Hindawi Publishing Corporation
EURASIP Journal on Applied Signal Processing
Volume 2006, Article ID 13623, Pages 1–11
DOI 10.1155/ASP/2006/13623

Extended δ-Regular Sequence for Automated Analysis of Microarray Images

Hee-Jeong Jin,[1, 2] Bong-Kyung Chun,[1, 2] and Hwan-Gue Cho[1, 2]

[1] *Department of Computer Engineering, Pusan National University, San-30, Jangjeon-dong, Keumjeong-gu, Pusan, 609-735, South Korea*
[2] *Research Institute of Computer, Information, and Communication, Pusan National University, San-30, Jangjeon-dong, Keumjeong-gu, Pusan, 609-735, South Korea*

Received 3 May 2005; Revised 24 August 2005; Accepted 1 December 2005

Microarray study enables us to obtain hundreds of thousands of expressions of genes or genotypes at once, and it is an indispensable technology for genome research. The first step is the analysis of scanned microarray images. This is the most important procedure for obtaining biologically reliable data. Currently most microarray image processing systems require burdensome manual block/spot indexing work. Since the amount of experimental data is increasing very quickly, automated microarray image analysis software becomes important. In this paper, we propose two automated methods for analyzing microarray images. First, we propose the extended δ-regular sequence to index blocks and spots, which enables a novel automatic gridding procedure. Second, we provide a methodology, hierarchical metagrid alignment, to allow reliable and efficient batch processing for a set of microarray images. Experimental results show that the proposed methods are more reliable and convenient than the commercial tools.

1. INTRODUCTION

Microarray is a principal technology in molecular biology, because it results in hundreds and thousands of expressions of genotypes at once [1]. The microarrays are queried in a cohybridization assay using two or more fluorescently labeled probes prepared from the mRNA from the cellular phenotypes of interest [2]. The kinetics of hybridization allows expression levels to be determined relative to the ratio with which each probe hybridizes to an individual array element. Hybridization is assayed using a confocal laser scanner to measure fluorescence intensities, which allow the simultaneous determination of the relative level of expression of all the genes represented in the array.

The first step of a microarray experiment is to generate a raw image, which consists of spots (genes) that form regular arrays (blocks). Figure 1 shows a typical microarray image which consists of 4 × 4 blocks and each block is composed of 24 × 24 spots [3]. In order to measure the level of expression of each spot, the location of each block and spot must be identified in a process called "*gridding*," and then the area of each spot is determined. Finally, the intensity of both the true spot and the background is estimated; this is called "*spots quantification*." The gridding procedure must be performed correctly to quantify all spots precisely, but the huge number of spots makes this procedure difficult to be done manually. In order to overcome this, many automated and/or semiautomated gridding methods and metagridding methodologies have been proposed.

There are many automatic gridding algorithms for computing the exact location of each block and spot. Steinfath has proposed a robust automatic imaging system for microarray experiments [4]. One drawback of Steinfath's system is that if the spot expression rate is less than 70% or the microarray image is skewed, it does not guarantee acceptable performance. Roberto has proposed an automatic gridding method by mathematical morphology [5]. However, it is not a fully automatic method since it requires manual work to correct image rotation, and the suggested gridding method using horizontal and vertical projection may be sensitive to noise. Generally, the other automatic gridding methods cannot be fully or correctly implemented if the image has a lot of noise or a low level of expression [6–8].

In this paper, we propose two methods. One is an automatic gridding algorithm that computes the extended δ-regular sequence by allowing extra pseudopoints. The other

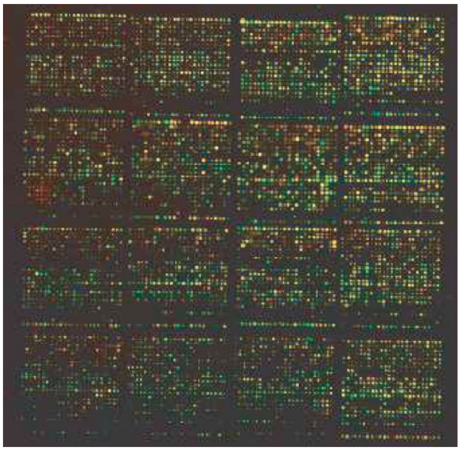


Figure 1: A typical raw image of a microarray: it consists of 4 × 4 blocks and each block is composed of 24 × 24 spots [3].

is a hierarchical metagrid alignment methodology to provide reliable and efficient batch processing for a set of microarray images. Figure 2 shows a flow chart of our automated image analysis system. The basic idea of our automatic gridding algorithm is as follows. In a microarray image, if the spots are in a single block, it is highly likely that they are in the form of a regular sequence. So we compute a set of regular sequences for an entire microarray image and then cluster the "near" regular point patterns, which form a spot grid in a single block. However, handling microarray images with a low expression rate and/or experimental error is difficult. Our model easily overcomes this problem. In batch processing, we align the metagrid to a real image according to the structure of the metagrid which consists of a chipbox, blocks, and spots.

The organization for this paper is as follows. Section 2 explains the concept of the extended δ-regular point sequence and how to apply this procedure to locate the block/spot index. Hierarchical metagrid alignment will be discussed in Section 3. Finally, experimental work and results are given in Sections 4 and 5.

2. EXTENDED δ-REGULAR SEQUENCE

The task of detecting regular spatial sequences in images arises in many computer vision applications, including scene analysis, military applications, and other areas [9, 10]. The general problem is one of recognizing equally spaced collinear subsets in a given set of points. In an ideal microarray image, all of the spots in a row or column in each block should be included in the exact regular sequence. However, it is not desirable to apply an ideal definition of regular pattern sequence to a microarray image, since a microarray image contains much noise and the location of each spot varies somewhat in practice due to the mechanical error in the microarray production machine (spotter). So we propose a new *relaxed* algorithm to compute the regular sequences and to correctly locate the positions of spots and blocks.

2.1. *Preliminary*

In the ideal microarray image, all spots in a row/column are collinear and equally spaced. So we give formal definitions of "collinear" and "equally spaced" for the given finite set of points.

Definition 1. $\overline{P} = \{p_1, p_2, \dots, p_n\}$ is called *collinear* if the area $\triangle(p_i, p_j, p_k) = 0$ and $|\overline{P}| \geq 3$. In a similar way, $\overline{P} = \{p_1, p_2, \dots, p_n\}$ is called *equally spaced* if $|p_i - p_{i-1}| = |p_{i+1} - p_i|$, for $2 \leq i \leq n-1$. Note that $|p - q|$ denotes the Euclidian distance between points p and q.

Now, we define a regular sequence as follows.

Definition 2. $P = \{p_1, p_2, \dots, p_n\}$ is a *regular point sequence* if P is collinear and equally spaced.

A maximal regular sequence of a set of points is one that is not properly contained as a contiguous subsequence in any other regular sequence. Based on the definition of a regular sequence, we define the δ-regular sequence as follows.

