DNA Microarray Images: Processing, Modelling, Compression

DNA MICROARRAY IMAGES: PROCESSING, MODELLING, COMPRESSION

BY

NASER FARAMARZPOUR

APRIL 2004

A THESIS
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING
AND THE SCHOOL OF GRADUATE STUDIES
OF MCMASTER UNIVERSITY
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE

Master of Applied Science (2004)	McMaster University
(Electrical & Computer Engineering)	Hamilton, Ontario

TITLE:	DNA Microarray Images: Processing, Modelling, Compression
AUTHOR:	Naser Faramarzpour B.Sc. (Electrical Engineering) Sharif University of Technology, Tehran, Iran
SUPERVISORS:	Dr. S. Shirani and Dr. M. J. Deen
NUMBER OF PAGES:	x, 74

Acknowledgments

I would like to thank Dr. S. Shirani and Dr. M. J. Deen for their guidance and for giving me the opportunity for academic advancement. Without their careful supervision, encouragement, assistance and feedback, this thesis fwould not have been completed. I am also thankful to Profs. Kumar, Li, and Wu for agreeing to be on my thesis examining committee and for their comments on my research.

I would also like to thank Natural Sciences and Engineering Research Council (NSERC) and MICRONET for financial support of this work.

Finally, I wish to thank my family and friends for their encouragement and support in my academic pursuits.

Abstract

DNA Microarray is an innovative tool for gene studies in biomedical research. It is capable of testing and extracting the expression of large number of genes in parallel. Its applications can vary from cancer diagnosis to human identification. A DNA microarray experiment generates an image which has the genetic data embedded in it. Fast, accurate, and automatic routines for processing and compression of these images do not exist.

For processing and modelling of micoarray images, we introduce a new, fast and accurate approach in this thesis. A new lossless compression method for microarray images is introduced that provides an average compression ratio of 1.89:1, and that outperforms other lossless compression schemes and the work of other researchers in this field. For the lossy compression, our new method has overcome the rate-distortion curve of JPEG. A new scanning method called spiral path, and a new spatial transform called C2S are introduced in this thesis for lossless and lossy compression of microarray images.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

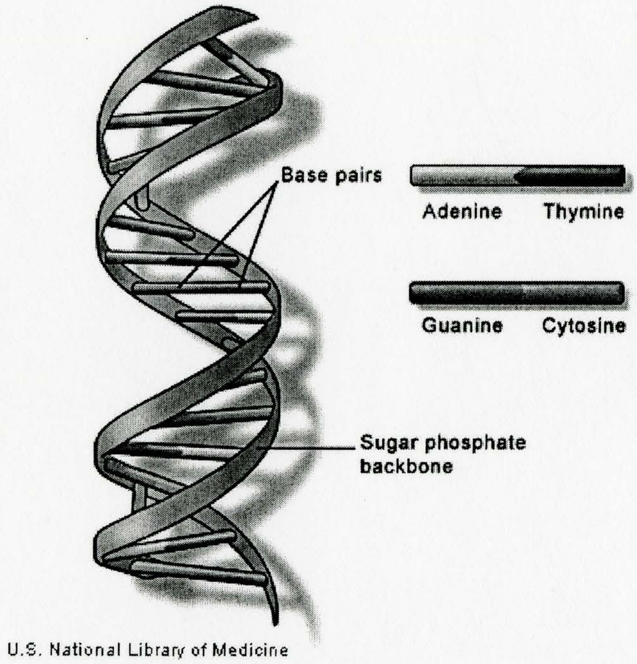
1.1 Background

Deoxyribonucleic acid or DNA, is the hereditary material in humans and all living organisms. Nearly every cell in a person's body has the same DNA. Most DNA is located in the cell nucleus where it is called nuclear DNA, but a small amount of DNA can also be found in the mitochondria where it is called mitochondrial DNA or mtDNA [1].

The information in DNA is stored as a code made up of four chemical bases: adenine (A), guanine (G), cytosine (C), and thymine (T). Human DNA consists of about 3 billion bases, and more than 99 percent of those bases are the same in all people. The order or sequence of these bases determines the information available for building and maintaining an organism, similar to the way in which letters of the alphabet appear in a certain order to form words and sentences. A model of DNA is shown in Fig. 1.1.

DNA bases pair up with each other, A with T and C with G, to form units called base pairs. Each base is also attached to a sugar molecule and a phosphate molecule. Together, a base, sugar, and phosphate are called a nucleotide. Nucleotides are arranged in two long strands that form a spiral called a double helix. The structure of the double helix is somewhat like a ladder, with the base pairs forming the rungs of the ladder and the sugar and phosphate molecules forming the vertical sidepieces of the ladder.

An important property of DNA is that it can replicate, or make copies of itself. Each strand of DNA in the double helix can serve as a pattern for duplicating the sequence of bases. This is critical when cells divide because each new cell needs to have an exact copy of the DNA present in the old cell [16].



U.S. National Library of Medicine

Figure 1.1: DNA structure [3].

A gene is the basic physical and functional unit of heredity. Genes, which are made up of DNA, act as instructions to make molecules called proteins. In humans, genes vary in size from a few hundred DNA bases to more than 2 million bases. The Human Genome Project has estimated that humans have between 30,000 and 40,000 genes (Fig. 1.2).

Every person has two copies of each gene, one inherited from each parent. Most genes are the same in all people, but a small number of genes (less than 1 percent of the total) are slightly different in each individual. Alleles are forms of the same gene with small differences in their sequence of DNA bases. These small differences contribute to each person's unique physical features [5].

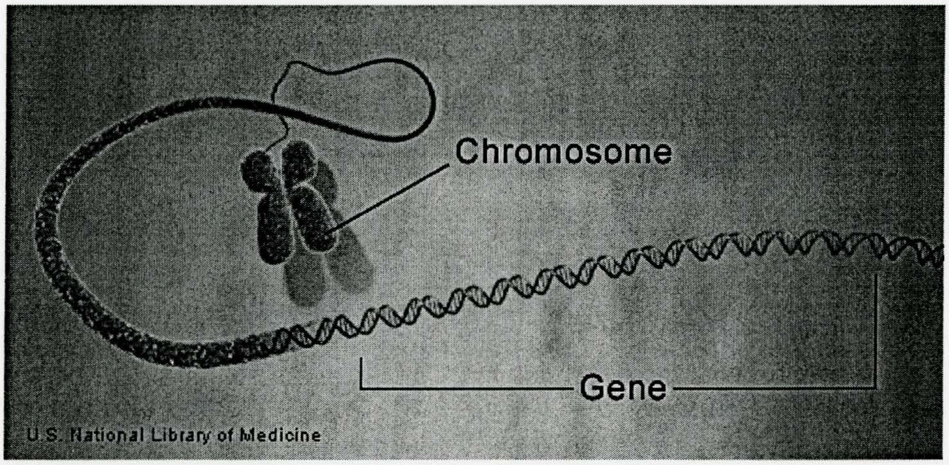


Figure 1.2: Gene in a DNA sequence [3].

Most genes contain the information needed to make functional molecules called proteins. A few genes produce other molecules that help the cell assemble proteins. The journey from gene to protein is complex and tightly controlled within each cell. It consists of two major steps: transcription and translation. Together, transcription and translation are known as gene expression.

During the process of transcription, the information stored in a gene's DNA is transferred to a similar molecule called ribonucleic acid (RNA) in the cell's nucleus.

Both RNA and DNA are made up of a chain of nucleotide bases, but they have slightly different chemical properties. The type of RNA that contains the information for making a protein is called messenger RNA (mRNA) because it carries the information, or message, from the DNA out of the nucleus into the cytoplasm.

Translation is the second step in getting from a gene to a protein. In eukaryotes translation takes place in the cytoplasm. The mRNA interacts with a specialized complex called a ribosome, which reads the sequence of mRNA bases. Each sequence of three bases, called a codon, usually codes for one particular amino acid. (Amino acids are the building blocks of proteins.) A type of RNA called transfer RNA (tRNA) assembles the protein, one amino acid at a time. Protein assembly continues until the ribosome encounters a *stop* codon that is a sequence of three bases that does not code for an amino acid.

The flow of information from DNA to RNA to proteins is illustrated in Fig 1.3. This flow is one of the fundamental principles of molecular biology. It is so important that it is sometimes called the *central dogma* [2].

1.2 Motivation towards microarrays

Though most cells in our bodies contain the same genes, not all of the genes are used in each cell. Some genes are turned on, or *expressed* when needed. Many genes are used to specify features unique to each type of cell. Liver cells, for example, express genes for enzymes that detoxify poisons, while pancreas cells express genes for making insulin. To know how cells achieve such specialization, scientists need a way to identify which genes each type of cell expresses.

Before, scientist had to study genes in a single cell once at a time. Microarray technology now allows scientists to look at many genes simultaneously and to determine which are expressed in a particular cell type [47, 48]. DNA molecules

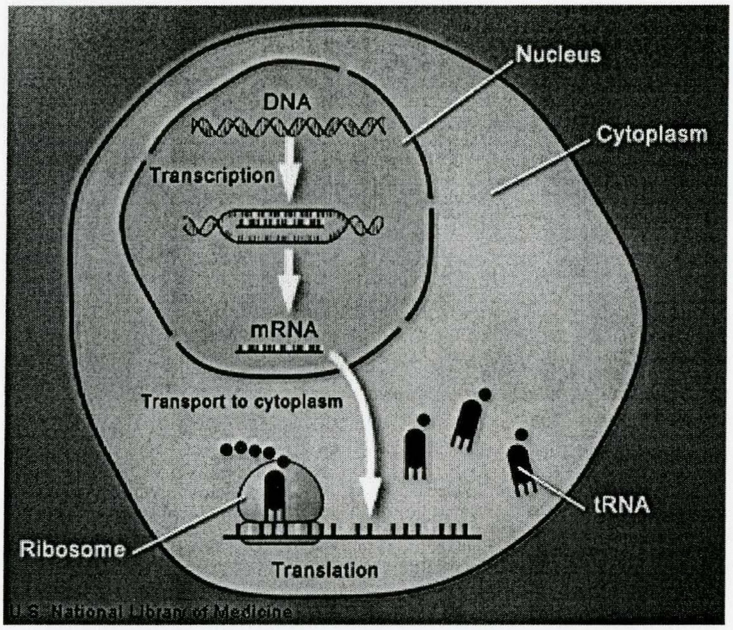


Figure 1.3: Production of protein in a cell [3].

representing many genes are placed in discrete spots on a microscope slide. This is called a microarray, like the one shown in Fig. 1.4. Thousands of individual genes can be spotted on a single square inch slide. Each gene is single stranded, amplified in number, and put on the slide to form a spot. Sample solution has to be prepared as well. Messenger RNA, the working copies of genes within cells and thus an indicator of which genes are being used in these cells, is purified from cells of a particular type. The RNA molecules are then *labelled* by attaching a fluorescent dye that allows us to detect them later, and added to the DNA dots on the microarray.

Due to a phenomenon termed base-pairing, RNA will stick to the gene it came from. This process is called hybridization and is shown in Fig. 1.5. After washing away all of the unstuck RNA, light is shone over the microarray and it is scanned by optical detector devices to get a fluorescent image. We can look at the microarray

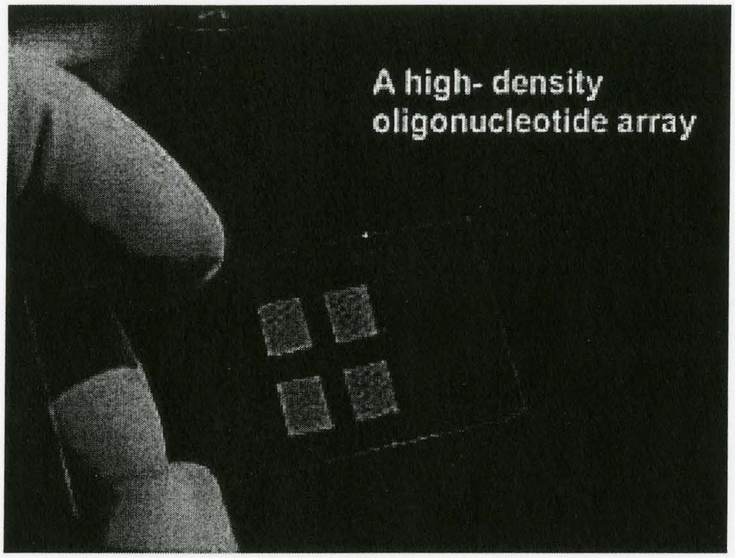


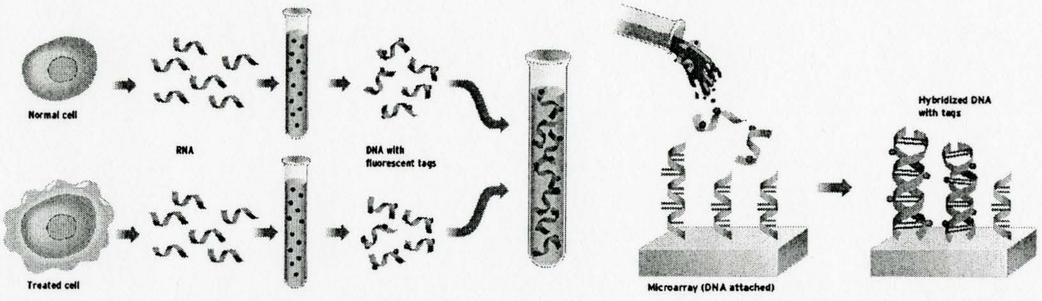
Figure 1.4: DNA microarray [4].

image and see which RNA remains stuck to the DNA spots. Since we know which gene each spot represents, and the RNA only sticks to the gene that encoded it, we can determine which genes are turned on in the cells. Some researchers are using this powerful technology to learn which genes are turned on or off in diseased versus healthy human tissues. The genes that are expressed differently in the two tissues may be involved in causing the disease [21].

1.3 Microarray technologies

There are many companies involved in making microarrays, each having its own manufacturing technology [20]. Depending on the manufacturer, microarrays can have different shapes, sizes, and be used for different applications. Most of the microarray manufacturing technologies fall into one of two categories: *printing*, and *in situ*.

Perhaps the most straightforward array-making method is by contact printing. A



(a) (b)

Figure 1.5: (a) Sample preparation and (b) hybridization in a microarray experiment [20].

pin is first dipped into a solution containing pieces of DNA of uniform sequence that have been synthesized in the lab. The pin is then pressed to the array surface leaving behind a droplet of solution. The pin can be rewetted after each deposition, or can have a small reservoir of fluid.

In an in situ fabrication, DNA sequences are made during the process of manufacturing the microarray and not before. Photolithography is used as an example of in situ fabrication. In photolithography, light at 365 nm is shone through a mask to illuminate a subset of regions on a substrate, which is coated with a photosensitive capping chemical. The light releases the capping chemical, exposing parts of the substrate. A solution containing a single type of nucleotide attached to a photosensitive chemical is then washed over the substrate. The nucleotides attach to the unprotected sites, adding their own capping layer. The process is repeated, building up sequences of DNA, as shown in Fig. 1.6.

Another in situ method is inkjet technology used in the Santa Clara plant. It is essentially the same as that found in a desktop printer. Jets of fluid are pressed through nozzles and broken into uniform droplets by the print head. In the in situ synthesis, the four colors of ink are replaced with the four nucleotides of DNA. This

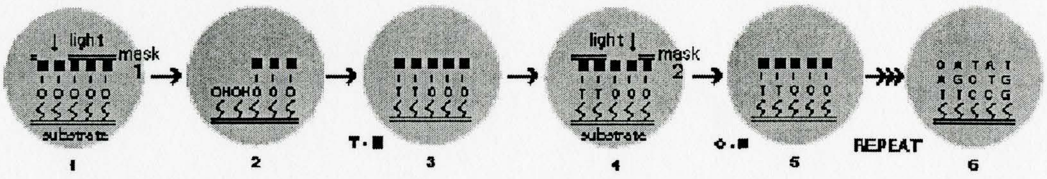


Figure 1.6: Photolithography method [6].

system can build lengths of DNA up to to 60 nucleotides long.

Some producers of microarrays, biochips, and lab-on-a-chip devices [18] are listed alphabetically below. For each company, a short description of the technology applied is described.

ACLARA Bio Sciences, Inc. (Mountain view, CA) - LabCard: Device uses electric fields to move fluids through capillaries on chips for the miniaturization and integration of complex, multi-step biochemistry processes [7, 8].

Affymetrix, Inc. (Santa Clara, CA) - GeneChip: High-density arrays produced by photolithographic process for gene expression and geno-typing [9, 10].

Caliper Technologies Corp. (Mountain view, CA) - LabChip: Micro-fluidic devices with active fluid control and parallel processing for variety of genome analysis procedures including molecular purification and high-speed DNA separations [11].

Clinical Micro Sensors (Pasadena, CA) - Low density array with electrochemical sensing for point-of-care diagnostic applications. This company was bought by Motorola later.

Hewlett Packard Company (Palo Alto, CA) - A thermal jet spotting system to produce cDNA microarrays.

Hysep, Inc. (Sunnyvale, CA) - Hyseq HyChip: A universal sequencing chip based on sequencing by hybridization technology [12, 13].

