that the resolution of the reconstructed image is doubling by each step.

When we have a set of logical variables and the elements of this dataset are correlated to each others, then we are able to predict any element of it higher than 50% accuracy. To predict the value of an element usually we compare it to its direct neighbor or neighbors. The principle of wavelet transformation is - and some other transformation's too- the comparison between its elements. The result of this kind of transformations is usually a perfect binary tree. The higher is the predictability of the elements, the higher is the percentage of zeros than the ones in the transformed image. The aim of any image transformation to binary images is, to reduce the number of 1-s compared to zeros. This could increase the efficiency of any compression method.

Thesis 4

A preliminary kinetic model was set up to imitate the physical movement of transfer RNA (tRNA) in the cytoplasm of *Escherichia coli* bacteria. The obtained simulation results verify the hypothesis that tRNA molecules are not able to approach the ribosome by random Brownian motion at a rate which would be sufficient to maintain the speed of protein synthesis.

Transfer RNAs (tRNAs) can recognize a specific amino acid from a possible pool of 20. They are able to transport these protein building-blocks to the ribosome, the site where amino acids assemble into protein chains. Accurate and rapid selection of tRNAs by the ribosome is critical for cell survival. The aim of this thesis is to develop a preliminary and simple model of tRNA molecular movement in the bacterial (*Escherichia coli*—*E. coli*) cytoplasm. Spatial movement/placement of aminoacyl-tRNAs (aa-tRNA or charged tRNA) were examined in the cytoplasm—viewed from the perspective of that particular aa-tRNA. To achieve this goal, a kinetic model of the interaction between messenger RNA, ribosome, and RNA molecules is developed. The purpose of the simulation is to examine the conditions necessary for the tRNA to deliver a particular amino acid to the ribosome within a biological timeframe. Simulation results show that it is unlikely that tRNAs are able to reach the “A site” of the ribosome by random movement.

Furthermore three potentially probable mechanisms were proposed to explain tRNA pre-selection (to distinguish it from initial selection, it is referred in this presentation as “pre-selection”) in prokaryotes.

10.9 Theses in Hungarian - Tézisek

1. Tézis

Amennyiben egy többsejtű élőlényt egy háromdimenziós objektumnak tekintünk, úgy a sejtosztódás és differenciálódás progressziója leírható wavelet transzformációval. Ennek alátámasztására:

- 1.a. egy sejtszámozási rendszert állítottam fel, amely megalapozza a wavelet transzformáció alkalmazhatóságát,**
- 1.b. igazoltam, hogy létezik olyan wavelet transzformáció, ami hasonló binárisfával ábrázolható, mint a sejtosztódás progressziója,**
- 1.c. megmutattam, hogy a sejtosztódás térbeli irányultságához hasonlóan lehet a wavelet transzformációhoz szükséges „felbontás csökkentésnek” (downsampling) is különféle térbeli irányultsága.**

Egy sejt számozási rendszert javasolva képesek lehetünk megnevezni és azonosítani minden egyes előd kromoszóma példányt vagy prokarióta sejt leszármazottját (zigóta, gyakorlatilag az első sejt szexuális sejtosztódás vagy crossing over után). Feltételezhetjük, hogy a sejtosztódás (aszexuális) jól koreografált, így feltehető, hogy a DNS kettős láncának pontosan ugyanaz a fele marad mindig a régi sejtben és mindig a másik megy az új sejtbe (halhatatlan szál hipotézis). A sejtvonalak és ezen számozási rendszer összekapcsolása segíthet abban, hogy megértsük, hogyan lehet tárolva a genomban a test felépítésére vonatkozó információ. Minden sejt a néhány száz sejtípus egyikébe tartozik. A keletkező új sejtek új helyet, pozíciót foglalnak el a 3-dimenziós térben, úgy tekinthető, mintha egy kép lenne. Tekintve a sejtosztódást követő differenciálódás mintázatát, ezt a folyamatot a legjobban egyfajta wavelet transzformációként lehet meghatározni. A sejtosztódás és differenciálódás leírása közelebb visz ahhoz, hogy megértsük, hogyan vannak tömörítve a testi felépítésre vonatkozó adatok, mi a lehetséges struktúrája ennek az információnak. Ezek alapján a szerző kifejlesztett egy adathalmaz transzformációs és tömörítési algoritmust, tárolási és átviteli célokra.

