Gridding Spot Centers of Smoothly Distorted Microarray Images

Jinn Ho, Wen-Liang Hwang, Henry Horn-Shing Lu, and D. T. Lee, *Fellow, IEEE*

***Abstract*—We use an optimization technique to accurately locate a distorted grid structure in a microarray image. By assuming that spot centers deviate smoothly from a checkerboard grid structure, we show that the process of gridding spot centers can be formulated as a constrained optimization problem. The constraint is equal to the variations of the transform parameter. We demonstrate the accuracy of our algorithm on two sets of microarray images. One set consists of some images from the Stanford Microarray Database; we compare our centers with those annotated in the Database. The other set consists of oligonucleotide images, and we compare our results with those obtained by GenePix Pro 5.0. Our experiments were performed completely automatically.**

***Index Terms*—Microarray image, spot gridding.**

I. Introduction

THE NEW technique of DNA microarray imaging provides a systematic method that can simultaneously measure gene expression levels of the genomic scale. Because of its high throughput capacity, the technique has great potential for biological, medical, and industrial applications. Two of the major categories of microarrays are cDNA [7]–[9], [25], and oligonucleotide microarrays [19], [20]. Specific cDNA, or oligonucleotide fragments of genes that are of interest, are spotted or printed on an array matrix as probes to detect gene expression. Samples of mRNAs are then reverse transcribed to cDNAs, which are labeled with fluorescent dyes to act as targets. The labeled cDNA targets are hybridized to probes by complementation, and the unhybridized targets are then washed out. A laser scanner detects the fluorescent intensities in proportion to the contents of the hybridized pairs of targets and probes. This generates microarray images with pixel intensities for various samples, indicating the relative expression levels of the genes. Finally, image processing techniques and statistical methods are applied to determine the expression levels of the spots of microarrays in order to perform gene expression analysis [6], [27].

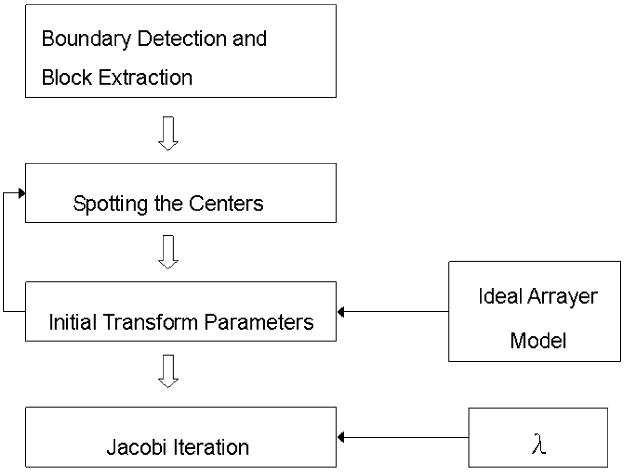

Boundary Detection and Block Extraction
Spotting the Centers
Initial Transform Parameters
Ideal Arrayer Model
Jacobi Iteration
λ

Fig. 1. Block diagram of our approach. We use an ideal-arrayer model and multithresholds to find the initial solution and solve the optimization problem by the Jacobi iteration that uses the parameter λ.

An important first step in gene expression analysis is detecting the position of a spot center, and labeling its corresponding coordinate in an micro-arrayer [9], [21]. This is called the spot gridding problem [5], [16], [30]. Even though micro-arrayers arrange spots on a relatively regular checkerboard grid, spotting error irregularities that occur during the array manufacturing process makes accurate gridding of spot centers difficult. Deviations from microarray regularities are attributable to different causes, such as center-to-center spacing deviations of an arrayer, varied surface properties of the substrate, and imprecise movement of manufacturing devices [26]. Spots can also vary in size and position due to noise in the sample preparation and hybridization processes [28]. Dealing with spot center variations is the principal source of complexity in solving the gridding problem.

Some image analysis softwares for spot gridding found in the Stanford Microarray Database (SMD) are ScanAlyze [24], GenePix [11], and Koadarray [15]. These softwares require parameters and, at times, manual intervention to locate exact spot centers. In GenePix, for example, a user must assign the layout of blocks, the number of spots in each block, and the distances of spot centers between adjacent rows and columns. An interactive graphic-based environment may be provided to adjust the location of blocks before gridding spot centers. Meanwhile, Koadarray only uses two parameters, namely, the block number and spot number in each block. Although simpler than GenePix, it produces weak results for slightly rotated images.

Manuscript received February 1, 2004; revised February 12, 2005. This work was supported in part by the National Science Council, Taiwan, R.O.C., under Grants NSC-93-2213-E-001-013, NSC-92-3112-B-001-018-Y, NSC-92-3112-B-001-021-Y, NSC 93-2752-E-002-005-PAE, NSC 93-2213-E-001-009 NSC-93-2118-M-009-001, and NSC-93-3112-B-001-021-Y. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Attila Kuba.

J. Ho and D. T. Lee are with the Institute of Information Science, Academia Sinica, Taipei, Taiwan, R.O.C., and also with The Genomics Research Center, Academia Sinica, Taiwan, R.O.C. (e-mail: hjinn@math.ntu.edu.tw; dtlee@iis.sinica.edu.tw).

W.-L. Hwang is with the Institute of Information Science, Academia Sinica, Taipei, Taiwan, R.O.C. (e-mail: whwang@iis.sinica.edu.tw).

H. H.-S. Lu is with the Institute of Statistics, National Chiao-Tung University, Taiwan, R.O.C. (e-mail: hslu@stat.nctu.edu.tw).

Digital Object Identifier 10.1109/TIP.2005.860610

As an improvement to methods that require parameters and manual intervention, we propose an algorithm that is accurate and fully automatic. This is very hard to achieve without imposing some constraints on the possible solution. Thus, we only specify one constraint, which assumes that spot centers deviate from a sequence of similarity transformations whose parameters vary smoothly. With this constraint, we can formulate the spot center gridding problem as a constrained optimization problem by combining a quantitative criterion that measures gridding result correctness with a constraint that reduces local parameter variation. The problem can be solved numerically by an iterative algorithm.

In the block diagram of our approach shown in Fig. 1, the ideal arrayer block is a checkerboard grid, where the number of spots in a block is known. We begin with block boundary detection to extract the block layout. A Bayesian approach, based on a multithreshold Markov model [4], is combined with a model-based recognition method to successively refine the spot centers from a sequence of thresholds. This provides the initial transform parameters of a subblock. If the parameters of the subblock are inconsistent with those of its neighboring subblocks, we then use a tree-based algorithm to correct the subblock's parameters so that robust initial transform parameters of each subblock can be obtained. After obtaining the initial transform parameters, the Lagrange multiplier λ controls the balance between the correctness of the estimated transform parameters and the smoothness of the transforms in the final solution.

Of the many approaches that can be used to solve the spot gridding problem, graph matching [1] and Bayesian grid matching [12] find the best solution, the former by dynamic programming, and the latter by simulated annealing. However, we have found that these approaches are too complex for our goal. Our proposed method is based on an optical flow estimation approach [14] in which a template structure is deformed to numerically estimate the deformation.

We illustrate our results using images manufactured with different techniques and different signal-to-noise (SNR) levels. Note that our resultant images are obtained completely automatically from the experimental images. That is to say, we use the same set of parameters for all experiments. In Section II, the spot addressing problem is formulated as a constrained optimization problem and solved numerically. Section III presents a method to obtain a robust initial solution. In Section IV, the accuracy of our method is illustrated. Finally, in Section V, we present our conclusions.

II. Gridding Problem

The first step of microarray data analysis is spot gridding, which identifies the spot centers of a microarray image. Many approaches for solving the gridding problem require parameters, as well as human intervention. The usual parameters are: the block layout structure, the width and height of each block, and the distances of spot centers between adjacent rows and columns. Even if these parameters are given, human intervention is still needed to adjust overly-deviated spot centers.

Analyzing the causes of deviations and creating a fully automatic system is difficult. A reasonable approach is to assume that spot center deviations can be modeled as smoothly varying transformations, which can be characterized by a few local parameters. Let $\mathbf{x}$ and $\mathbf{y}$ be the coordinates of a pair of matched centers in an ideal micro-arrayer model and an image, respectively. We use $\mathcal{T}_{\mathbf{x},\mathbf{y}}$ to represent the transformation from $\mathbf{x}$ to $\mathbf{y}$ and assume that the transformation can be approximated by a similarity transform. Thus, $\mathbf{x}$ and $\mathbf{y}$ are related by the matrix form

$$\mathbf{y} = \mathcal{T}_{\mathbf{x},\mathbf{y}}(\mathbf{x}) \approx A(\mathbf{x},\mathbf{y})\mathbf{x} + \mathbf{b}(\mathbf{x},\mathbf{y})$$

where

$$\begin{aligned} A(\mathbf{x},\mathbf{y}) &= \begin{bmatrix} a(\mathbf{x},\mathbf{y}) & -b(\mathbf{x},\mathbf{y}) \\ b(\mathbf{x},\mathbf{y}) & a(\mathbf{x},\mathbf{y}) \end{bmatrix} \\ &= s(\mathbf{x},\mathbf{y}) \begin{bmatrix} \cos\theta(\mathbf{x},\mathbf{y}) & -\sin\theta(\mathbf{x},\mathbf{y}) \\ \sin\theta(\mathbf{x},\mathbf{y}) & \cos\theta(\mathbf{x},\mathbf{y}) \end{bmatrix} \end{aligned} \tag{1}$$

and

$$\mathbf{b}(\mathbf{x},\mathbf{y}) = \begin{bmatrix} c(\mathbf{x},\mathbf{y}) \\ d(\mathbf{x},\mathbf{y}) \end{bmatrix} \tag{2}$$

