An Original Genetic Approach to the Fully-Automatic Gridding of Microarray Images

Eleni Zacharia, and Dimitris Maroulis, *Member, IEEE*

***Abstract*—Gridding microarray images remains, at present, a major bottleneck. It requires human intervention which causes variations of the gene expression results. In this paper, an original and fully-automatic approach for accurately locating a distorted grid structure in a microarray image is presented. The gridding process is expressed as an optimization problem which is solved by using a Genetic Algorithm. The Genetic Algorithm determines the line-segments constituting the grid structure. The proposed method has been compared with existing software tools as well as with a recently published technique. For this purpose, several real and artificial microarray images containing more than one million spots have been used. The outcome has shown that the accuracy of the proposed method achieves the high value of 94% and it outperforms the existing approaches. It is also noise-resistant and yields excellent results even under adverse conditions such as arbitrary grid rotations, and the appearance of various spot sizes.**

***Index Terms*—Genetic algorithm, Image analysis, Microarrays, Spot gridding.**

I. INTRODUCTION

CDNA microarrays is a fundamental and powerful biotechnology tool which enables scientists to simultaneously monitor the expression levels of thousands of genes over different samples [1]. It has been utilized in a wide variety of different biomedical application areas, such as: (i) cancer research (i.e. determination of molecular differences between normal and abnormal cells, classification of tumors), (ii) infectious disease diagnosis and treatments (i.e. determination of risk factors, monitoring treatment during different disease stages), and (iii) pharmacology research (i.e. determination of correlations between the genetic profiles of patients and their therapeutic responses to drugs) [2].

In cDNA microarrays [3], a set of DNA probes that are of particular interest are placed on a glass slide, creating an invisible array of DNA dots. Two distinct populations of mRNA, are reversely transcribed into cDNAs, which in turn are colored with fluorescent dyes. The cDNA populations are then hybridized with the slide's DNA dots. The hybridized glass slides are fluorescently scanned, and two digital images are produced, one for each population of mRNA. Each digital image contains a number of spots (corresponding to the DNA-cDNA dots) of various fluorescence intensities. Given that the intensity of each spot is proportional to the hybridization level of the cDNAs and the DNA dots, the gene expression information is obtained by analyzing the digital images.

An important first stage in microarray image analysis is gridding, which is the process of segmenting a microarray image into numerous compartments, each containing one individual spot and the background. Although this process may seem relatively straightforward, it is in fact rather complicated since the quality of images suffers from the existence of noise (i.e. dust on the slide), artifacts (i.e. inner holes and scratches) and uneven background, while some spots are poorly contrasted and ill-defined [4]. In addition, given that spots vary in size and position due to the presence of noise during the sample preparation and hybridization processes, there may be rotations, misalignments and local deformations of the ideal rectangular grid [5].

As a result, the available gridding software programs (i.e. ScanAlyze [6], Dapple [7], ImageGene [8], and SpotFinder [9]) require human intervention in order to specify input parameters as well as to adjust properly the location of the grid structure. Automating this part of the process is essential because: (i) it will allow rapid high throughput analysis of the expression levels of thousands of genes, and also (ii) it will prevent variations in the results of gene expression levels. Indeed, the experiment reported in [10] shows that for the same microarray slide, human intervention in the gridding procedure leads to significant discrepancies in the gene expression levels.

Other well-known approaches to gridding microarray images are based on axis projections [11], or on morphological filtering [12]. Both of them require user intervention in order to manually adjust the grid location. The Hill-Climbing approach for automatic gridding [13] can perform gridding properly only if misalignments and rotations of the ideal grid are not present. Markov random field (MRF) [14] and graph-based grid approaches [15] have been also applied to gridding. A drawback of these approaches is that they require input

Manuscript received September 12, 2007. This work was supported by the Greek general Secretariat of Research and Technology and the European Social Fund, through the PENED 2003 program (grant no, 03ED332).

E. Zacharia, and D. Maroulis are with the Department of Informatics and Telecommunications, University of Athens, GR 15784 Panepistimiopolis, Ilisia, Greece (e-mails: eezacharia@gmail.com, dmaroulis@di.uoa.gr rtsimage@di.uoa.gr).

parameters.

A variety of different methodologies have been proposed with the intension to solve rotation and misalignment problems. Bajcsy [16] suggested an exhaustive search of all the expected rotation angles. Steinfath [17] proceeded by estimating the rotation angle. A drawback of the latter is that it introduces pixel distortions when the rotation angle is small. Brandle et al. [18] utilized the discrete Radon transformation to estimate the angle rotation. As it is computationally expensive, the process is accelerated by constraining the range of rotation angles. Ho et al. [19] expressed the gridding process as an optimization problem based on the Jacobi iteration. However, this method is efficient only when the grids are smoothly distorted. Giuiliano et al. [20] recommended a gridding procedure based on stochastic search algorithms. Although it deals with rotations effectively, it requires manual intervention in order to define the radius of the spots. As a result, it is not efficient when the microarray image contains various spot sizes.

In this paper, an original, fully-automatic and unsupervised approach to gridding microarray images is presented. It relies on a Genetic Algorithm, which determines very effectively the line-segments, constituting the borders between adjacent blocks or spots. The proposed method can deal with rotations, misalignments and local deformations of the ideal rectangular grid, as line-segments may have various angles in the microarray image. It is also noise-resistant and it is efficient even under adverse conditions such as the appearance of various spot sizes or the absence of spots.

The rest of this paper is structured in four sections as follows: In Section II, a typical cDNA microarray image is portrayed and a brief description of Genetic Algorithms is provided. In Section III, the proposed gridding method is presented. The gridding process is expressed as an optimization problem which then is solved by using a specific Genetic Algorithm. In Section IV experiments are presented that test the proposed gridding method and compare it to existing software packages for microarray image analysis, as well as to a recently published technique. To this extent, artificial microarray images, and real microarray images were used. Our conclusions are apposed in section V.

II. Background Material

A. A typical cDNA microarray image

A typical cDNA microarray image usually contains one or more distinct, rectangular or square blocks, each one containing equal number of spots. These blocks are arranged in a 2D array layout. Under magnification, spots belonging to the same block are arranged in a 2D array layout too. Fig. 1 illustrates a typical microarray image which is composed of 6 blocks in a 3x2 layout. Each block contains equal number of spots which are located in a 16x17 block layout.