Illumina, Inc. (San Diego, CA) - BeadArray: Fiber-optic self-assembled addressable arrays are filled with optically encoded libraries of $3 - 5\mu m$ diameter beads to simultaneously process millions of assays [14].

Orchid BioSciences, Inc. (Princeton, NJ) - Developing micro-fluidic glass chips to enable high-throughput chemical synthesis, genomics, DNA analysis, screening, and diagnostics [15].

Packard Instrument Company (Meriden, CT) - BioChip Arrayer with four PiexoTip piezoelectric drop-on-demand tips that provide non-contact dispensing for arraying onto glass, filters, and HydroGel substrates.

Rosetta Inpharmatics, Inc. (Kirkland, WA)- FlexJet inkjet-based microarrays available only through scientific collaborations [16].

Sequnome (San Diego, CA) - High-throughput resequencing array that uses a MALDI-TOF mass spectrometer for subsequent detection and identification of DNA fragments, used for SNP and geno-typing [17].

Vysis, Inc. (Downers grove, IL) - GenoSensor: a genomic array system for addressing gene copy number.

1.4 Outline of thesis

This thesis is divided into four main chapters, plus two chapters for the introduction and the conclusions.

1. The first chapter introduces DNA microarrays. It gives a minimum required background about DNAs and genomics for this thesis. Then it discusses the motivation behind microarrays, and presents different technologies used for manufacturing them.

2. The second chapter is on the processing of DNA microarray images. It starts with an introduction of the basics of processing microarray images. Three basic steps of gridding, segmentation, and quantification are discussed. Then, more specific

aspects of microarray imaging are presented.

3. The third chapter is on modelling of spots in a microarray image. In that chapter, we discuss different aspects of mathematical modelling of spots, and how model parameters should be extracted from a microarray image.

4. The fourth chapter of this thesis is about lossless compression of microarray images. It concentrates on the method introduced by the author for lossless compression of microarray images, and it will describe in detail, different parts of the algorithm implemented for this purpose. Spot extraction, spiral path fitting, pixel prediction, and sequence coding are the four major parts of the proposed algorithm that are discussed. Results obtained from this method will also be presented and compared with the results of other compression schemes.

5. The fifth chapter is on lossy compression of microarray images. The method of the authors for lossy compressing microarray images will be introduced in this chapter. Four main parts of the algorithm are: spot extraction, model parameter extraction, C2S transform, and DCT, quantization, and encoding are presented. The results using the new lossy compression algorithm will be presented and compared with JPEG.

6. Chapter six presents the conclusions and recommendations for future research.

Chapter 2

Processing

2.1 Introduction

After a microarray experiment is performed, its results are embedded in a microarray image. Genetic information in a microarray experiment has to be extracted from the microarray image. Processing of a DNA microarray image is a critical step in a microarray experiment [22]. As briefly mentioned in previous chapter, every spot in a microarray image represents information about the abundance of the corresponding gene in the solutions of two test cells (Fig. 1.5). The relative abundance of the mRNAs in the two test samples is approximated with the relative intensity of spots in the red and green images. This number will be handed to the data mining step for further processing. So the task of the microarray image processing unit is to extract the intensity of each spot in the microarray image. A real microarray image consists of two components, red and green. Every one of these images will be processed independently as a monochrome image. As the image processing is the same for red and green images, from now on, we will assume that the images are monochrome.

Fig. 2.1 shows a microarray. It is compared in size with an American quarter. On the right, you can see the original microarray image obtained after laser scanning the microarray. The obtained image is usually huge in size. The microarray of Fig. 2.1 consists of 4 × 8 grids. Each grid is approximately made up of 24 × 26 spots. Now, assume that each spot is scanned into a square region of a minimum of 10 × 10 pixels. This means that our image consists of approximately 20,000 spots, and the corresponding color image has a size of at least 12 Mbytes. This is excluding the background and margin areas of the image. This image is handed over to the processing unit that is supposed to construct two sets of matrices for red and green colors. For every color, there should be a set of 4 × 8 matrices, each matrix with a size of 24 × 26. The entry of these matrices are preferred to be 16-bits numbers. So the output data will have a size of approximately 80 Kbytes.

Different kinds of noise and artifacts [23] can be seen in the microarray image of Fig. 2.1. There are black regions around the image, which means that some of the spots have been lost during the scanning. There are dust particles all around the image, which are seen as bright, irregular points around the image. In Fig. 2.1, there are regions with a high level of background illumination, for example on the right side of the image, and there are problems with the focusing on left side of the image. These are some of the factors that the image processor unit should consider during the process of extracting spot intensities of a microarray image.

This chapter starts with the basics of processing a microarray image: gridding, segmentation, and quantization. These fundamental steps will be explained in detail. Then, deviations from the ideal cases assumed in the fundamental steps will be introduced and solutions for them will be presented. These deviations are introduced in section 2.2 under the title of non-ideal effects.

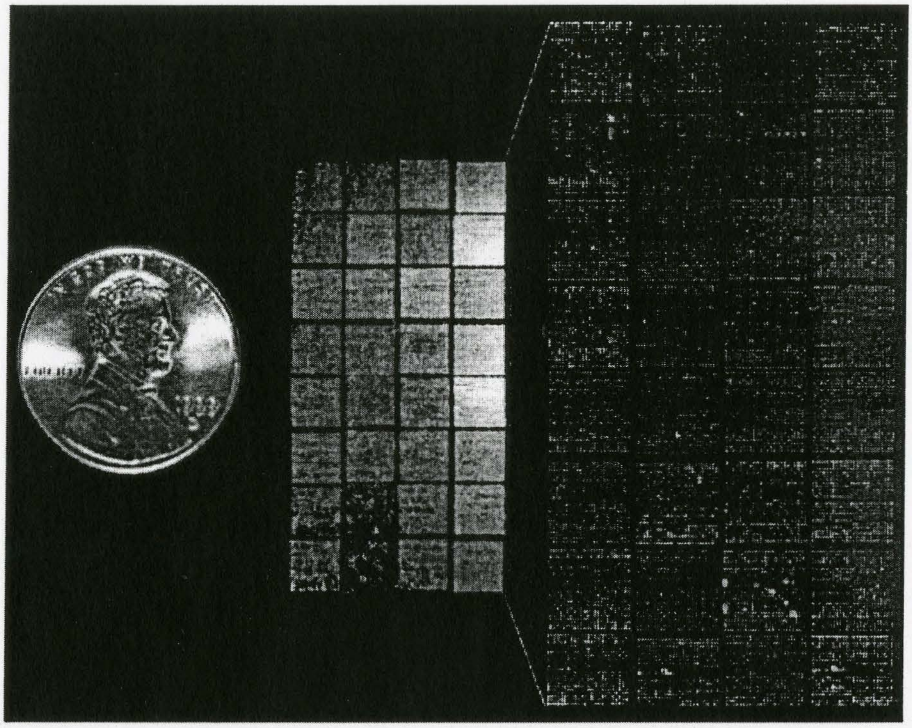


Figure 2.1: A microarray compared to a coin, and a two color microarray image for the corresponding microarray [19].

2.2 Basics of processing microarray images

There are some basic steps in the processing of a microarray image [24]. Every microarray, independent of its manufacturing technology, should be subject to three steps in its image processing [26]. The first step, gridding, is to assign coordinates to every element of the spot array. The second step, segmentation, is to classify a group of pixels as spot pixels. The third step, quantification, deals with measuring the intensity of the spot signal and the background. These steps are described in more detail in the following subsections.

2.2.1 Gridding

Gridding is to localize each spot in a microarray image. It is a geometric process that deals with the relative array spacing of the spots in an image. There are many approaches for localizing spots in a microarray image. The simplest method is to sweep a template over the image, and then to scan for the similarity of the template and the underlying part of the image. The template can be either one of the spots of the microarray itself, or the average of some of the spots throughout the microarray. A measure of the similarity can be the autocorrelation function, the mean square error, or the absolute value error. One application of this type of gridding is introduced in the third chapter of this thesis.

One widely used method for microarray gridding starts with integrating the microarray image along its axis. The integrals obtained will have maxima along the spot coordinates and minima along the background regions. These points can help in extracting the grid structure. This method is explained in more detail in chapter 4.

A method based on discrete Fourier transform (DFT) exists for gridding. The motivation for this method lies in the fact that spots are expected to be evenly spaced along rows and columns. However, they can vary from experiment to experiment. The DFT allows us to compute the spacings of the spots from an image. If we sum the pixel intensities along rows and columns, then we obtain two vectors x_v and x_h of real numbers. A visualization of these vectors show peaks which are spaced by some periods. In the frequency domain, this regular spacing should result in a local maximum M at the frequency f_M. The DFTs X_v and X_h may have many local maxima. However, M should be in a band-limited frequency domain $[f_{min}, f_{max}]$. The band limits are corresponding to two extreme cases of distribution of the spots. In the first case, the targets are spaced in a regular but maximum spacing. This case corresponds to a minimum frequency limit f_{min}. In the second case, spots are juxtaposed and a maximum frequency limit f_{max} is deduced. Once f_M has been determined, the period p_M can be computed. This period will be used to find the grid parameters of the image. In another similar approach, a two dimensional DFT can be used. The peak of the 2D DFT will correspond to the two dimensional period of the spots in the image, which can be used to localize them.

Another method is based on the Hough transform (HT) [29] that is an image processing technique originally used to detect lines and circles. However, the method has been generalized so that it can detect objects of arbitrary shapes of a reasonable size [28]. In our context, we are interested in the method to find circles, the circular Hough transform (CHT). The first step in the HT is to compute an intensity gradient image at all pixel locations. A gradient image is an image of the first intensity derivative of each pixel with its neighboring pixels. It is obtained by convolution of a small operator with the image and the goal is to detect the edges in an image. A large number of operators exist for this purpose, including those by Sobel, Roberts, Prewitt [30]. The gradient image is then thresholded to keep the significant edge points. In the second step, a parameter space is computed. In the case of the linear Hough transform LHT, for each edge pixel (x, y), all the lines going through this point in the (m, c) space with $y = mx + c$ are plotted. The polar (r, θ) space with $r = x \times cos(\theta) + y \times sin(\theta)$, where r is the length of a normal from the origin to this line and θ is the orientation of r with respect to the X-axis, is more often used. The highest accumulator points in (r, θ) space correspond to the strongest line edges in the image. In the circular Hough transform, for each edge point, all the possible center locations at a distance R are accumulated in a parameter space R where R is an anticipated radius for our circles. A pixel that has been accumulated a large number of times is most probably a spot center. Even though this method is the most precise one, it is computationally expensive and is not usually used in practice.

2.2.2 Segmentation

After applying the gridding step, we have each spot *approximately* localized. Approximately, because a spot is not a single pixel, or a perfect circle, to be exactly localized. Depending on the method we have used for griding, we either have a center for the spot, or a region (usually a square) in which the spot is located. It is yet to be decided where the actual location of the spot is, and which pixels are inside the spot, rather than in the background. To have a better idea about this problem, some sample spots from different microarrays are shown in Fig. 2.2.

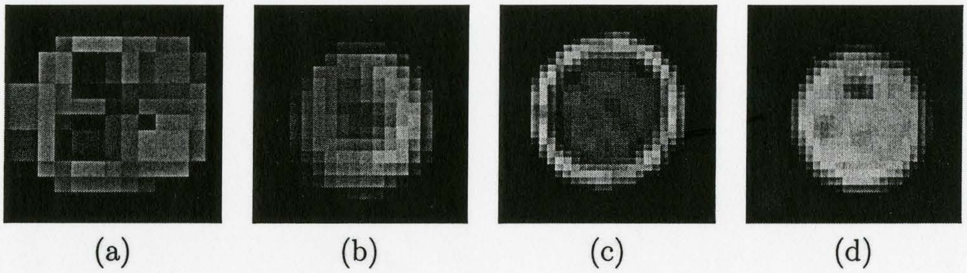


Figure 2.2: Spot samples from different microarrays. (a) Low resolution irregular spot. (b) Spot with smooth variation in boundaries. (c) Sharp varying spot with high intensity edges. (d) Sharp, high resolution, disk-shaped spot which is the closest to the ideal case.

Fig. 2.2 shows that spots can vary significantly from one microarray to another. They can vary in resolution, which is the number of pixels allocated to each spot. They can vary in shape, from a full disk in the ideal case to a number of connected pixels in some other cases. They can also vary in their relative size to the corresponding sub-image. The relative space each spot takes in a microarray sub-image is determined by how close spots are put on the microarray during the manufacturing process, and also the original volume of droplets which make each spot during the printing processes of manufacture.

The segmentation unit should be able to automatically determine the spot region in sub-images like the ones in Fig. 2.2. It should be able to perform independently of the variations in the manufacturing technologies. There are many methods to do this, like calculating the mean of the pixel values, edge detection, fitting circles, or parametric model fitting.

The simplest solution to segmentation is not to do segmentation. By assuming that the spot sub-image is presenting the spot, one can simply take the mean of the pixel values in the sub-image. This method is very preliminary and is not recommended any more.

A more realistic method is feature extraction, which is applying image processing methods to extract the feature (spot) in the sub-image. The most popular method in feature extraction is based on finding edges of the object in the image. For example, for the spot in fig 2.3a which has sharp edges, one can first calculate the gradient function for all pixels in the image, as shown in Fig 2.3b.

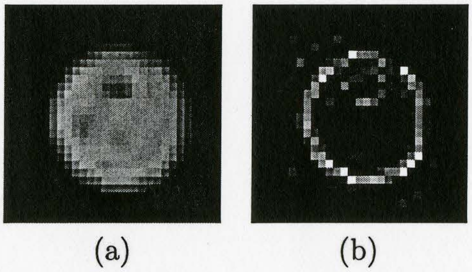


Figure 2.3: (a) A microarray spot and (b) its gradient image.

Then, a routine removes the isolated points in the gradient image to achieve a cleaner image which will approximately describe the borders of the microarray spot. Even though this method is well-known and commonly used for many applications, it is not considered appropriate for segmentation of microarray spots. Also, not all spots have distinguishable borders. An example is the spot shown in Fig. 2.2a. Also some spots vary very smoothly from the internal region to the background region, like the one in Fig 2.2b. In these cases the edge extraction usually fails.

The other commonly used method is to fit a circle to each spot, and use the pixels inside the matched circle as spot pixels in the next quantification step. This method is better than the previous two methods, and it is applied in this research work in chapter 5 of the thesis for lossy compression of microarray images. A detailed explanation of the circle fitting method is presented in section 5.2.2. The idea is as follows. First, find the initial values for spot center and radius using mathematical approaches. Then, optimize them for the best circle matching. In this case, matching is quantified by the difference between the average pixel values inside and outside the circle. A larger difference will result in a better matching.

Finally one can use the model fitting method for segmentation. Model fitting is simply a generalization of circle fitting, in which the spot is modelled by a circle. The modelling of spots is described in detail in chapter 3 of this thesis.

2.2.3 Quantification

Quantification is the process which deals with measuring the spot signal and background intensity values. Under idealized conditions, the relative total florescent intensities from two images of a spot is proportional to the corresponding gene expression [31]. These idealized conditions are now listed.

1- The probe DNA concentration in the solution is proportional to that of the cell.

2- The hybridization is done appropriately so that the amount of DNA binding on the spots is proportional to the DNA in the solution.

3- The amount of DNA deposited on each spot during fabrication is constant.

4- There is no contamination in the spots.

5- The signal pixels are correctly scanned and analyzed by software.

There are different methods for spot quantification, including total, mean, median, and volume. In the *total* method, it is the total signal intensity of the pixels in a spot that is assigned to each spot. The existence of contamination and variations in the size of spots make this method inaccurate for most applications.

The *mean* signal intensity is the average intensity of the spot pixels. This parameter has certain advantages over the *total*, such as making it independent of the spot size. The problem with this method lies in differences in the fabrication methods of microarrays, which make spots vary in shape and content. You can see that in the spot in Fig 2.2a, there are black regions inside the spot. It may make the quantification process more accurate if these pixels are not included in the averaging.

The *median* of the signal intensity is the intensity value that splits the sorted intensity of the signal pixels in halves. This method has problems similar to the ones for the *mean* approach.

The *volume* approach deals with the volume of the model which is matched to the spot. This method is applied in the approach proposed in this thesis and it is described in chapter 3. We strongly believe that this method is one of the best methods for quantification, considering the fact that the DNA abundance over each spot in reality has a 3D description.

2.3 Non-ideal effects

There are deviations from the ideal assumptions we have made so far for processing microarray images. In this section, these cases will be introduced, and solutions, when they exist, will be provided.

2.3.1 Uneven background illumination

During the scanning step of a microarray experiment, The microarray is exposed to light. In addition to the spots, glass substrate of the microarray can fluoresce. This is one of the major sources of noise in microarray images. Fluorescing is a property of glass and is a systematic error in microarray imaging systems. The main issue with this noise is that it can have a relatively high variation throughout the image. An example of a full microarray image is shown in Fig. 2.4a. You can see that the background noise on the right side of the image is higher than the same noise on the left side. This property makes the background noise different from other white noise sources in the image.