is a translation matrix. The parameters of a similarity transform are a scaling factor $s(\mathbf{x},\mathbf{y})$, a rotation angle $\theta(\mathbf{x},\mathbf{y})$, and a translation vector $\mathbf{b}(\mathbf{x},\mathbf{y})$. We denote the centers in the model and image as $\{\mathbf{x}\}$ and $\{\mathbf{y}\}$, respectively, and use $[\mathbf{x},\mathbf{y}]$ to denote that $\mathbf{x}$ and $\mathbf{y}$ are a pair of matched centers. Because the structure of an ideal arrayer is known, we can obtain all points of the arrayer from $\{\mathbf{x}\}$. However, as $\{\mathbf{y}\}$ is obtained by processing a microarray image, it may not contain all the spot centers. These two sets may, therefore, have different numbers of elements. Thus, we use the active set $\mathcal{A}\{\mathbf{x}\}$ of $\{\mathbf{x}\}$ to represent the largest subset of $\mathbf{x}$, in which any element only has a matched center in $\{\mathbf{y}\}$. The mean-squared error of all matched centers $\{[\mathbf{x},\mathbf{y}]\}$ is

$$e_e(\{[\mathbf{x},\mathbf{y}]\}) = \frac{1}{|\mathcal{A}\{\mathbf{x}\}|} \sum_{\mathbf{x}\in\mathcal{A}\{\mathbf{x}\}} ||A(\mathbf{x},\mathbf{y})\mathbf{x} + \mathbf{b}(\mathbf{x},\mathbf{y}) - \mathbf{y}||^2 \tag{3}$$

where $|\mathcal{A}\{\mathbf{x}\}|$ is equal to the number of all matched pairs. Usually, the distortion $\mathcal{T}_{\mathbf{x},\mathbf{y}}$ varies smoothly in an image. Thus, we can impose a smoothness constraint by minimizing the variations of $A(\mathbf{x},\mathbf{y})$ and $\mathbf{b}(\mathbf{x},\mathbf{y})$. Let us assume that $\mathbf{y}_i$ and $\mathbf{y}_j$ are paired with $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively. If $\mathbf{x}_i$ and $\mathbf{x}_j$ are neighboring grids, then according to the smoothness constraint, the parameters of $A(\mathbf{x}_i,\mathbf{y}_i)$, $\mathbf{b}(\mathbf{x}_i,\mathbf{y}_i)$ and $A(\mathbf{x}_j,\mathbf{y}_j)$, $\mathbf{b}(\mathbf{x}_j,\mathbf{y}_j)$ should have similar values. We use $\mathcal{N}(\mathbf{x}_i)$ to denote the set of neighbors of $\mathbf{x}_i$ in the active set. A simple measurement of the smoothness of the parameters is

$$\begin{aligned} e_s(\{[\mathbf{x},\mathbf{y}]\}) = \frac{2}{|\mathcal{A}\{\mathbf{x}\}|} &\sum_{\mathbf{x}_i\in\mathcal{A}\{\mathbf{x}\}} \frac{1}{|\mathcal{N}\{\mathbf{x}_i\}|} \\ &\cdot \Bigg[\sum_{\mathbf{x}_j\in\mathcal{N}\{\mathbf{x}_i\}} ||A(\mathbf{x}_i,\mathbf{y}_i) - A(\mathbf{x}_j,\mathbf{y}_j)||_F^2 \\ &\quad + ||\mathbf{b}(\mathbf{x}_i,\mathbf{y}_i) - \mathbf{b}(\mathbf{x}_j,\mathbf{y}_j)||^2 \Bigg] \end{aligned} \tag{4}$$

where $||.||_F$ is the Frobenius norm. The Frobenius norm of matrix B is defined as $\sqrt{\sum_i \sum_j |b_{i,j}|^2}$, where $b_{i,j}$ is an element in matrix B. According to (3) and (4), we need to find the set of matched pairs between $\{\mathbf{x}\}$ and $\{\mathbf{y}\}$ that minimizes $e_e + \lambda e_s$, where λ is a nonnegative parameter that weights the matched pair error relative to the departure from smoothness of the transform parameters.

A. Numerical Solution

To find a numerical solution of $e_e + \lambda e_s$, we use a finite-element method because it is easy to implement and achieves a satisfactory solution. Let $\mathbf{x}$ and $\mathbf{y}$ be paired, with $\mathbf{x}$ in the active set $\mathcal{A}\{\mathbf{x}\}$. If the coordinate of $\mathbf{x}$ is $[x_1(k,l)\ x_2(k,l)]^T$ and the coordinate of $\mathbf{y}$ is $[y_1(k,l)\ y_2(k,l)]^T$, then—to simplify the formulation of a numerical method—we denote $\mathbf{x}$ as $\mathbf{x}_{k,l}$, and $\mathbf{y}$ as $\mathbf{y}_{k,l}$; the parameters in $A(\mathbf{x}_{k,l}, \mathbf{y}_{k,l})$ as $a_{k,l}$ and $b_{k,l}$; and the parameters in $\mathbf{b}(\mathbf{x}_{k,l}, \mathbf{y}_{k,l})$ as $c_{k,l}$ and $d_{k,l}$. The mean-squared error measurement in (3) is, therefore

$$e_e = \frac{1}{|\mathcal{A}\{\mathbf{x}\}|} \sum_{\mathbf{x}_{k,l} \in \mathcal{A}\{\mathbf{x}\}} D_e(\mathbf{x}_{k,l})$$

where $|\mathcal{A}\{\mathbf{x}\}|$ is the size of the active set and $D_e(\mathbf{x}_{k,l}) = \{(a_{k,l}x_1(k,l) - b_{k,l}x_2(k,l) + c_{k,l} - y_1(k,l))^2 + (b_{k,l}x_1(k,l) + a_{k,l}x_2(k,l) + d_{k,l} - y_2(k,l))^2\}$.

We also use $\mathcal{N}(\mathbf{x}_{k,l})$ to denote the set of neighbors of $\mathbf{x}_{k,l}$ in the active set. In the system of first-order neighbors, a neighbor of $\mathbf{x}_{k,l}$ in the active set will be either $\mathbf{x}_{k+1,l}$, $\mathbf{x}_{k-1,l}$, $\mathbf{x}_{k,l+1}$, or $\mathbf{x}_{k,l-1}$ whose coordinates are, respectively

$$\begin{bmatrix} x_1(k+1,l) \\ x_2(k+1,l) \end{bmatrix}, \quad \begin{bmatrix} x_1(k-1,l) \\ x_2(k-1,l) \end{bmatrix}, \quad \begin{bmatrix} x_1(k,l+1) \\ x_2(k,l+1) \end{bmatrix} \quad \text{or} \begin{bmatrix} x_1(k,l-1) \\ x_2(k,l-1) \end{bmatrix}.$$

Equation (4) then becomes

$$e_s = \frac{2}{|\mathcal{A}\{\mathbf{x}\}|} \sum_{\mathbf{x}_{k,l} \in \mathcal{A}\{\mathbf{x}\}} \frac{1}{|\mathcal{N}\{\mathbf{x}_{k,l}\}|} \sum_{\mathbf{x}_{k+i,l+j} \in \mathcal{N}\{\mathbf{x}_{k,l}\}} D_s(\mathbf{x}_{k,l}) \tag{5}$$

where $i,j \in \{-1,1\}$ and $D_s(\mathbf{x}_{k,l}) = [2(a_{k,l} - a_{k+i,l+j})^2 + 2(b_{k,l} - b_{k+i,l+j})^2 + (c_{k,l} - c_{k+i,l+j})^2 + (d_{k,l} - d_{k+i,l+j})^2]$. We need $\{a_{k,l}\}$, $\{b_{k,l}\}$, $\{c_{k,l}\}$, and $\{d_{k,l}\}$ to minimize

$$e = e_e + \lambda e_s. \tag{6}$$

To solve this, we differentiate e with respect to $a_{k,l}$, $b_{k,l}$, $c_{k,l}$, and $d_{k,l}$ and set the derivatives to zero. The resultant equations are formed as a matrix representation and can be solved by the Jacobi iterative scheme. In Appendix I, we present the Jacobi iterative method for the optimal solution of the parameters $a^*_{k,l}$, $b^*_{k,l}$, $c^*_{k,l}$, and $d^*_{k,l}$. The final quality of the Jacobi iterative solution depends on the quality of the initial solution. In Section III, we propose a method that finds a robust initial solution.

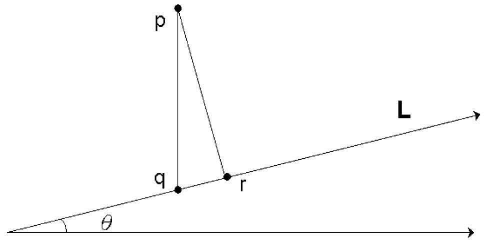


Fig. 2. For a small θ, the distance from p to L is approximated as the vertical distance between p and q.

III. Finding a Robust Initial Solution

We use a sequence of robust image processing methods for a more automated process of finding effective initial transform parameters.

A. Boundary Detection and Block Extraction

In a microarray, spots are grouped into blocks, and each block must be delineated in order to identify the spot centers within the blocks. In [10], the blocks of a microarray are delineated from the vertical and horizontal projection profiles of the image. This method works well, provided that the microarray image has no rotation deviation. We propose a method that can extract the boundaries of a slightly rotated image. From the projection profiles at the perpendicular direction to each boundary, the blocks of the rotated image can be separated from each other.

Boundary Detection: To extract the boundaries of a slightly rotated image, we use the line equation $y = x \cdot A_H + B_H$ to represent the top or bottom boundary. We only discuss the method to find either the top or bottom boundary, as the left and right boundaries can be found by a similar method using the line equation $x = y \cdot A_V + B_V$.