Definition 3. A sequence of points is δ-regular if each of its points can be displaced by at most δ along each axis to yield a regular sequence; that is, given a fixed $\delta \geq 0$, a sequence of points $P = \{p_1, p_2, \dots, p_n\} \subset E^2$ is a *δ-regular sequence* if $\overline{P} = \{\overline{p}_1, \overline{p}_2, \dots, \overline{p}_n\}$ and $\delta \geq |x_i - \overline{x}_i|$, $\delta \geq |y_i - \overline{y}_i|$, for all $1 \leq i \leq n$, where $p_i = (x_i, y_i)$ and $\overline{p_i} = (\overline{x_i}, \overline{y_i})$ [10].

Definition 4. A *maximal δ-regular sequence* is one that is not properly contained as a contiguous subsequence in any other δ-regular sequence.

A regular sequence should be one of a δ-regular sequence with $\delta = 0$. Figure 3 shows an example of a maximal δ-regular sequence. In order to show a more relaxed form of a regular sequence, we define an *extended δ-regular sequence* for analyzing microarray images.

Definition 5. A set of δ-regular sequences is called an *extended δ-regular sequence* if we can make them a single δ-regular sequence by adding pseudopoints in between them.

Figure 4 shows how to construct extended δ-regular sequences from input points. Simply, the extended δ-regular sequence is constructed by concatenating two adjacent and collinear δ-regular sequences by inserting pseudopoints between them.

2.2. *Automatic block indexing*

A maximal δ-regular sequence helps calculate the rotational angle θ and unit distance d_u of a given microarray image, and it identifies block structure. In order to reduce time to find all maximal δ-regular sequences and extended δ-regular sequences, we only consider the horizontal (or vertical) δ-regular sequences.

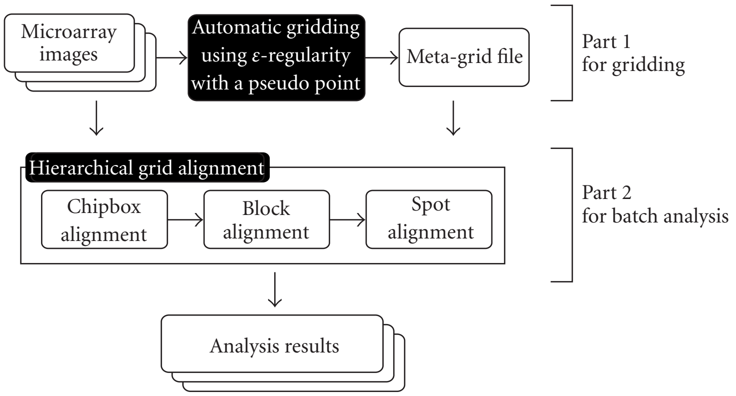


FIGURE 2: A flow chart of our automatic image analysis system. Part 1 is responsible for performing automatic gridding by using the extended δ-regular sequence and part 2 aligns the metagrid to the images using hierarchical metagrid alignment for batch processing.

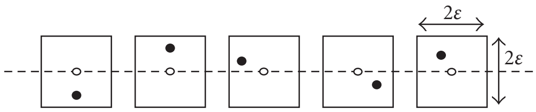


FIGURE 3: A δ-regular sequence: a sequence (solid dots) whose points are within δ of the corresponding (ideal) points of a regular sequence [10].

In our method, we first construct a set of points $\{P_i\}$ by image segmentation and by computing the geometric center of the spots. From this, we will find maximal δ-regular sequences. Let $step_i$ denote the distance between adjacent points in a δ-regular sequence r_i. Algorithm 1 shows the method for calculating the rotational angle and the unit distance. In Algorithm 1, we use the spots detected by segmentation methods, and then we perform spot filtering to remove spurious spots. A set of spots may contain several spurious spots after filtering. However, since the probability of the maximal occurrence of a δ-regular sequence which is composed of spurious spots is very low, they are not generally considered. In extreme cases, some spurious spots may not be eliminated by our algorithm, but every automated image analysis system fails to provide for some type of some extreme case. Handling spurious spots is an important aspect of the process. However, previous systems have not consistently responded to these spots.

Next, we generate a block of $\{P_i\}$ with rotational angle, θ, and unit distance, d_u. Algorithm 2 shows the steps for block construction. Figure 5 shows an *extended δ-regular sequence* in a microarray image. Figure 5(a) shows the input point set, Figure 5(b) is the δ-regular sequences of Figure 5(a), and Figure 5(c) is the *extended δ-regular sequence*.

Let the number of expressed spots be n and the valid cells be m. The work of Andrew implies an $\Theta(n^2)$ time algorithm for all maximal regular sequences in two dimensions [9]. We calculate all of the maximal regular sequences of a set of points of *valid cells* to get the rotational angle and unit distance of the microarray image.

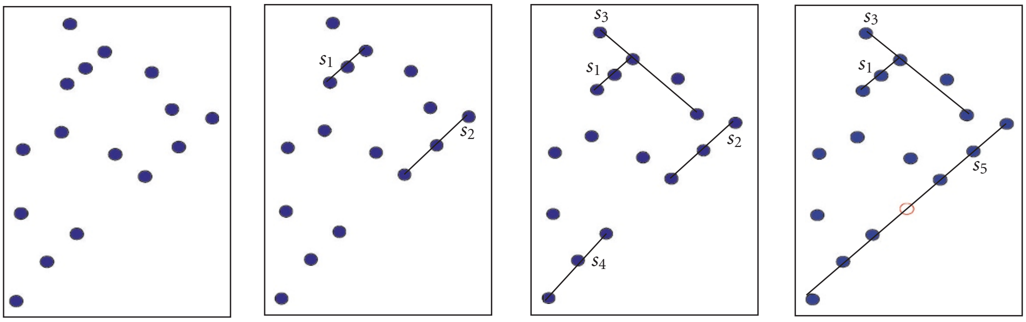
3. HIERARCHICAL METAGRID ALIGNMENT

Since the date of a single microarray experiment consists of 10 ~ 20 scanned images obtained from an identical microarray, the grid structure computed for the first image can be applied to all of the following images (especially for the duplicate experiment data). So it is reasonable to use batch processing for the scanned images obtained from an identical chip. Therefore most commercial systems (such as GenePix [11] and ImaGene [12]) provide a metagrid file to enable batch processing. A metagrid file is a template file that contains the properties (e.g., dimension, location, size, etc.) of the blocks and spots in a microarray image. Without a metagrid file, an experiment must find the spot/block index for every raw microarray image one by one. Batch processing with a metagrid proceeds as follows.

(1) Generate a metagrid template based on a base microarray image.
(2) Load one raw image file and a ready-made metagrid template.
(3) Compute the signal intensity of the image segmentation bounded by a metagrid circle for a spot.

It should be noted that the geometric properties (the physical locations of the blocks and spots) of a scanned image differ slightly from each other although they have all been obtained from the same microarray slide. This is due to the mechanical errors of scanners and experimental (manual work) error. Figure 7 shows the result of metagridding using an identical GAL file (a metagrid file provided in GenePix). Figure 7(b) shows a typical case of metagrid displacement. Clearly, the metagridding method requires some manual work.