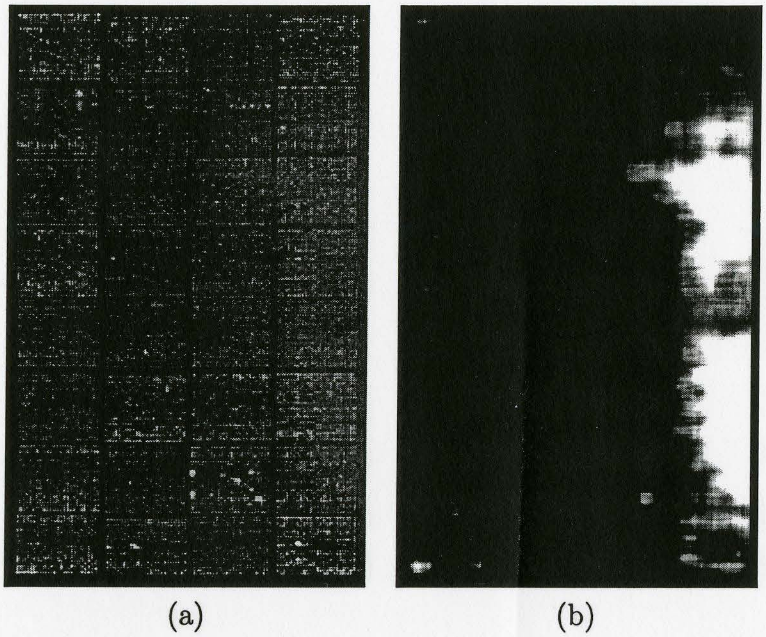


Figure 2.4: (a) A microarray with uneven background noise. (b) Background noise detected by our routine shown in the white regions.

There are well-known methods to eliminate uneven background noise. The fact is that this noise varies very slowly throughout the image. This means that other features of the image have a relatively higher frequency (in the spatial domain) than this noise. Therefore, one can use a high pass filter to eliminate this noise. The band-width should be carefully chosen to avoid loosing information about the spots, especially the region inside the spot which can have slow variations as well. Another similar method is to first low-pass filter the image and obtain an image similar to Fig. 2.4b. Then one can subtract this image from the original image to get rid of the background. As mentioned before, none of these methods can guarantee that there would be no loss in spot data.

Another method [27] uses a recursive approach to eliminate the background. The gridding provides us with the locations of spots. As a first approximation of the background, we can subtract the pixels belonging to the spots from the microarray image. Then, we use a hierarchical interpolation method based on Gaussian image pyramids to find a better estimate of the background. This method consists of two steps.

First step is performed after segmentation and before quantification. A pyramid of the microarray image $S[l]$, and a pyramid of the synthetic spot image $G[l]$ are built. Each level of the pyramid is made by transforming the image of the previous level into a lower resolution image, which represents the same data. One simple example can be replacing each block of four pixels by one pixel, so every level will have a size 1/4th of the lower level. At the level l_{max}, the resolution of the images is so low that the guide spot grid structure is no longer present in the images, meaning that the guide spots are merged. In this level, we subtract $G[l_{max}]$ from $S[l_{max}]$ which means that the spot intensities are approximately removed from the total image. A subsequent smoothing with a median filter is applied and this results in the background image $B[l_{max}]$. The level of the background is then decreased to get back to the original resolution. This can be done by an interpolation method. This process is illustrated in Fig. 2.5.

In the second step, knowing an estimate of the background, spots in G are calculated with a better approximation. Having an updated G, we repeat the calculations for a new B.

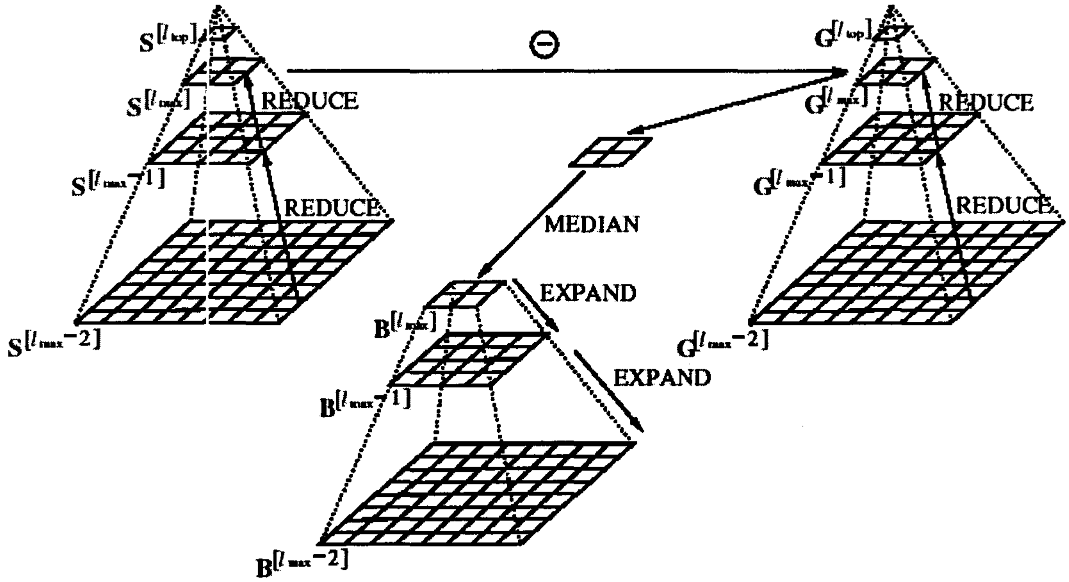


Figure 2.5: Background estimation using Gaussian pyramids. S is the pyramid based on the original image, G is based on a synthesized image which consists of the estimated spots. B is the difference between S and G which converges to the background noise image [27].

After two or three iterations, B will be a very accurate estimation of the background noise. The frequency range of the background noise achieved is determined by the median filter used in the method.

2.3.2 Dust and artifacts

Although microarrays are manufactured and used with a great amount of care, dust and other particles are always present on the microarray. These artifacts usually appear in the microarray image in the form of dots, or very small bright areas. Fig. 2.6 shows a microarray image with some of artifacts highlighted.

Removing these artifacts is a difficult task. Dust particles can have various shapes

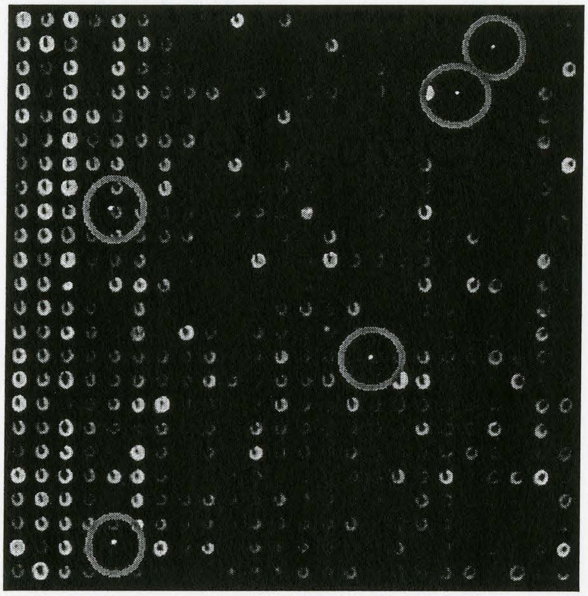


Figure 2.6: A microarray with its artifacts shown inside circles.

and their distribution is also completely random throughout the image. As these artifacts can be located inside a spot area, removing it requires special care in order to avoid data loss in the spot.

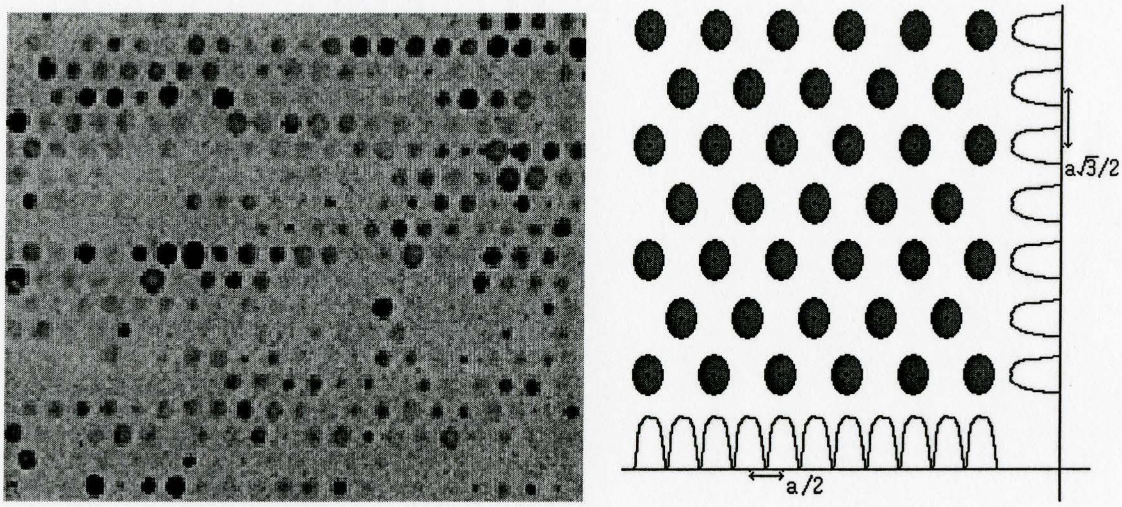
One straightforward approach for removing these artifacts is simply searching for small bright features in the image. There can be a threshold defined for the radius of spots in the image. Any feature with a smaller size can be considered as a dust. Since we may have dust particles of varying sizes, then this approach may not always work.

Another approach is to take into account the regular array structure of the spots, and then to consider that any feature that does not fit into that structure has to be an artifact. This is a good approach if there is a well defined model for spots that help us extract artifacts within the spots.

A solution similar to what we did for uneven background removal can also be considered. This time we are dealing with features that have a high frequency in the spatial domain. So a well defined low pass filter can help to remove them. Or we can have a high pass filter to extract the noise features, and then subtract it from the original image.

2.3.3 Hexagonal grids

Fig. 2.7a shows a microarray in which spots are located on a hexagonal grid. Some microarrays are made in this manner because of its better surface area efficiency.



(a) (b)

Figure 2.7: (a) A microarray image with hexagonal grid. (b) Computed integrals for a perfect hexagonal grid.

There can be automated routines for detecting this type of grid structure. In one approach, the image can be integrated along its rows and columns to obtain two sequences. The periods of these sequences can then be analyzed to detect the grid type. Fig. 2.7b shows a sample hexagonal array. Assuming that $T_{horizontal}$ and

$T_{vertical}$ are the periods of horizontal and vertical sequences, the type of the grid can be detected as follows:

$$GridType = \begin{cases} Hexagonal_{horizontal} & T_{horizontal}/T_{vertical} > 1.32 \\ Hexagonal_{vertical} & T_{horizontal}/T_{vertical} < 0.76 \\ Square & Otherwise \end{cases} \tag{2.1}$$

In the proposed formula, a judgement on the shape of the grid is made based on the relative periods of the integrated sequences. For the case of the perfect square array, $T_{horizontal}/T_{vertical}$ will be equal to 1. For the case of perfect horizontal hexagonal, it will be $\sqrt{3} = 1.73$, and for the case of perfect vertical hexagonal, it will be $1/\sqrt{3} = 0.58$. The ranges used in the mentioned formula are the average values of these numbers. The proposed formula is valid only for non-rotated grids.

2.3.4 Grid rotation

Microarrays can be very small in size. The process of manufacturing microarrays involves several steps, some of which may not be geometrically perfect. Therefore, it is highly probable, especially in older technologies, to have grids that are slightly rotated. Fig. 2.8 shows a microarray grid which is rotated relative to the axis of its scanner.

A rotation such as that shown in Fig. 2.8 is enough to make the basic gridding approach introduced before ineffective. As we calculate the projection of spots on the x and y axes, the rotated spots from different rows overlap and interfere with each other. This results in a projection in which the pattern of local minima cannot be seen anymore. Fig. 2.9 shows the projection of a rotated microarray on its y-axis

One of the methods to detect this rotation is to use a standard two dimensional FFT routine [24]. Any global rotation of the spots can be detected by looking at the maximum of the radially integrated spectrum given by

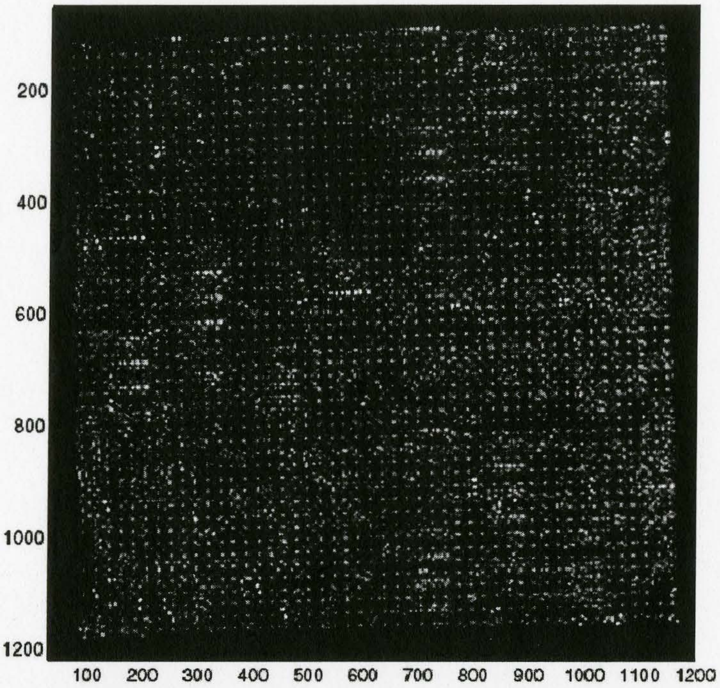


Figure 2.8: A rotated microarray image [27].

$$S(\theta) = \int |F(k, \theta)| \mathrm{d}k \tag{2.2}$$

where F is the Fourier transform of the image. This function of θ will be maximized when θ corresponds to the angle that the lines of spots make with the coordinates axes. This method suggests that the image be rotated by the value of angle obtained to retrieve the original image.

Another approach uses the discrete Radon transform [27]. This method does a spot amplification step before estimating the rotation angle of the image. It uses a matched filter (MF). A MF is a filter whose shape matches the shape of the signal one is trying to find. Having the matched filter M, the image S is amplified as follows: if $s[m,n]$ denotes an image patch around a pixel $S[m,n]$ in the image, then $s[m,n]$

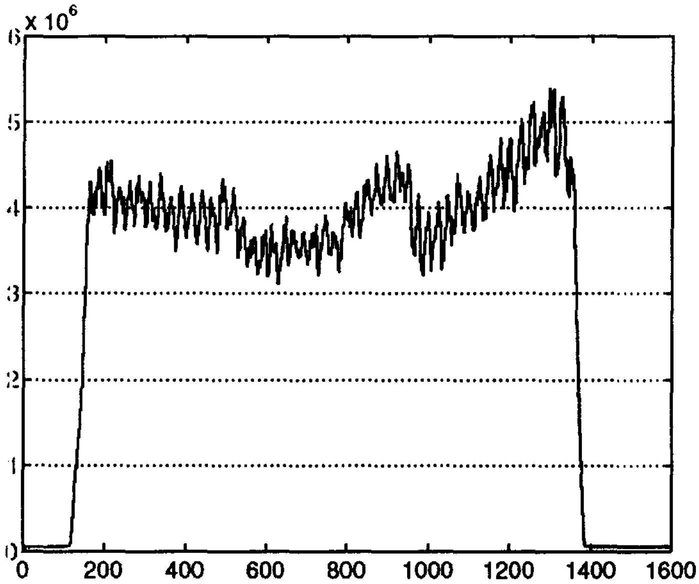


Figure 2.9: Projection of a rotated microarray image. Maximum and minimum patterns for spot localization can no longer be seen.

is first normalized to the local intensity mean, to obtain $e[m, n]$. Then the matched filter response value is equal to the dot product e and M:

$$R^M[m, n] = e[m, n].M \tag{2.3}$$

This corresponds to the similarity or statistical covariance between the image patch and the matched filter. R^M will be a zero mean image with features similar to the original microarray image. Then, the global rotation angle θ_g can be estimated with projections of the spot amplification response image R^M. Fig. 2.10 shows a projection $P(R^M, \theta_g)$ (discrete Radon transform) through the grid columns at the correct angle. The correct rotation angle corresponds to the index for which the maximum median value of the projection sequences is achieved.