Fig. 2 shows that the distance d_L from a point (i,j) to a line $L : y = x \cdot A_H + B_H$ can be approximated as the difference of the y-coordinates $d_L \approx i - (j \cdot A_H + B_H)$, as long as there is only a small rotation angle. To find the boundary line L that is tangent to the edges of spot centers located at a boundary, we use the Gaussian-like weighting function $W(d_L) = W(i,j) \cdot \exp(-{d_L}^2)$, in which $W(i,j) = |\partial I(i,j)/\partial i|$ and I is the image. This function gives more weight to a pixel that is closer to line L, or to one that has a greater absolute intensity gradient. To find the line, we look for the A_H and B_H that minimize the weighted-squared error

$$Err_H(A_H, B_H) = \sum_{i,j} W(d_L) \cdot ({d_L}^2).$$

By differentiating the above equation with respect to A_H and B_H and setting the results to zero, we have

$$\begin{aligned} \frac{\partial Err_H}{\partial A_H} &= \sum_{i,j} \frac{\partial W(d_L)}{\partial A_H} \cdot (d_L)^2 + (-2j) \cdot W(d_L) \cdot (d_L) \\ &= 0 \\ \frac{\partial Err_H}{\partial B_H} &= \sum_{i,j} \frac{\partial W(d_L)}{\partial B_H} \cdot (d_L)^2 + (-2) \cdot W(d_L) \cdot (d_L) \\ &= 0. \end{aligned}$$

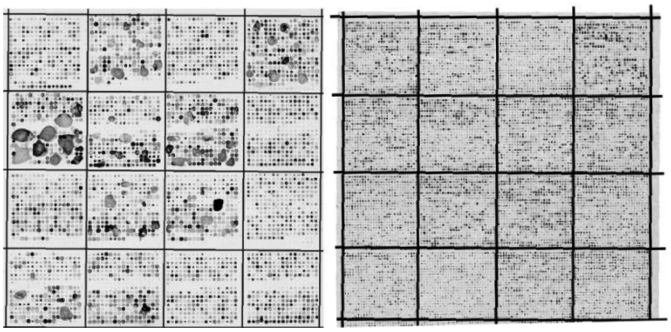


Fig. 3. Blocks in each image are enclosed by a rectangular bounding box. The left image (size 1 824 × 1 828 pixels) is hp7005a, while the right image (size 1 896 × 2 032 pixels) is the block detection result of an image of LC23N085 rotated 1.5°.

Since $\partial W(d_L)/\partial A_H = (-2j)d_L \cdot W(d_L)$ and $\partial W(d_L)/\partial B_H = (-2)d_L \cdot W(d_L)$, these equations can be simplified to

$$\sum_{i,j} jW(d_L) \cdot d_L \left\{{d_L}^2 + 1\right\} = 0 \text{ and}$$
$$\sum_{i,j} W(d_L) \cdot d_L \left\{{d_L}^2 + 1\right\} = 0.$$

If a point is close to line L, then we have $d_L \to 0$. This means that $\left({d_L}^2 + 1\right)$ can be approximated as $\exp\left({d_L}^2\right)$. If we denote $\sum_L$ to be the summation of all points near L, we have the approximations of the above two equations, which are

$$\sum_L jW(i,j) \cdot d_L = 0 \text{ and}$$
$$\sum_L W(i,j) \cdot d_L = 0.$$

The solutions of the above equations are

$$A_H = \left(\frac{1}{d}\right) \cdot \left\{\sum_L jW(i,j) \sum_L iW(i,j) - \sum_L W(i,j) \cdot \sum_L j \cdot i \cdot W(i,j)\right\} \text{ and} \tag{7}$$

$$B_H = \left(\frac{1}{d}\right) \cdot \left\{\sum_L jW(i,j) \sum_L j \cdot i \cdot W(i,j) - \sum_L j^2 W(i,j) \cdot \sum_L iW(i,j)\right\} \tag{8}$$

where $d = \left(\sum_L jW(i,j)\right)^2 - \sum_L W(i,j) \cdot j^2 \sum_L W(i,j)$.

Block Extraction: After extracting four boundary lines, we slightly modify them such that they form a rectangular box. To extract blocks, we project along each boundary line and select the blocks from the projection profile, as described in [10]. Fig. 3 shows the block extraction results of two SMD images.

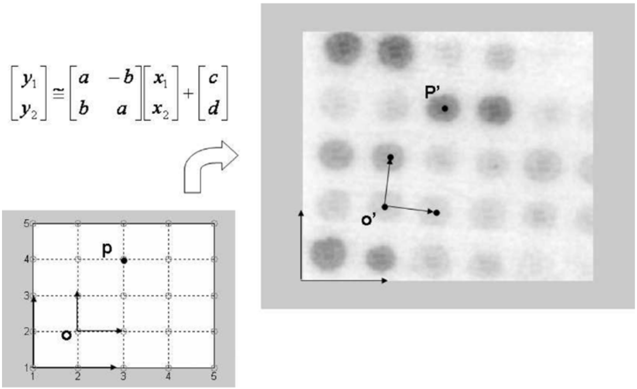

$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \cong \begin{bmatrix} a & -b \\ b & a \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} c \\ d \end{bmatrix}$
P'
O'
p
o
1
2
3
4
5
1
2
3
4
5

Fig. 4. Coordinate frame O is related to the frame O' by a similarity transformation. The parameters of the transformation are obtained from the corresponding coordinate frames and the centers of the frames. The coordinates of points P and P' are the same with respect to frames O and O'.

B. Initial Distortion Estimation

After the blocks are delineated, we estimate an initial distortion of the spot centers in a block. A good initial estimation of block distortion is obtained by dividing a block into subblocks, and assuming that the transform within a subblock is the same anywhere in that subblock. This allows us to efficiently apply the geometric hashing algorithm to obtain transform parameters. There is a tradeoff between the solutions of geometric hashing and computation time, i.e., a subblock with more spots yields a better result at the expense of higher computation time. Experiments show that the geometric hashing algorithm estimates acceptable transform parameters of a subblock if the size of the subblock contains 16 to 64 spots.

Geometric Hashing to Find Matched Pairs in a Subblock: We assume that local distortion within a subblock can be approximated by a similarity transform. That is, model point $\mathbf{x}$ and subblock point $\mathbf{y}$ are related by the matrix transformation

$$\mathbf{y} \approx \begin{bmatrix} a & -b \\ b & a \end{bmatrix} \mathbf{x} + \begin{bmatrix} c \\ d \end{bmatrix}. \tag{9}$$

Geometric hashing can find the parameters of this similarity transform between the model points and subblock points, according to an invariant property. We define a frame from two model points, which form a basis, and assign the coordinate $[0\ 0]^T$ to one point and $[1\ 0]^T$ to the other. The coordinates of all other points with respect to the same basis will be preserved after applying any similarity transform to the points. Fig. 4 shows invariant coordinates after a similarity transform is applied to the points in the left subfigure. In this way, if model points and subblock points are related by a similarity transform, we can derive the parameters of the similarity transform from the matched basis in the model and subblock. A detailed discussion of geometric hashing can be found in [18], [23], [29].

Note that in order to estimate the transform parameters with the geometrical hashing algorithm, we need to find subblock points $\mathbf{y}$ first. These points are obtained by thresholding a subblock image so that it yields black and white pixels. The black pixels that are connected to each other form disjoint connected

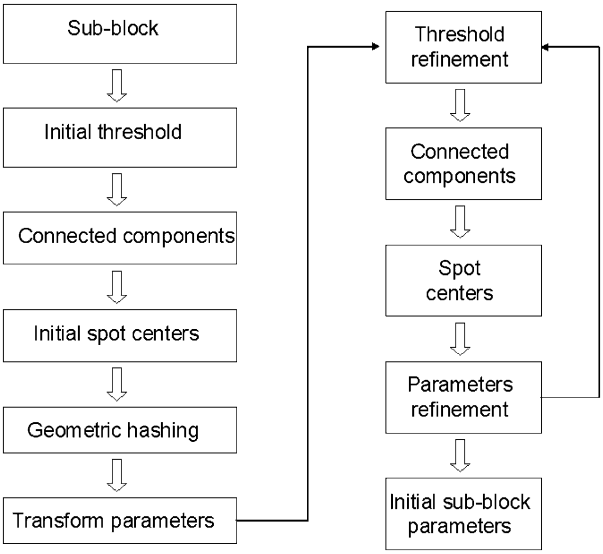


Fig. 5. Acquisition of subblock transform parameters.

components. After removing connected components that are unlikely to be spots, the centers of the resultant components are the subblock points $\mathbf{y}$. The algorithm for finding the connect components and their centers is presented in Appendix II.

Gridding Centers Using a Multithreshold Markov Model: Because there are various signal intensities and noise levels in a microarray, using a threshold to distinguish signals from noise may yield either a spot pattern with insufficient signal information, or one with too much noise information. Thus, we use a simple multithreshold approach on a Markov model to locate spots.

We begin with a coarse threshold to binarize a subblock image, compute the connected components of the resultant image, and find the center of each component. The transform between the centers of the model and the subblock image is obtained by the geometric hashing algorithm. We then refine the threshold and repeat the above procedure, but replace the geometric hashing algorithm with a Bayesian approach that uses a Markov model to refine the initial transform parameters of the subblock.

The block diagram for finding the initial subblock parameters is shown in Fig. 5. To illustrate the process, let Y be a subblock image, X be the set of model points, and $Y(\tau_i)$ represent the set of spot centers detected from the subblock Y, using a gray scale threshold level τ_i. The spot centers $Y(\tau_i)$, detected by using threshold τ_i, binarize the subblock image. We then utilize the algorithm in Appendix II to find the connected components and calculate the center of each component. We use $\pi_\tau : X \rightarrow Y(\tau)$ to represent the similarity transform (9) obtained from mapping model points X to spot centers $Y(\tau)$. From the Bayes theorem and the assumption that the transform parameters of π_τ follow equal prior distribution, the transform π_τ^* that maximizes the posterior probability $P(\pi_\tau \mid Y(\tau), X)$ is equal to the transform that maximizes the likelihood probability $P(Y(\tau) \mid \pi_\tau, X)$. From a set of coarse to fine thresholds $\{\tau_0, \tau_1, \tau_2, \ldots\}$, the estimation of

$$Max_\pi P(Y(\tau_0), Y(\tau_1), Y(\tau_2), \ldots \mid \pi, X)$$

can be simplified by a Markov random field approach, which yields

$$\text{Max}_\pi P(Y(\tau_0) \mid \pi, X) \cdot \left\{ \prod_i P(Y(\tau_{i+1}) \mid Y(\tau_i), \pi, X) \right\}. \quad (10)$$

The transform π_0^* of the initial threshold τ_0 is obtained by applying the geometrical hashing algorithm to spot centers $Y(\tau_0)$ and X. The threshold is iteratively modified in order to refine the parameters of the transform π that maximizes the above equation. To simplify the notation, we illustrate below the transform parameter refinement from π_0^* to π_1^*. Further refinement of the parameters with a finer threshold can be derived in a similar way.