2. Tézis

A tézisben ismertetett biológiai indíttatású többdimenziós kép, diszkrét adathalmaz wavelet transzformációjára és tömörítésére szolgáló eljárás megalkotása során:

2.a. Analógiát mutattam ki a sejtosztódás kromoszóma szintű binárisfája és egy többdimenziós kép vagy diszkrét adatbázis 1. tézis szerinti wavelet transzformációja között. Ez a transzformált adathalmaz lesz alkalmas további tömörítésre.

2.b. Definiáltam egy képrendezettségre vonatkozó entrópia jellegű fogalmat. Ezt felhasználva kidolgoztam egy hierarchikus diszkrét adathalmaz vagy képtömörítési módszert.

Ez a tézis egy algoritmus, diszkrét adathalmaz transzformálására és tömörítésére tárolási és átviteli célokra. A transzformáció ugyanazon az algoritmuson alapul, ahogy egy többsejtű élőlény kialakul egyetlen sejtől egymást követő sejtosztódások és differenciálódásokon át. Ez az algoritmus gazdaságos módszer korrelált vagy erősen korreláló diszkrét, kvantált analóg jel, kép vagy többdimenziós adathalmaz kezelésére. Ennek az algoritmusnak egy speciális, fejlődésbiológia ihlette binárisfa struktúra az alapja, hasonlóan, ahogy egy többsejtű szervezet kialakul egyetlen sejtől. Ez a természetben megfigyelhető folyamat tulajdonképpen egy inverz transzformáció, ami alapján egy ennek megfelelő wavelet transzformációs módszert is lehet készíteni.

Különböző módszerek léteznek digitális jelek, adatbázisok vagy képek tömörítésére, mindegyiknek van valamilyen különleges tulajdonsága. Az itt bevezetett eljárás egy entrópia alapú adaptív tömörítési módszer, fő jellemzője a hierarchikus felépítés és az entrópia alapú tömörítés. Egy adatbázis wavelet transzformáltját el lehet rendezni egy binárisfa szerű struktúrává, ami a kiindulási pontja ennek a tömörítésnek. Egy kép összetettségének foká változó, különböző részeinek különböző az információ tartalma. Részlet dús régiók bőségesebb leírást igényelnek, mint a nyugodt, kiszámítható részekből állók, amihez kevesebb is elég. A feladat megoldására egy új típusú változó került bevezetésre, ami a kép egy adott területének entrópiáját testesíti meg. Minél magasabb az értéke ennek az entrópiára utaló változónak, annál nagyobb a terület részletegyütthatóinak jelentősége.

3. Tézis

Egy gén állapota voltaképpen egy logikai változónak tekinthető, sejtosztódások közötti megváltozása binárisfa alakban ábrázolható, szintúgy egy bináris logikai adathalmaz, bittérkép vagy monokromatikus kép is, ezért:

3a. Kidolgoztam egy algoritmust, amellyel leírható a gén szintű sejtosztódás és differenciálódás, és amivel egy több dimenziós bitkép is binárisfa alakba transzformálható. Ez a logikai változós transzformáció strukturális hasonlóságot mutat a 2. tézisben ismertetett diszkrét wavelet transzformációval.

3b. Ezen binárisfa transzformációhoz olyan entrópia fogalmat alakítottam ki, mely alkalmas logikai adathalmazok, bináris képek tömörítésére. Ez alapján egy hierarchikus entrópia alapú képtömörítési módszert dolgoztam ki bittérképek vagy bináris képek tömörítésére Boole-algebra és többértékű logikai változók felhasználásával.