As it can be easily observed, a typical microarray image has the following properties:

--All blocks inside the microarray image contain equal number of spots being arranged in identical 2D array layouts.

--Adjacent blocks and adjacent spots are clearly separated. However, the edge-to-edge distance between two adjacent blocks is larger than the distance between two adjacent spots within each block.

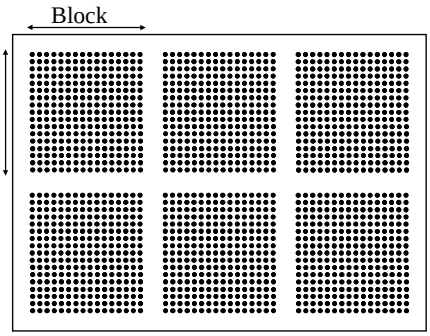


Fig. 1. A typical microarray image containing 6 blocks each one having 272 spots.

B. Genetic Algorithms

Genetic Algorithms (GAs) are powerful, stochastic non-linear optimization tools based on the principles of natural selection and evolution [21]. Compared to traditional search and optimization tools (such as Blind Search Algorithms), GAs demonstrate superior performance, given that they are robust optimizers, suitable for solving problems for which there is little or no a priori knowledge of the underlying processes.

Given a specific optimization problem, a typical GA searches for the optimal solution as follows: Firstly, it creates a finite number of potential solutions encoded as alpha-numerical sequences called Chromosomes. These Chromosomes constitute an initial Population Pop_1. Subsequently, the GA produces a new Population Pop_2 according to the following: The Chromosomes constituting the Pop_1 are evaluated using a Fitness Function. Thereafter, the GA evolves the Population Pop_1 into a new Population Pop_2 using the three Genetic Operators: Reproduction, Crossover, and Mutation. This Evolutionary Cycle from one Population to the next (Pop_1 to Pop_2, Pop_2 to Pop_3 and so forth) continues until a specific termination criterion is satisfied. Subsequently, the essential elements of the GA are: Chromosome representation, Chromosome evaluation, the Evolutionary cycle, and the Termination criteria.

A Chromosome is often represented as a simple alpha-numerical sequence which encodes the values of variables defining a possible solution to the optimization problem at hand. Although a traditional GA uses a binary number in order to encode these variables, in the present application, a Real-Coded Genetic Algorithm (RCGA), which uses real values, is applied. The reason is that real-coded Chromosomes exhibit various advantages over binary-coded Chromosomes as they can use large or unknown domains for the variables they encode. On the other hand, assuming that the Chromosome has

a fixed length, binary implementations cannot increase the domain without sacrificing precision [22].

The evaluation of the Chromosome is based on a Fitness Function which assigns to the Chromosome a Fitness Value measuring the quality of the solution that the Chromosome represents. Naturally, the Fitness Function depends on the particular optimization problem at hand and on the Chromosome representation.

Reproduction, Crossover and Mutation are the three Genetic Operators used for the creation of new Chromosomes [22]. All of them have been implemented in several, distinct fashions depending on the Chromosome representation.

Common terminating criteria are: (i) A solution that satisfies the defined minimum standards, (ii) The attainment of a maximum number of Populations, (iii) The attainment of a fixed number of Populations for which the Fitness Value of the best Chromosome remains the same, and (v) Combinations of the above [23].

III. The Proposed Genetic Approach to Gridding Microarray Images

Due to the nature of the alignment of blocks inside the microarray image and the arrangement of spots inside the blocks, the gridding procedure is divided into two stages:

STAGE I: The microarray image is segmented into blocks, by determining (drawing) a set of line-segments L_{BG} whose members are the line-segments constituting the borders of adjacent blocks.

STAGE II: Each block (from Stage I) is segmented into single-spot compartments, by determining (drawing) a set of line-segments L_{SG} whose members are the line-segments constituting the borders between adjacent spots.

In detail, let G be a microarray image or a block, which makes it quadrilateral in shape. Its boundaries are named as "*Line_Above*", "*Line_Below*", "*Line_Left*" and "*Line_Right*" (Fig. 2).

In the particular case when G is an MxN microarray image, it becomes rectangular in shape. "*Line_Above*" is defined then by the end points $B(0,0)$ and $C(0,N\text{-}1)$, "*Line_Below*" is defined by the end points $A(M\text{-}1,0)$ and $D(M\text{-}1,N\text{-}1)$, "*Line_Left*" is defined by the end points $B(0,0)$ and $A(M\text{-}1,0)$ and "*Line_Right*" is defined by the end points $C(0,N\text{-}1)$ and $D(M\text{-}1,N\text{-}1)$. Otherwise, when G is a block, "*Line_Above*", "*Line_Below*", "*Line_Left*" and "*Line_Right*" are four line-segments determined in stage I of the gridding procedure. In the latter case, the borders of G can have any possible direction on the two-dimensional Cartesian plane.

Each set of line-segments (L_{BG} or L_{SG}) can be divided into a sub-set of line-segments L_V whose members are the line-segments defined by "*Line_Above*" and "*Line_Below*" of G and into a sub-set of line-segments L_H whose members are the line-segments defined by "*Line_Left*" and "*Line_Right*" of G.

The determination of line-segments which are included in either the L_V or the L_H sub-sets can be viewed as an optimization problem which is tackled by using the proposed Genetic Algorithm which determines the exact values of the variables of all the optimal line-segments included in both sub-sets, one sub-set at a time. For this purpose, the following elements of GAs must be determined: Chromosome representation, Chromosome evaluation, the Evolutionary cycle, and the Termination criteria.

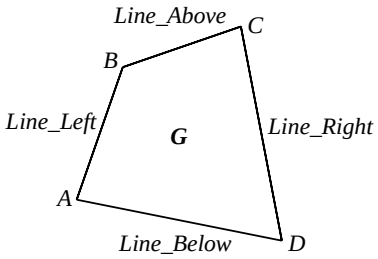


Fig. 2. "*Line_Above*", "*Line_Below*", "*Line_Left*" and "*Line_Right*" of a microarray image or block G.

A. Chromosome representation

The Chromosome m represents all line-segments L_i, i=1,…,$N(m)$ belonging either to the L_V sub-set or the L_H sub-set. $N(m)$ is the number of the line-segments belonging to the respective sub-set.