We denote the matched pair $\{[\mathbf{x}, \mathbf{y}]\}$ of π_0^* as Q_0. Given the new spot centers $Y(\tau_1)$, a matched pair $[\mathbf{x}, \mathbf{y}]$ in Q_0 can be modified as $[\mathbf{x}, c(\mathbf{y})]$ by replacing $\mathbf{y} \in Y(\tau_0)$ with the closest center $c(\mathbf{y}) \in Y(\tau_1)$. We denote this modified matched pair and the new set of matched pairs between X and $Y(\tau_1)$ as Q_1^1 and Q_1^2, respectively, and let $Q_1 = Q_1^1 \bigcup Q_1^2$ be all the matched pairs between X and $Y(\tau_1)$. We assume that the joint probabilities $P(Y(\tau_0) \mid \pi, X)$ and $P(Y(\tau_1) \mid Y(\tau_0), \pi, X)$ can be factored as the product of the marginal probability of each matched pair element in Q_0 and Q_1, respectively. We further assume that the marginal probability of each element is a normal distribution, determined from the distance between the two corresponding spot centers by the transform π. Then, we have

$$\begin{aligned} P(Y(\tau_0) \mid \pi, X) &\propto \prod_{[\mathbf{x},\mathbf{y}] \in Q_0} e^{(-\|\pi(\mathbf{x}) - \mathbf{y}\|^2)/\sigma^2} \text{ and} \\ P(X(\tau_1) \mid Y(\tau_0), \pi, X) &\propto \prod_{[\mathbf{x},c(\mathbf{y})] \in Q_1^1} e^{(-\|\pi(\mathbf{x}) - c(\mathbf{y})\|^2)/\sigma^2} \\ &\quad \prod_{[\mathbf{x}',\mathbf{y}'] \in Q_1^2} e^{(-\|\pi(\mathbf{x}') - \mathbf{y}'\|^2)/\sigma^2} \end{aligned}$$

where σ^2 is the variance, which is assumed to be constant for all matched pairs. The transform π_1^* that maximizes the likelihood probability

$$P(Y(\tau_0) \mid \pi, X) P(Y(\tau_1) \mid Y(\tau_0), \pi, X)$$

equals the transform that minimizes

$$\begin{aligned} E(\pi) = \sum_{[\mathbf{x},\mathbf{y}] \in Q_0} \|\pi(\mathbf{x}) - \mathbf{y}\|^2 + \sum_{[\mathbf{x},c(\mathbf{y})] \in Q_1^1} \|\pi(\mathbf{x}) - c(\mathbf{y})\|^2 \\ + \sum_{[\mathbf{x}',\mathbf{y}'] \in Q_1^2} \|\pi(\mathbf{x}') - \mathbf{y}'\|^2. \quad (11) \end{aligned}$$

The parameters of the optimal similarity transform π_1^* can be obtained by taking the derivative of the above equation with respect to each parameter of the transform and setting it to zero. Each term in (11) is a quadratic function of its parameters.

Hence, each term's derivative is a linear function of the parameters, and the optimal parameters can be obtained by solving a system of linear equations. Although the above procedure determines the optimal parameters of a subblock image for two thresholds, it can be extended effortlessly to any sequence of thresholds.

Tree-Based Outlier Correction: The initial parameters of a subblock, obtained from the block diagram in Fig. 5, may be inconsistent with those of its neighboring subblocks. If this is the case, we say that the subblock is an outlier. Because local distortion varies smoothly, the transform parameters of neighboring subblocks should have similar values. An error in a previous parameter estimation can, thus, be adjusted, based on the estimated parameters of neighboring subblocks. An outlier subblock can be determined by comparing the rotation θ and scale s of the subblock parameters to those of its neighboring subblocks. We use the following simple method to define an outlier: if θ is outside of the proper range $[\theta_l, \theta_h]$, or $(s/s_m) \geq \epsilon$, where s_m is the median scale of its neighboring inlier subblocks and ϵ is a given threshold, then the subblock is an outlier.

The transform parameters of an outlier subblock can be corrected by the parameters of its neighboring inlier subblocks. We use a quadtree structure to organize all the subblocks. Each subblock is a leaf of the tree, and four neighboring subblocks have a parent node. The spot centers of the parent node are the union of the spot centers of its children nodes. Four neighboring parent nodes are then grouped and assigned to a parent node. This process continues recursively until the root of the quadtree is reached. Any node can be associated to a set of transform parameters. The transform is obtained from the transform parameters of the node's inlier children. Let I be the set of inlier children of the node p and let $\{[\mathbf{x}_{i,k}, \mathbf{y}_{i,k}]\}$ be the set of matched pairs between model X and subblock i, which is an inlier child of p. Further, let k be the index of an element in the set. The transform of node p can then be obtained by finding the transform π_p that minimizes

$$\sum_{i \in I} \sum_{k} \|\mathbf{y}_{i,k} - \pi(\mathbf{x}_{i,k})\|^2. \tag{12}$$

The optimal similarity transform parameters can be derived by taking a partial derivative of the formula of each parameter and setting each result to zero. The new transform parameters of the outlier children of node p are the optimal parameters of the node p. Fig. 6 shows a quadtree of a microarray block. An outlier node (4, 2) in the figure can be corrected by its inlier sibling nodes, (3, 1), (3, 2), and (4, 1), as described by (12). The new transform of the outlier node (4, 2) is inherited from the transform of its parent node p.

The quadtree structure allows us to extend this simple example to correct any number of outlier subblocks of different sizes. If some children of a node p are inliers, the parameters of the inlier children nodes are used to calculate the parameters of the parent node p, according to (12). The resultant parameters are passed to all outlier children of p, and become the new parameters of the outlier nodes. If all the children of p are outliers, then p is an outlier. We can use the parameters obtained from the inlier sibling of p as the new parameters of p. These parameters replace those of the children of p. Details of this tree-based outlier correction algorithm are given in Appendix III. Fig. 7 shows a block diagram, using a tree to correct subblock parameters in order to obtain the final initial transform parameters.

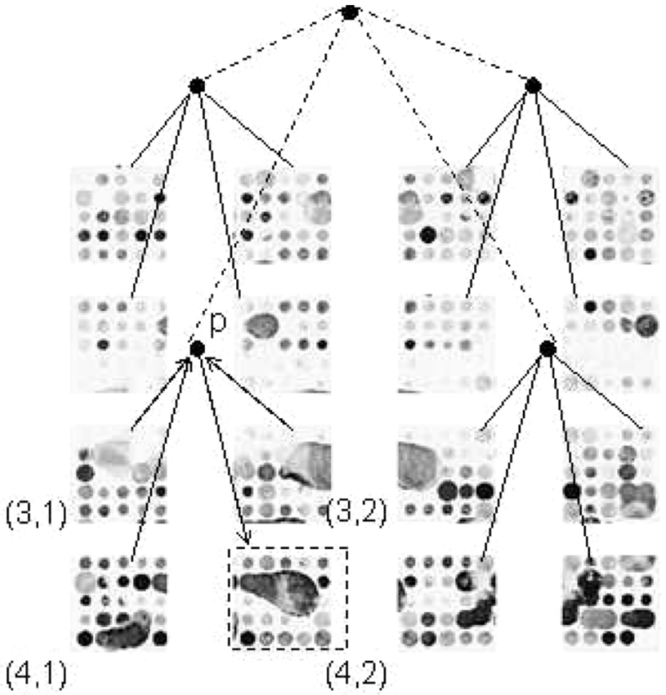


Fig. 6. Subblock quadtree, where p is the parent node of subblocks (3, 1), (3, 2), (4, 1), and (4, 2). Subblock (4, 2) is an outlier whose transform parameters pass through p and are obtained from nodes (3, 1), (3, 2), and (4, 1).

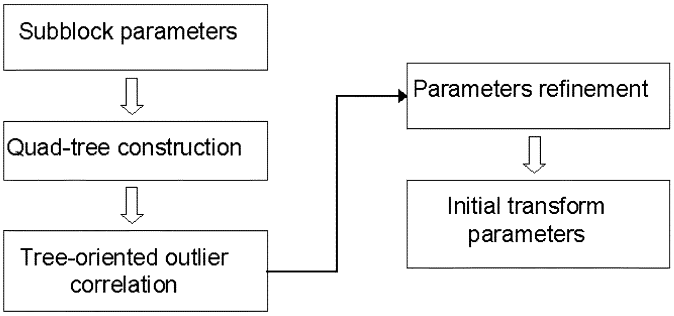


Fig. 7. Acquisition of initial transform parameters from the initial subblock parameters of Fig. 5.