Bármely többsejtű élő organizmus egyetlen sejtből fejlődik ki, egymást követő sejtosztódásokon keresztül. Sejtosztódás után az egyik leánysejt megtartja a kiindulási sejt tulajdonságait, a másik lehet azonos vagy különböző. Egy élőlény minden sejtje ugyanazt a genetikai információt tartalmazza. Egy sejt típusát a gén expressziós profilja határozza meg, egyes gének transzkripciója engedélyezve van, míg másoké epigenetikus módosításokon keresztül elhallgattatva. Ez a tézis egy algoritmust ír le, ami logikai adatbázis transzformálására lett kifejlesztve, mint egy adott gén állapotai, tárolási és átviteli célokra. Ez az algoritmus gazdaságos módja korrelált vagy erősen korreláló logikai adathalmaz kezelésére. Ez az algoritmus egy speciális fa struktúra segítségével alakít át egy többdimenziós adatbázist bináris fa szerkezetté, egy természetes algoritmust követve. Továbbiakban bemutat egy potenciálisan lehetséges megközelítést, hogyan lehet tömöríteni ezt a fajta bittérképet. A legfőbb előnye ennek a veszteségmentes módszernek, hogy a rekonstruált kép felbontása duplázódik minden egyes lépéssel.

Ha van egy logikai változókból álló halmazunk és az adatbázis elemei összefüggnek egymással, akkor képesek vagyunk megjósolni bármely elemét több mint 50%-os pontossággal. Egy elem értékének becsléséhez azt általában a közvetlen szomszédjával vagy szomszédjaival vetjük egybe. A wavelet transzformáció alapelve - és néhány más transzformációé is- az elemei közötti összehasonlítás. Az ilyen típusú átalakítások eredmény rendszerint egy tökéletes binárisfa. Minél nagyobb az elemek kiszámíthatósága, annál nagyobb százalékban találhatók nullák a transzformált képben, mint 1-esek. Minden bináris képtranszformáció célkitűzése az, hogy csökkentsék az 1-sek számát a nullákéhoz képest. Ez növeli bármely tömörítési módszer hatékonyságát.

4. Tézis

Egy előzetes kinetikai modellt állítottam fel a szállító RNS (tRNS) fizikai mozgásának modellezésére *Escherichia coli* baktérium citoplazmájában. A kapott szimulációs eredmények azt a hipotézist igazolják, hogy a tRNS molekulák véletlenszerű Brown-mozgással nem tudják olyan ütemben megközelíteni a riboszómát, ami elegendő lenne a fehérjeszintézis sebességének fenntartásához.

Transzfer RNS-ek (tRNS-ek) képesek felismerni egy adott aminosavat a lehetséges 20-as halmazból. Ezek szállítják a fehérje építőköveit a riboszómához, a helyhez, ahol az aminosavak összeállnak fehérje láncokká. A sejtek túlélésének kritériuma, hogy a riboszóma pontosan és gyorsan válassza ki a tRNS-eket. A tézis

célkitűzése, hogy kidolgozzon egy előzetes és egyszerű modellt a tRNS molekulák mozgására a baktérium (*Escherichia coli-E. Coli*) citoplazmájában. Aminoacyl-tRNS (aa-tRNS vagy töltött tRNS) térbeli mozgását/elhelyezkedését vizsgáltuk a citoplazmában- az adott aa-tRNS szemszögéből nézve. E cél elérése érdekében a hírvivő RNS, a riboszómák és a tRNS molekulák közötti kölcsönhatásról egy kinetikus modell lett kidolgozva. A szimuláció célja megvizsgálni a szükséges feltételeit annak, hogy a tRNS egy adott aminosavat a riboszómához szállítson a biológiailag elégséges időkereten belül. A szimulációk eredményei azt mutatják, nem valószínű, hogy a tRNS-ek képesek elérni a riboszóma "A" helyét véletlenszerű mozgással.

Továbbá javasolok három potenciálisan lehetséges mechanizmust a tRNS kiválogatódás magyarázatára a prokariotákban (megkülönböztetésül az "initial selection"-től ebben az értekezésben úgy nevezzük "pre-selection").

EPILOGUE

Studying nature could help us to utilize this knowledge in seemingly unrelated fields of science too, example we can employ this wavelet interpretation in digital image processing.

The aim of natural sciences is to find out the laws of nature. To find out how body plan information is stored, we have to find regularities in species development. To follow copies of DNA strands seems to be one of the most promising ways. We have to study more how chromosome scaffold change shape during cell divisions, its role in regulating gene expression, the role of centriole and spindles in correct segregation of chromosomes.