The line-segments, in both sub-sets, are represented on a two-dimensional Cartesian system. Thus, they can be described algebraically by the following linear equation:

$$y = y_1 + \frac{y_2 - y_1}{x_2 - x_1}(x - x_1) \tag{1}$$

where (x_1, y_1), (x_2, y_2) denote the coordinates of the two end-points of the line-segment, and x is the independent variable of the equation, whose range is defined as:

$$x \in [x_1, x_2] \tag{2}$$

The Cartesian system used for the representation of line-segments belonging to the L_V sub-set (Fig. 3a) always differs to the one used in the L_H sub-set (Fig. 3b). The reason is that (1) is not defined when x_1 is equal to x_2.

The first line-segment represented by the Chromosome m is L_1. In the case when the Genetic Algorithm searches for the optimal line-segments included in the L_V sub-set, L_1 is defined as the line-segment which is left to the first column of blocks (or spots), nearest to the line-segment "*Line_Left*" ($line_{V1}$, Fig. 4). Likewise, in the case when the Genetic Algorithm searches for the optimal line-segments included in the L_H sub-set, L_1 is defined as the line-segment which is above the first row of blocks (or spots), nearest to the line-segment "*Line_Above*" ($line_{H1}$, Fig. 4).

The rest of the line-segments L_i, i=2,…,$N(m)$, represented

by the Chromosome *m*, are defined as line-segments which are parallel to L_1, and their distance from L_1 is equal to $(i-1)\cdot d$. d is the distance between two adjacent line-segments. Thus, the line-segments L_i, i=2,…,N(m) can be described by the following linear equation, in which the unknown variables are the line-segment L_1 and the distance d:

$$L_i = L_1 + (i-1)\cdot d \tag{3}$$

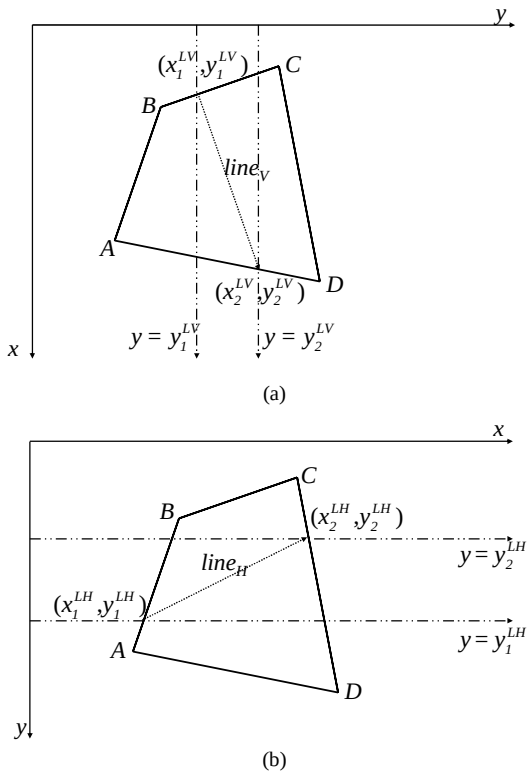


Fig. 3. Two dimensional Cartesian systems used for the representation of line-segments and graphical explanation of the computation of the x-coordinates of the end points of line-segments: (a) belonging to the L_V subset, (b) belonging to the L_H subset.

As previously mentioned, the Chromosome *m* represents all the line-segments L_i, i=1,…,N(m) belonging either to the L_V sub-set or to the L_H sub-set. Consequently, the Chromosome should encode the end-points $[(x_1^{L1}, y_1^{L1}), (x_2^{L1}, y_2^{L1})]$ of the line-segment L_1 and, the distance d. However, instead of this, the Chromosome encodes only: (i) the y-coordinate (y_1^{L1}) of the start point of the line-segment L_1, (ii) the y-coordinate (y_2^{L1}) of the end point of the line-segment L_1 and, (iii) the distance d between two adjacent line-segments (Fig. 5).

It is worth pointing out that the x-coordinates (x_1^{L1} and x_2^{L1}) of the end-points of the line-segment L_1 have not been included in the Chromosome. The reason is that they can be computed from the y-coordinates. For instance, the x-coordinates of the line-segment "$line_V$" (Fig. 3a) can be computed given that: (i) the intersection point of the line $y = y_1^{LV}$ and the line-segment "*Line_Above*" (*BC*) is the point (x_1^{LV}, y_1^{LV}), and (ii) the intersection of the line $y = y_2^{LV}$ and the line-segment "*Line_Below*" (*AD*) is the point (x_2^{LV}, y_2^{LV}). Equivalent, the x-coordinates of the line-segment "$line_H$" (Fig. 3b) can be computed given that: (i) the intersection point of the line $y = y_1^{LH}$ and the line-segment "*Line_Left*" (*BA*) is the point (x_1^{LH}, y_1^{LH}), and (ii) the intersection of the line $y = y_2^{LH}$ and the line-segment "*Line_Right*" (*CD*) is the point (x_2^{LH}, y_2^{LH}).

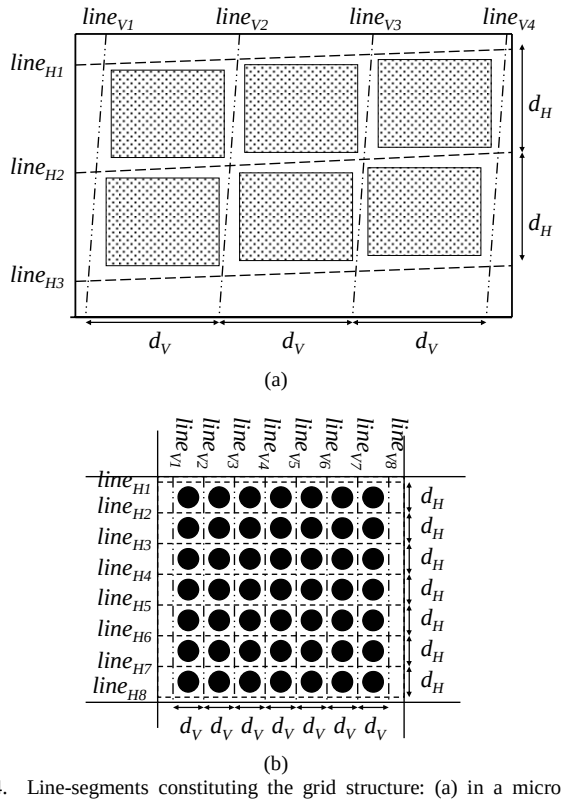


Fig. 4. Line-segments constituting the grid structure: (a) in a microarray image, (b) in a block.