IV. Performance Evaluation

We evaluate our spot gridding algorithm by comparing our results with those obtained by other algorithms on two sets of microarray images. One set contains some poor quality images from SMD, while the other set contains Agilent 60-mer oligonucleotide microarrays whose specifications are on the related web pages [2]. The Agilent's microarrays are some of the best quality oligonucleotide chips currently available commercially. We use these sets of images to demonstrate that our method can accurately grid the spot centers of images of varying quality produced by different technologies. We implement our algorithm using the Windows XP platform and all images are processed in the Matlab environment. The gray

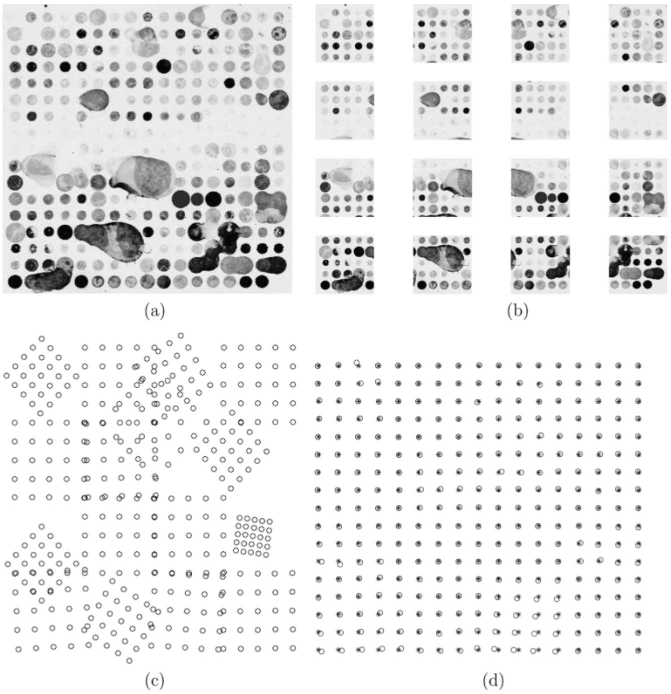


Fig. 8. (a) Block containing various types of noise. (b) Block (2, 2) of hp7004b divided into 16 subblocks. (c) Result of geometric hashing. (d) Alignment of the initial and final centers. The initial centers are marked by asterisks and the final centers by circles.

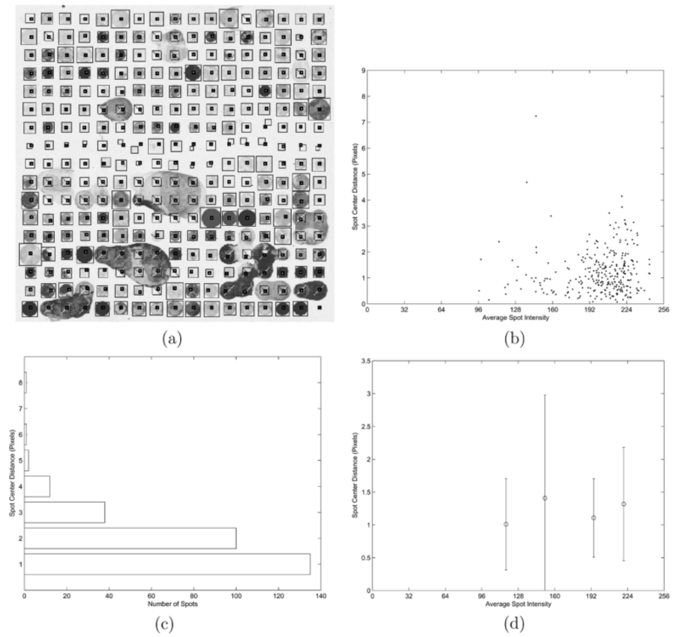


Fig. 9. (a) Our final centers are super-imposed on the block. All but one center are within box boundaries. The boxes are from SMD. (b) The scatter plot of the distribution of the pixel distance of our spot centers and those of SMD versus the average intensity of the spots. (c) The histogram of the distance between our spot centers and those in SMD. The average distance between adjacent SMD spot centers is 15 pixels. (d) The distribution in (b) is partitioned into four sections of equal distance and average spot intensity. The spot intensity of each section is either $[96, 133)$, $[133, 170)$, $[170, 207)$, or $[207, 244]$, where 96 and 244 are, respectively, the smallest and the largest spot intensities. The mean and the standard deviation of each section are plotted in this subfigure.

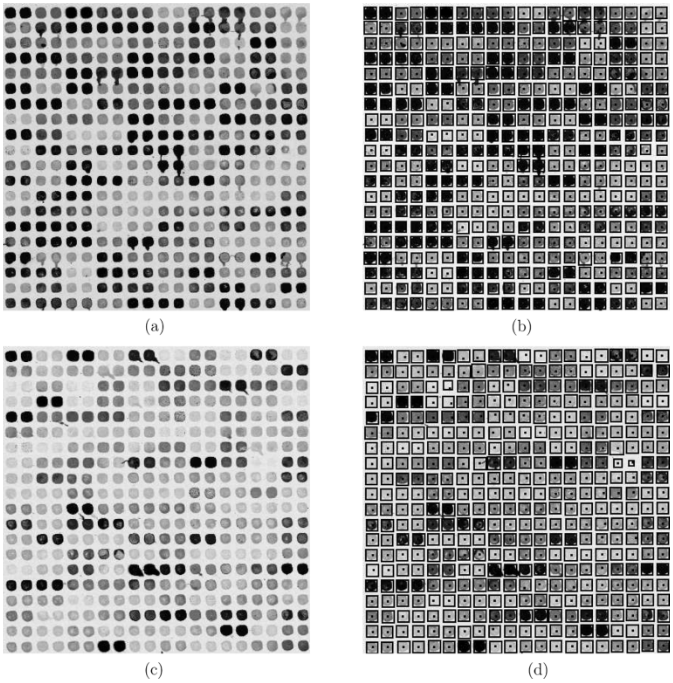


Fig. 10. (a), (c) Two image blocks of oligonucleotide chips; (b) is obtained by super-imposing our spot centers on (a), while (d) is obtained by super-imposing our spot centers on (c). The rectangular boxes in the images are obtained by using GenePix Pro 5.0. The distance between adjacent spot centers of the two images is 10 pixels.

level image in SMD takes eight bits, while that of Agilent's image takes sixteen bits. Throughout our experiments, we use grayscale images and set the control parameter λ to 1/16. Our experiments show that this setting is robust for different microarray images.

In the following experiment, we first demonstrate the step-by-step results of applying our procedure to an image from SMD. The top half of Fig. 8 shows a block and the subblocks of image *hp7004b* from SMD. The result of mapping the model points using the transform obtained by geometric hashing is shown in Fig. 8(c). The initial spot centers (marked by asterisks) are aligned with the final spot centers (marked by circles), as shown in Fig. 8(d). Comparing the initial and final centers, we find that the distances between the centers of strong signals are shorter than those between weak signals. Thus, the Jacobi iterations refine the spot centers of weak signals. Fig. 9(a) shows the superimposition of our centers on the block images. The only center that lies outside of a box corresponds to a very weak signal. Fig. 9(c) shows the histogram of the distances of our centers and the centers provided by SMD. The mean and standard deviation of the histogram are 1.2 and 0.9 pixels, respectively. The average distance of adjacent spot centers of both SMD and our method is 15 pixels. Thus, our method accurately locates the spot centers of this image. Fig. 9(b) shows the distribution of the spot center distance versus the average spot intensity. The distance between the spot centers with the largest average intensity and those with the smallest average intensity is then uniformly divided into four sections. The mean and the standard deviation of the distance between our spot centers and those of SMD with spots, whose average intensities are located within each section, are calculated and plotted in Fig. 9(d). As this subfigure shows, the variation of the mean and the standard deviation of different spot intensities can be neglected, which indicates that the accuracy of our method is consistent for all spot intensities. This is

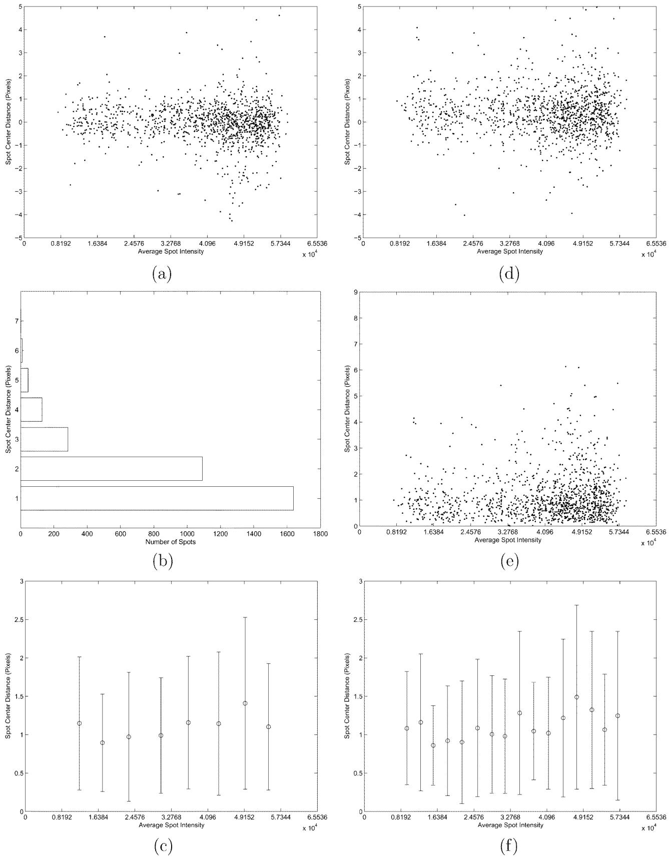


Fig. 11. Comparison of the spot centers of our method and GenePix Pro 5.0. (a), (b), and (d) are, respectively, the distributions of the horizontal, the vertical, and the distance of the spot-center-difference versus the spot intensity; (c) is the plot of the number of points versus the spot center distance of (d); (e) and (f) are the mean and the standard deviation of each segment. The statistics are calculated from all the points in (d) that are located in the segment.