In this section, we propose a new algorithm, *hierarchical metagrid alignment* (HMA). HMA consists of three sequential steps: chipbox, block, and spot alignments. The problem of aligning a metagrid to a given image could be considered as a point set matching problem [13]. But it is an expensive algorithm, running in $O(n^3)$ (n is point number). Applying the hierarchical alignment concept, we can easily get suboptimal alignment results by matching an image from the metagrid and an image from the chipbox area to the spot area.



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

Figure 4: (a) Input points, (b) regular sequences, (c) δ-regular sequences, (d) extended δ-regular sequences. An empty circle denotes an inserted pseudopoint.

```
Input: (i) $\{P_i\}$; a set of center points of expressed spots
(ii) $\delta$; a threshold constant given by user.
Output: rotational angle $\theta$ and unit distance $d_u$.
(1) Divide a microarray image into cells whose sizes are $2*\delta$
    by $2*\delta$ for the given $\delta$-value.
    Let $c_v$ (valid cell) refer to a cell that contains at least one
    spot of $P$.
(2) Construct a set of center points $\overline{P}$ of $c_v s$ and compute all
    maximal $\delta$-regular sequences $R = \{r_1, r_2, \dots, r_n\}$ of $\overline{P}$.
    Figure 6 shows an example of valid cell, $P_i$, and center
    points of valid cells.
    We construct only the $r_i$ that have $step_i$ smaller than the
    $step_j$ $(1 \le j \le i-1)$. If $step_i = 2 \cdot \delta$, we select $r_i$ and exit
this procedure.
(3) Select maximal $\delta$-regular sequences $\overline{R}$ which has
    the smallest step.
(4) Set $\theta$ = the angle of $\overline{R}$ to horizontal line,
    $d_u$ = distance between the adjacent points in the $\overline{R}$.
```

Algorithm 1: Computing the rotational angle of a given image and the unit distance between two adjacent spots.

3.1. Chipbox alignment

Let *ChipBox* be the minimum rectangular area including all blocks in a chip. There are two kinds of chipboxes: one is *mChipBox* from the metagrid, and the other is *fChipBox* from *fSpots* (the real image given). In the following, *mBlock* (*mSpot*) denotes the block (spot) of a metagrid and similarly, *fBlock* (*fSpot*) denotes the block (spot) of a scanned image. The *ChipBox* alignment is to determine the *fChipBox*, so *mChipBox* is aligned with *fChipBox* by matching the left upper points of the two regions. We first determine the *fChipBox* to calculate MBR of all expressed spots and then align the *mChipBox* with the *fChipBox*.

Figures 8(a) and 8(b) show the before/after snapshots of the chipbox alignment. In Figures 8(a) and 8(b), the yellow objects indicate target spots, the red rectangle is a *fChipBox* and the cyan objects are *mBlocks*.

```
Input: (i) $\{\overline{P_i}\}$; a set of center points of $c_v$,
(ii) $\theta$; rotational angle of a microarray image given,
(iii) $d_u$; unit distance of a given microarray image.
Output: block index of an input microarray.
(1) Rotate the whole image by $-\theta$ degree.
    $\{\overline{P_i}'\}$ = Rotation $(\{\overline{P_i}\}, -\theta)$.
(2) Construct the set of extended maximal regular sequences
    $R_e = \{r_{e1}, r_{e2}, \dots, r_{en}\}$ of $\overline{P}'$.
(3) Make a simple graph $G(V, E)$ from $R_e$.
    If the point $p_i$ of $r_i$ is equal to $p_j$ of $r_j$, they are connected.
    $e(v_i(\equiv p_i), v_j(\equiv p_j))$.
(4) Apply the MBR (minimum boundary rectangle) of
    each graph to the block.
(5) Re-rotate the whole image by $+\theta$ degree.
    $MBRs'$ = Rotation $(MBRs, +\theta)$.
```

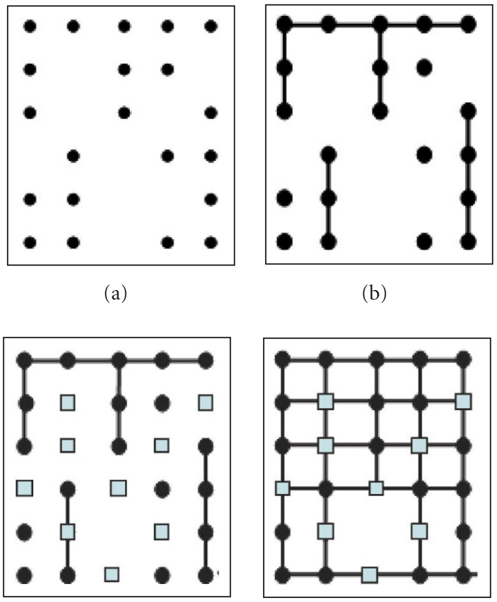
Algorithm 2: Block gridding.

3.2. Block alignment

Assume that the chipbox alignment has already been performed. Figure 9(a) shows a situation in which the *mBlock* does not correctly match the *fBlock*. So we have to align every block. Block alignment is similar to chipbox alignment. First, we divide the chipbox into *uBlocks* for detecting *fBlocks* using the gap between the nearest blocks and the block size from metagrid. Second, we assign the *fBlocks* to MBR of expressed spots in the *uBlocks* and then align the *fBlocks* with the *mBlocks*. We perform the following two steps to align the *mBlocks* with the *fBlocks*.

(1) We align the *fBlocks* which are similar in size to *mBlocks*. In this case, we align the *fBlock* to fit $mSpot_{0,0}$, which is the upper left spot in the *mBlock*, with the upper left position of the fBlock.

(2) After the first step, we calculate the upper left position of the *fBlock* using neighboring *mBlocks* which have already been aligned. And then, the *mBlock* is aligned to fit the upper left position of the *fBlock*.


(a) (b)

(c) (d)

FIGURE 5: A simple example of a grid structure constructed from an extended δ-regular sequence with input points: (a) input points, (b) a graph by δ-regular sequence merging, (c) adding pseudopoints between δ-regular sequences, (d) a grid graph by merging all extended δ-regular sequences. The shaded box points are the inserted pseudopoints.