Two gridding results are shown in Fig. 4. In the case when the Genetic Algorithm searches for the exact values of the variables of the optimal line-segments defined by "*Line_Above*" and "*Line_Below*", its Chromosome will encode the y-coordinates of the end-points of "$line_{V1}$" and "d_V". In the case when the Genetic Algorithm searches for the exact values of the variables of the optimal line-segments defined by "*Line_Left*" and "*Line_Right*", its Chromosome will encode the y-coordinates of the end-points of "$line_{H1}$" and "d_H".

y_1^{L1}	y_2^{L1}	d

Fig. 5. The parameters encoded in the real-value Chromosome used in the GA.

Obviously, the Genetic Algorithm can deal with rotations, misalignments and local deformations of the ideal rectangular

grid. Indeed, given that the angle of the line-segment is depended on its end points, the Genetic Algorithm can determine line-segments which have various angles as it determines the end points of the line-segments.

B. Chromosome evaluation

Each Chromosome m in every Population is evaluated using a Fitness Function, $F(m)$, which assigns to it a degree of how appropriate a solution to the gridding optimization problem it is. The higher the value of the Fitness Function, the more appropriate the Chromosome is. As far as the gridding optimization problem, the Chromosome evaluation contains the following two objectives: (i) Maximization of the number of line-segments which are determined simultaneously; (ii) Maximization of the probabilities of all the determined line-segments to be part of the grid.

Hence, it becomes essential to define the probability of a line-segment to be part of the grid, before the mathematical definition of the Fitness Function.

Probability of a line-segment to be part of the grid

A line-segment which is part of the grid is located in an area empty of spots. The pixels of this area are part of the background and their intensities are generally lower than the intensities of the pixels constituting spots.

As a result of the above observation, we define a region R_{Li} which is located on either side of the line-segment L_i. More precisely, R_{Li} is defined as:

$$R_{Li} = \{ p \mid (p \in G) \wedge d(p, L_i) \le w \} \tag{4}$$

where p are the pixels contained within the quadrilateral G and $d(p,L_i)$ denotes the distance of the pixel p to the line-segment L_i. w is a constant integer which controls the width of the region R_{Li}, on either side of the line-segment L_i.

The probability $P(L_i)$ of a line-segment L_i to be part of the grid is defined as the percentage of background pixels $f_B(L_i)$ located in R_{Li} minus the percentage of spot pixels $f_S(L_i)$ located in R_{Li}. It is computed by the following equation:

$$P(L_i) = f_B(L_i) - f_S(L_i) \tag{5}$$

The real-value functions $f_B(L_i)$ and $f_S(L_i)$ are computed as follows:

$$f_B(L_i) = \frac{\#\{ p \mid ((p \in R_{Li}) \wedge (I(p) \le I_B))\}}{\#\{ p \mid p \in R_{Li} \}} \tag{6}$$

and

$$f_S(L_i) = \frac{\#\{ p \mid ((p \in R_{Li}) \wedge (I(p) > I_B))\}}{\#\{ p \mid p \in R_{Li} \}} \tag{7}$$

where the symbol # denotes the number of the elements of the set that is defined by the brackets {}. $I(p)$ denotes the intensity of the pixel p. I_B is the intensity value that is present in most pixels of G. An example of I_B is illustrated in Fig. 6.

It is self-explanatory that the majority of pixels of G forms the background. Therefore, I_B corresponds to a threshold. Pixels with intensity lower or equal to I_B have a distinct probability to belong to the background, while pixels with intensity higher than I_B have a distinct probability to belong to the spots.

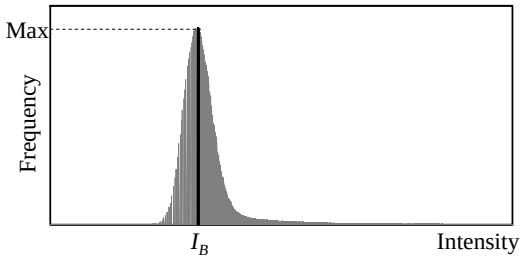


Fig. 6. A typical histogram of G, where I_B is depicted.

Fitness Function

The Fitness Function, $F(m)$, of a Chromosome m that encodes a possible solution to the particular optimization problem is defined by the following equation:

$$F(m) = \begin{cases} S_p(m) \cdot N(m), \; if \; f_{LS}(m) \le f_{Max} \\ S_p(m), \; otherwise \end{cases} \tag{8}$$

where

$$S_p(m) = \sum_{i=1}^{N(m)} P(L_i) \cdot q_i \,, \tag{9}$$

and

$$q_i = \begin{cases} 1, & if \; P(L_i) > P_{MA} \\ 0, & otherwise \end{cases} . \tag{10}$$

Also,

$$f_{LS}(m) = \frac{\sum_{i=1}^{N(m)} k_i}{N(m)}, \tag{11}$$

and

$$k_i = \begin{cases} 1, & if \; P(L_i) \le P_{Low} \\ 0, & otherwise \end{cases} . \tag{12}$$

$S_p(m)$ denotes a total sum of the probabilities $P(L_i)$ of the line-segments L_i, i=1,…,$N(m)$, that are represented by the Chromosome m, and have a higher than a threshold P_{MA} probability $P(L_i)$ to be part of the grid. P_{MA} is a threshold which expresses the minimum acceptable probability of a line-

segment to be part of the grid. Therefore, it controls which of the line-segments L_i participate in the sum $S_p(m)$.

$f_{LS}(m)$ denotes the percentage of the line-segments L_i, i=1,…,$N(m)$, that are represented by the Chromosome m, and have a lower than (or equal to) a threshold P_{Low} probability $P(L_i)$ to be part of the grid, where $P_{Low} < P_{MA}$. $N(m)$ denotes the total number of the line-segments L_i which are represented by the Chromosome m.