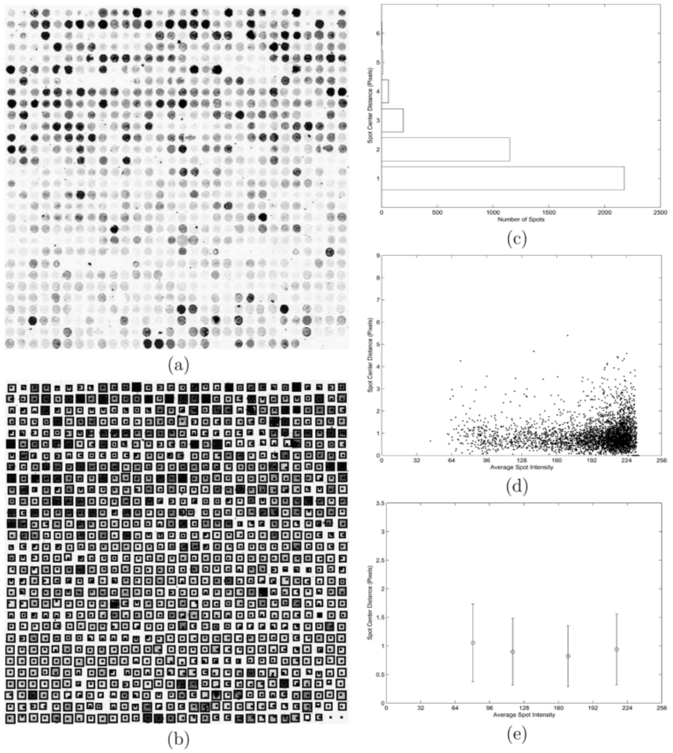


Fig. 12. Spot centers compared with those provided by SMD; (a) and (b) are, respectively, a block in image SHDR146 of SMD and the super-imposition of our spot centers on the image. The spots provided by SMD images are shown as boxes, while detected spot centers are marked by small, bold squares; (c) shows the distance histogram of four randomly chosen blocks of SHDR146. The average distance between adjacent spot centers is about 15 pixels, while the mean and the standard deviation of the spot center difference, are 0.9 and 0.6 pixels, respectively; (d) is the distribution of the spot center distance verus the average spot intensity. This distribution is divided into four segments of equal length. The mean and the standard deviation in each segment are plotted in (e).

because the initial spot centers, found by using the modeling approach and a sequence of robust procedures, are close to the accurate spot centers. Moreover, the spot centers of weak signals, which are usually less accurate in the initial solution, are refined by Jacobi's iterations. Consequently, our method for gridding spot centers renders an accuracy that is independent of the intensity of a spot.

Fig. 10(a) and (c) represents two image blocks of Agilent 60-mer oligonucleotide microarrays. There are 400 spots in the image and the distance between adjacent spot centers is ten pixels. The spot centers of our method in Fig. 10(a) and (c) is super-imposed on the images and shown in Fig. 10(b) and (d), respectively. The rectangular boxes are obtained by using GenePix Pro 5.0, and statistical measurement is used to demonstrate that our spot centers and those of GenePix Pro 5.0 are matched. Fig. 11(a) and (b) is the respective distributions of the horizontal (X) and vertical (Y) differences between our spot centers and those provided by GenePix Pro 5.0, compared to the average spot intensity. The distributions are obtained by accumulating the results of eight blocks of Agilent's microarrays with typical blocks, as shown in Fig. 10. Measuring the mean and standard deviations of all the spot differences of these images, the horizontal mean and standard deviation of spot-center-differences are 0.1 and 1.0 pixels, respectively, while those of vertical spot-center-differences are 0.4 and 1.04 pixels, respectively. Fig. 11(c) is the histogram of the number of pixels versus the distance of spot centers for our spot centers and those of GenePix. The maximal value of the histogram occurs at approximately one pixel. The reason may be that if we assume the marginal probabilities of horizontal and vertical spot-center-differences are of normal density with mean zero and variance one, then the distance of spot centers becomes a Rayleigh distribution whose maximal value occurs at one pixel [22]. The distribution of the distance between the spot centers of our method and those of GenePix Pro 5.0 is plotted in Fig. 11(d). To measure the effects of the average spot intensity on the spot center distance, we partition the points in the distribution according to their intensities into a different number of segments. We then measure the mean and standard deviation of the points located in each segment. The results of eight and sixteen segments are shown in Fig. 11(e) and (f), respectively. The intensity interval of each segment is calculated by using a similar method to that used to obtain the intensity interval of Fig. 9(d), where the range between the minimal and the maximal intensity values is divided into segments of equal length. From Fig. 11(e) and (f), the spot center differences between our method and those of GenePix appears to be independent of the average intensity of the spots on these images.

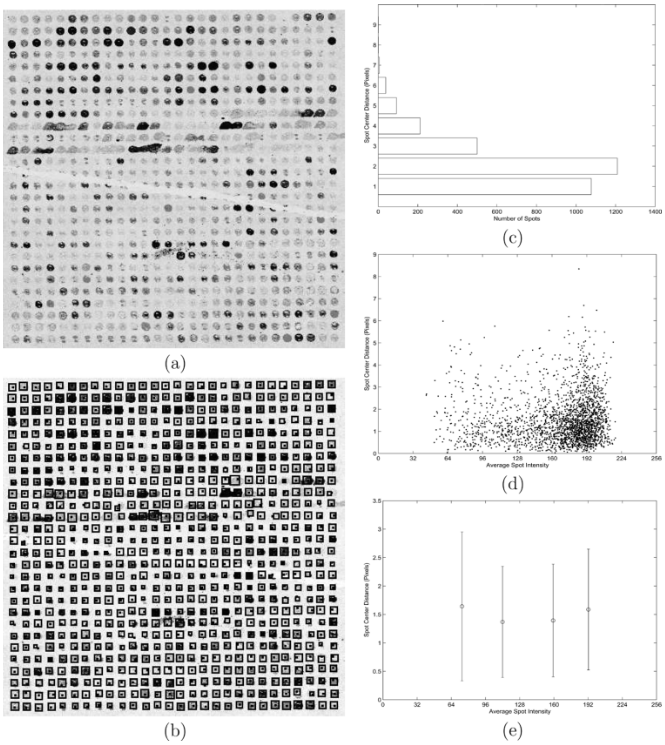


Fig. 13. Block (1, 2) of LC23N085 is shown in (a), and the super-imposition of our spot centers is shown in (b). The spots provided by SMD images are shown as boxes, while detected spot centers are marked by small, bold squares; (c) shows the distance histogram of four randomly chosen blocks of LC23N085. The average distance between SMD adjacent spot centers is about 15 pixels, while the mean and the standard deviation of the spot center distance, are 1.5 and 1.0 pixels, respectively; (d) is the distribution of spot center distance versus the average spot intensity. This distribution is divided into four segments of equal length. The mean and the standard deviation in each segment are plotted in (e).

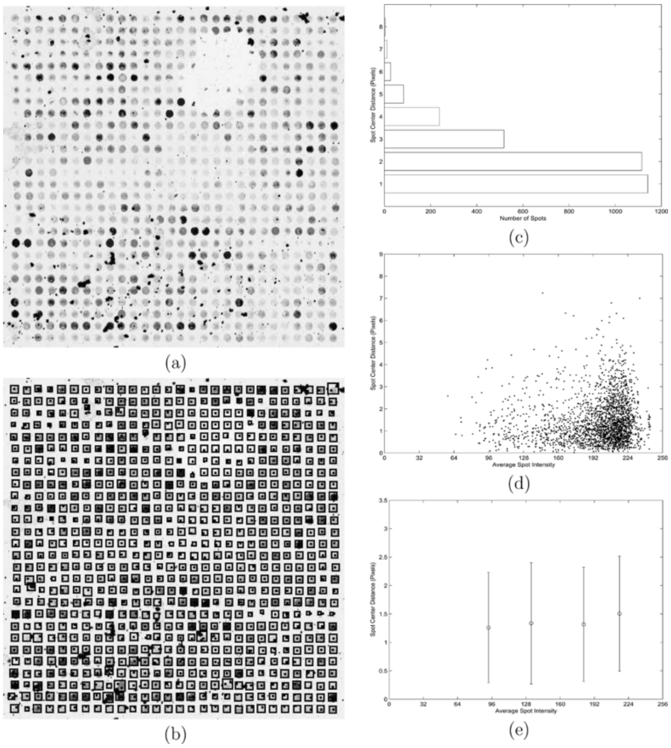


Fig. 14. Block (3, 4) of lc30n008 is shown in (a), and the super-imposition of our spot centers is shown in (b). The spots provided by SMD images are shown as boxes, while detected spot centers are marked by small, bold squares; (c) shows the distance histogram of four randomly chosen blocks of lc30n008. The average distance between SMD adjacent spot centers is about 15 pixels, while the mean and the standard deviation of the spot center distance, are 1.4 and 1.0 pixels, respectively; (d) is the distribution of spot center distance versus the average spot intensity. This distribution is divided into four segments of equal length. The mean and the standard deviation in each segment are plotted in (e).

We use the array notation (r, c) to denote the block at the rth row and the cth column of a microarray image. Fig. 12(a) shows the block (2, 4) in the microarray image *SHDR146* of SMD, while Fig. 12(b) shows a super-imposition of our detected spot centers over the image block provided by SMD. All spot centers detected by our method are located within their corresponding boxes. Fig. 12(c) plots the histogram of the distance between our spot centers and those provided by SMD for four blocks in *SHDR146*, including the block (2, 4). To show the dependence of our method on the average spot intensity, we plot the distribution of the distance of spot centers versus the average spot intensity in Fig. 12(d). The range of the spot intensity is divided into four segments of equal length. The mean and the standard deviation of points in the distribution located in each segment are plotted in Fig. 12(e).

Figs. 13–15 show other examples of processing noisy SMD images. Distance histograms of our spot centers and those of SMD are given in Figs. 13(c)–15(c), respectively. The mean and standard deviation of each segment of distance distribution versus average spot intensity, as shown in Figs. 13(d)–15(d), are plotted in Figs. 13(e)–15(e), respectively. The average adjacent spot distance of these images is 15 pixels, while the mean and the standard deviation are each less than two pixels. Moreover, the distance between our spot centers and those of SMD is irrelevant to the spot intensity.

From the results of processing oligonucleotide images and noisy SMD images, we show that our method can accurately detect the spot centers of images of varying quality that are manufactured by different techniques. The accuracy of our spot gridding method is irrelevant to the intensity of a spot. The method is robust and automatic because, in our experiments, we do not use any manual adjustments to find spot centers after a block is delineated. Our algorithm takes less than 15 min to grid an image of about 450 × 450 pixels on a CPU whose speed is 2.4 GHz per second. The slowest part of our algorithm occurs when computing connected components. This can be improved when implemented in an environment that is more efficient than Matlab.

