Automatic Microarray Spot Segmentation Using a Snake-Fisher Model

Jinn Ho and Wen-Liang Hwang

***Abstract*—Inspired by Paragious and Deriche's work, which unifies boundary-based and region-based image partition approaches, we integrate the snake model and the Fisher criterion to capture, respectively, the boundary information and region information of microarray images. We then use the proposed algorithm to segment the spots in the microarray images, and compare our results with those obtained by commercial software. Our algorithm is automatic because the parameters are adaptively estimated from the data without human intervention.**

***Index Terms*—Microarray Image, Spot Segmentation.**

I. Introduction

As the DNA microarray can measure thousands of gene expression levels on the genomic scale simultaneously, it has enormous potential for biological, medical, and industrial applications [1], [2]. Specific cDNA, or oligonucleotide fragments of genes, are spotted or printed on an array matrix as probes to detect gene expressions. Meanwhile, samples of mRNAs are reverse transcribed to cDNAs, which are labelled with fluorescent dyes to act as targets. The labelled cDNA targets are then hybridized to probes by complementation. After washing out the unhybridized targets, a laser scanner is used to detect the fluorescent intensities, which are proportional to the contents of hybridized pairs of targets and probes. This process generates microarray images that indicate the relative expression levels of the genes. Finally, image processing techniques and statistical methods are applied to determine the expression levels of the spots in the microarrays in order to perform gene expression analysis.

According to Yang et al. [3], the processing of microarray images involves spot gridding, segmentation, and intensity extraction. The spot gridding task detects the positions of the spot centers and identifies their coordinates [4]. Existing commercial software provides semi-automatic algorithms to deal with the problem. An accurate and automatic algorithm for the case where the spot centers are smoothly distorted is provided in [5]. The goal of segmentation is to classify a pixel as either foreground inside the spot, or as background outside the spot. A number of segmentation techniques have been proposed [6], [7], some of which assume that the geometry of a spot is either a fixed circle or an adaptive circle [8]. However, the assumption is incorrect because a spot's morphology is not always a circle. Other techniques use hypothesis testing to segment the foreground and background [9], but this necessitates modeling the pixel intensity distributions, which is a difficult task. Region growing based on the watershed algorithm proposed in *Spot* [3] can segment regions of irregular shape and the technique does not need to model a region's probability; however, the segmentation results are not necessarily an optimization of some class separation criteria. The objective of the intensity extraction task is to calculate and normalize the spot intensity in order to derive quality measurements [10]. The segmentation task is the focus of the present study.

Objects or homogeneous regions can be segmented by a region-based approach, which usually minimizes a class separation criterion according to the statistics of the regions [11]. Another popular approach based on the boundary information uses the gradients along a contour to segment objects. The pioneering work in [12] integrated two approaches within a minimization framework that finds the boundary of deformable objects in medical images. Related works that integrate boundary-based and region-based approaches for image partitioning can be found in [13], [14], [15]. Paragios and Deriche [16] proposed a systematic framework that combines the region-based and boundary-based segmentation approaches. The model can be generalized by incorporating specific functions that can capture boundary and region properties. For the spot segmentation task, we propose using the snake model to capture boundary information and the Fisher criterion to capture region information. The snake model [17] is very effective in segmenting objects whose boundaries can be approximately delineated by a set of large gradient points along a contour. The spot boundary is such an example. The Fisher criterion is based on discriminate analysis in statistics, which uses between-class and within-class statistics to form a criterion for class separation [18]. We adopt the Fisher criterion because it is simple and can be analyzed mathematically.

The solution of a snake model depends on the initial contour and the parameter values selected. Because an image has various signal-to-noise (SNR) levels, a good initial contour and the parameter values must be determined manually. Even after extensive study [19], [20], [21], these problems have not been resolved completely. Thus, the snake operation is a semi-automatic process. Since microarray images contain an enormous number of spots, a semi-automatic process severely degrades the throughput of microarray analysis. To resolve these difficulties, we first modify the Markov chain Monte

Jinn Ho is with the Institute of Information Science and Genomics Research Center, Academia Sinica, Taiwan

Wen-Liang Hwang is with the Institute of Information Science, Academia Sinica, Taiwan

Carlo-based Climber algorithm to find a good initial contour [22], [23], and then estimate the values of our parameters from that contour. Although the proposed approach does not resolve the problems theoretically, experiments on several synthesized and natural images show that it can find a good initial contour and estimate quality parameters. Using the proposed method, we segment the spots in microarray images with various SNR levels, and compare our results with those of $GenePix\ Pro$ 5.0 [8] and $Spot$ 2.0 [3].

The remainder of the paper is organized as follows. In the next section, we introduce our model. In Section III, we present an automatic algorithm that finds a solution for our model. In Section IV, we validate the model by comparing it with other approaches. Finally, in Section V we present our conclusions.