The Fitness Function $F(m)$ of a Chromosome m equals to $S_p(m)$ or $S_p(m)\cdot N(m)$, according to the value of the percentage $f_{LS}(m)$. If the percentage $f_{LS}(m)$ of the Chromosome m is higher than a threshold f_{Max}, it means that the line-segments, represented by the Chromosome m, are ill-defined because they are located in areas containing spots instead of background (1st case). On the other hand, if the percentage $f_{LS}(m)$ of the Chromosome m is lower or equal to the threshold f_{Max}, it means that the line-segments, represented by the Chromosome m, are well-defined because they are located in background areas, some of which may be contaminated with noise (2nd case). Using the Fitness Function $F(m)$, the Genetic Algorithm can assign to the Chromosome m of the 1st case a lower Fitness Value than to the one of the 2nd case. Moreover, the multiplication $S_p(m)\cdot N(m)$ in (8) prevents the Genetic Algorithm from converging to a local solution that would be not an efficient one (this would lead to termination without determining all the line-segments belonging to the L_H or the L_V sub-sets). Indeed, the higher the Fitness Value of the Chromosome is, the greater number $N(m)$ of the line-segments L_i which have a high probability $P(L_i)$ to be part of the grid it represents.

It is worth pointing out that the probabilities $P(L_i)$ of the line-segments L_i which are less or equal to the threshold P_{Low}, are not taken into account for the calculation of (9) given that $P_{Low} < P_{MA}$. As a result, the Fitness Function $F(m)$ exploits only the percentage of the line-segments L_i (11) and not the exact values of their probabilities $P(L_i)$.

C. Evolutionary circle -Termination Criteria

Let Pop_n be a Population of Chromosomes which consists of N_{pop} Chromosomes, where n stands for the consecutive number of Populations. A new Population Pop_{n+1} of an equal number of Chromosomes (N_{pop}) is created through the following stages: (i) Reproduction stage: P_r% of the best Chromosomes of the current Population Pop_n are carried over to the new Population Pop_{n+1}. (ii) Crossover-Mutation stage: The Chromosomes needed to complete the new Population Pop_{n+1} are produced through iterations of the following: Four Chromosomes of the Population Pop_n are selected using the tournament selection method [24]; These Chromosomes are subsequently subjected in turn to a Crossover operator (according to a P_c% probability) and then to a Mutation operator (according to a P_m% probability). The best two of the four resulting Chromosomes (the two with the best Fitness Value) proceed to the new Population Pop_{n+1}.

It should be noted that the Crossover operator applied, is the joint application of the BLX-a, and the Dynamic Heuristic Crossover as it is the most promising Crossover application [25]. Moreover, the Mutation operator applied, is the wavelet-Mutation as it exhibits a fine-tune ability as opposed to other Mutation operators [26].

New Populations are thus produced until at least one of the following two criteria is met: (i) the Genetic Algorithm is executed up to a maximum number of Populations G_{Max}; (ii) the Genetic Algorithm is executed up to a maximum number of Populations G_{Fit} for which the best Fitness Value has remained unchanged.

IV. Results

Several experiments were executed so as to evaluate the performance of the proposed method for gridding cDNA microarray images. It should be noted that the following pre-processing step was applied before gridding. The reason being that, microarray images may contain low-intensity spots which are not clearly visible. Hence, in order to study them, we adjusted the pixels' intensity of the microarray images so that low-intensity spots are amplified and high-intensity spots are not saturated. To this extent, the Box-Cox transformation was applied as a pre-processing step, before gridding, as it has been proven useful for adjusting microarray spot intensities [27].

The microarray images used in the experiments are divided in two different datasets:

The first dataset contains 25 real microarray images from the Stanford Microarray Database (SMD) [28], which is publicly available. The images are digitized at ~ 5000 x 2000 pixels at 16-bit grey level depth and they are stored in tiff format. Each one of them encloses 48 blocks, each block containing 864 spots. The microarray images have been produced by comprehensively analyzing the gene expression profiles in 54 specimens of acute lymphoblastic leukemia, 37 positive and 17 negative to BCR-ABL [29]. BCR-ABL is a fusion gene product resulting from translocation between the 9th and the 22th chromosomes.

The second dataset contains the data used for the evaluation of the gridding algorithm described by Blekas et al.[30]. More precisely, it contains ten microarray blocks, which have been arbitrarily selected from ten microarray images, artificially created or obtained from publicly available microarray databases. These blocks are stored in tiff format, at 16-bit grey level depth and they have been obtained from paper [30] upon request from its authors.

Using the two datasets, it is demonstrated that the proposed method can accurately determine the grid structure even if the images have been produced by different technologies or even if the images contain microarray spots of varying quality. The enormous number of spots which are contained in the microarray images used in the experiments additionally support this argument. Indeed, the microarray images contain 1,040,300 microarray spots from which 1,036,800 spots are contained in the first dataset and 3500 spots are contained in

the second dataset. Moreover, using the second dataset, the performance of the proposed method is compared with the one proposed in [30], as well as with other well-known software programs (ScanAlyze and SpotFinder).

During the experiments, the efficiency of the proposed method was analyzed by means of a statistical analysis. The statistical analysis resembles to the one described in [30]. More precisely, each microarray spot existing in the microarray images was classified in one of the following three categories: "perfectly", "marginally" and "incorrectly" gridded. A spot was "perfectly" gridded if the entire spot area was contained inside the equivalent compartment of the grid. This means that even if one pixel of the microarray spot was outside the compartment then it could not be classified in this category. A spot was "marginally" gridded if at least 80% of the entire spot area was contained inside the equivalent compartment of the grid. A spot was "incorrectly" gridded if less than 80% of the entire spot area was contained inside the equivalent compartment of the grid. All of the spot areas of the microarray spots were either the ones annotated in the SMD results, for the first dataset, or were defined, for the second dataset, in the same manner, as it is described at [30].

In all the experiments, the population size of the Genetic Algorithm was set to 100. This size is high enough to reduce the possibility of the Genetic Algorithm to prematurely converge to a local solution that would not be an efficient one. Meanwhile, it does not increase the time required for the population to converge to an efficient solution [31]. The percentage of each Population which was reproduced was relatively small (P_r=10%) as the reproduction was used only for the best Chromosomes of the Population to be preserved in the next Population. In accordance with [32] the high Crossover probability of 80% was chosen (P_c=80%). The Mutation probability was experimentally adjusted to 80% too (P_m=80%). In [33], it is suggested that the Real-Coded GAs may take advantage of high Mutation rates. Our experiments have confirmed this advantage of high Mutation rates as a low mutation rate could in fact cause the Genetic Algorithm to find a solution that is not efficient. The reason is that the Real-Coded GAs do not provide enough diversity through the Crossover operation alone. Mutation on the other hand can select a new real value within the allowable range of each of the designed genes of the Chromosome. The Termination criterion was satisfied when the Genetic Algorithm was executed for 1000 Populations (G_{Max}=1000) or when the best fitness value remained unchanged for 200 Populations (G_{Fit}=200).