Advanced molecular biology research techniques and methods make it possible now to follow individual chromosomes and DNA strands; for example it could be utterly interesting to follow the fate of each and every individual DNA strands of a model organism like *C. elegans*.

Our knowledge seems to be not sufficient yet for the understanding of this kind of biological processes, so the time has not yet come for conscientiously engineering genetically modified organisms (GMO).

“Ontogeny recapitulates phylogeny”, Ernst Haeckel formulated this biogenetic law in 1872 [67], legendarily declaring, that embryonic development repeats similar steps as it could have happened during evolution.

ACKNOWLEDGEMENTS

I would like to express my gratitude to everyone who helped me in my studies leading to this thesis work. Utmost thank to my supervisor, Professor Péter Korondi, head of the Department of Mechatronics, Optics and Mechanical Engineering Informatics, Budapest University of Technology and Economics, who provided an excellent environment to accomplish my aspirations and to the members of the department.

This research was started while I was a graduate student at American University of Sharjah, special thanks to Dr. Ghaleb A. Hussein. Before that Prof. Zoltán Papp from Medical and Health Science Centre, University of Debrecen smoothed my first steps in the world of biology. My work was also supported by University of Debrecen, Faculty of Engineering and by my colleagues from the Department of Electrical and Mechatronics Engineering.

The research of S. Piros was supported by the European Union and the State of Hungary, co-financed by the European Social Fund in the framework of TÁMOP 4.2.4. A/2-11-1-2012-0001 'National Excellence Program'.

Special thanks to my family supporting me in my studies all the way to reach here.

BIBLIOGRAPHY

- [1] [Online]. Available:
http://www.nobelprize.org/educational_games/medicine/dna/index.html.
- [2] M. Nirenberg and P. Leder, "RNA codewords and protein synthesis," *Science*, no. 145, pp. 1399-407, 25 Sep 1964.
- [3] G. M. Cooper, *The Cell; A Molecular Approach*, 2000.
- [4] H. Lodish, A. Berk, S. L. Zipursky, P. Matsudaira, D. Baltimore and J. E. Danell, *Molecular Cell Biology*, 2000.
- [5] K. Sanbonmatsu, S. Joseph and C. Tung, "Simulating movement of tRNA into the ribosome during decoding," *PNAS*, pp. 15854-15859, 2005.
- [6] C. S. Fraser and J. W. Hershey, "Movement in ribosome translocation," *Journal of Biology*, vol. 4, no. 2.
- [7] R. Polikar, "The Wavelet Tutorial," Rowan University, College of Engineering Web Servers, 12 1 2001. [Online]. Available:
<http://users.rowan.edu/~polikar/WAVELETS/WTtutorial.html>.
- [8] Fadil Santosa, "Second Generation Wavelets: Theory and Application," University of Minnesota, Institute for Mathematics and its Applications, 06 Oct. 2011. [Online]. Available:
http://www.ima.umn.edu/industrial/97_98/sweldens/fourth.html. [Accessed 16 Apr. 2013].
- [9] J. W. Woods, *Multidimensional Signal, Image, and Video Processing and Coding*, Waltham: Elsevir Inc., 2012.
- [10] J. Mukhopadhyay, *Image and Video Processing in the Compressed Dimain*, Boca Raton: CRC Press, 2011.
- [11] T. Stearns, "Stem cells: A fateful age gap," no. 461, pp. 891-892, 15 October 2009.
- [12] M. R. P. G. T. F. Stewart EJ, "Aging and Death in an Organism That Reproduces by Morphologically Symmetric Division," *PLoS Biol*, vol. 3, no. 2, 2005.
- [13] C. E. H. A. L. P. Falconer E, "Chromosome orientation fluorecence in situ hybridization to study sister chromatid segregation in vivo," *NATURE PROTOCOLS*, vol. 5, no. 7, pp. 1362-1377, 2010.