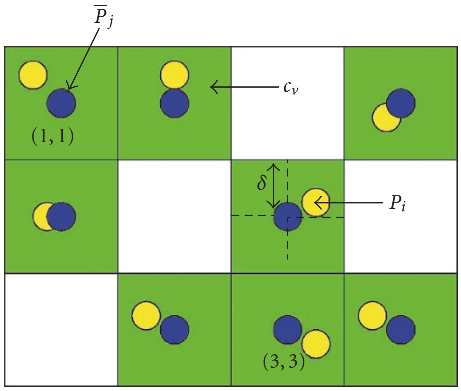
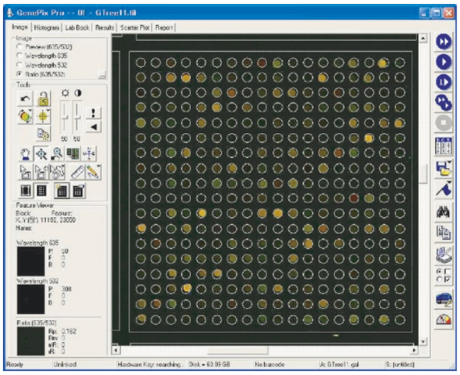


FIGURE 6: An example of $\{P_i\}$, valid cells and center points of valid cells. We construct a set of center points $\overline{P}$ of c_v (*valid cell*)s and compute all maximal δ-regular sequences $R = \{r_1, r_2, \ldots, r_n\}$ of $\overline{P}$.

Figure 9 shows the before/after snapshots of the block alignment. The red rectangle denotes the *fBlock* and the cyan objects denote the *mBlock*.

3.3. *Spot alignment*

This step assumes that the chipbox and the block are already aligned. Now we want to align the metagrid spot (*mSpot*) to the real grid spot (*fSpot*). The spot alignment in each block is the last step of HMA. In this alignment, we first have to classify *mSpots* into *active spots* and *nonactive spots*. Figure 10 shows an active spot and a nonactive spot. A *mSpot* is an active spot if it has one fSpot within the distance, *d*. After identifying all active spots, we align the *active spots* to the corresponding *fSpots*.


(a) Image I_a

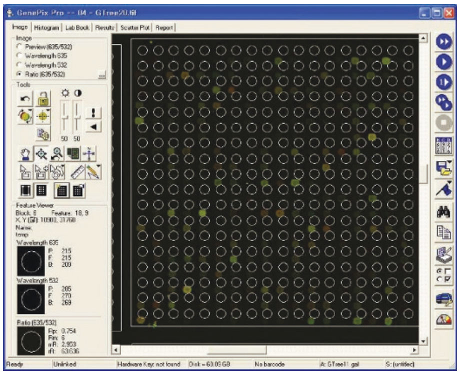

(b) Image I_b

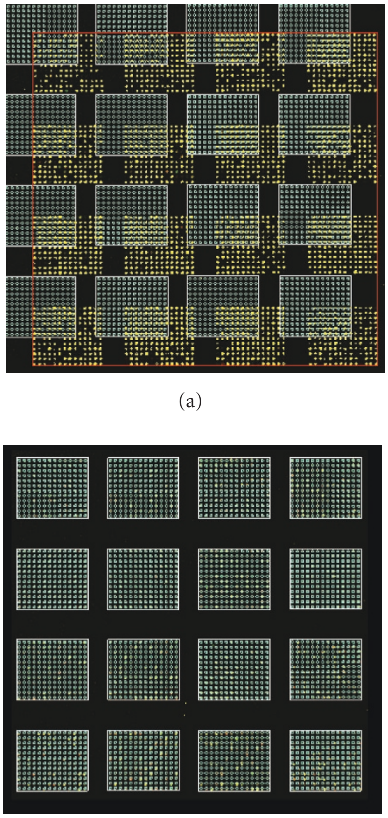
FIGURE 7: The metagridding snapshots of two images using the same GAL file in GenePix. I_a shows that the metagrid fits the given image correctly. I_b shows an image that has a discrepancy between the metagrid and the given image. Manual work is required for I_b.

Figure 11 shows the result of spot alignment. In Figure 11(b), the red circles denote *active spots* and the cyan objects are *nonactive spots*.

4. EFFECTIVENESS OF EXTENDED δ-REGULAR SEQUENCE

Generally, the length (the number of points) of an extended δ-regular sequence is expected to be longer than that of a δ-regular sequence. First, we need to know the expected size of the *extended δ-regular sequence* and the *δ-regular sequence*.

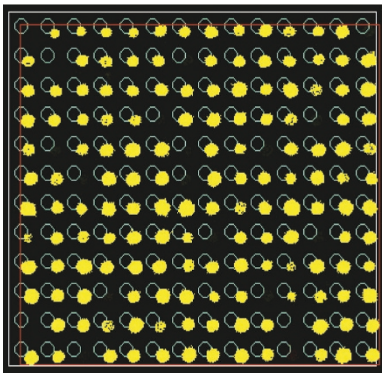
Let P_δ denote the probability that there exists at least one δ-regular sequence with length i in a point sequence S, $|S|=n$. Let s denote the expression rate of spots, and let q be the probability that a point is located in the δ box (see Figure 2). Such a δ-regular sequence can start (end) with the first (last) spot, or can be located in the middle of S. Accordingly, the



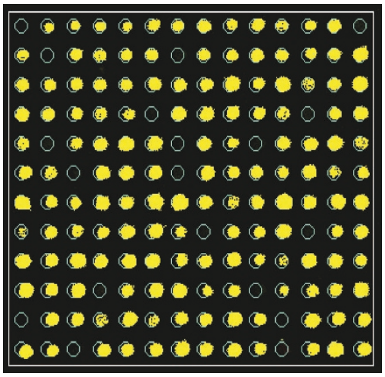
(a)

(b)

Figure 8: (a) A snapshot before chipbox alignment, (b) a snapshot after chipbox alignment. The yellow object is a *fSpot*, the red rectangle is a *fChipBox*, and the cyan object is a *mBlock*.



(a) A snapshot before block alignment



(b) A snapshot after block alignment

Figure 9: (a) A snapshot before block alignment, (b) a snapshot after block alignment. In (a) and (b), the red rectangle denotes the *fBlock* and the cyan (yellow) object denotes *mSpot* (*fSpot*), respectively.

probability, P_δ, is given as follows:

$$\begin{aligned} P_\delta &= \text{Prob}(\text{boundary}\ \delta - \text{regular sequence}) \\ &\quad + \text{Prob}(\text{middle}\ \delta - \text{regular sequence}) \\ &= \sum_{i=3}^{n-2} (s \cdot q)^i (1 - s \cdot q) + \sum_{i=3}^{n-1} (s \cdot q)^i (1 - s \cdot q)^2 \\ &= (s \cdot q)^3 (1 - s \cdot q) \\ &\quad \times \frac{\left(1 - (s \cdot q)^{n-4}\right) + (1 - s \cdot q)\left(1 - (s \cdot q)^{n-3}\right)}{(1 - s \cdot q)}. \end{aligned} \tag{1}$$