Fitness parameters have been experimentally adjusted. The value of the afore-mentioned margin (w) depends on which of the two stages of the gridding procedure (section III) the Genetic Algorithm is executed. This is due to the fact that the edge-to-edge distance between two adjacent blocks is larger than the distance between two adjacent spots within each block. As a result, the margin (w) was set to 8 when the Genetic Algorithm was searching for line-segments constituting the borders between two adjacent blocks (Stage I of the gridding procedure). Respectively, when the Genetic Algorithm was searching for line-segments constituting the borders between two adjacent spots, the margin (w) was set to 2 (Stage II of the gridding procedure). A minimum acceptable probability P_{MA} of 0.7 was adopted. A threshold P_{LOW} of 0.5 and a threshold f_{Max} of 0.2 were adopted as the most appropriate in order to distinguish the Chromosomes which represent line-segments located in background areas from the ones which represent line-segments located in spot areas.

Table I summarizes the Genetic Algorithm parameters used in the experiments.

TABLE I
PARAMETERS VALUES USED FOR THE GENETIC ALGORITHM

Parameters	Values
Population parameters	
Number of chromosomes (N_{pop})	100
Genetic- operators parameters	
Reproduction percentage (P_r%)	10%
Crossover probability (P_c%)	80%
Mutation probability (P_m%)	80%
Fitness parameters	
Margin (w)	8 or 2
Minimum acceptable probability (P_{MA})	0.7
Threshold of low probability (P_{LOW})	0.5
Threshold f_{Max}	0.2
Termination-criteria parameters	
Maximum number of populations (G_{Max})	1000
Maximum number of populations (G_{Fit})	200

The evaluation results of the proposed method are shown in table II. The first row corresponds to the results obtained using the first dataset while the second row corresponds to the results obtained from the second dataset. As it can be easily observed, the proposed method determines the grid structure almost perfectly, irrespective of which dataset was used. In details, more than 94% of the microarray spots have been perfectly gridded and only a small percentage of them, less than 1%, have been incorrectly gridded.

TABLE II
PERFORMANCE OF THE PROPOSED GRIDDING METHOD

	Perfect %	Marginal%	Incorrect%
1st dataset	94.6	4.8	0.6
2nd dataset	94.4	5.1	0.5

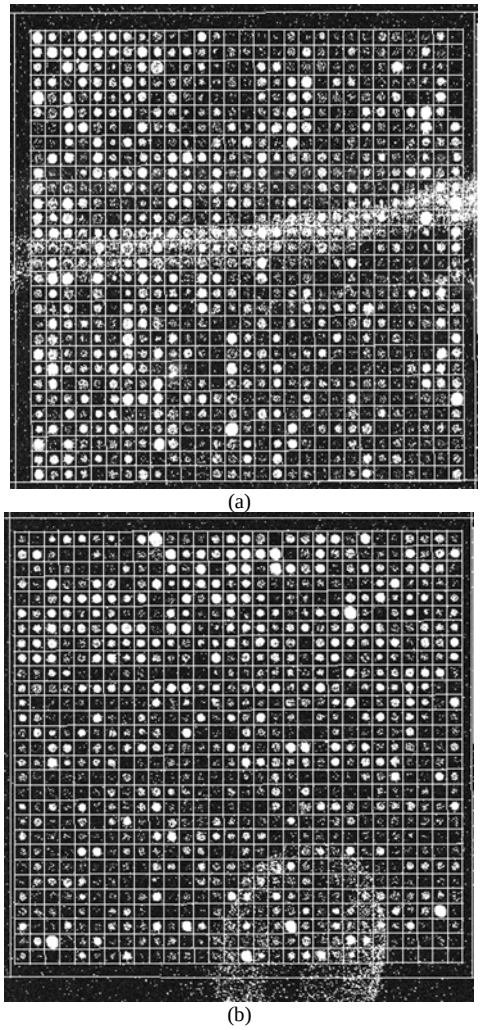
The evaluation results of the proposed method have been also compared to the ones reported by Blekas et al. [30]. To this extent, we appose the results reported by Blekas et al. to table III. Comparing the results of the proposed method using the 2nd dataset (2nd row of the table II) with the results reported by Blekas et al. (table III), it is obvious that the proposed method outperforms the method proposed by Blekas

et al. Moreover, the results of the proposed method are significantly more successful than the ones of ScanAlyze and SpotFinder software programs.

TABLE III
PERFORMANCE OF GRIDDING METHODS BY BLEKAS ET AL.[30]

Gridding methods	Perfect %	Marginal %	Incorrect %
Method proposed by Blekas et al.	89.6	9.2	1.2
ScanAlyze	48.7	22.6	28.7
SpotFinder	72.8	14.3	12.9

Fig. 7 represents the gridding results of two microarray blocks using the proposed method. Each block belongs to the first dataset. Both of them are contaminated with noise, and they contain uneven background. It is worth noticing that the proposed method can accurately determine the grid structure. Even the compartments of the grid which bound microarray spots fully-contaminated with noise, are very effectively depicted.



(a)

(b)

Fig. 7. Gridding results in two blocks contaminated with noise.

Fig. 8 depicts the gridding results obtained by applying the proposed method to one of the blocks of the 2nd dataset. It is evident that although the block contains several under-expressed and high-expressed spots, the gridding results are almost perfect. This example indicates that the accuracy of the proposed method is not influenced by spot intensities and sizes.

To evaluate the gridding method when rotation exists, we rotated the microarray images by seven degrees and we applied the proposed method. Fig. 9 presents the gridding result of a rotated sub-image. It is demonstrated that the proposed method can very efficiently determine the rotated grid structure.

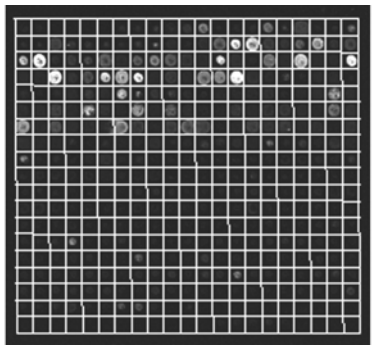


Fig. 8. Gridding results in a microarray block which contains several under expressed and high expressed spots.