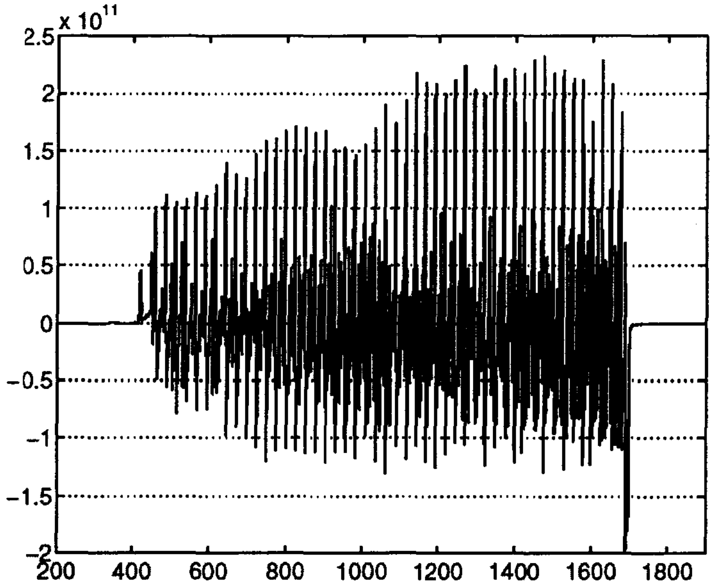


Figure 2.10: Projection of a rotated amplified microarray at the correct angle.

2.3.5 Other sources of noise

A DNA microarray experiment is a complicated multi-step process. Sources of noise in a microarray process are also varied and complicated [32]. From the many sources of noise affecting the final result of a microarray process, below are listed some important ones that are categorized by their source.

Test sample preparation: To start a microarray experiment, geneticists isolate two samples of mRNAs- a control (reference) and a test (experimental) sample. Samples are obtained from cells under different conditions. The mRNA samples are reverse transcribed into cDNA samples. The process of extraction can always have variations. There can also be systematic errors in the chemical processes involved. Therefore, uncertainties begin to show up from the first step of the experiment. However it should be mentioned that there are scientists performing this step with considerably high standards to minimize the uncertainties.

PCR amplification: cDNA samples obtained from cells are not enough in number to start the hybridization. They have to be amplified by orders of magnitude. Amplification is performed via a process called polymer chain reaction (PCR). Fig. 2.11 explains some of the details of PCR.

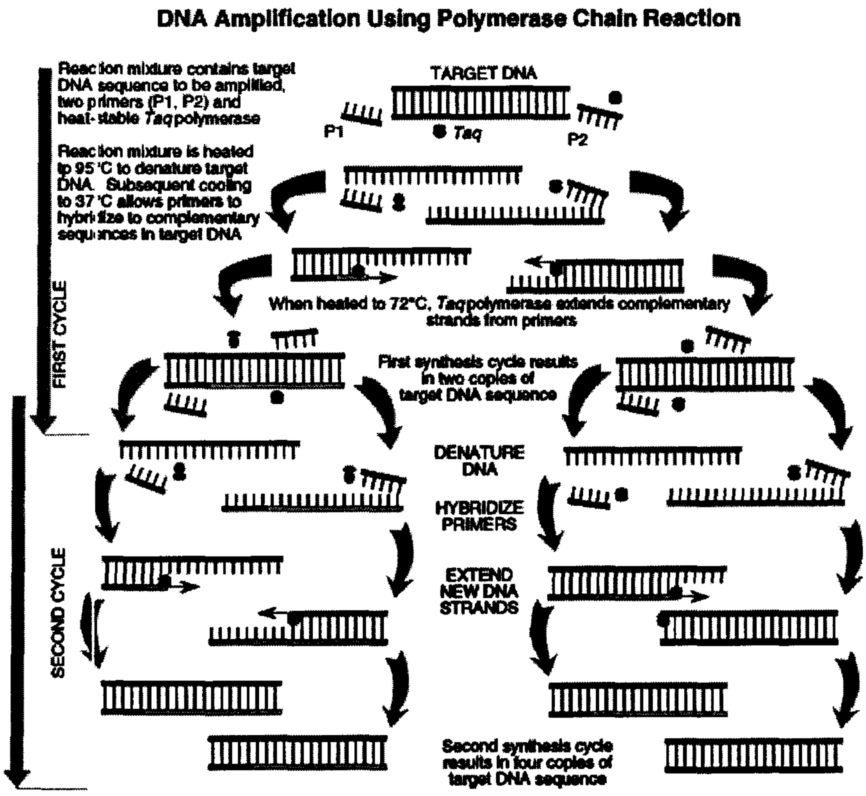

DNA Amplification Using Polymerase Chain Reaction
Reaction mixture contains target DNA sequence to be amplified, two primers (P1, P2) and heat-stable *Taq* polymerase
TARGET DNA
P1
Taq
P2
Reaction mixture is heated to 95 °C to denature target DNA. Subsequent cooling to 37 °C allows primers to hybridize to complementary sequences in target DNA
When heated to 72°C, *Taq* polymerase extends complementary strands from primers
FIRST CYCLE
First synthesis cycle results in two copies of target DNA sequence
DENATURE DNA
HYBRIDIZE PRIMERS
EXTEND NEW DNA STRANDS
SECOND CYCLE
Second synthesis cycle results in four copies of target DNA sequence

Figure 2.11: Polymer chain reaction [25].

There is a high amount of inconsistency in the gain of PCR. Variations in the gain of amplification of cDNAs will result in uncertain relative amplitude detection of the spot samples.

Fluorescent labelling: Samples from two cells are tagged in different colors. Cy-3 and Cy-5 are the typical fluorescent tags used. The tagging process is a chemical one that involves systematic errors and environmental uncertainties. There may be sequences that get labelled twice and some may not be labelled. Yet from a statistical point of view, the population of samples is high enough to have a consistent behavior.

Variations in pin geometry: Pins of an array spotter are made in tiny sizes. It is very difficult to keep strict measures in such tiny sizes. Also the shape of the head of the pin can vary and this contributes significantly to the shape of microarray spots.

Random fluctuations in target volume and shape: Even among spots which are put with the same pin, there are fluctuations in some physical aspects. The two most important ones are the volume and shape of the spot. The volume of the spot left by the pin on the slide depends on many factors and usually varies randomly around an average value. The number of cDNAs spotted on the array is directly proportional to the volume of the spot. On the other hand, the image processing routine of a microarray process has a limited tolerance for variations in the shape of the spot. Therefore, if the shape of the spot varies, then the intensity values read by the image processing unit may also vary.

Slide inhomogeneities: The slide over which a microarray is made is not perfectly smooth. It is not perfectly flat either. This can affect the shape and volume of spots in different regions. It can affect the amount of background noise in different regions of the microarray. It may also introduce focusing problems for the image grabber, as its lens is focused for a certain depth.

Hybridization parameters: Hybridization is a chemical process. Hybridization is performed for several hours in special chambers. The time spent for hybridization is just one of the factors involved in the process. The pressure in the chamber, the temperature and humidity, and the pH level are some of the deciding factors of the quality of hybridization inside a chamber.

Over-shining and saturation effects: There can be points on the microarray with a high density of fluorescent tags. These points can be unwanted jammed pieces of cDNA sequences, or simply highly hybridized and shiny spots. In either case, the photo detector can become saturated in the corresponding spots and non-linearity can be introduced into the image.

Non-linear transmission characteristics: Fluorescent transmission is not linearly proportional to the density of fluorescent tags. This can cause problems in processing of spots, considering the fact that red and green intensities are measured relatively for each spot. This effect can be reduced by careful studying of the non-linear behavior of spot illumination and cancelling it in software, similar to a channel equalizer.

Scanner noises: The image scanner consists of a photo detecting device like a CCD (charge coupled device) or a CMOS image detector. There are many types of noises which are introduced into the image during the photo detection. Some of them are: shot noise, smear, blooming, fixed pattern noise, dark current noise, reset noise, and internal amplifier noise. These are just some of many noises in a photo detector. It should be clarified here that there exists many excellent photo detectors with great performance with respect to the mentioned noises. However there is still room for custom made photo detectors suitable for this application.

Processing imperfections: In the final stage of a microarray experiment, the microarray image is handed over to the image processor. The processing unit has to extract spot intensities. Due to variations in spot shape and spacing, it is impossible for the processing unit to perfectly describe the spot. Therefore, there can always be errors introduced in the results just because of the imperfect modelling of the spots in the processing unit, or inaccurate gridding and quantification of the microarray image.

Chapter 3

Modelling

3.1 Introduction

The final goal in processing a microarray image is to assign every spot in the array a number representing its illumination level. There have been some routines that do so, all of them relying on some simplifying assumptions about the image. For example, spots are perfect circles, or they are perfectly aligned [24]. In these methods, noise effects are either neglected or assumed to be in very special forms, like substrate defects or dust particles [34, 35]. Moreover, in all existing methods spot densities are calculated regardless of the underlying technology and physical properties of the spots, for example by simply taking the mean in a square region around a spot. None of these methods is accurate or reliable.

In this chapter we propose a new and general method for detecting and assigning illumination levels to spots in a microarray image without making any assumptions about the geometry of the spots. We introduce a model for DNA spots and employ image recognition methods. Relying on some hidden convex behavior in the process of spot identification, we use convex optimization algorithms which lead to very fast and accurate results.

3.2 Method

The process of making microarrays involves wetting a printing pin with the solution containing many copies of a DNA sequence and ejecting the droplet onto a solid substrate [33] as shown in Fig. 3.1.

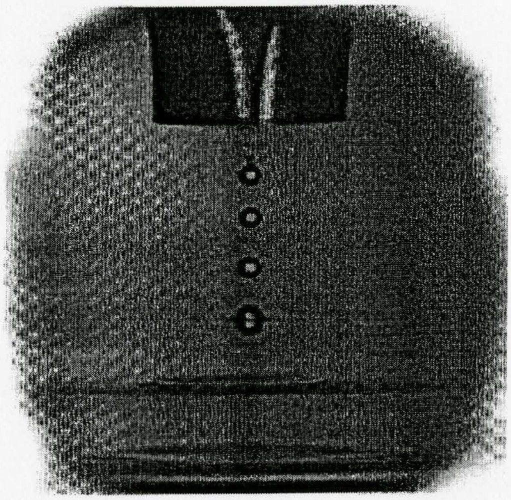


Figure 3.1: BioChip manufacturing process. A liquid droplet with volume of 300pL is ejected at 2m/sec velocity on a solid substrate [33].

We found that a spot made in such a way has a predictable shape which is a function of droplet volume velocity, and density. Trying many possible mathematical models, the best model that could simulate the shape of the spot is

$$f(r, r_{in}, t) = e^{-(r-r_{in})^2/t^2} + e^{-(r+r_{in})^2/t^2} \tag{3.1}$$

where r is the distance from the center of the spot, and r_{in} and t are shape parameters that define the diameter of the spot and the thickness of its lobe. Fig. 3.2 shows how we can have different spot shapes by adjusting these parameters.

In any microarray manufacturing process, the volume of the droplet is constant.

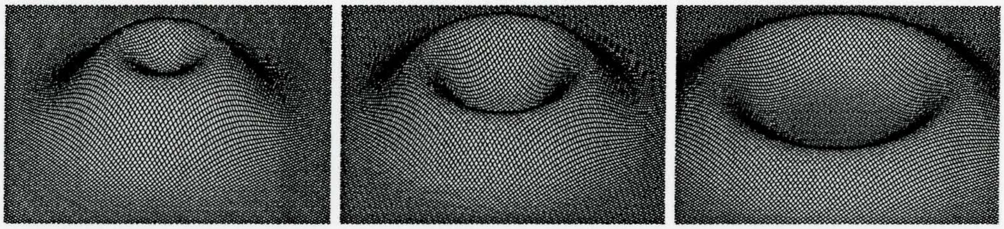


Figure 3.2: Volume normalized spot models with different shape variables.

So the model we propose should have a constant volume. For our model, the volume can be formulated as follows:

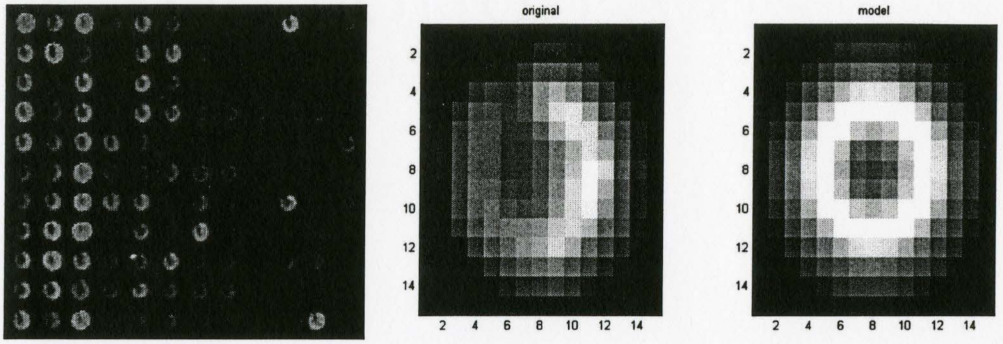
$$V(r_{in}, t) = \int_0^\infty (e^{-(r-r_{in})^2/t^2} + e^{-(r+r_{in})^2/t^2}) 2\pi r \mathrm{d}r. \tag{3.2}$$

So the normalized model based on r_{in} and t (Fig. 3.2) will be:

$$f_{norm}(r, r_{in}, t) = f(r, r_{in}, t)/V(r_{in}, t). \tag{3.3}$$

Fig. 3.3a shows a typical DNA microarray image after hybridization and laser scanning. What we should do with this image is to extract a matrix (in this example, a 11×12 matrix) whose entries represent the illumination levels of the corresponding spots. These numbers will also be directly referred to as the density of the corresponding spots' DNA sequences in the solution under test.

In order to extract this illumination level, we try to match every spot in the original image with the model spot we have proposed. This model has five variables: x and y which are the coordinates of the center of the model, r_{in} and t which are the shape parameters, and c which is the amplitude factor of the normalized model spot, and is the output of our image processor. To find the best matching, we minimize the least square image difference of our original spot and our model spot. Fig. 3.3b compares an original spot with its corresponding model spot.



(a) (b)

Figure 3.3: (a) A sample microarray image. (b) One of the spots and its corresponding model.

Our objective function to minimize should somehow represent the difference between our original spot and our model spot. We use the least square difference for this minimization and sample our smooth model to build a sum of differences of our sample values and our input image pixels.

$$F_{obj}(x, y, r_{in}, t, c) = \sum_{i,j}[cf_{norm}(\sqrt{(i-x)^2 + (j-y)^2}, r_{in}, t) - Im(i,j)]^2 \tag{3.4}$$

where $Im(i,j)$ is the gray scale intensity level of a pixel of the original image. As mentioned before, x and y are coordinates of the center of the model spot and the summation on i's and j's is performed in a region around x and y which contains the spot we are estimating.

From this formulation, it is obvious that x and y do not need to be integers and a spot can be coordinated with infinite precision. Considering the fact that laser scanning is usually done with low resolution, this is a valuable feature (Fig. 3.3b).

So our problem can be summarized as follows:

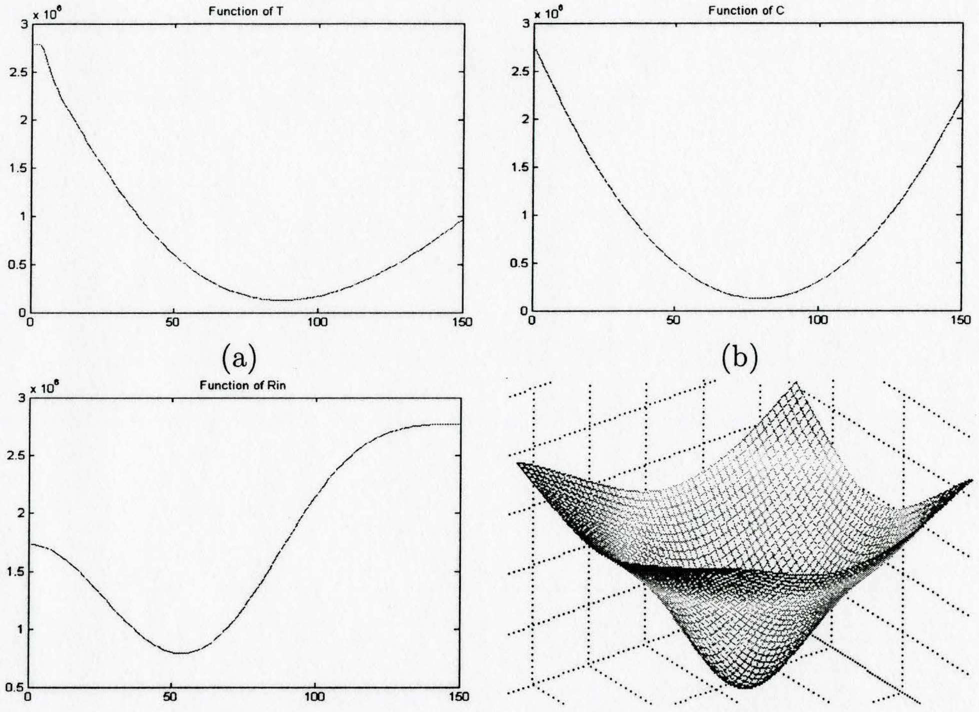
$$\begin{aligned} \text{Minimize } & F_{obj}(x, y, r_{in}, t, c) \\ \text{s.t.} \quad & 0 < x, y < \text{Image size} \\ & r_{in}, t, c > 0. \end{aligned}$$

For the case of a single spot image, there is only one local minimum for F_{obj} which is also a global minimum. On the other hand, for a multiple spot image, such as the one in Fig. 3.3a, we are looking for all local minima of F_{obj}. For example in this case there are 11x12 local minima. It is not practical to run a complete search on all the variables because the input image is usually large and the range of the other variables is usually image dependent. Examining the F_{obj}, we find an interesting convex behavior for it around its local minima (Fig. 3.4). Keeping r_{in}, t, and c constant, F_{obj} shows a quasi-convex behavior around its minima (Fig. 3.4d). The other variables also have the same property.