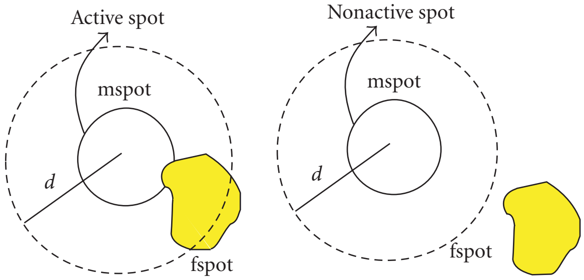
Let P_E be the probability that there exists at least one extended δ-regular sequence of length i in S. An extended δ-regular sequence includes two types of sequences: one from the original δ-regular sequence and another from the concatenation of two adjacent δ-regular sequences by adding one pseudopoint in the middle of those two δ-regular sequences. So P_E should be

$$\begin{aligned} P_E &= P_\delta + \text{Prob}(\delta - \text{regular sequence extends}) \\ &= P_\delta + \sum_{i=3}^{n-2} (s \cdot q)^{i-1} (1 - s \cdot q)^3 + \sum_{i=3}^{n-1} (s \cdot q)^{i-1} (1 - s \cdot q)^2 \\ &= P_\delta + (s \cdot q)^2 (1 - s \cdot q)^2 \\ &\quad \times \frac{(1 - s \cdot q)\left(1 - (s \cdot q)^{n-5}\right) + \left(1 - (s \cdot q)^{n-4}\right)}{(1 - s \cdot q)}. \end{aligned} \tag{2}$$

We compute the expected length of the δ-regular sequence and the extended δ-regular sequence in S with $|S| = n$, $E(L_\delta) = \sum_{k=3}^{n} k \cdot P_\delta$, and $E(L_E) = \sum_{k=3}^{n} k \cdot P_E$.

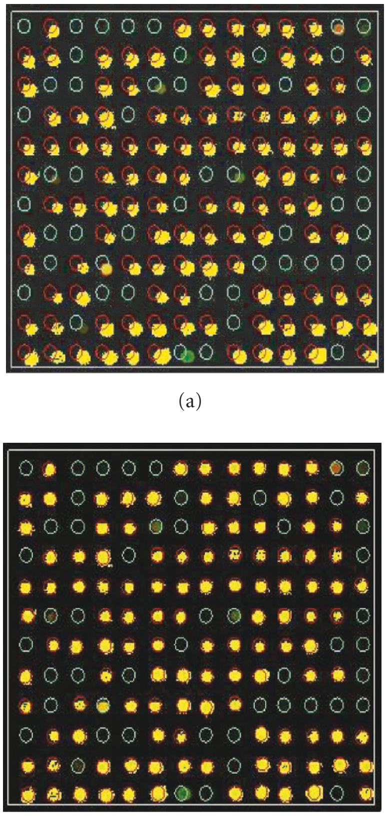
This calculation reveals the effectiveness of the "extended" δ-regular sequence versus the δ-regular sequence. The calculation shows that the extended δ-regular sequence is more than twice the length of the δ-regular sequence if the spot expression rate is low (see Table 1 and Figure 12). If the expression rate is high, there is no difference. This means that if the expression rate, s, and tolerance probability, q, approach 1, the extended δ-regular sequence should be the same as the δ-regular sequence.

In practice, the spot expression rate is about 40% to 60% (see Table 2). So we can say that this extended δ-regular sequence greatly helps to identify the block structures in practice, especially for microarray images with low expression rates.



(a) (b)

FIGURE 10: A *mSpot* is an active spot if it has one *fSpot* within distance, d. (a) An active spot, (b) a nonactive spot.



(a)

(b)

FIGURE 11: The progress of *spot alignment*. (a) A snapshot of active spots, (b) a snapshot after spot alignment. The red circles and cyan objects denote *active spots* and *nonactive spots*, respectively.

TABLE 1: Comparison of the expected length of the extended δ-regular sequence ($= E(L_E)$) and the expected length of the δ-regular sequence ($= E(L_\delta)$), where the number of point rows ($n = 20$), columns ($n = 20$), and $q = 0.9$.

$s \cdot q$	$E(L_\delta)$	$E(L_E)$	$E(L_E)/E(L_\delta)$
0.40	1.06	2.7	2.50
0.45	1.26	2.82	2.22
0.50	1.49	3.00	2.00
0.55	1.77	3.22	1.81
0.60	2.10	3.50	1.67
0.65	2.50	3.85	1.54
0.70	3.02	4.32	1.43
0.75	3.72	4.96	1.33
0.80	4.69	5.86	1.25
0.85	6.13	7.21	1.18
0.90	8.42	9.37	1.11
0.95	12.42	13.08	1.05
1.00	20.00	20.00	1.00

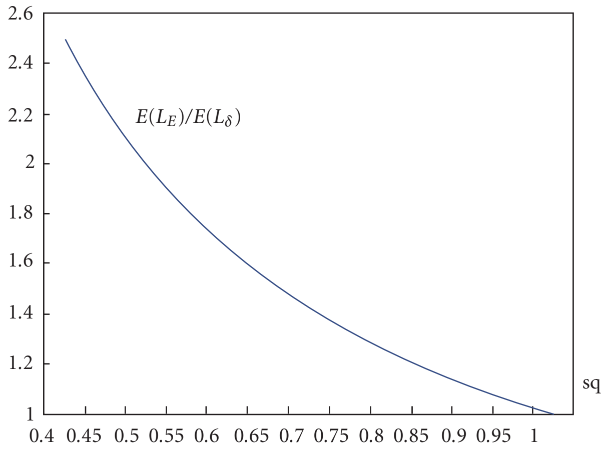


FIGURE 12: $E(L_E)/E(L_\delta)$, the ratio of the expected length of the extended δ-regular sequence to the δ-regular sequence.

We have shown that a δ-regular sequence can be extended to another longer δ-regular sequence by inserting one pseudopoint in between two disjoint and collinear δ-regular sequences. Now we will show the effectiveness of this extended δ-regular sequence for the spot indexing procedure.

Let P_a be a set of points obtained by spot image segmentation for a microarray image. As we explained above, we construct a geometric grid graph $G(P_a)$ after spot segmentation by adding edges among them. Figure 13 shows an example for segmented spots and its corresponding $G(P_a)$. If $G(P_a)$ is connected and the minimum bounding box (MBR) of $G(P_a)$ is the same as to the MBR of a single block of microarray image given, then we call $G(P_a)$ successful since we correctly separate each block in the whole microarray image. It is crucial to get a successful $G(P_a)$, which enables us to index (i, j) of each spot automatically. Otherwise, if $G(P_a)$ is disconnected or MBR of $G(P_a)$ does not cover a block region, then $G(P_a)$ is called unsuccessful for the given microarray image since we do not automatically index the spots.

Let $G_\delta(P_a)$ ($G_{e\delta}(P_a)$) be a grid graph obtained from a set of δ-regular sequences (extended δ-regular sequences). Figure 14(b) shows an unsuccessful example of $G_\delta(P_a)$ and Figure 14(d) shows a successful example of $G_{e\delta}(P_a)$. In order to give the spot index of an unsuccessful G_δ, manual intervention is required to set the index of spot.

TABLE 2: Probability functions $p_\delta(s)$ and $p_{e\delta}(s)$.