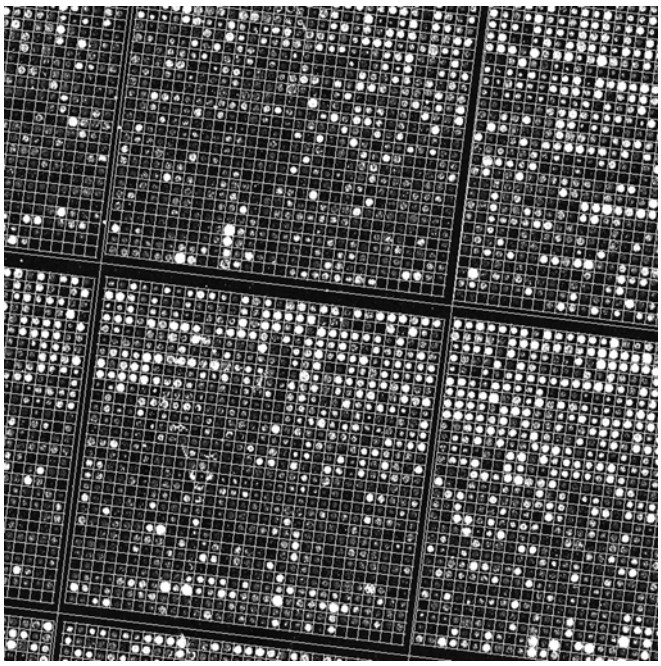


Fig. 9. Gridding results in a rotated microarray sub-image.

Fig. 10a presents the gridding results of a microarray image which contains four blocks which are misaligned. The misalignment can be easily observed in Fig. 10b which depicts an enlargement of the central part of the image shown in Fig. 10a. The ideal rectangular grid is deformed since the edge-to-edge distance between the two top blocks is smaller than the edge-to-edge distance between the two bottom blocks. However, the proposed method has very effectively determined the grid structure.

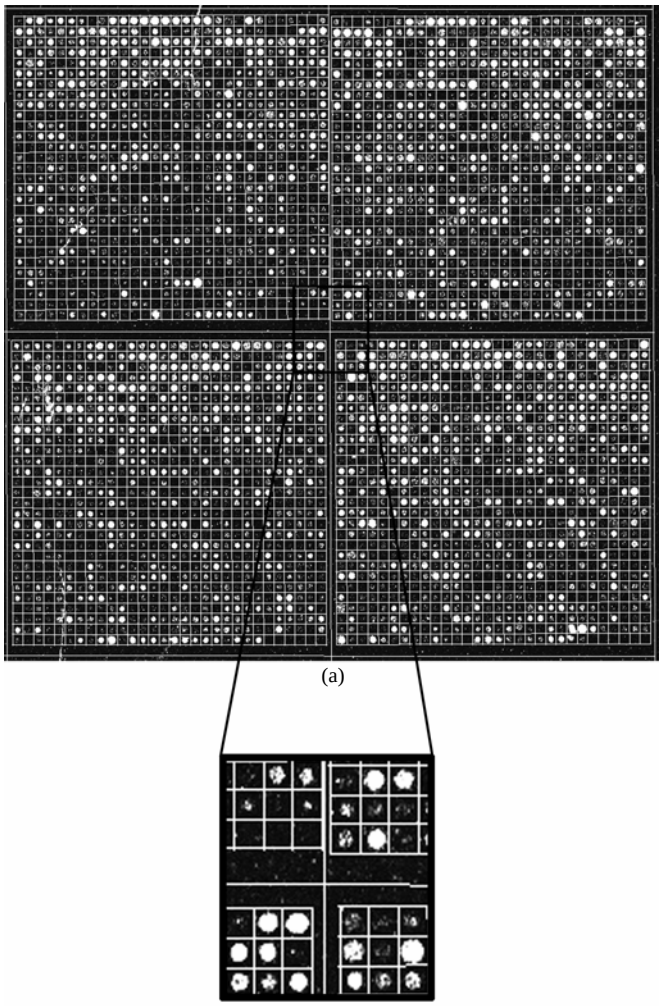


Fig. 10. (a) Gridding results in four adjacent misaligned microarray blocks of a microarray image. (b) Enlargement of the central part of the microarray sub-image.

V. Conclusions

Gridding is the first important stage in microarray image analysis. In this paper, the gridding procedure is expressed as an optimization problem which is tackled by using a Genetic Algorithm, which determines the line-segments constituting the grid structure. The proposed method can very efficiently deal with various kinds of perturbations such as arbitrary rotations, local deformations and missing spots. It is also noise-resistant and it is efficient even under adverse conditions such as the appearance of various spot sizes or the absence of spots. Last but not least, it is fully-automatic since it does not require any input parameter or human intervention in order to adjust properly the grid structure. The experimental results over synthetic and real images demonstrate that it is very efficient and effective. It outperforms the existing software program methods as well as recently published techniques. After applying it to several images containing 1,040,300 microarray spots, the proposed method achieved an accuracy of more than 94%. To our knowledge, this percentage is much higher than the ones obtained from state-of-the-art gridding techniques.

Acknowledgment

The authors would like to extend their gratitude to N. P. Galatsanos, K. Blekas, A. Likas, and I. E. Lagaris for providing the microarray images of the second dataset used for the evaluation of the proposed gridding method.

References

[1] Y. F. Leung and D. Cavalieri, “Fundamentals of cDNA microarray data analysis,” *Trends in Genetics*, vol. 19, no. 11, pp. 649-659, Nov. 2003.

[2] E. K. Lobenhofer, P. R. Bushel, C. A. Afshari, and H. K. Hamadeh, “Progress in the Application of DNA Microarrays,” *Enviromental Health Perspectives*, vol. 109, no. 9, pp. 881–891, Sept. 2001.

[3] A. M. Campbell and L. J. Heyer, *Discovering Genomics, Proteomics & Bioinformatics*, 2nd ed., Pearson Benjamin Cummings, 2007, pp. 233-238.

[4] W. B. Chen, C. Zhang, and W. L. Liu, “An Automated Gridding and Segmentation Method for cDNA Microarray Image Analysis,” in *Proc. 19th IEEE Symp. Computer-Based Medical Systems*, Salt Lake City, 2006, pp. 893-898.