V. Conclusion

A large proportion of grid distortion can be approximated by a locally smooth distortion. In this manuscript, an optimization approach is proposed to grid the exact spot centers of a microarray image, whose grids are slowly varying similarity transforms. A Bayesian approach and a multithreshold Markov model are used to find robust initial parameters. The initial parameters are refined by Jacobi iterations, which solves our optimization problem. Experiments show that our method can robustly extract accurate spot centers from microarrays with local smooth grid distortions. In practice, however, grid distortions

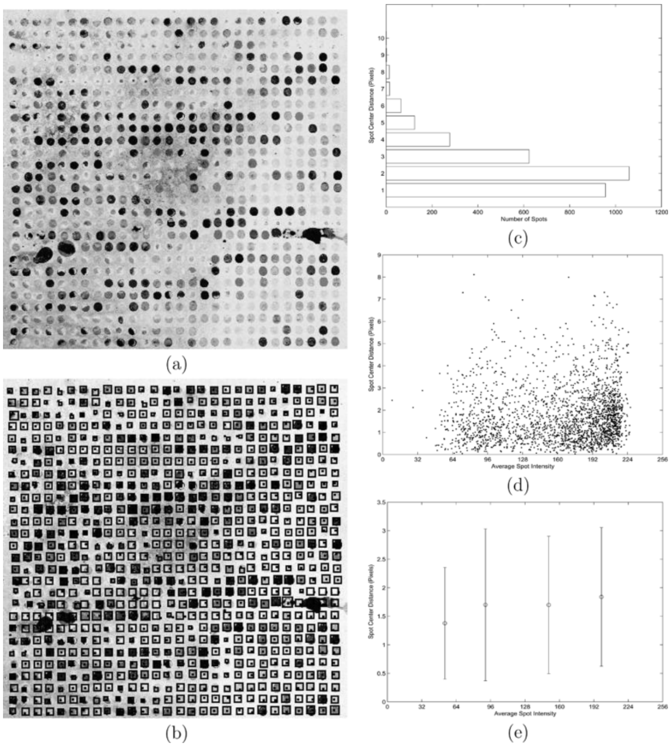


Fig. 15. Block (1, 1) of lc30n010 is shown in (a), and the super-imposition of our spot centers on the block is shown in (b); (c) shows the distance histogram of four randomly chosen blocks of lc30n010. The average distance between SMD adjacent spot centers is about 15 pixels, while the mean and the standard deviation of the spot center distance are 1.76 and 1.2 pixels, respectively; (d) is the distribution of the spot center distance verus the average spot intensity. This distribution is divided into four segments of equal length. The mean and the standard deviation in each segment are plotted in (e).

can be discontinuous. Improving our method for images of discontinuous distortion grids is an issue worth further study.

Appendix I
Finding the Optimal Parameters by the Jacobi Iterative Scheme

We use $N(k,l)$ to represent the number of neighbors of $\mathbf{x}_{k,l}$ that are in the active set $\mathcal{A}\{\mathbf{x}\}$. Differentiating e in (6) with respect to $a_{k,l}$, $b_{k,l}$, $c_{k,l}$, and $d_{k,l}$ yields

$$\begin{aligned}\frac{\partial e}{\partial a_{k,l}} &= \frac{2}{|\mathcal{A}\{\mathbf{x}\}|}\\ &\quad\cdot \big[a_{k,l}(x_1(k,l)^2 + x_2(k,l)^2) + c_{k,l}x_1(k,l)\\ &\qquad + d_{k,l}x_2(k,l) - (x_1(k,l)y_1(k,l)\\ &\qquad + x_2(k,l)y_2(k,l)) + 8\lambda(a_{k,l} - \bar{a}_{k,l})\big]\end{aligned}$$

$$\begin{aligned}\frac{\partial e}{\partial b_{k,l}} &= \frac{2}{|\mathcal{A}\{\mathbf{x}\}|}\\ &\quad\cdot \big[b_{k,l}(x_1(k,l)^2 + x_2(k,l)^2) - c_{k,l}x_2(k,l)\\ &\qquad + d_{k,l}x_1(k,l) + (x_2(k,l)y_1(k,l)\\ &\qquad - x_1(k,l)y_2(k,l)) + 8\lambda(b_{k,l} - \bar{b}_{k,l})\big]\\ \frac{\partial e}{\partial c_{k,l}} &= \frac{2}{|\mathcal{A}\{\mathbf{x}\}|}\\ &\quad\cdot \big[a_{k,l}x_1(k,l) - b_{k,l}x_2(k,l) + c_{k,l} - y_1(k,l)\\ &\qquad + 4\lambda(c_{k,l} - \bar{c}_{k,l})\big]\\ \frac{\partial e}{\partial d_{k,l}} &= \frac{2}{|\mathcal{A}\{\mathbf{x}\}|}\\ &\quad\cdot \big[a_{k,l}x_2(k,l) + b_{k,l}x_1(k,l) + d_{k,l} - y_2(k,l)\\ &\qquad + 4\lambda(d_{k,l} - \bar{d}_{k,l})\big]\end{aligned}$$

where $\bar{a}_{k,l}$, $\bar{b}_{k,l}$, $\bar{c}_{k,l}$, and $\bar{d}_{k,l}$ are the local averages of $a_{k,l}$, $b_{k,l}$, $c_{k,l}$, and $d_{k,l}$, respectively. Extremum values occur when the above derivatives of e equal zero. To simplify the formulation, let B be the matrix in the equation at the bottom of the page. The resultant equations can be combined in the following matrix form:

$$\begin{aligned}B\cdot\begin{bmatrix}a_{k,l}\\ b_{k,l}\\ c_{k,l}\\ d_{k,l}\end{bmatrix} &= 4\lambda\begin{bmatrix}2&0&0&0\\0&2&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}\bar{a}_{k,l}\\ \bar{b}_{k,l}\\ \bar{c}_{k,l}\\ \bar{d}_{k,l}\end{bmatrix}\\ &\quad+\begin{bmatrix}x_1(k,l)y_1(k,l) + x_2(k,l)y_2(k,l)\\ x_1(k,l)y_2(k,l) - x_2(k,l)y_1(k,l)\\ y_1(k,l)\\ y_2(k,l)\end{bmatrix}.\end{aligned}$$

After multiplying both sides of the above equation by B^{-1} (if B^{-1} does not exist, we can use the pseudo inverse of B), we have

$$\begin{aligned}\begin{bmatrix}a_{k,l}\\ b_{k,l}\\ c_{k,l}\\ d_{k,l}\end{bmatrix} &= 4\lambda B^{-1}\begin{bmatrix}2&0&0&0\\0&2&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}\bar{a}_{k,l}\\ \bar{b}_{k,l}\\ \bar{c}_{k,l}\\ \bar{d}_{k,l}\end{bmatrix}\\ &\quad+B^{-1}\begin{bmatrix}x_1(k,l)y_1(k,l) + x_2(k,l)y_2(k,l)\\ x_1(k,l)y_2(k,l) - x_2(k,l)y_1(k,l)\\ y_1(k,l)\\ y_2(k,l)\end{bmatrix}.\end{aligned}$$

$$\begin{bmatrix}x_1(k,l)^2 + x_2(k,l)^2 + 8\lambda & 0 & x_1(k,l) & x_2(k,l)\\ 0 & x_1(k,l)^2 + x_2(k,l)^2 + 8\lambda & -x_2(k,l) & x_1(k,l)\\ x_1(k,l) & -x_2(k,l) & 1+4\lambda & 0\\ x_2(k,l) & x_1(k,l) & 0 & 1+4\lambda\end{bmatrix}$$

We can solve the above equation for $a_{k,l}$, $b_{k,l}$, $c_{k,l}$, and $d_{k,l}$ by the Jacobi iterative scheme

$$\begin{bmatrix} a_{k,l}^{(n+1)} \\ b_{k,l}^{(n+1)} \\ c_{k,l}^{(n+1)} \\ d_{k,l}^{(n+1)} \end{bmatrix} = 4\lambda B^{-1} \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \bar{a}_{k,l}^{n} \\ \bar{b}_{k,l}^{n} \\ \bar{c}_{k,l}^{n} \\ \bar{d}_{k,l}^{n} \end{bmatrix} + B^{-1} \begin{bmatrix} x_1(k,l)y_1(k,l) + x_2(k,l)y_2(k,l) \\ x_1(k,l)y_2(k,l) - x_2(k,l)y_1(k,l) \\ y_1(k,l) \\ y_2(k,l) \end{bmatrix}.$$

The new values of $a_{k,l}$, $b_{k,l}$, $c_{k,l}$, and $d_{k,l}$ are equal to the average of the surrounding values multiplied by a matrix, and the addition of an adjustment term. The final quality of the Jacobi iterative scheme solution depends on the quality of the initial solution.

Appendix II
Finding Connected Components and Their Centers

The region growing technique proposed in [3] is a popular method for finding the connected components of black pixels from a binarized image. The algorithm is not optimal, but it is efficient and simple to implement. The initial step of the algorithm is to un-label all black pixels. It then iteratively performs the following steps until all the black pixels are labeled.

```
Step 1. Choose an un-labeled black pixel as the seed for a newly connected component and label it.
Step 2. Any un-labeled pixel that neighbors a pixel in the connected component is labeled and included in the connected component.
Step 3. Step 2 is repeated until there are no un-labeled pixels neighboring any pixel in the component.
Step 4. If any un-labeled pixels still exist, return to Step 1.
```

The center of a connected component is the mean of the coordinates of the pixels in the connected component. Criteria, such as the compactness of a spot [17], could be used to determine whether a connected component is associated with a spot. However, in our algorithm, we use the size of a connected component to determine whether it is likely to be a spot.

We assume that an image subblock composed of m by n spots has u by v pixels. The numbers m and n can be obtained from the model subblock. Thus, a spot in the image takes at most $A =$ (uv/mn) pixels. We remove a connected component whose size is either too small for A, or too close to A. A connected component that is too small is likely to occur because of noise variations, while one that is too large may yield a larger error in center computation.