It is possible to apply convex optimization algorithms to solve this problem. To do so, good initial values for variables are needed that are close enough to the local minimum. For initializing coordinates, which is probably the most challenging one, we use a simple image recognition method. We extract one of the spots of image as a template and sweep it all over the original image. At any point in the process, we calculate the least square difference of the template and the underlying part of the image. As a result, we will have a 2D function which its local minima can be used to extract the initial values for the coordinates, as shown in Fig. 3.6.

Having initial values for the coordinates, then the initial values for c can be calculated by taking mean of the image pixels intensity levels in proximity of the initial coordinates. The initial values for r_{in} and t are less critical and can be assumed constant.

F_{obj} is a complicated function and its derivatives are not easily available. So the



(a) (b) (c) (d)

Figure 3.4: F_{obj} due to variation of (a) t, (b) c, (c) r_{in}, (d) x, y.

convex optimization algorithm chosen here has to be a derivative free one. We have chosen the pattern search method [37]. For this application eight search direction vectors are chosen which are:

$$\begin{aligned} D(x, y, r_{in}, t, c) = {} & (0, 1, 0, 0, 0), \\ & (\sqrt{3}/2, -0.5, 0, 0, 0), \\ & (-\sqrt{3}/2, -0.5, 0, 0, 0), \\ & (0, 0, 0, 1, 0), \\ & (0, 0, \sqrt{3}/2, -0.5, 0), \\ & (0, 0, -\sqrt{3}/2, -0.5, 0), \\ & (0, 0, 0, 0, 1), \\ & (0, 0, 0, 0, -1). \end{aligned}$$

Searching with even less number of direction vectors (down to 6) is possible, but it doesn't necessarily guarantee a faster approach. After any successful iteration, the step size is increased by a factor of 1.1 and if no direction decreases F_{obj}, then the step size is divided by 2. Searching is repeated until the step size is less than or equal to 0.003.

3.3 Results

First, we report the results of employing our pattern search method on single spot images. We compare our pattern search approach with a simple full search approach. The full search method simply checks all possible values in a discrete manner for the F_{obj}'s variables, and then extracts its minimum. This method is too slow for the full size image case. We initialize our pattern search method with the coordinates of the center of the image. We have tried 4 different spots with different illumination levels (Fig 3.5).

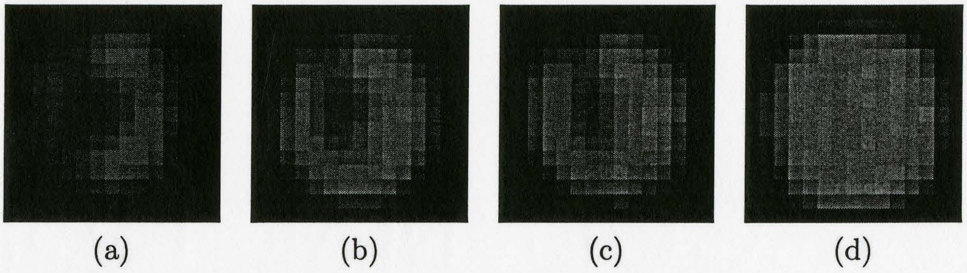


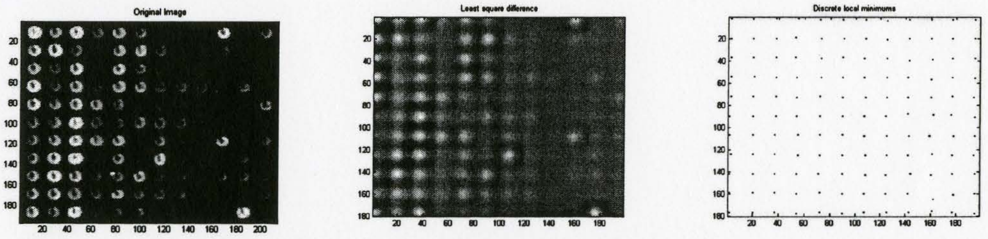
Figure 3.5: Sample single spot images.

In Table 3.1, which shows the results, C.S. is for Complete Search and P.S. represents Pattern Search. In the first five rows, we have the optimization variables and

Sample	(a)		(b)		(c)		(d)	
Search	C. S.	P. S.	C. S.	P. S.	C. S.	P. S.	C. S.	P. S.
c	110	110.0	167	166.4	199	199.2	325	324.9
x	8	7.73	8	8.36	8	8.19	8	8.27
y	8	7.93	8	7.68	8	8.22	8	8.28
r_{in}	32	32.85	34	34.38	33	32.16	33	32.89
t	300	291.58	274	265.39	279	284.80	339	338.53
F_{obj}	60K	57.0K	126K	99.1K	117K	105.9K	103K	70.7K

Table 3.1: Results for single spot images of Fig. 3.5.

their optimum values. The first row, c, shows the illumination levels of the spots. The optimum values for F_{obj} are given in the last row. Surprisingly, the results of C.S. are even worse (greater values for F_{obj}) because of the finite precision nature of the discrete search.



(a) (b) (c)

Figure 3.6: (a) Original image, (b) least square difference for a constant template, and (c) its local minima used as initial coordinates.

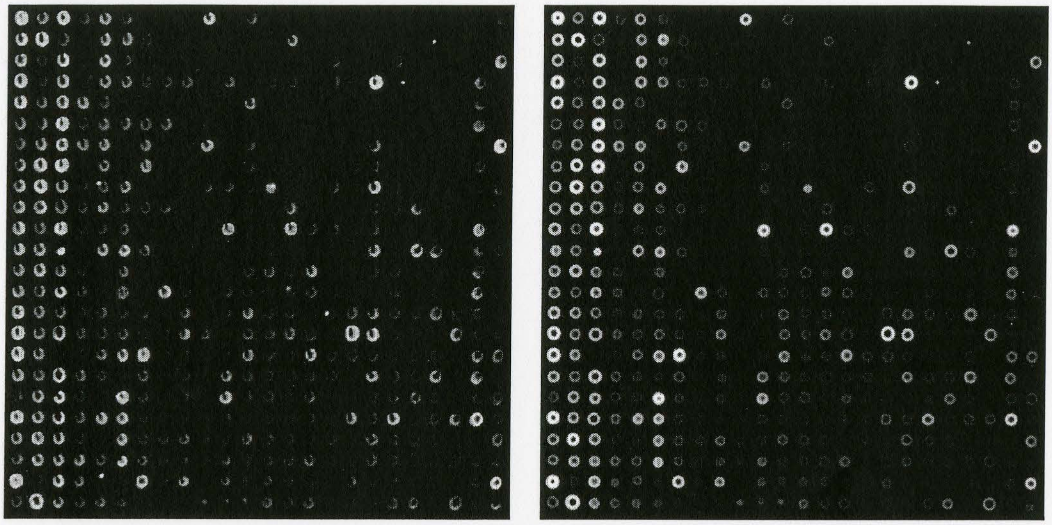
Next, we report the results of our approach for a full image with multiple spots. First, the initial values for the coordinates of the spots are extracted from the image as explained in section 3.2. Fig. 3.6c shows the initial coordinate values. We assign initial values for other variables as mentioned in section 3.2. Then we run the pattern search algorithm, once for every initial point we have. Converging to local minima in their proximity, the pattern search approach produces precise values for the optimum points x^*, y^*, r_{in}^*, c^*, and t^*. The optimum values for c form a matrix for all spots which will be the output of our process. Table 3.2 shows this matrix for the image of Fig. 3.3a.

318.81	134.83	289.54	89.54	145.12	73.19	33.95	12.76	14.98	186.82	25.09	69.99
199.94	260.76	97.57	23.02	141.46	191.19	70.16	23.93	10.40	52.67	29.49	17.38
166.96	64.93	207.54	18.17	165.91	63.76	41.69	25.23	14.27	7.15	16.49	15.99
294.77	74.74	179.57	48.32	177.66	120.11	60.66	66.55	43.55	8.46	62.53	24.90
225.60	105.99	201.14	138.92	78.57	32.92	44.71	21.66	19.42	16.62	29.04	72.56
91.77	104.99	296.24	84.66	49.47	98.42	85.15	56.37	10.31	12.91	27.54	31.38
92.68	138.67	253.93	162.38	145.78	25.36	81.12	25.72	21.87	119.96	38.63	50.85
106.64	231.55	318.84	37.42	100.00	36.41	198.34	32.94	28.11	15.22	51.60	28.91
129.26	280.42	166.47	51.42	94.93	143.40	46.08	35.89	10.23	34.19	50.82	37.29
127.71	163.93	151.28	61.00	114.16	61.62	61.53	46.03	29.02	4.48	12.19	32.01
191.11	105.15	277.82	47.14	54.14	57.54	36.68	23.20	18.27	7.56	236.43	36.28

Table 3.2: Results for original image of Fig. 3.3a.

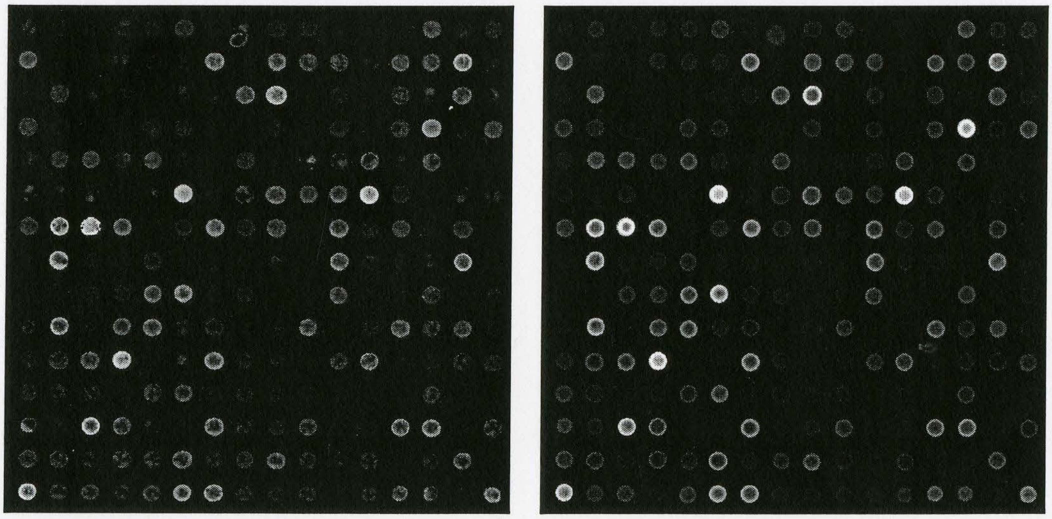
The values in table 3.2 were rounded because of lack of space. Another interesting feature of this method is its very high sensitivity. For the image of Fig. 3.3a, for example, we have a spot with illumination level of 319 and another spot with illumination level of 4. This property is a result of our model and its noise cancelling properties. As in Fig 3.2, the exponential decrease in the amplitude of our model cancels noise effects in the outer regions. However, in ordinary methods where the illumination level is extracted by taking mean in a square region containing the spot, the noise effects are accumulated constantly.

Figures 3.7, 3.8, and 3.9 show that our model is very good in describing different spots in different microarrays. They illustrate microarray images made with spots that are defined with our proposed models, and compare them to the original images.



(a) (b)

Figure 3.7: (a) Original image and (b) model image.



(a) (b)

Figure 3.8: (a) Original image and (b) model image.

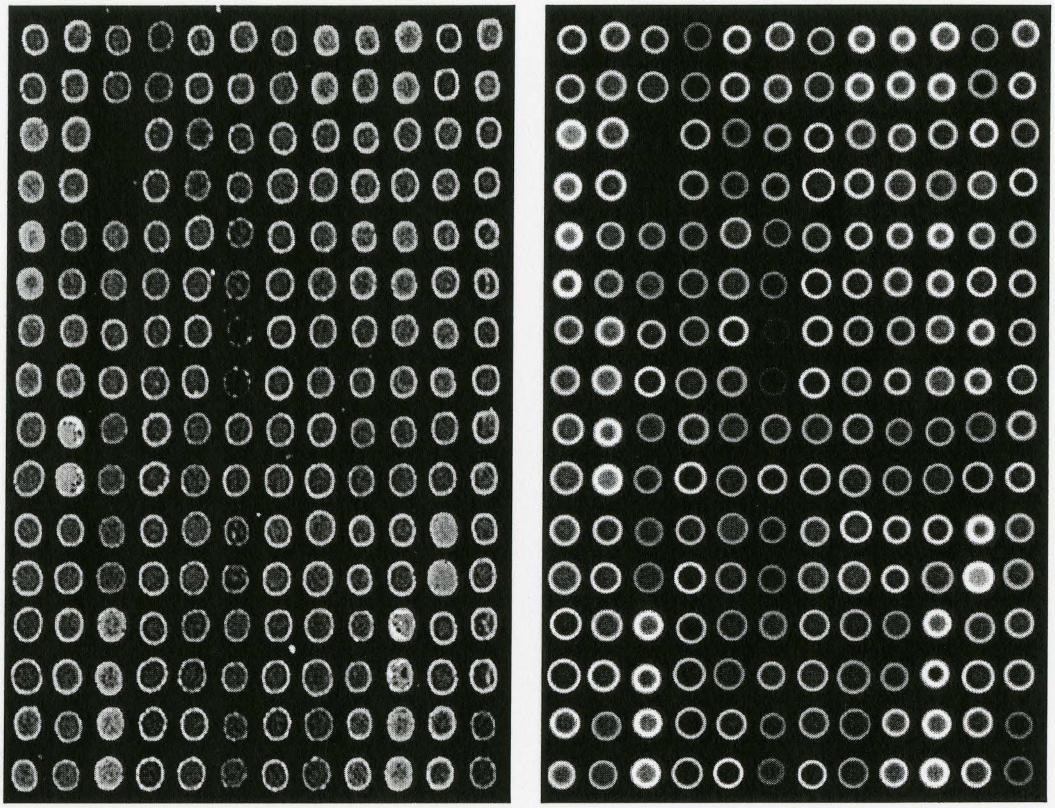


Figure 3.9: (a) Original image and (b) model image.

Chapter 4

Lossless Compression

4.1 Introduction

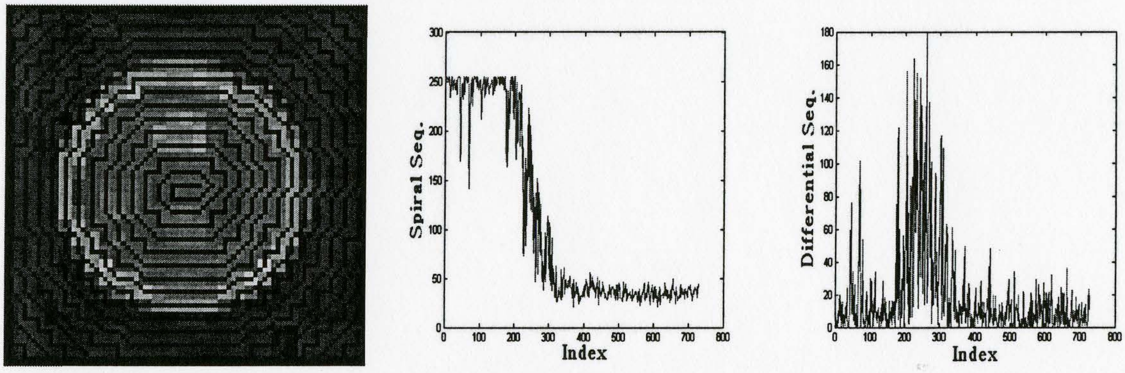
Microarray images are usually massive in size. Assume that a microarray that consists of 20,000 spots. Now assume that each spot has a size of at least 8 × 8 pixels, which means a spot sub-image of about 12 × 12 pixels. Therefore, color image with 16-bits per color will have a size of at least 17.3MBytes. If we add the size of background and margins, then we will have an image of 20MBytes. As various organizations are establishing databases for sharing microarray images [20], image data compression seems to be essential.

In this chapter, we propose a new technique for lossless compression of DNA microarray images. We first introduce the spiral path that is a scanning technique designed for this application. Then, we explain the algorithm we have implemented for lossless compression of microarray images. A detailed overview of the steps involved in the algorithm is then described. Some mathematical concepts including interpolation, entropy analysis, and convex optimization methods are applied in the algorithm. Finally the results are compared with other conventional compression schemes and with the work of other groups in this area.

4.2 Spiral path method

Spiral path is a new scanning method. It can be used for spatial scanning of any image, but it is more useful for disk-shaped or ring-shaped images. The idea is to convert a 2D image into a highly correlated 1D sequence to be coded subsequently. If we scan the image of a circle in the conventional raster scheme, we enter and exit the circle several times during the process. One way to avoid this is to extract the boundary of the circle and scan along it [36]. Another approach used in our spiral path method is to start scanning at the center of the circle, scan the entire area inside it, get out of it once and scan the remaining area outside the circle.