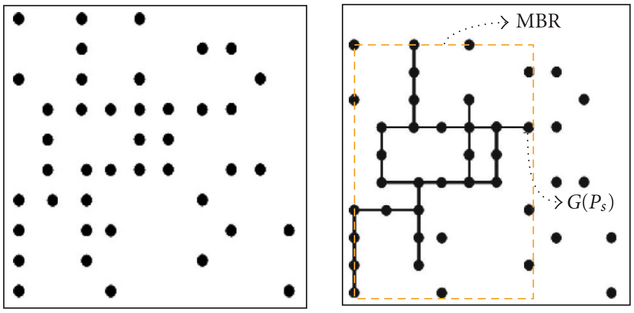
s	$p_\delta(s)$	$p_{e\delta}(s)$
0.100	0.0000	0.0000
0.200	0.0000	0.0002
0.250	0.0000	0.0018
0.275	0.0004	0.0174
0.300	0.0010	0.0578
0.325	0.0050	0.2186
0.350	0.0380	0.3832
0.375	0.0800	0.6412
0.400	0.2224	0.7784
0.425	0.3626	0.9094
0.450	0.5842	0.9544
0.475	0.7200	0.9856
0.500	0.8636	0.9922
0.525	0.9222	0.9974
0.550	0.9688	0.9982
0.575	0.9860	0.9994
0.600	0.9966	0.9994
0.625	0.9984	1.0000
0.650	0.9990	1.0000
0.675	0.9994	1.0000
0.700	1.0000	1.0000
0.900	1.0000	1.0000

Let $p_\delta(s)$ and $p_{e\delta}(s)$ be the probability functions of the expression rate s that grid graphs $G_\delta(P_a)$ and $G_{e\delta}(P_a)$ are successful, respectively. Now we want to compare the probabilities $p_\delta(s)$ and $p_{e\delta}(s)$. Since the expression rate of each spot is a random variable, $G(P_a)$ should be a kind of a random graph where the existence of each edge is dependent on a probabilistic model.

There are so many interesting results on random graph [14]. One of interesting results is about the characteristics in the probability of graph connectedness. That is the famous Erdös and Renyi theorem [15]. Let p be the probability of edge existence with n vertices graph. It is known that $p = \log n/n$ is the threshold function for graph connectedness. That means if $\lim p/(\log n/n) = 0$ implies $\lim p_\delta(s) = 0$ and if $\lim p/(\log n/n) = 1$ implies $\lim p_{e\delta}(s) = 1$, this threshold function property is one important feature of random graph.

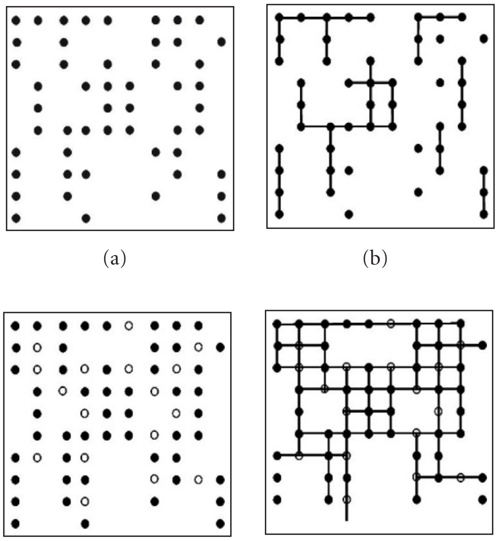
So far we did not find any probabilistic graph model which is exactly the same as microarray grid graphs. We believe that constructing a rigorous probabilistic model for this grid graph is very hard. Instead of this, we tried to apply a Monte-Carlo method to estimate the threshold value for the connectedness of random grid graph by using 50 000 artificial grid graphs.

First we generate 50 000 sets of artificial spot to simulate microarray images for each expression rate $s = 0.1, 0.2, \ldots, 0.7, 0.9$. Let P_U be the set of 50 000 point sets. Next we construct $G_\delta(P_s)$ and $G_{e\delta}(P_s)$, for each $P_s \in P_U$.



(a) (b)

FIGURE 13: (a) A set of spot points, P_a, obtained from a microarray image segmentation. (b) A case of unsuccessful $G(P_a)$ since the MBR of $G(P_a)$ does not cover the whole block.



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

FIGURE 14: Example for unsuccessful and successful cases where expression rate $s = 0.51$. (a) A set of input points, P_a. (b) A case of unsuccessful $G_\delta(P_a)$, due to 7 disconnected components. (c) A point set $P_b = P_a \cup \{\text{pseudopoints inserted}\}$. Each "∘" denotes a pseudopoint. (d) Successful $G_{e\delta}(P_a)$, which is one connected component and its MBR covers the whole block.

Table 2 shows the $p_\delta(s)$ and $p_{e\delta}(s)$ values and Figure 15 shows $p_\delta(s)$ and $p_{e\delta}(s)$ curves according to the expression rate s. In Figure 15, solid curve and dotted curve denote $p_\delta(s)$ and $p_{e\delta}(s)$, respectively. As was noted, we can see the sharp hill of the threshold value for graph connectedness. Interestingly, we can see that our extended regular sequence gives the much higher successful probability of $G_{e\delta}(P_s)$ in expression rate interval $s[0.3, 0.5]$ compared to $G_\delta(P_s)$.

It is also interesting to see that there is no difference between $p_\delta(s)$ and $p_{e\delta}(s)$, if the spot expression rate is less than 0.3 or higher than 0.7. In practice, we know that the expression rate is normally between 0.3 and 0.7. This means our extended δ-regular sequence is very helpful and effective to enable the automatic spot indexing. The ratio $r_s = p_\delta(s)/p_{e\delta}(s)$ is plotted in Figure 16.

Determining the number of pseudopoints to be inserted in a δ-regular sequence is crucial to the gridding of the whole microarray index. The more pseudopoints are allowed, the higher the probability of connectedness for a single block becomes. But this leads to an undesirable situation in which two adjacent blocks are connected into a single component, which prevents identification of the block structure. Therefore, the number of pseudopoints must be based on the distance between blocks in a scanned real microarray image.

5. EXPERIMENTAL RESULTS

We tested our method using images of four different chips (from a medical center, a university, and a biocompany). Table 3 shows the specifications of the test data set. #B and #S are the dimensions of the block and the spot of a given chip, respectively, and d_s is the diameter of the spot. gap_b and gap_s are the gap distances of adjacent blocks and spots, respectively. # Img indicates the number of images to be tested per chip.

Figure 17 shows the three different grid structures after adjusting the ratio $r_x = \text{gap}_b/(2\cdot\text{gap}_s + d_s)$. This figure shows that r_x is an important characteristic constant in obtaining a successful block/spot gridding. It is the same $r_x = 0.5$ as the distance between adjacent center points of expressed spots in the ideal microarray image. In Figure 17, $r_x = 0.25$ is too small to detect each single block and $r_x = 1.0$ is too large resulting in merged blocks. Optimal block gridding occurs when $r_x = 0.5$. This implies that the block gridding performance goes best when the distance between points in both sides of the pseudopoint in an extended δ-regular sequence.