II. Description of the Model

For simplicity, we assume there are only two regions to be delaminated. However, the proposed model can be easily extended to delaminate more than two regions simultaneously. For example, let us assume there are two non-overlapping foreground regions, F_1 and F_2, with a background region B, and that we have an algorithm that can delaminate both a foreground region and a background region. We can estimate F_1 approximately and obtain $\hat{F_1}$, and then use the algorithm to delaminate F_2 and $B - \hat{F_1}$. After that, we use the result of estimating F_2 and the algorithm to delaminate F_1 and $B - F_2$. Clearly, if we have an algorithm that can delaminate a foreground and a background region, it can also delaminate an image with more than two regions. Hence, in the following analysis we focus on detecting the most significant foreground region in an image.

A. Energy Form

We define $R = \{I(x, y)\}$ as an image of gray value pixels. A simple closed curve $\Gamma = \Gamma(s)$ on R divides the image into $\{R_1, R_2\}$, where $R = R_1 \cup R_2$ and $\Gamma = \partial R_1 \cap \partial R_2$. We denote M_1 and M_2 as the expected values of pixels in R_1 and R_2, respectively. The total energy induced by contour Γ is defined as the sum of the snake's energy and the region's energy. The former measures the properties along the contour, while the latter measures the statistical differences between the regions separated by the contour. The total energy E_{total} is written as

$$E_{total} = E_{snake} + \tilde{\gamma} E_{region}. \tag{1}$$

We use the two-class Fisher discriminate criterion to represent an E_{region}. In such cases, (1) can be expressed as

$$E_{total}(\Gamma) = \int_\Gamma (\frac{\alpha}{2}|\Gamma_s|^2 + \frac{\beta}{2}|\Gamma_{ss}|^2 - \| \bigtriangledown I\|^2)ds$$
$$+\tilde{\gamma}\frac{\iint_{R_1}(I-M_1)^2dxdy + \iint_{R_2}(I-M_2)^2dxdy}{(M_1-M_2)^2}, \tag{2}$$

in which

$$E_{snake}(\Gamma) = \int_\Gamma (\frac{\alpha}{2}|\Gamma_s|^2 + \frac{\beta}{2}|\Gamma_{ss}|^2 - \| \bigtriangledown I\|^2)ds, \tag{3}$$

$$E_{region}(\Gamma) = \frac{\iint_{R_1}(I-M_1)^2dxdy + \iint_{R_2}(I-M_2)^2dxdy}{(M_1-M_2)^2} \tag{4}$$

where (3) and (4) are respectively the snake energy model and the Fisher criterion model. The latter can be written as $E_{region} = E_{within}/E_{between}$, where the within-class distance E_{within} measures the scatter of samples in R_1 and R_2 around their expected values as follows:

$$\begin{aligned}
E_{within} &= \iint_{R_1}(I-M_1)^2dxdy + \iint_{R_2}(I-M_2)^2dxdy \\
&= \iint_{R_1}(I-M_1)^2dxdy + \iint_{R-R_1}(I-M_2)^2dxdy \\
&= \iint_{R_1}(I-M_1)^2dxdy + \iint_{R}(I-M_2)^2dxdy \\
&\quad - \iint_{R_1}(I-M_2)^2dxdy \\
&= \left[\iint_{R_1}(I-M_1)^2dxdy - (I-M_2)^2dxdy\right] \\
&\quad + \iint_{R}(I-M_2)^2dxdy,
\end{aligned} \tag{5}$$

while the between-class distance $E_{between} = (M_1 - M_2)^2$ is the difference between the expected gray levels of R_1 and R_2.

We propose an iterative algorithm to find the solution contour of (2). The algorithm begins with an initial contour; then, at each iteration, a new contour is obtained by alternating the subsequent stages. In the first stage, by fixing the values of the between-class distance $E_{between}(\Gamma)$, $\iint_R (I - M_2)^2dxdy$ (the rightmost term in (5)), and the model's parameters α, β, and $\tilde{\gamma}$, the algorithm finds the curve $\hat{\Gamma}$ that minimizes

$$\begin{aligned}
E_{total}(\Gamma) &= \int_\Gamma (\frac{\alpha}{2}|\Gamma_s|^2 + \frac{\beta}{2}|\Gamma_{ss}|^2 - \| \bigtriangledown I\|^2)ds \\
&\quad +\frac{\tilde{\gamma}}{E_{between}}\iint_{R_1}[(I-M_1)^2 - (I-M_2)^2]dxdy \\
&\quad +\frac{\tilde{\gamma}}{E_{between}}\iint_{R}(I-M_2)^2dxdy.
\end{aligned} \tag{6}$$

In the second stage, we update $E_{between}$ and $\iint_R (I - M_2)^2dxdy$ with respect to $\hat{\Gamma}$. The parameter values are then estimated by minimizing the mean square error (MSE) of the Euler equation in (6). Note that (6) can be solved by using other methods, such as the level set approach [24], [25]. The Euler equation is derived as follows.