- [14] S. Tajbakbsh, P. Rocheteau and I. Le Roux, "Asymmetric Cell Divisions and Asymmetric Cell Fates," *ANNUAL REVIEW OF CELL AND DEVELOPMENTAL BIOLOGY*, vol. 25, pp. 671-699, 2009.
- [15] S. F. Gilbert, *Developmental Biology*, 2010.
- [16] A. Cedilnik, J. Baumes, L. Ibanez, S. Megason and B. Wylie, "Integration of information and volume visualization for analysis of cell lineage and gene expression during embryogenesis," in *Visualization and Data Analysis 2008*, San Jose, CA, USA, 2008.
- [17] R. A. Cartwright and D. Graur, "The Multiple Personalities of Watson and Crick Strands.," *Biology Direct*, Feb. 2011.
- [18] A. M. DESHPANDE and C. S. NEWLON, "The ARS Consensus Sequence Is Required for Chromosomal," *MOLECULAR AND CELLULAR BIOLOGY*, vol. 12, no. 10, pp. 4305-4313, Oct. 1992.
- [19] P. Lansdorp, "Immortal Strands? Give Me a Break," *Cell*, no. 129, pp. 1244-1247, 29 June 2007.
- [20] P. Karpowicz, M. Pellikka, E. Chea, D. Godt, U. Tepass and D. van der Kooy, "The germline stem cells of *Drosophila melanogaster* partition DNA non-randomly," *European Journal of Cell Biology*, vol. 88, no. 7, pp. 397-408, Jul. 2009.
- [21] P. H. Thorpe, J. Bruno and R. Rothstein, "Kinetochore asymmetry defines a single yeast lineage," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 106, no. 16, p. 66736678, 21 Apr. 2009.
- [22] J. Gerhart and M. Kirschner, *Cells, Embryos and Evolution*, 1997.
- [23] E. G. Conklin, *Journal of the Academy of Natural Sciences of Philadelphia*, vol. 13, pp. 1-119, 1905.
- [24] E. B. Wilson, "Science is Beauty," [Online]. Available: <http://scienceisbeauty.tumblr.com/post/37104503090/general-view-of-cells-in-the-growing-root-tip-of>. [Accessed 30 07 2014].
- [25] UT Southwestern Medical Center , "HOW FAT CELLS ARE CREATED:," [Online]. Available: <http://www.wealthofhealth.co.za/how-fat-cells-are-created>. [Accessed 30 07 2014].
- [26] M. Ankerst, G. Kastenmüller, H.-P. Krieger and T. Seidl, "Nearest Neighbor Classification in 3D Protein Databases," in *ISMB-99 Proceedings*, 1999.
- [27] P. J. Hines and S. Kadereit, "Asymmetric Cell Division," 2004.

- [28] R. Ghosh and C. Tomlin, "Symbolic Reachable Set Computation of Piecewise Affine Hybrid Automata and its Application to Biological Modelling: Delta-Notch Protein Signalling".
- [29] G. Marnellos, G. A. Deblandre, E. Mjolsness and C. Kintner, "Delta-Notch lateral inhibitory patterning in the emergence of ciliated cells in *Xenopus*: experimental observations and a gene network model.," 2000.
- [30] A. & S. K. Ralston, "Gene expression regulates cell differentiation.," 2008.
- [31] J. Beutel, Y. Kim and S. C. Horii, Handbook of Medical Imaging: Display and PACS, 2000.
- [32] K. H. Talukder and K. Harada, "Haar Wavelet Based Approach for Image Compression and Quality Assessment of Compressed Image," *International Journal of Applied Mathematics*, vol. 36:1, no. 1, 2007.
- [33] S. J. Piros and P. Korondi, "Developmental Biology from Informatics Point of View," in *11th International Symposium on Computational Intelligence and Informatics (CINTI)*. , Budapest, 2010.
- [34] K. F. S. L. Nick D. Panagiotacopulos, "Lossy Image Compression Using Wavelets," *Journal of Intelligent and Robotic Systems*, vol. 28, no. 1-2, pp. 39-59, 2000.
- [35] S.-G. P. L. G. ., C. C. Radu-Daniel VATAVU, "Modeling Shapes for Pattern Recognition: A Simple Low-Cost Spline-based Approach," in *9th International Conference on DEVELOPMENT AND APPLICATION SYSTEMS*, Suceava, Romania, May 2008.
- [36] M. A. El-Sayed and T. A.-E. Hafeez, "New Edge Detection Technique based on the Shannon Entropy in Gray Level Images," *International Journal on Computer Science and Engineering (IJCSE)*, vol. Vol. 3, no. No. 6 June, June 2011.
- [37] C. Croonenbroeck, "Local Entropy Based Image Reconstruction," *Discussion Paper No. 312*, January 2012.
- [38] S. Deutsch, A. Averbuch and S. Dekel, "Adaptive compressed image sensing based on wavelet modeling and direct sampling," in *"SAMPTA'09*, Marseille : France, 2009.
- [39] N. Shakhakarmi, "Quantitative Multiscale Analysis using Different Wavelets in 1D Voice Signal and 2D Image," *IJCSI International Journal of Computer Science Issues*, Vols. Vol. 9., no. Issue 1, No 2., January 2012.