Figure 18 shows an successful gridding result of microarray with 2×2 grid structures using our automatic gridding. In Figure 18, microarray image leans about 2 degrees. Our algorithm finds successfully the locations of all blocks and spots.

Now we will explain the efficacy of HMA. Let ΔM denote the total displacement distance of all spots to metagridding spots. And let ΔH denote the total displacement distance needed for HMA. ΔM and ΔH are computed as

$$\Delta M = \sum_{b\in\text{blocks}} \sum_{i,j\in\text{spots}} \left| m_{ij}^{(b)} - s_{ij}^{(b)} \right|, \tag{3}$$

where $m_{ij}^{(b)}$ is the physical position of a metagrid spot whose index is (i, j) in block b, and where $s_{ij}^{(b)}$ is the physical position of a real spot in the block in a scanned image.

$$\begin{aligned} \Delta H = {} & \Delta\,\text{ChipBox} + \sum_{b\in\text{blocks}} \Delta\,\text{Block}_b \\ & + \sum_{b\in\text{blocks}} \sum_{i,j\in\text{spots}} \left| \tilde{m}_{ij}^{(b)} - s_{ij}^{(b)} \right|, \end{aligned} \tag{4}$$

where $\tilde{m}_{ij}^{(b)}$ is the physical position of a metagrid spot after chipbox and block alignments. ΔChipBox and ΔBlock are the displacement distances of the chipbox and block alignments, respectively. Figure 19 compares the ΔM and ΔH, which were computed with four chips using HMA, and a straightforward metagrid overlapping. We can see that HMA drastically reduces the displacement distance by more than 90% as compared to the straightforward simple metagridding.

Figure 20 shows the rate of displacement sum after each procedure (chipbox, block, spot alignments). In Figure 20,

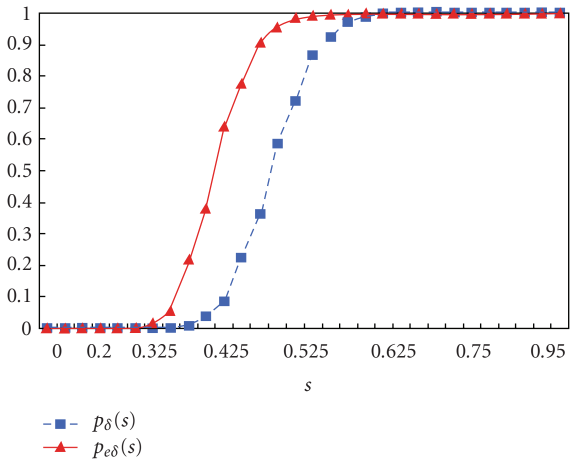


FIGURE 15: The probability functions $p_\delta(s)$ and $p_{e\delta}(s)$ according to the expression rate s. Solid and dotted curves denote the $p_\delta(s)$ and $p_{e\delta}(s)$, respectively.

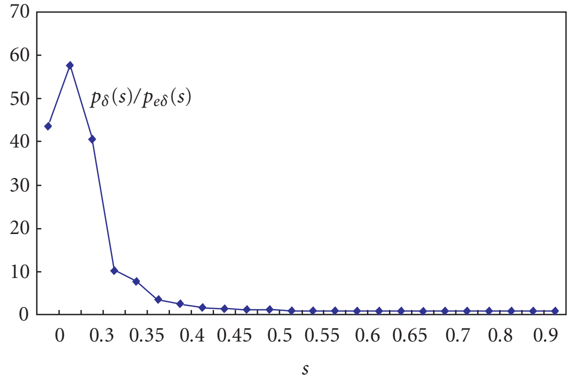
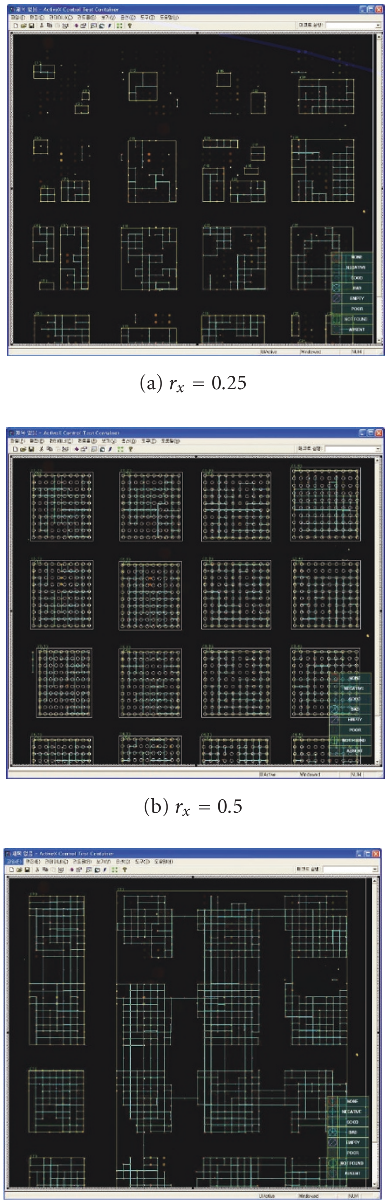


FIGURE 16: The ratio $r_s = P_\delta(s)/P_{e\delta}(s)$ with respect to the expression rate s (horizontal axis). If s is near 0.3, the $P_{e\delta}(s)$ is 60 times higher than $P_\delta(s)$. If $s = 0.325$, the r_s is about 40.

TABLE 3: The specification of the test data set.

Chip	#B	#S	d_s	gap_b	gap_s	# Img
A	4 × 4	12 × 14	14	98.58	12.20	5
B	4 × 4	10 × 10	16	90.75	18.90	8
C	4 × 4	18 × 18	14	34.13	8.53	8
D	4 × 4	10 × 10	16	91.66	18.81	8



(a) $r_x = 0.25$

(b) $r_x = 0.5$

(c) $r_x = 1.0$

Figure 17: r_x and its corresponding autogridding result in chip B. (a) $r_x = 0.25$ does not detect the block clusters. (b) $r_x = 1.0$ is too large resulting in merged blocks. (c) $r_x = 0.5$ gives the optimal block gridding.

the metagridding results before HMA (none) have many displacement distances, but the further the alignment processes in HMA proceed, the better the results are. HMA finally attains the ideal gridding results.

6. CONCLUSION

It is very important to develop an automated and intelligent system for analyzing microarray images. The contributions of this paper are as follows.

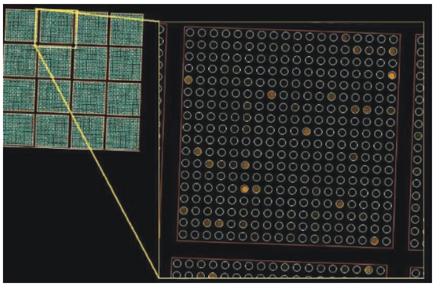


Figure 18: The correct gridding result. This image consists of 4 × 4 blocks with 18 × 18 spots each, and leans about 2 degrees.