Fig. 4.1a shows a typical spiral path superimposed over a microarray spot image. Fig. 4.1b shows the pixel values if we scan the image along the spiral path. The spot area and the background area can easily be distinguished in this sequence. The sequence is highly compressible due to small changes in its consecutive values.



(a) (b) (c)

Figure 4.1: (a) Spiral path superimposed on a spot; (b) spiral sequence and (c) its differential sequence.

A continuous spiral (with polar representation of $r = k\theta$) does not match the Cartesian discrete system. Finding an image scanning method which has a spiral shape and preserves the spatial continuity so that it covers all image pixels is a rather big challenge. Our method for building a spiral path is as follows. First, a center for the path, with coordinates x_{Center} and y_{Center}, is chosen. Our method for building a spiral path can accept real values for its center coordinates, which provides great flexibility to the shape of the spiral as can be seen in the difference between images in Figures 4.1a and 4.2a. This fact will also be helpful later during the optimization of the center of the spiral path. After choosing the center, a matrix P is formed, which consists of rounded values of the Euclidian distances of each of the image pixels to the center. Table 4.1 shows the matrix P for a 18 × 19 pixels image. The values of elements of P are calculated using

$$P[i,j] = Round(\sqrt{(i - x_{Center})^2 + (j - y_{Center})^2}). \tag{4.1}$$

The procedure of building a path starts with initializing a distance parameter, d, to 0. In each step, pixels with coordinates (i, j) for which $P[i,j] = d$ are scanned in a counterclockwise direction starting at the right most pixel. After a full circle is made, d is increased by 1. This will continue until the entire image area is covered. As seen in Fig. 4.1a, the spiral path will inevitably break near the borders of the image. Since this always happens in the background area, it will not affect the quality of compression of microarray images.

As illustrated in Fig. 4.2, if the center of a spiral path is not chosen carefully, when the path reaches the boundary of the spot, it swings several times inside and outside of the spot until it gets completely outside of the spot. This phenomenon, that we will call *edge effect*, drastically decreases the compressibility of a spot image. In section 4.2.2, we will explain our method for extracting the center of a spot to minimize the edge effect.

Fig. 4.3 shows the flow-chart of the algorithm proposed in this thesis for lossless compression of microarray images. Four main steps are involved in this algorithm.

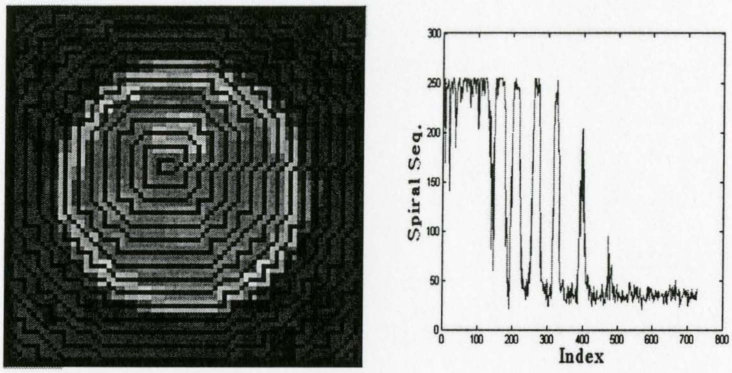
Table 4.1: Matrix P calculated in (4.1) for an 18 × 19 *image.*

13	12	11	11	10	10	9	9	9	9	9	9	9	10	10	11	11	12	13
12	11	11	10	9	9	9	8	8	8	8	8	9	9	9	10	11	11	12
11	11	10	9	9	8	8	7	7	7	7	7	8	8	9	9	10	11	11
11	10	9	8	8	7	7	6	6	6	6	6	7	7	8	8	9	10	11
10	9	9	8	7	6	6	5	5	5	5	5	6	6	7	8	9	9	10
10	9	8	7	6	6	5	4	4	4	4	4	5	6	6	7	8	9	10
9	9	8	7	6	5	4	4	3	3	3	4	4	5	6	7	8	9	9
9	8	7	6	5	4	4	3	2	2	2	3	4	4	5	6	7	8	9
9	8	7	6	5	4	3	2	1	1	1	2	3	4	5	6	7	8	9
9	8	7	6	5	4	3	2	1	0	1	2	3	4	5	6	7	8	9
9	8	7	6	5	4	3	2	1	1	1	2	3	4	5	6	7	8	9
9	8	7	6	5	4	4	3	2	2	2	3	4	4	5	6	7	8	9
9	9	8	7	6	5	4	4	3	3	3	4	4	5	6	7	8	9	9
10	9	8	7	6	6	5	4	4	4	4	4	5	6	6	7	8	9	10
10	9	9	8	7	6	6	5	5	5	5	5	6	6	7	8	9	9	10
11	10	9	8	8	7	7	6	6	6	6	6	7	7	8	8	9	10	11
11	11	10	9	9	8	8	7	7	7	7	7	8	8	9	9	10	11	11
12	11	11	10	9	9	9	8	8	8	8	8	9	9	9	10	11	11	12

First, spots in the image are located and extracted in rectangular regions which we will call *spot sub-image.* Second, a spiral path is matched to each individual spot sub-image and the sub-image is scanned into a 1D sequence of pixel values. Third, each sequence is differentially encoded using prediction techniques developed for this application. The second and third steps cooperatively minimize the entropy of residual sequence obtained from differential coding. Finally, all residual sequences from all spot sub-images are concatenated after being divided into spot and background parts. The spot and background parts are coded independently using a variable length coding scheme. In the following, we explain each of these four steps in more detail.

4.2.1 Spot extraction

In our proposed method for lossless DNA microarray image compression, spots in the microarray image should be processed individually. Therefore, the first step in our method is to localize and extract microarray spots. Gridding is described in detail in chapter 2 of this thesis. We use the same idea in here. The array arrangement of



(a) (b)

Figure 4.2: (a) Edge effect for a poorly matched spiral path. (b) The swings in intensities along the path are easily seen.

spots in a microarray image can help us to do the gridding. The microarray image is integrated along its rows and columns using

$$Int_x[i] = \sum_{j=1}^{n} Im[i,j]$$
$$Int_y[j] = \sum_{i=1}^{m} Im[i,j] \qquad (4.2)$$

where $Im[i,j]$ is the image pixel value. Figures 4.4a and 4.4b show the corresponding integrals for the microarray image of Fig. 4.4c. Discrete Fourier transforms of Int_x and Int_y are then calculated. Spatial frequencies corresponding to the non-dc peaks of these DFT sequences give us estimates for periods of Int_x and Int_y sequences. Int_x and Int_y are then divided into intervals of corresponding periods and minimum points are found in each interval. These local minima will then form two vectors which will become the coordinates of rectangular regions in which the spots are located. It is worth mentioning that as coordinate calculations are based on integrated pixel values, then the effects of noisy spots and other artifacts like dust, will be filtered. Fig. 4.4c

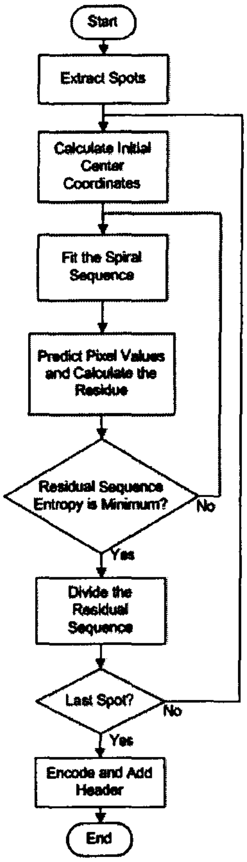
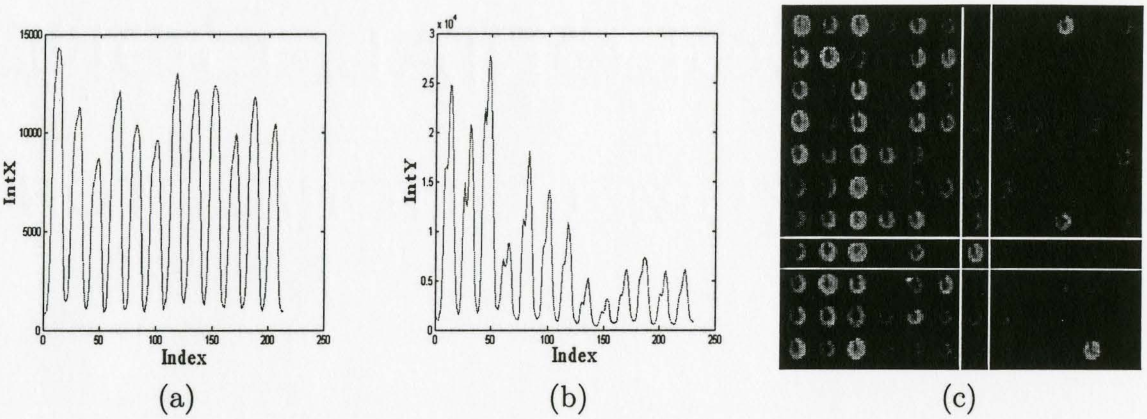

Start
Extract Spots
Calculate Initial Center Coordinates
Fit the Spiral Sequence
Predict Pixel Values and Calculate the Residue
Residual Sequence Entropy is Minimum?
No
Yes
Divide the Residual Sequence
Last Spot?
No
Yes
Encode and Add Header
End

Figure 4.3: Flow-chart of our algorithm for lossless DNA microarray image compression.

shows a typical extracted spot sub-image.

4.2.2 Spiral path fitting

The property that spots are more or less circular is what makes the spiral path scanning method suitable for microarray image compression. In fact, this is the property which makes microarray images more compressible compared to ordinary images. The spiral path we consider is uniquely defined by its center coordinates.



(a) (b) (c)

Figure 4.4: (a) Int_x and (b) Int_y calculated for the microarray image shown in (c). White lines show how spot sub-images are extracted.

The center of the path should be carefully chosen in order to avoid edge effect.

Our method locates the center of a spiral path in a spot in two steps. First, the initial coordinates are calculated using the geometric properties of the image. Second, the coordinates are locally tuned to optimize the path location. In the following, we explain these two steps in detail.

The initial coordinates are found by integrating the spot sub-image along its rows and columns and calculating their mean values using

$$
\begin{aligned}
Center_x &= \frac{\sum_{i=1}^{m_{Sub}} SubInt_x[i]i}{\sum_{i=1}^{m_{Sub}} SubInt_x[i]} \\
Center_y &= \frac{\sum_{i=1}^{n_{Sub}} SubInt_y[i]i}{\sum_{i=1}^{n_{Sub}} SubInt_y[i]}
\end{aligned}
\tag{4.3}
$$

where m_{Sub} and n_{Sub} are the size of extracted spot sub-image. $SubInt_x$ and $SubInt_y$ are calculated for the sub-image in the same way that Int_x and Int_y were calculated in (4.2). The initial coordinates found in (4.3) are usually close to the optimum ones. We are searching for coordinates as close to the geometric center of the spot as possible, in order to form a highly correlated sequence of pixel values. So an optimization is performed on the the center of the spiral to minimize the entropy of residual sequence obtained after the prediction coding. We have employed the 2D pattern search method [37] to perform this real-valued optimization. As will be explained in section 4.2.4, the residual sequences of spots will be concatenated together. Minimizing the entropy of each of these sequences individually will not necessarily minimize the entropy of the concatenated sequence. Nevertheless, the concatenated sequence will be highly compressible because of the fact that spots in a microarray image have similar statistical behavior.

4.2.3 Pixel prediction and differential coding

A special form of prediction is developed for our compression method. We start by moving along the spiral path found in the previous step. A prediction for a pixel value is made based on the previous pixel value(s) which is (are) available to both encoder and decoder. Differences between the predicted values and the pixel values form a residual sequence which will be subsequently variable length coded. Better prediction results in residues with smaller values, which are more compressible.

The simplest prediction for a pixel's value is the value of the previous pixel on the spiral path. Fig. 4.1c shows the residual sequence of this simple prediction scheme for the sequence of Fig. 4.1b. Although slow variation of pixel values on the spiral path results in small differences in adjacent pixel values along the path, higher compression ratios can be achieved by performing a more advanced prediction. In fact, without the prediction scheme we will introduce in the next paragraph, spiral path scanning by itself cannot outperform competing compression methods.

When we are coding a pixel's value, all the pixels' values on the spiral path up to that pixel are already coded. This means that we can use more neighbors to predict the value of pixel being coded. Fig. 4.5 shows two typical pixels on the spiral path with their neighbors, with spatial distances up to 2 that have already been

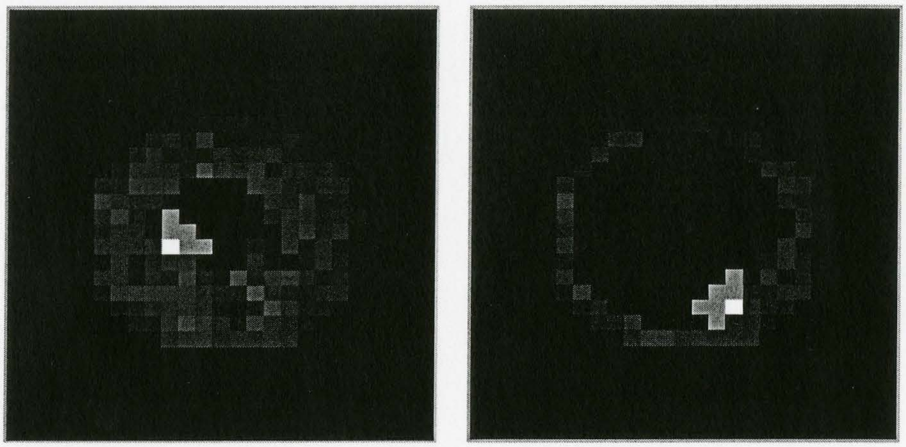


Figure 4.5: White pixels are neighbors with a spatial distance up to 2 inside the spiral for two points on different locations along the path. Black pixels inside the spot area are pixels along the spiral path that have already been coded.

coded. Unlike ordinary prediction coding methods, the number and arrangement of the neighbors used in the prediction of a pixel's value changes depending on the position of the pixel on the spiral path. An effective prediction method should also take into account the radial behavior of a spot. We expect pixels with the same distance from the center to have similar pixel values. Also if in a neighborhood, the values of pixels are decreasing (or increasing) with the distance from the center, then the predictor should be able to capture this trend and generate a suitable prediction. Therefore, we approximate the behavior of pixels in a small neighborhood to be linear with respect to their distances from the center. We gather previously encoded neighbors with spatial distances up to 2 from pixel being coded. We form (y_i, r_i) pairs for these neighbors, y_is being their pixel values and r_is being their real-valued Euclidian distances from the center of spiral. Then we find their linear interpolation function:

$$\hat{y} = y_0 + ar_0 + \beta \tag{4.4}$$

and use $\hat{y}$ to predict the intensity of our pixel based on r_0, its distance to the center. Equation (4.4) is a linear interpolation formulation and the parameters in (4.4) can be calculated from

$$y_0 = \frac{\begin{vmatrix} \sum_{i=1}^{n_N} y_i & \sum_{i=1}^{n_N} r_i \\ \sum_{i=1}^{n_N} y_i r_i & \sum_{i=1}^{n_N} r_i^2 \end{vmatrix}}{\begin{vmatrix} n_N & \sum_{i=1}^{n_N} r_i \\ \sum_{i=1}^{n_N} r_i & \sum_{i=1}^{n_N} r_i^2 \end{vmatrix}} \quad a = \frac{\begin{vmatrix} n_N & \sum_{i=1}^{n_N} y_i \\ \sum_{i=1}^{n_N} r_i & \sum_{i=1}^{n_N} y_i r_i \end{vmatrix}}{\begin{vmatrix} n_N & \sum_{i=1}^{n_N} r_i \\ \sum_{i=1}^{n_N} r_i & \sum_{i=1}^{n_N} r_i^2 \end{vmatrix}} \tag{4.5}$$

where $|..|$ is the determinant, n_N is the number of $(y_i,\ r_i)$ pairs, and β is an experimentally determined offset, equal to -0.4. Pairs corresponding to neighbors with spatial distance of 1 are duplicated prior to calculating (4.4) and (4.5). Numerical experiment shows that putting emphasis on immediate neighbors in this way improves final results.

4.2.4 Sequence coding

After performing the above explained steps we will have a residual sequence with the length $m_{Sub}n_{Sub} - 1$ (which we will call L) for a $m_{Sub} \times n_{Sub}$ spot sub-image. The pixel value of the center of the spiral path will be sent separately in the header part of the compressed file. Fig. 4.6a shows a typical spiral path sequence and Fig. 4.6b shows its predictive residual sequence.