Appendix III
Tree-Based Outlier Correction Algorithm

We correct all outliers by first using the Bottom-up Algorithm, then the Top-down Algorithm.

```
Bottom-up Algorithm
Step 0. Build the quadtree.
Step 1. Calculate the similarity transform of each subblock. Subblocks are represented as the leaves of the tree.
Step 2. Mark subblocks that have inconsistent parameters as outliers.
Step 3. Use (12) and the parameters of the inlier subblocks to calculate the transform parameters of the parent nodes. If all the children of a node are outliers, mark the node as an outlier.
Step 4. If all nodes at the current level are inliers, we stop at this level. Otherwise, we proceed with Steps 3 and 4 one leaf higher than the current level until the root of the tree is reached.
```

```
Top-down Algorithm
Step 0. Begin at the root of the tree, (which is assumed to be an inlier).
Step 1. If we reach the lowest level of the tree, we stop. Otherwise, for each node at a level, if any of its children nodes are outliers, we use the parameters of the node as the new parameters of its outlier children and remove the outlier mark on them.
Step 2. Repeat Step 1 one level down.
```

Acknowledgment

The authors would like to thank the reviewers for their comments, as well as P.-F. Shang, who helped them implement the geometric hashing algorithm; Prof. C.-F. Chang of the Chinese Culture University, who posed the problem to them; and Prof. W.-H. Li of the University of Chicago for his insightful suggestions.

References

[1] Y. Amit and A. Kong, "Graphical templates for model registration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 2, pp. 225–235, Feb. 1996.

[2] [Online]. Available: http://www.chem.agilent.com/scripts/generic.asp?lpage=10 692&indcol=N&prodcol=Y

[3] D. H. Ballard and C. M. Brown, *Computer Vision*. Englewood Cliffs, NJ: Prentice-Hall, 1982.

[4] C. A. Bouman and M. Shapiro, "A multiscale random field model for Bayesian image segmentation," *IEEE Trans. Image Process.*, vol. 3, no. 2, pp. 162–177, Mar. 1994.

[5] J. Buhler, T. Ideker, and D. Haynor. (2000) Dapple: Improved Techniques for Finding Spots on DNA Microarray. [Online]. Available: http://www.cs.wustl.edu/jbuhler/research/dapple/

[6] Y. Chen, E. R. Dougherty, and M. L. Bittner, "Ratio-based decisions and the quantitative analysis of cDNA microarray images," *J. Biomed. Opt.*, vol. 2, pp. 364–367, 1997.

[7] V. G. Cheung, M. Morley, F. Aguilar, A. Massimi, R. Kucherlapati, and G. Childs, "Making and reading microarrays," *Nat. Genet.*, vol. 21, no. 1, pp. 15–9, Jan. 1999.

[8] D. J. Duggan, M. Bittner, Y. Chen, P. Meltzer, and J. M. Trent, "Expression profiling using cDNA microarrays," *Nat. Genet.*, vol. 21, no. 1, pp. 10–4, Jan. 1999.

[9] M. B. Eisen and P. O. Brown, "DNA arrays for analysis of gene expression," *Meth. Enzymol. 303*, pp. 179–205, 1999.

[10] R. Fabbri, L. da F. Costa, and J. Barrera, "Toward nonparametric gridding of microarray images," in *Proc. 14th Int. Conf. Digital Signal Processing*, vol. 2, Jul. 1–3, 2002, pp. 623–626.

[11] [Online]. Available: http://www.axon.com/gn_GenePixSoftware.html

[12] K. Hartelius and J. M. Carstensen, "Bayesian grid matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 2, pp. 162–173, Feb. 2003.

[13] J. Gollub, C. Ball, G. Binkley, K. Demeter, D. Finkelstein, J. Hebert, T. Hernandez-Boussard, H. Jin, M. Kaplper, J. Matese, M. Schroeder, P. Brown, D. Botstein, and G. Sherlock, "The stanford microarray database: Data access and quality assessment tools," *Nucl. Acids Res.*, vol. 31, no. 1, pp. 94–96, Jan. 2003.

[14] B. K. P. Horn, *Robot Vision*. Cambridge, MA: The MIT Press, 1986.

[15] [Online]. Available: http://www.koada.com/koadarray

[16] C. Kooperberg, T. G. Fazzio, J. J. Delrow, and T. Tsukiyama, "Improved background correction for spotted DNA microarrays," *J. Comput. Biol.*, vol. 9, no. 1, pp. 55–66, 2002.

[17] A. K. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[18] Y. Lamdan, J. T. Schwartz, and H. J. Wolfson, "Affine invariant model-based object recognition," *IEEE Trans. Robot. Autom.*, vol. 6, no. 5, pp. 578–589, Oct. 1990.

[19] R. J. Lipshutz, S. P. Fodor, T. R. Gingeras, and D. J. Lockhart, "High density synthetic oligonucleotide arrays," *Nat. Genet.* , vol. 21, no. 1, pp. 20–4, Jan. 1999.

[20] D. J. Lockhart, H. Dong, M. C. Byrne, M. T. Follettie, M. V. Gallo, M. S. Chee, M. Mittmann, C. Wang, M. Kobayashi, H. Horton, and E. L. Brown, "Expression monitoring by hybridization to high-density oligonucleotide arrays," *Nat. Biotechnol.*, vol. 14, no. 13, pp. 1675–80, Dec. 1996.

[21] D. J. Lockhart and E. A. Winzeler, "Genomics, gene expression and DNA arrays," *Nature*, vol. 405, pp. 827–836, Jun. 2000.

[22] A. Papoulis, *Probability, Random Variables, and Stochasrtic Process*. New York: McGraw-Hill, 1984, pp. 138–138.

[23] I. Rigoutsos and R. Hummel, "A Bayesian approach to model matching with geometric hashing," *Comput. Vis. Image Understand.*, vol. 61, no. 7, pp. 11–26, Jul. 1995.

[24] [Online]. Available: http://rana.lbl.gov/EisenSoftware.htm

[25] M. Schena, D. Shalon, R. W. Davis, and P. O. Brown, "Quantitative motoring of gene expression patterns with a complementary DNA microarray," *Science*, vol. 270, no. 5235, pp. 467–70, Oct. 1995.

[26] M. Schena, *Microarray Analysis*. New York: Wiley, 2003.

[27] G. K. Smyth, Y. H. Yang, and T. Speed, "Statistical issues in cDNA microarray data analysis," in *Functional Genomics: Methods and Protocols*, M. J. Brownstein and A. B. Khodursky, Eds. Totowa, NJ: Humana, 2002.

[28] Y. Tu, G. Stolovitzky, and U. Klein, "Quantitative noise analysis for gene expression microarray experiments," in *Proc. PNAS*, vol. 99, Oct. 2002, pp. 14 031–14 036.

[29] H. J. Wolfson and I. Rigoutsos, "Geometric hashing: An overview," *IEEE Comput. Sci. Eng. Mag.*, vol. 4, no. 4, pp. 10–21, Oct.–Dec. 1997.

[30] Y. H. Yang, M. J. Buckley, S. Dudoit, and T. P. Speed, "Comparison of methods for image analysis on cDNA microarray data," *J. Comput. Graph. Stat.*, vol. 11, pp. 108–136, 2002.

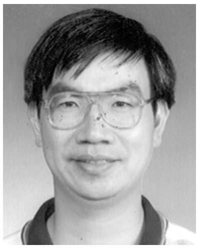
Jinn Ho received the M.S. degree in mathematics from National Taiwan University, Taipei, Taiwan, R.O.C., in 1997.

Currently, he is a Research Assistant with the Institute of Information Science, Academia Sinica, Taipei, Taiwan. His research interests include PDE methods for image processing, numerical methods, and statistical analysis for gene expression.

Wen-Liang Hwang received the B.S. degree in nuclear engineering from National Tsing-Hua University, Hsinchu, Taiwan, R.O.C., the M.S. degree in electrical engineering from the Polytechnic Institute of New York, New York, and the Ph.D. degree in computer science from New York University.

He was a Postdoctoral Researcher with the Department of Mathematics, University of California, Irvine, in 1994. In January 1995, he became a member of the Institute of Information Science, Academia Sinica, Taipei, Taiwan, where he is currently a Research Fellow. He is co-author of the book *Practical Time-Frequency Analysis*. His research interests include wavelet analysis, signal and image processing, and multimedia compression and transmission.

Dr. Hwang he was awarded the Academia Sinica Research Award for Junior Research in 2001.



Henry Horn-Shing Lu received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, R.O.C., and the M.S. and Ph.D. degrees in statistics from Cornell University, Ithaca, NY, in 1990 and 1994, respectively.

He is currently a Professor at and the Director of the Institute of Statistics, National Chiao-Tung University, Hsinchu, Taiwan. He has been a visiting scholar at the University of California, Los Angeles; Harvard University, Cambridge, MA; and the University of Chicago, Chicago, IL. His research interests include statistics, medical images, and bioinformatics.

D. T. Lee (F'92) received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, R.O.C., in 1971, and the M.S. and Ph.D. degrees in computer science from the University of Illinois at Urbana-Champaign, Urbana, in 1976 and 1978, respectively.

He has been with the Institute of Information Science, Academia Sinica, Taiwan, where he is a Distinguished Research Fellow, and where he has been the Director since July 1, 1998. Prior to joining the Institute of Information Science, he had been a Professor with the Department of Electrical and Computer Engineering, Northwestern University, Evanston, IL, since 1978. His research interests include the design and analysis of algorithms, computational geometry, VLSI layout, web-based computing, algorithm visualization, software security, bio-informatics, digital libraries, and advanced IT for intelligent transportation systems. He has published over 120 technical articles in scientific journals and conference proceedings. He also holds three U.S. patents and one Taiwan, R.O.C., patent. He is the Editor of *Algorithmica, Computational Geometry: Theory & Applications*, the *ACM Journal of Experimental Algorithmics*, the *International Journal of Computational Geometry & Applications*, the *Journal of Information Science and Engineering*, and Series Editor of *Lecture Notes Series on Computing*.

He is a Fellow of ACM, President of IICM, and an Academician of Academia Sinica.