- [40] USC-SIPI Image Database, "Girl (Lena, or Lenna)," [Online]. Available: <http://sipi.usc.edu/database/database.php?volume=misc&image=12#top>. [Accessed 30 07 2014].
- [41] N. Huck-Hui and A. Bird, "DNA methylation and chromatin modification," *Current Opinion in Genetics & Development*, vol. 9, no. 2, pp. 158-163, April 1999.
- [42] S. J. Piros and P. Korondi, "Biologically inspired informatics; algorithm for logical data processing," in *2nd International Conference on Cognitive Infocommunications*, Budapest.
- [43] A. M. Casto and C. Amid, "Beyond the Genome: genomics research ten years after the human genome sequence," *Genome Biol.*, vol. 11(11), 30 Nov 2010.
- [44] M. Stabno and R. Wrembel, "RLH: Bitmap compression technique based on run-length and Huffman encoding," vol. 34, no. 4-5, pp. 400-414, June-July 2009.
- [45] W.-H. Steeb, *Nonlinear Workbook: Chaos, Fractals, Cellular Automata, Neural Networks, Genetic Algorithms, Gene Expression*, World Scientific Publishing Company, 2008.
- [46] D. M. Miller and M. A. Thornton, *Multiple valued logic: concepts and representations*, Morgan & Claypool Publishers, 2008.
- [47] D. A. HUFFMAN, "A Method for the Construction of Minimum-Redundancy Codes," *Proceedings of The Institute of Radio Engineers*, vol. 40, no. 9, pp. 1098-1101, 1952.
- [48] K. P. S. G. J. A. L. Zany Péter, "Image based automatic object localisation in iSpace environment.," in *10th International Symposium of Hungarian Researchers on Computational Intelligence and Informatics (CINTI 2009)*, Budapest., 2009.
- [49] M. Schwehm, "Parallel Stochastic Simulation of Whole-Cell Models," 2001.
- [50] Institute for Biomolecular Design, "E.coli Statistics ; Project CyberCell," [Online]. Available: http://redpoll.pharmacy.ualberta.ca/CCDB/cgi-bin/STAT_NEW.cgi. [Accessed 2007].
- [51] M. Elowitz, M. Surette, P. Wolf, J. Stock and L. S., "Protein Mobility in the Cytoplasm of Escherichia coli," *Journal of Bacteriology*, pp. 197-203, 1999.
- [52] J. Deich, E. M. Judd, H. H. McAdams and W. E. Moerner, "Visualization of the movement of single histidine kinase molecules in live Caulobacter cells," *PNAS*, vol. 101, no. 45, pp. 15921-15926, 9 Nov. 2004.