[5] Y. Tu, G. Stolovitzky, and U. Klein, “Quantitative noise analysis for gene expression microarray experiments,” in *Proc. National Academy of sciences*, USA, 2002, pp.14031-14036.

[6] M. B. Eisen. (1999). (a) ScanAlyze. [Online]. Available: http://rana.lbl.gov/EisenSoftware.htm

[7] J. Buhler, T. Ideker, and D. Haynor, “Dapple: improved techniques for finding spots on DNA microarrays,” UW CSE Technical Report UWTR 2000-08-05, pp. 1-12, Aug. 2000.

[8] Biodiscovery Inc. (2005). ImaGene. [Online]. Available: http://www.biodiscovery.com/imagene.asp

[9] P. Hegde et al., “A concise guide to cDNA microarray analysis,” *Biotechniques*, vol. 29, no. 3, pp. 548–562, Sept. 2000.

[10] N. D. Lawrence, M. Milo, M. Niranjan, P. Rashbass, and S. Soullier, “Reducing the variability in cDNA microarray image processing by Bayesian inference”, *Bioinformatics*, vol. 20, no. 4, pp. 518–526, Mar. 2004.

[11] N. Deng and H. Duan, “The Automatic Gridding Algorithm based on projection for Microarray Image”, in *Proc. Int. Conf. Intelligent Mechatronics and Automation*, Chendu, 2004, pp. 254-257.

[12] A. W. C. Liew, H. Yan, and M. Yang, “Robust adaptive spot segmentation of DNA microarray images,” *Pattern Recognition*, vol. 36, no. 5, pp. 1251–1254, May 2003.

[13] L. Rueda and V. Vidyadharan, “A Hill-Climbing Approach for Automatic Gridding of cDNA Microarray Images,” *IEEE/ACM Trans. Comp. Biology and Bioinformatics*, vol. 3, no. 1, pp. 72-83, Jan. 2006.

[14] G. Antoniol and M. Ceccarelli, “A markov random field approach to microarray image gridding,” in *Proc. 17th Int. Conf. Pattern Recognition*, Cambridge, 2004, pp. 550-553.

[15] H. Y. Jung and H. G. Cho, “An automatic block and spot indexing with k-nearest neighbors graph for microarray image analysis,” *Bioinformatics*, vol. 18, no.1, pp. 141–151, Oct. 2002.

[16] P. Bajcsy, “Gridline: automatic grid alignment in DNA microarray scans,” *IEEE Trans. Image Processing*, vol. 13, no. 1, pp. 15–25, Jan. 2004.

[17] M. Steinfath, W. Wruck, H. Seidel, H. Lehrach, U. Radelof, and J. O’Brien, “Automated image analysis for array hybridization experiments,” *Bioinformatics*, vol. 17, no.7, pp. 634–641, Jul. 2001.

[18] N. Brandle, H. Bischof, and H. Lapp, “Robust DNA Microarray image analysis,” *Machine Vision and Applications*, vol. 15, no.1, pp. 11–28, Oct. 2003.

[19] J. Ho, W. L. Hwang, H. H. S. Lu, D. T. Lee, “Gridding Spot Centers of smoothly distorted microarray images,” *IEEE Trans. Image Processing*, vol. 15, no. 2, pp. 342-353, Feb. 2006.

[20] G. Antoniol and M. Ceccarelli, "Microarray image gridding with stochastic search based approaches," *Image and Vison Computing*, vol. 25, no. 2, pp. 155-163, Feb.2007.
[21] D. E. Goldberg, *Genetic Algorithms in Search, Optimization & Machine Learning*, Boston: Addison-Wesley, Reading, 1989, ch. 1.
[22] F. Herrera, M. Lozano, and J. L. Verdegay, "Tackling Real Coded Genetic Algorithms: Operators and Tools for Behavioural Analysis," *Artificial Intelligence Review*, vol. 12, no. 4, pp. 265–319, Nov. 1998.
[23] C. S. M. Hayes, "Generic Properties of the Infinite Population Genetic Algorithm," Ph.D. dissertation, Dept. Mathematics, Montana State Univ., Bozeman, Montana, USA, 2006.
[24] B. L. Miller, and D. E. Goldberg, "Genetic Algorithms, Tournament selection, and the Effects of Noise," Complex Systems, vol. 9, no. 3, pp. 193-212, 1995.
[25] F. Herrera, M. Lozano, and A. M. Sanchez, "Hybrid crossover operators for real-coded genetic algorithms: An experimental study," *Soft Computing*, vol. 9, no. 4, pp. 280-298, Apr. 2005.
[26] S. H. Ling, and F. H. F. Leung, "An improved genetic algorithm with average-bound crossover and wavelet mutation operations," *Soft Computing*, vol. 11, no. 1, pp. 7-31, 2007.
[27] C. T. Ekstrom, S. Bak, C. Kristensen and M. Rudemo, "Spot shape modelling and data transformations for microarrays", *Bioinformatics*, vol. 20, no. 14, pp. 2270-2278, Sep. 2004.
[28] Standford Microarray Database. [Online]. Available: http://genome-www5.stanford.edu/
[29] D. Juric *et al.*, "Differential Gene Expression Patterns and Interaction Networks in BCR-ABL-Positive and -Negative Adult Acute Lymphoblastic Leukemias," *Journal of Clinical Oncology*, vol. 25, no. 11, pp. 1341-1349, April 2007.
[30] K. Blekas, N. P. Galatsanos, A. Likas, and I. E. Lagaris, "Mixture Model Analysis of DNA Microarrray Images," *IEEE Trans. Medical Imaging*, vol. 24, no. 7, pp. 901-909, July 2005.
[31] S. Achiche, L. Baron, and M. Balazinski, "Real/binary-like coded versus binary coded genetic algorithms to automatically generate fuzzy knowledge bases: a comparative study," *Engineering Applications of Artificial Intelligence*, vol. 17, no. 4, pp. 313-325, 2004.
[32] M. T. Miller, A. K. Jerebko, J. D. Malley, and R. M. Summers, "Feature selection for computer-aided polyp detection using genetic algorithms," *in Proc. of SPIE*, Santa Clara, 2003, pp. 102-110.
[33] C. Z. Janikow, Z. Michalewicz, "An experimental comparison of binary and floating point representations in genetic algorithms," in *Proc. 4th Int. Conf. Genetic Algorithms*, San Diego, 1991, pp. 31–6.