It can bee seen in Fig. 4.6b that the residual sequence can be divided into two parts. Inside or spot part, and outside or background part. The statistical characteristics of the spot and background parts are very different. For example, the spot part usually has a larger mean and an approximately uniform distribution. On the other hand, the background part has a smaller mean and a more skewed distribution. By coding these parts separately, we will improve the performance of our compression method. Fig 4.6c and 4.6d show extracted spot and background parts of the sequence in Fig. 4.6b. To divide the residual sequence into spot and background parts, the difference in the distribution of two regions is used. The expected cumulative length of the coded sequences, F, is minimized based on where the residual sequence might be cut:

$$F(i) = Ent(RList[1..i])i + Ent(RList[i+1..L])(L-i) \tag{4.6}$$

where $RList$ is the residual sequence, and $Ent()$ is the entropy function of a given sequence. We perform a semi-Newton optimization on $F(i)$ to find i^* which will be where the sequence should be divided. The minimization algorithm is initialized with:

$$i_{initial} = Round(\alpha\sqrt{\frac{\sum_{i=1}^{L} List[i]i^2}{\sum_{i=1}^{L} List[i]}}) \tag{4.7}$$

where $List$ is the 1D sequence of pixel values along the spiral path. α is a correction factor obtained by experiment, and it is 1.3 in our implementation.

The spot and background parts of all spot sub-images of the microarray image are concatenated to form two sequences. These sequences are then variable length coded independently using adaptive Huffman coding. The coded sequences, added with a header including information about spot coordinates, spiral path centers, spot/background division points in the spiral sequence, and intensity values of center pixels of spots form the lossless compressed file.

4.3 Results

Here, we show the results obtained for compression of microarray images with our algorithm. Table 3.2 shows the results obtained for compression of two cDNA microarray images. It shows the sizes of different parts of the compressed bitstreams obtained using our method and how they compare to the original sizes. It also shows that our algorithm works independent of the image size or its precision, as the first row in the table is for an 8bit microarray image with an original size of 187K and the second row corresponds to a 16bit image with an original size of 28.7M. The size of the header is proportional to the number of spots in the images and it is usually less than 2% of the compressed bit stream.

Table 4.2: Size of components of two compressed files in bytes.

Original	Header	Spot reg.		Background reg.		Comp-ressed
		Original	Coded	Original	Coded	
187,702	1,440	59,462	42,798	126,922	44,056	88,294
28.7M	252K	9.3M	6.4M	19.4M	6.9M	13.5M

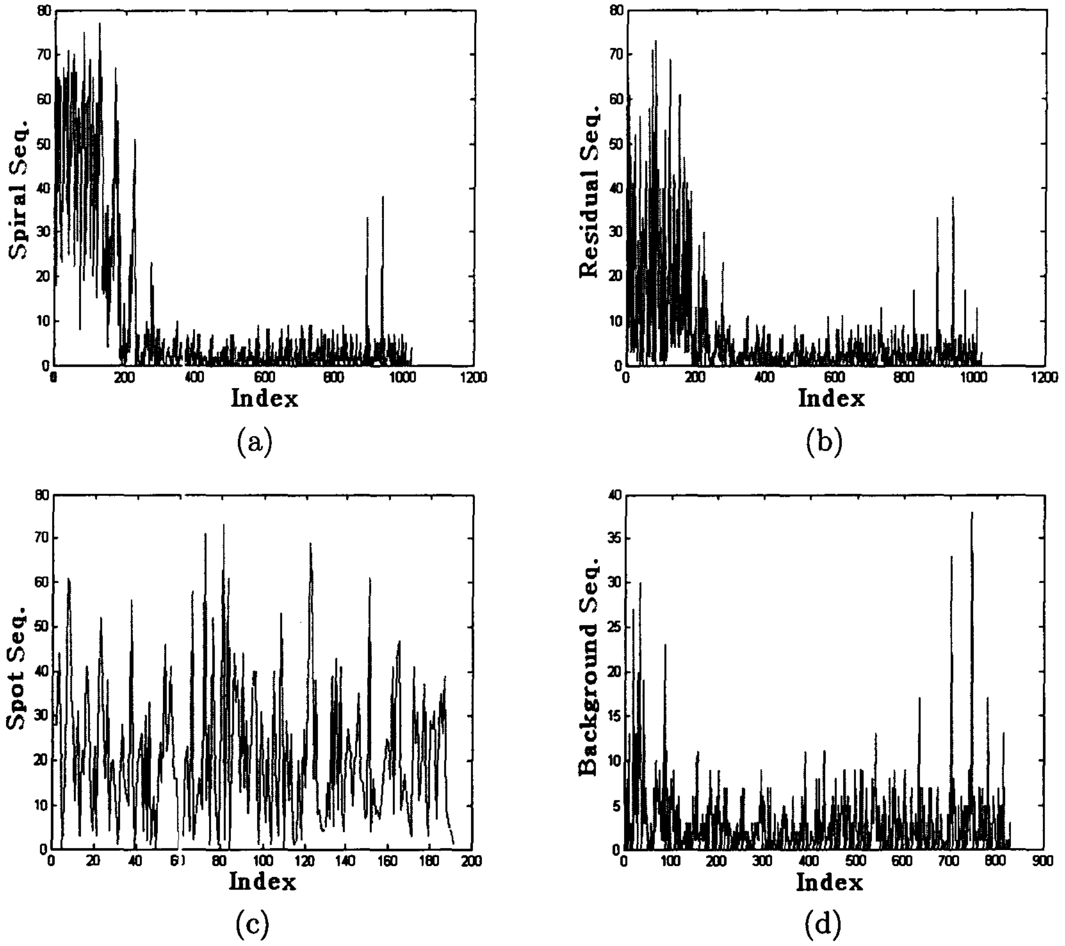
We also compare our method with conventional compression schemes and with the results of other published works on microarray image compression. We used a set of microarray images from different companies with different sizes and different precisions. Depending on the level of purity and noisy-ness of the image, we get compression ratios in the range of 1.45:1 to 2.15:1. The average compression ratio achieved by our algorithm is compared with the compression ratios of some conventional image and non-image compression algorithms in Table 4.3 [38]. One of the latest and most advanced implementations of JPEG-LS is used in this experiment[1]. As can be seen in Table 4.3, our proposed method outperforms all conventional compression schemes.

Table 4.3: Averaged compression ratio of our method compared to some other methods

Method	ZIP	GIF	TIFF	JPEG-2000	JPEG-LS	Our
Comp. ratio	1.47:1	1.35:1	1.35:1	1.54:1	1.80:1	1.89:1

[1] Apollo, PICTools version 2, Pegasus imaging corporation, April 2003

Our approach also outperforms the recent work of other groups on microarray image compression. Here we compare ours with two of the latest compression schemes. The first method proposed in [39] uses a LOCO compression method. The second uses object-based EBCOT coding [40]. The first method achieves an average compression ratio of 1.83:1, compared to 1.89:1 for our method. The second approach claims a 1.3% improvement with respect to JPEG-LS. For our approach, this improvement is 5.0% for a more advanced JPEG-LS implementation.



(a)

(b)

(c)

(d)

Figure 4.6: (a) Spiral path sequence, (b) residual sequence, (c) spot part, and (d) background part of residual sequence.

Chapter 5

Lossy Compression

5.1 Introduction

We mentioned in chapter 4 that microarray images are usually massive in size. An example of a typical DNA microarray image with a size of about 20MBytes was mentioned. Fig. 5.1 shows an example of a full size microarray image. One can choose either a lossless or a lossy method for compressing these images. Lossless compression methods, as their name says, do not change the image pixel values. For biomedical applications in which the pixel values can have critical information, this can be a good option. However, an important limitation of lossless compression is the low compression ratio that can be achieved for example 2.2:1 in good implementations [41]. On the other hand, using a lossy compression method, depending on the distortion introduced in the image, the desired compression can be achieved. In this work, we introduce an algorithm for lossy compression of microarray images.

5.2 Method

Fig. 5.2 shows the flow-graph of our proposed algorithm for lossy compression of DNA microarray images [42]. The first step is to localize and extract individual spots in the microarray image. This is necessary as some parts of our method work on each spot independently. Then, we match a circle to each of the spots. To do so, initial values for the center and the radius of such a circle is calculated by mathematical means. Then, these parameters are optimized for the best matching with the shape of spot. After having proper circle parameters matched to each spot, we perform a *circle-to-square* transform, C2S, to transform the area inside the circle for each spot to a corresponding square shaped image. The transform is designed for this application. Then, the resultant square images are put together, and are lossy compressed by means of the discrete cosine transform (DCT), quantization, and entropy coding.

5.2.1 Spot extraction

In our proposed method, the spots in the microarray image should be processed individually. Therefore, the first step of our method is to localize and extract microarray spots. The method used here is the same as the one used for lossless compression. Then, the spots are extracted into sub-images, and are processed later.

5.2.2 Parameters extraction

In order to apply our C2S transform to individual spots, the coordinates of the center of each spot and its radius should be extracted. To do so, we first find the initial approximations for the spot coordinates and radius, and then optimize them to best fit the spot. The initial coordinates are calculated as:

$$Center_x = \frac{\sum_{i=1}^{m_{Sub}} SubInt_x[i]i}{\sum_{i=1}^{m_{Sub}} SubInt_x[i]}$$

$$Center_y = \frac{\sum_{i=1}^{n_{Sub}} SubInt_y[i]i}{\sum_{i=1}^{n_{Sub}} SubInt_y[i]} \tag{5.1}$$

where m_{Sub} and n_{Sub} give the size of the spot sub-image. $SubInt_x$ and $SubInt_y$ are calculated in the same way as in (5.1), with the difference that they are calculated for each spot sub-image rather than the whole image. In (5.1), we are approximating the coordinates of the center of the spot with the 2D mean of the pixel values of the image. Experiment shows that (5.1) generates a good approximation. The initial value for the radius is given by:

$$R_p = (m_{sub} + n_{sub})/\gamma \tag{5.2}$$

where γ has a value between 7 and 8. Calculation of γ is based on both theoretical and experimental facts. First, it is obvious that $\gamma = 2$ will give an R_p equal to the average of the lengths of the spot sub-image's sides. As the shape of a spot sub-image is usually close to a square, this average is very close to either m_{sub} or n_{sub}. Second, as the radius of a spot is usually between 1/3 and 1/4 of the length of the sides of its sub-image, then the mentioned range for γ is obtained. The cost function used in our optimization is the difference between the mean values of the pixel intensities outside, and the pixel intensities inside the circle defined with parameters $Center_x$, $Center_y$, and R_p. The optimum choice of the circle parameters will result in an optimum separation of the pixels in the spot and background regions which will minimize the cost function. We have used a 2D pattern search method [37] for this optimization. As our initial values for optimization are usually close to the local optimum points, the pattern search method usually converges in 4 or 5 iterations. This optimization is done once for every spot in a microarray image. The cost function calculation has a run time in the order of the size of spot sub-image. So we conclude that the parameter generation has a run time in the order of image size, which is the order of the input of our algorithm. This maintains the fact that our method is relatively fast, despite having an optimization stage.

5.2.3 C2S transform

Most powerful transforms like DCT and wavelet are defined on rectangular shaped images, or matrices. In order to optimize their effect on our microarray images, we can transform spots into square shapes. This idea is implemented by designing a C2S transform to map the circular regions found in the previous step into square regions. Fig. 5.3 shows how our transform maps a circular area with radius R_p to a square with sides of length a. First, r and Θ are calculated for every pixel belonging to the square. Then we have:

$$L = \begin{cases} \frac{(a-Y_t)}{cos\Theta} & : \Theta \in [\pi/4, 3\pi/4] \\ \frac{X_t}{sin\Theta} & : \Theta \in [3\pi/4, 5\pi/4] \\ \frac{Y_t}{cos\Theta} & : \Theta \in [5\pi/4, 7\pi/4] \\ \frac{(a-X_t)}{sin\Theta} & : Otherwise \end{cases} \tag{5.3}$$

where (X_t, Y_t) is the center of the square. Then:

$$x = \frac{r}{L} R_p \tag{5.4}$$

is calculated and this gives us the distance of the point from the center we should be reading the desired value from. x is expected to have a value in the range of $[0, R_p]$. Having (x, Θ), we calculate the Cartesian coordinates of the pixels in the spot image whose intensity is assigned to the (r, Θ) pixel in the square image. Finally, these squares are arranged together to form an image. Fig. 5.4b shows how the transform works on a sample DNA microarray image.

To decode images which are compressed in this way, we need a S2C (Square to Circle) transform. The basics of this S2C transform are very similar to the C2S transform we introduced above. This time, we have the (R_p, Θ) pixel coordinates whose value should be read from the square image. Then L is calculated for the spot image and the calculation for r follows.

The joint application of C2S and S2C transforms can introduce losses into the microarray image which are independent from the losses resulting from subsequent quantization step. Assume that the square image that the spot image was transformed into has a small size. Obviously, some data corresponding to the spot image will be lost after the transform. The S2C transform will not perfectly reconstruct the spot image. An effect similar to spatial low-pass filtering will be observed after reconstruction. We will use this phenomenon for obtaining the rate-distortion characteristic of our algorithm.

5.2.4 DCT, quantization, and encoding

The image constructed by tiling the C2S transformed images together goes through the following stages. First, similar to JPEG, the image is divided into 8 × 8 blocks. These blocks are then DCT transformed. In order to reduce the blocking effect and improve the overall quality of compression, a in the C2S transform should preferably be set to a multiple of 8. After DCT is applied to each block, the transformed blocks are quantized and the whole image is variable length coded. Arithmetic coding is used for this application.

5.3 Results

Fig. 5.5 compares our results for lossy compression of a microarray image with lossy JPEG [38]. We have obtained this curve by fixing the quantization and coding parts of our method and changing the size of the C2S transform. This size is the parameter a in (5.4) and changing it subsequently changes X_t and Y_t. A larger value for a results in a larger size for the compressed image and a better quality for the decompressed image. We have chosen the mean square error for the measure of the distortion of images. The curve we obtain in this way is not very smooth and bumps can be seen in it in Fig. 5.6. The reason, as we also mentioned in the previous section, is whenever the parameter a is a multiple of 8, we get slightly better quality due to the reduced blocking effects of the coding steps. The rate distortion point corresponding to $a = 8$ can be found on the curve at the distortion value of 5000. The same is for $a = 16$ at the distortion value around 3300, and so on. All points of our results are positioned well below the rate distortion curve of JPEG.

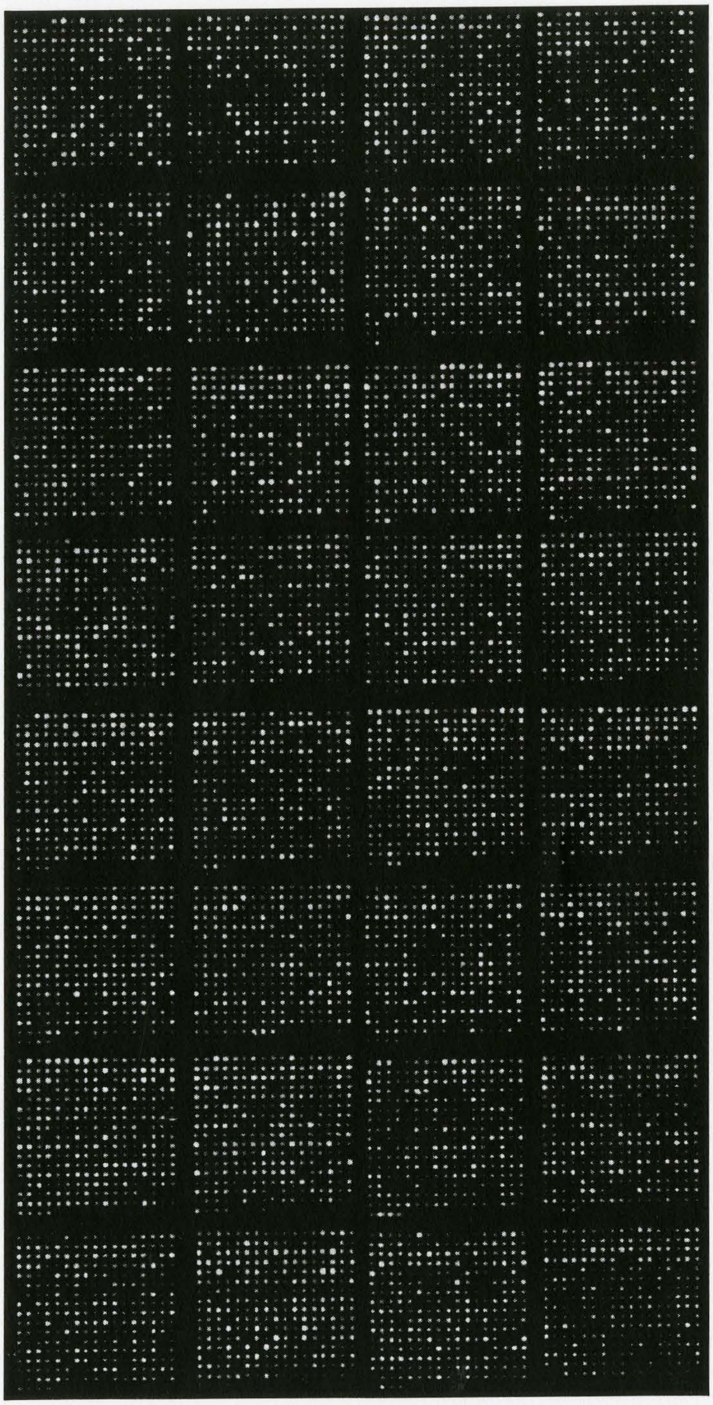


Figure 5.1: A full microarray image [46].