- [53] I. Golding and E. C. Cox, "Physical Nature of Bacterial Cytoplasm," *Physical Review Letters*, pp. 96-99, 2006.
- [54] G. Broderick, M. Ruaini, E. Chan and M. Ellison, "A Life-Like Virtual Cell Membrane Using Discrete Automata," *In Silico Biology*, pp. 163-178, 2005.
- [55] S. J. Plimpton and S. A., "Microbial Cell Modeling via Reacting Diffusive Particles," *Journal of Physics*, pp. 305-09, 2005.
- [56] K. Sanbonmatsu, "Alignment/misalignment hypothesis for tRNA selection by the ribosome," *Biochimie*, pp. 1075-1089, 2006.
- [57] K. Sanbonmatsu, "Energy landscape of the ribosomal decoding center," *Biochimie*, pp. 1053-1059, 2006.
- [58] S. J. Piros and G. A. Hussein, "Preliminary Modeling of Transfer RNA Kinetics in the Cytoplasm of Escherichia coli Bacteria," *ADVANCED SCIENCE LETTERS*, vol. 3, pp. 28-36, Mar 2010.
- [59] S. J. Piros and G. A. Hussein, "Possible Physical Mechanisms of tRNA Pre-Selection in the Cytoplasm of Escherichia coli Bacteria," *ADVANCED SCIENCE LETTERS*, vol. 3, p. 37-42, Mar 2010.
- [60] [Online]. Available: <http://www.lbl.gov/Science-Articles/Archive/sabl/2005/November/ribosome-groove.html>.
- [61] Colibri, "<http://genolist.pasteur.fr/Colibri/genome.cgi>," 1999. [Online].
- [62] J. Ninio, "Multiple stages in codon-anticodon recognition: double-trigger mechanisms and geometric constraints," *Biochimie*, pp. 963-992, 2006.
- [63] T. Pape, W. Wintermeyer and M. Rodnina, "Induced fit in initial selection and proofreading of aminoacyl-tRNA on the ribosome," *The EMBO Journal*, pp. 3800-3807, 1999.
- [64] T.-H. Lee, S. C. Blanchard, H. D. Kim, J. D. Puglisi and S. Chu, "The role of fluctuations in tRNA selection," *PNAS*, vol. 104, no. 34, pp. 13661-13665, 21 August 2007.
- [65] H. S. Zaher and R. Green, "Fidelity at the molecular level: lessons from protein synthesis," *Cell*, vol. 136, no. 4, pp. 746-762, 20 February 2009.
- [66] E. B. Kramer and P. J. Farabaugh, "The frequency of translational misreading errors in E. coli is largely determined by tRNA competition," *RNA*, no. 14, pp. 2407-2416, 1 November 2008.

- [67] L. Olsson, G. S. Levit and U. Hoßfeld, "Evolutionary developmental biology: its concepts and history with a focus on Russian and German contributions," *Naturwissenschaften*, vol. 97, no. 11, pp. 951-969, November 2010.

Related Publications of the Author

- [33] S. J. Piros and P. Korondi, "Developmental Biology from Informatics Point of View," in *11th International Symposium on Computational Intelligence and Informatics (CINTI)*, Budapest, 2010.
- [42] S. J. Piros and P. Korondi, "Biologically inspired informatics; algorithm for logical data processing," in *2nd International Conference on Cognitive Infocommunications*, Budapest.
- [58] S. J. Piros and G. A. Hussein, "Preliminary Modeling of Transfer RNA Kinetics in the Cytoplasm of Escherichia coli Bacteria," *ADVANCED SCIENCE LETTERS*, vol. 3, pp. 28-36, Mar 2010.
- [59] S. J. Piros and G. A. Hussein, "Possible Physical Mechanisms of tRNA Pre-Selection in the Cytoplasm of Escherichia coli Bacteria," *ADVANCED SCIENCE LETTERS*, vol. 3, p. 37-42, Mar 2010.
- [68] S. Piros and P. Korondi, "INFORMATIKA A BIOLÓGIÁBAN, BIOLÓGIA AZ INFORMATIKÁBAN," in *Informatika a felsőoktatásban 2011 konferencia: IF2011*, Debrecen, 2011.
- [69] S. J. Piros, "Tracing the Immortal Strand: Cell Numbering and Lineage Representation as Wavelet Transformation," in *Proceedings of the IASTED International Conference: Computational Bioscience*, Cambridge, 2011.
- [70] S. J. Piros and P. Korondi, "Compression Method for Binary Tree Like Bitmaps," in *Eighth International Symposium on Mechatronics and its Applications*, Sharjah, 2012.
- [71] S. J. Piros and P. Korondi, "Biologically inspired informatics; algorithm for discrete data and signal processing,," in *Institute of Electrical and Electronics Engi-neers(ed.) Proceedings IEEE/ASME International Conference on Advanced Intelligent Mechatronics.*, 2011.
- [72] S. J. Piros and P. Korondi, "Who said that pixels should be squares?," in *Proceedings of CERiS'13 - Workshop on Cognitive and Eto-Robotics in iSpace.*, Budapest, 2013.