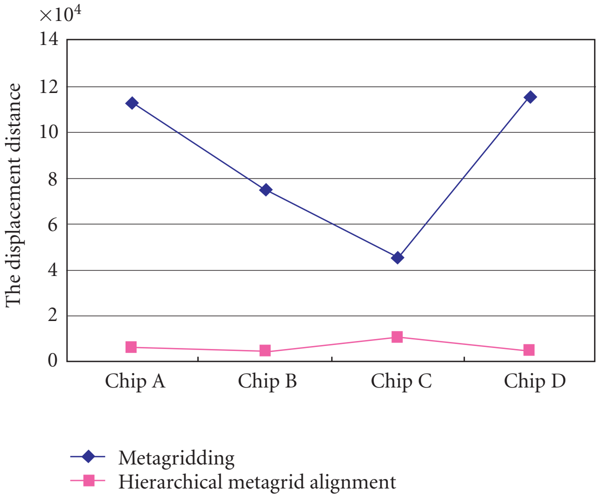


Figure 19: Total displacement distance required by metagridding and our HMA.

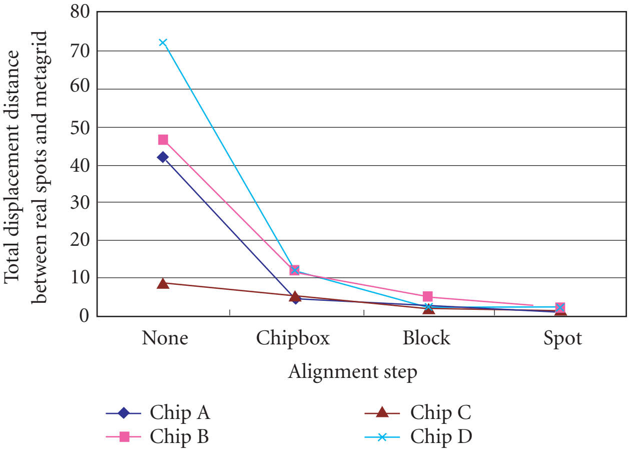


Figure 20: Reduction of displacement distances after {chipbox, box, spot} alignment.

(i) The autogridding method using the *extended δ-regular point sequence* is reliable for index blocks and spots, which is useful for minimizing manual work.

(ii) HMA (hierarchical metagrid alignment) is a novel method for processing microarray image batches between all real spots and metagrid spots, and reduces the total displacement distance by more than 90% as compared to straightforward metagridding methods (e.g., GenePix style).

We are developing a more rigid probabilistic model for the extended δ-regular sequence. It is well known that the probability of connectedness for a random graph approaches 1 when the edge probability of the random graph is above $O(\log n/n)$. It is easy to see that the graph model constructed from the microarray point sequences for block/spot indexing is a bipartite graph. So it is instructive to determine the probability of bipartite graph connectedness when the edge probability, p, is given. We also use Monte-Carlo simulation method to estimate the probability of the connectedness of the grid graph. It is also a very interesting problem to establish the complete probabilistic model for the grid graph obtained from microarray experiment. The supplemental information is available on our website (http://jade.cs.pusan.ac.kr/~gridding).

ACKNOWLEDGMENTS

This research was supported by a Grant (M10529000001-05N2900-00110) from Strategic National R&D Program of Ministry of Science and Technology. We gratefully credit the thoughtful reviewers who provided substantial constructive criticism on an earlier version of this paper.

REFERENCES

[1] D. J. Duggan, M. L. Bittner, Y. Chen, P. Meltzer, and J. M. Trent, "Expression profiling using cDNA microarrays," *Nature genetics*, vol. 21, pp. 10–14, 1999.

[2] D. Shalon, S. J. Smith, and P. O. Brown, "A DNA microarray system for analyzing complex DNA samples using two-color fluorescent probe hybridization," *Genome Research*, vol. 6, no. 7, pp. 639–645, 1996.

[3] A. A. Alizadeh, M. B. Eisen, R. E. Davis, et al., "Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling," *Nature*, vol. 403, no. 6769, pp. 503–511, 2000.

[4] 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, 2001.

[5] R. Hirata Jr., J. Barrera, R. F. Hashimoto, and D. O. Dantas, "Microarray gridding by mathematical morphology," in *Proceedings of 14th Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI '01)*, pp. 112–119, Florianópolis, Brazil, October 2001.

[6] 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. suppl 2, pp. S141–S151, 2002.

[7] G. Kauer and H. Blöcker, "Analysis of disturbed images," *Bioinformatics*, vol. 20, no. 9, pp. 1381–1387, 2004.

[8] T. Srinark and C. Kambhamettu, "A microarray image analysis system based on multiple-snake," *Journal of Biological Systems Special Issue*, vol. 12, no. 4, pp. 202–209, 2004.

[9] A. B. Kahng and G. Robins, "Optimal algorithms for extracting spatial regularity in images," *Pattern Recognition Letters*, vol. 12, no. 12, pp. 757–764, 1991.

[10] G. Robins, B. L. Robinson, and B. S. Sethi, "On detecting spatial regularity in noisy images," *Information Processing Letters*, vol. 69, no. 4, pp. 189–195, 1999.

[11] GenePix, http://www.axon.com.

[12] ImaGene, http://www.biodiscovery.com/imagene.asp.

[13] T. S. Caetano, T. Caelli, and D. A. C. Barone, "An optimal probabilistic graphical model for point set matching," in *Proceedings of Joint IAPR International Workshops on Structural, Syntactic, and Statistical Pattern Recognition (S+SSPR '04)*, pp. 162–170, Lisbon, Portugal, August 2004.

[14] J. Spencer, *Ten Lectures on the Probabilistic Method*, SIAM, Philadelphia, Pa, USA, 1990.

[15] B. Bollobás, *Random Graphs*, Cambridge University Press, Cambridge, UK, 2001.

Hee-Jeong Jin received her B.S. degree in 2000 from Pusan National University, South Korea, the M.S. degree in 2002 from Pusan National University, South Korea. From 2002 to 2003, she had been in National Genome Research Institute, KNIH, and since 2003, she has been a Ph.D. candidate in Pusan National University, South Korea. Her research interest is bioinformatics (analysis of ppi, comparative genomics, and microarray gridding).

Bong-Kyung Chun received his B.S. degree in 2003 from Pusan National University, South Korea, the M.S. degree in 2005 from Pusan National University, South Korea, and since 2005 he has been a Ph.D. student in Pusan National University, South Korea. His research interests are bioinformatics and computer graphics.

Hwan-Gue Cho received his B.S. degree in 1984 form Seoul National University, South Korea, the M.S. degree in 1986 from Korea Advanced Institute of Science and Technology, South Korea, and the Ph.D. in 1990 from Korea Advanced Institute of Science and Technology, South Korea. Since 1990 he has been a Professor in Pusan National University, South Korea. His research interests are graphics (visualization) and bioinformatics (sequence alignment and bionetwork analysis).