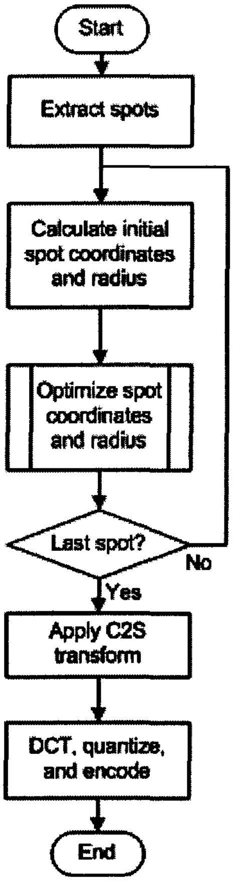

Start
Extract spots
Calculate initial spot coordinates and radius
Optimize spot coordinates and radius
Last spot?
No
Yes
Apply C2S transform
DCT, quantize, and encode
End

Figure 5.2: Flow-chart of our lossy compression algorithm.

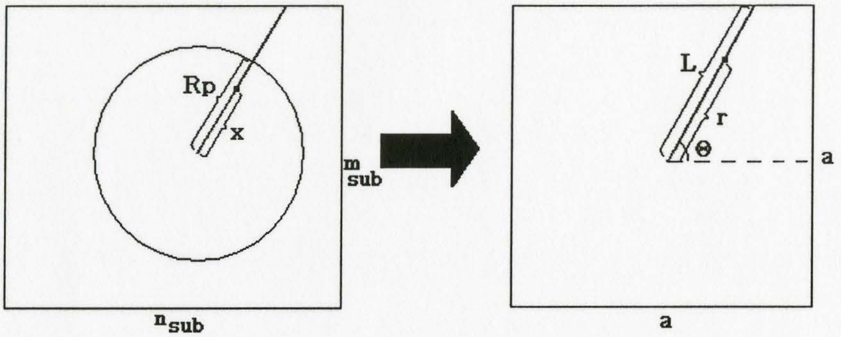


Figure 5.3: Geometric representation of C2S transform.

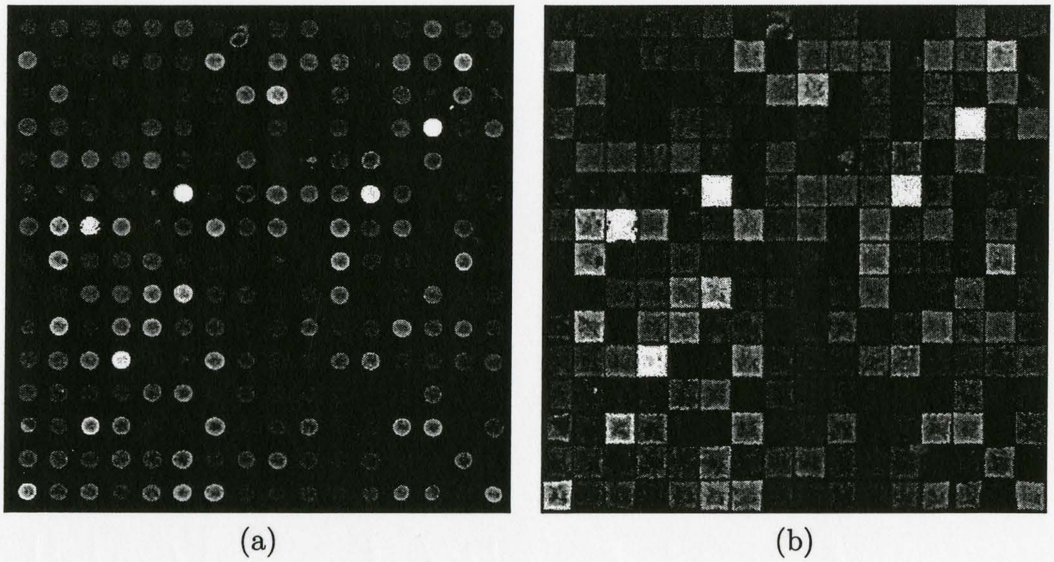


Figure 5.4: (a) A microarray image before, and (b) after applying the C2S transform to each of its spots.

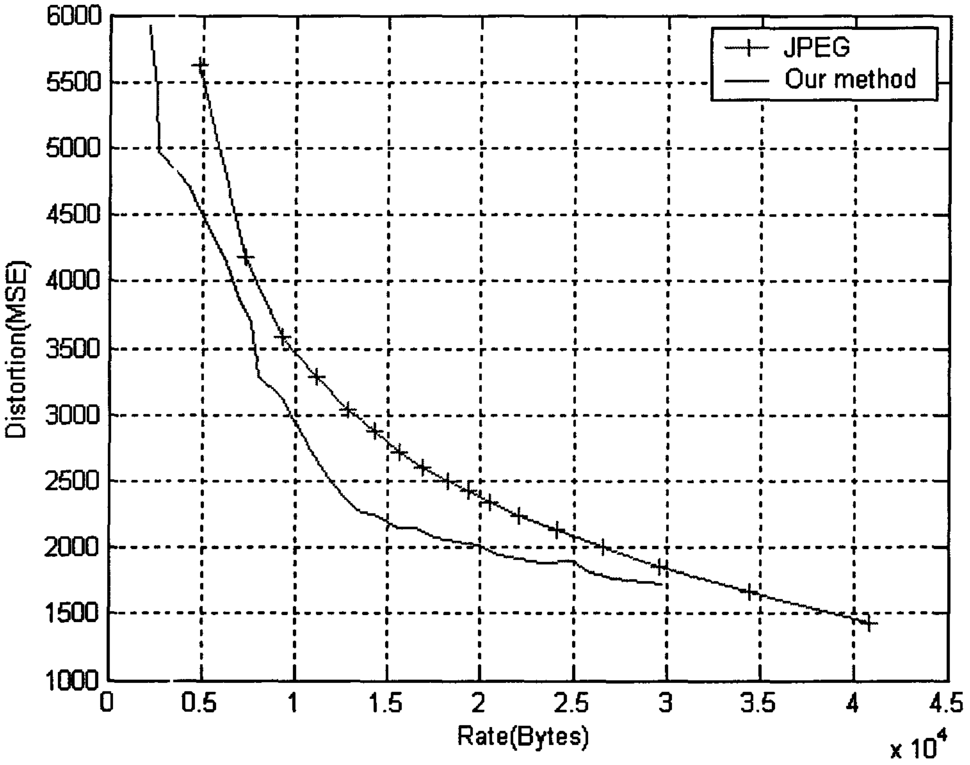


Figure 5.5: The rate distortion curve achieved by JPEG compared to curve achieved by our lossy compression method for the same test image.

Chapter 6

Conclusion and Recommendations

DNA microarray is a new and effective tool for studying of genes in biomedical science. In the work presented in this thesis, new ideas and algorithms for processing, modelling, and compression of microarray images have been introduced. For processing, this includes our routines for extracting red and green intensities of the spots and also their ratios removing background noises, dealing with dust and artifacts, and automatic detection of the grid type of the microarray. The routine introduced for background removal can be applied to any similar application in which data is stored in features of the image with spatially high frequencies.

For the modelling of spots in the microarray, the radially exponential model was introduced. It is a three dimensional mathematical model capable of describing the physical shape of the spot, according to the knowledge about its manufacturing process. The model is suitable for describing the behavior of the shape of droplets put on any flat surface. It can also be generalized to meet other existing physical constraints.

A new routine for lossless compression of microarray images has been introduced in this thesis. The idea of spiral path has been developed for the first time in this work. It is used for spatial scanning of the spots in the microarray, and it can be used for compression of any image which consists of circular features. It can also be upgraded to match virtually any geometric shape, with a good mathematical model. Spirals for ellipsoids and squares have already been developed. The routine also includes the new neighbor prediction feature, developed to work with the spiral path. It also uses the entropy optimization idea to divide data sequence into smaller pieces to get a better overall compression gain.

Lossy compression of microarray images has also been implemented. C2S transform is introduced for this application, to spatially transform the circle region, in which spot lies, into a square region prior to block codings. Experiment shows that a great improvement in the overall rate-distortion curve of the compression can be achieved by applying C2S.

There are many subjects mentioned in this thesis for which more research can be done. Some of the major subjects are listed now.

- The image processor presented in this thesis has covered many major and minor aspects of DNA image processing. For the gridding, more cases of irregular arrays can be added to the processor depending on the application. Shape recognition methods were not used in this implementation. One can try pattern recognition methods and transforms like Hough or Radon that may outperform this implementation.

- The model introduced in this thesis for microarray spots in chapter 3 is relatively fast and accurate, but it still has room for some improvement. One can expand it for asymmetric spot shapes. It can also be upgraded to cover not normalized volumes for applications in which the volume of the droplet in the process can be variable.

- In some microarray experiments, time variations of the intensities of the spots are important. Our approach can easily be upgraded to cover this type of experiments. As soon as our model parameters are extracted for the first image, they will remain constant and it will only be the parameter c which should be updated every time. The output curve which is the variation of c in time can also be easily extracted.

- When the spiral sequence for an image is extracted, the sample values on it are non-uniform samples of the original image. This is an immediate consequence of the discrete shape of the spiral path proposed in this work. Ordinary Fourier or DCT analysis of this sequence theoretically fails. For the best performance, non-uniform sampling theory can be added to the work, especially in the prediction and coding part.

- It is possible to upgrade the residual sequence division routine of our lossless approach, to produce three regions instead of two. The regions will be inside, boundary, and outside the spot. This may cause some improvements in the compression ratio of our routine. Though we don't promise much, as there will also be a new parameter for each spot which should be added to the header part.

- C2S transform can be generalized to fit the applications in which the data outside of the circle shouldn't be neglected. This can be done either by designing another function to transform the area outside of the circle into another square, or by upgrading C2S to a Square to Square (S2S) transform in which a circular area within the square is emphasized in the output.

- For lossy compression of circular shapes, another interesting approach which is not implemented yet is to design a transform coding scheme, like DCT or Wavelet, that can consider the radial behavior of the feature being coded. This will cancel the need for a C2S transform. Gabor and Zak transforms can be examples.

- The routines introduced in this thesis can be adjusted and simplified for hardware implementation [45]. Hardware implementation is specially important for remote biosensor applications, or lab-on-a-chip projects.

Bibliography

[1] Hummel S., *Ancient DNA Typing: Methods, Strategies and Applications*, Springer-Verlag, New York, 2002.

[2] http://www.cc.ndsu.nodak.edu/instruct/mcclean/plsc731/dna/dna1.htm.

[3] http://ghr.nlm.nih.gov/info=basics/section/dna.

[4] http://www.acefesa.es/microarray/asper/asper.htm.

[5] Snedden R., *DNA and Genetic Engineering*, Heinemann Library, 2002.

[6] http://www.affymetrix.com.

[7] Fodor S.P. et. al., "Light-directed, spatially addressable parallel chemical synthesis", Science 251, 767-73, 1991.

[8] Pease A.C. et. al., "Light-generated oligonucleotide arrays for rapid DNA sequence analysis", Proc. Natl. Acad. Sci. USA 91, 5022-26, 1994.

[9] Lockhart D.J. et. al., "Expression monitoring by hybridization to high-density oligonucleotide arrays", Nat. Biotechnol., 14, 1675-80, 1996.

[10] Gunderson K.L. et. al., "Mutation detection by ligation to complete n-mer DNA arrays", Genome Res., 8, 1142-53, 1998.

[11] Li L, Garden R.W., Sweedler J.V., "Single-cell MALDI: a new tool for direct peptide profiling", Trends Biotechnol., 18, 151-60, 2000.

[12] Walt D.R., "Techview: Molecular biology. Bead-based fiber-optic arrays.", Science, 287, 451-52, 2000.

[13] Heller M.J., "An active microelectronics device for multiplex DNA analysis", IEEE Trans. Eng. Med. Biol., 15, 100-3, 1996.

[14] Sosnowski R. et. al., "Rapid determination of single base mismatch mutations in DNA hybrids by direct electric field control", Proc. Natl. Acad. Sci., 94, 1119-23, 1997.

[15] Tang K., "Chip-based geno-typing by mass spectrometry", Proc. Natl. Acad. Sci., 96, 10016-20, 1999.

[16] Schena M., *DNA Microarrays: A Practical Approach*, Oxford Univ. Press, 1999.

[17] Rampal J.B., *Methods in Molecular Biology Vol. 170: DNA Arrays Methods and Protocols*, Humana, 2001.

[18] Heller M. J. "DNA microarray technology: Devices, systems, and applications", Annu. Rev. Biomed. Eng., 4, 129-53, 2002.

[19] http://www.medscape.com.

[20] Moore S.K., "Making Chips", IEEE Spectrum, pp. 54-60, March 2001.

[21] http://www.mcb.arizona.edu/wardlab/microarray.html.

[22] Hunter P., "Microarray data analysis: Separating the curd from the whey", Scientist, 50-1, Aug. 2003.

[23] T. Tu et al., "Quantitative noise analysis for gene expression microarray experiments", Proc. Natl. Acad. Sci., 99, 14031-6, 2002.

[24] Baumgartner R., Booth S., Bowman C., "Automated analysis of gene-microarray images", IEEE Canadian Conf. on Electrical and Computer Engineering, Vol. 2, pp. 1140-1144, 2002.

[25] www.makingindiagreen.org.

[26] Jouenne V. Y., *Critical Issues in the Processing of cDNA Microarray Images*, Virginia Polytechnic Institue, 2001.

[27] Brandle R. et al., "Robust DNA microarray image analysis", Machine vision and applications, 15, 11-28, 2003.

[28] Ballard D.H., Brown C.M. "Generalizing the Hough transform to detect arbitrary shapes", Pattern Recogn., 13, 111-122, 1981.

[29] Duda R. O., Hart P. E., "Use of the Hough tranformation to detect lines and curves in pictures", Comm. of the ACM, 15, 11-15, 1972.

[30] Gonzalez R.C., Woods R.E., *Digital image processing*, Addison-Wesley, 2nd Ed., 1992.

[31] Kamberova G., Shah S., "DNA arrays image analysis, nuts and bolts", DNA Press, Eagleville, PA, 2002.

[32] Russo E., "Chip critics countered", Scientist, 30-31, Aug. 2003.

[33] Bonner M. R. et al., "BioChip SNP analysis assay: Development of a 3-D microarray system", Motorolla BioChip Systems, Arizona.

[34] Plataniolis K.N., Smolka B., Szczepanski M.K., Venetsanopoulos A.N., "Enhancement of the DNA microarray chip images", IEEE 14th Conf. DSP, Vol. 1, pp. 395-398, 2002.

[35] Arena P., Fortuna L., Occhipinti L., "A CNN algorithm for real time analysis of DNA microarrays", IEEE Trans. Circuits Syst. I, Vol. 40, pp. 335-340, March 2002.

[36] Jornsten R., Yu B., "Comprestimation: Microarray Images in Abundance", Conf. on Information Science and Systems, CA, 2000.

[37] Polak E., Wetter M., "Generalized pattern search algorithms with adaptive precision function evaluations.", Technical report LBNL-52629, Lawrence Berkeley national library, Berkely, CA, 2003.

[38] Sayood K., *Introduction to data compression*, 2nd Edition, Morgan Kaufmann Pub., CA, 2000.

[39] Jornsten R., Yu B., Wang W, Ramchandran K., "Compression of cDNA microarray images", IEEE Int. Sym. on Biomedical Imaging, pp 38-41, 2002.

[40] Hua J., Xiong Z., Wu Q., Castleman K., "Fast segmentation and lossy-to-lossless compression of DNA microarray images", Workshop on Genomic Signal Proc., GENSIPS, North Carolina, 2002.

[41] Faramarzpour N., Shirani S., Deen M. J., "Spiral coding for lossless DNA microarray image compression", Revised submittion to IEEE Trans. on Image Processing, Feb. 2004.

[42] Faramarzpour N., Shirani S., Deen M. J., "Lossy compression of DNA microarray images", IEEE Canadian Conference on Electrical and Computer Engineering, 2004.

[43] Faramarzpour N., Shirani S., "Lossless and lossy compression of DNA microarray images", IEEE Data Compression Conference, Snowbird, UT, Mar. 2004.

[44] Faramarzpour N., Shirani S., Bondy J., "Lossless DNA micro- array image compression", IEEE Asilomar Conf. on Signals, Systems, and Computers, Pacific Groves, CA, Nov. 2003.

[45] Samavi S., Shirani S., Karimi N., Faramarzpour N., "DNA microarray image compression by pipeline architecture", IEEE Asilomar Conference on Signals, Systems, and Computers, Pacific Groves, CA, Nov. 2003.

[46] http://dir.niehs.nih.gov/microarray/images/.

[47] Wang, J., "Survey and Summary from DNA biosensors to gene chips", Nucleic Acids Research, Vol. 28, No. 16, pp. 3011-3016, 2000.

[48] Baldi P., Hatfield G. W., *DNA Microarrays and Gene Expression: From Experiments to Data Analysis and Modeling*, Cambridge University Press, UK, 2002.