B. Euler Equation

Let us rewrite (6) as

$$E_{total}(\Gamma) = E(\Gamma) + \frac{\tilde{\gamma}}{E_{between}}\iint_R (I-M_2)^2dxdy,$$

where

$$\begin{aligned}
E(\Gamma) &= \int_\Gamma (\frac{\alpha}{2}|\Gamma_s|^2 + \frac{\beta}{2}|\Gamma_{ss}|^2 - \| \bigtriangledown I\|^2)ds \\
&\quad +\frac{\tilde{\gamma}}{E_{between}}\iint_{R_1}[(I-M_1)^2 - (I-M_2)^2]dxdy.
\end{aligned} \tag{7}$$

Because M_2 is the mean value of region R_2, the region does not change much if the initial contour is good. Therefore, the value can be approximated as a constant and we can omit the term $\frac{\tilde{\gamma}}{E_{between}}E_{between}\iint_R (I-M_2)^2dxdy$ from the following analysis.

Green's theorem states that

$$\begin{aligned}
\iint_{R_1}(\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y})dxdy &= \int_{\partial R_1} Pdx + Qdy \\
&= \int_{\partial R_1}(Px_s + Qy_s)ds.
\end{aligned} \tag{8}$$

By applying the theorem to the last term in (7) and setting $\gamma = \frac{\tilde{\gamma}}{E_{between}}$, we obtain

$$\begin{aligned} E(\Gamma) &= \int_\Gamma (\frac{\alpha}{2}|\Gamma_s|^2 + \frac{\beta}{2}|\Gamma_{ss}|^2 - \| \bigtriangledown I\|^2) ds \\ &\quad + \gamma \int_\Gamma L(s; v_s, v_{ss}) ds, \\ &= \int_\Gamma F ds, \qquad (9) \end{aligned}$$

where

$$\begin{aligned} F(s; v, v_s, v_{ss}) &= (\frac{\alpha}{2}|\Gamma_s|^2 + \frac{\beta}{2}|\Gamma_{ss}|^2 - \| \bigtriangledown I(v)\|^2) \\ &\quad + \gamma L(s; v, v_s) \qquad (10) \end{aligned}$$

in which $v : [0, 1] \rightarrow \mathbb{R}^2$; $v(s) = (x(s), y(s)) = \Gamma(s)$; and $x, y \in C^2([0, 1])$. Using functional calculus with the detailed deviations given in Appendix I, the Euler equation becomes

$$-\frac{\partial\| \bigtriangledown I\|^2}{\partial x} - \alpha x_{ss} + \beta x_{ssss} + \gamma[(I - M_1)^2 - (I - M_2)^2]y_s = 0,$$

$$-\frac{\partial\| \bigtriangledown I\|^2}{\partial y} - \alpha y_{ss} + \beta y_{ssss} - \gamma[(I - M_1)^2 - (I - M_2)^2]x_s = 0.$$

To evaluate the third term in each of the above equations requires a fourth derivative. However, if that derivative is implemented in the discrete domain, there may be a large amount of numerical noise [26], [27]. Thus, we implement our method in a special case of the above, where $\beta = 0$, which has better numerical stability. The Euler equation for $\beta = 0$ is

$$-\frac{\partial\| \bigtriangledown I\|^2}{\partial x} - \alpha x_{ss} + \gamma[(I - M_1)^2 - (I - M_2)^2]y_s = 0, \qquad (11)$$

$$-\frac{\partial\| \bigtriangledown I\|^2}{\partial y} - \alpha y_{ss} - \gamma[(I - M_1)^2 - (I - M_2)^2]x_s = 0. \qquad (12)$$

The solution of the above two equations can be obtained by an iterative procedure similar to that in [17]; thus, we do not state it here. Because $\Gamma_{ss} = [x_{ss} \quad y_{ss}] = \kappa\vec{n}$, where κ is the curvature, and $\vec{n}$ is parallel to $[y_s \quad - x_s]$, (11) and (12) can be written as one equation:

$$-\nabla\| \bigtriangledown I\|^2 - \alpha\kappa\vec{n} - \gamma'[\frac{(I - M_1)^2 - (I - M_2)^2}{(M_1 - M_2)^2}]\vec{n} = 0. \qquad (13)$$

This equation requires that a point on the optimal contour must satisfy (14) in the tangent direction ($\vec{t}$), and (15) in the normal direction ($\vec{n}$):

$$\begin{aligned} \nabla\| \bigtriangledown I\|^2 \cdot \vec{t} &= 0, \qquad (14) \\ -\nabla\| \bigtriangledown I\|^2 \cdot \vec{n} &= \alpha\kappa + \gamma'[\frac{(I - M_1)^2 - (I - M_2)^2}{(M_1 - M_2)^2}]. \qquad (15) \end{aligned}$$

Equation (15) indicates that the optimal contour balances three terms: the first term is provided by the normal component of the gradients of the image, the second term is proportional to the curvature, while the last term measures the class separation.

III. Solution of Our Model

To solve the Euler equation, we need the initial contour and the model's parameters. First, we describe the methods used to obtain the initial contour and estimate the parameters. We then present an iterative algorithm that derives the solution of our model.

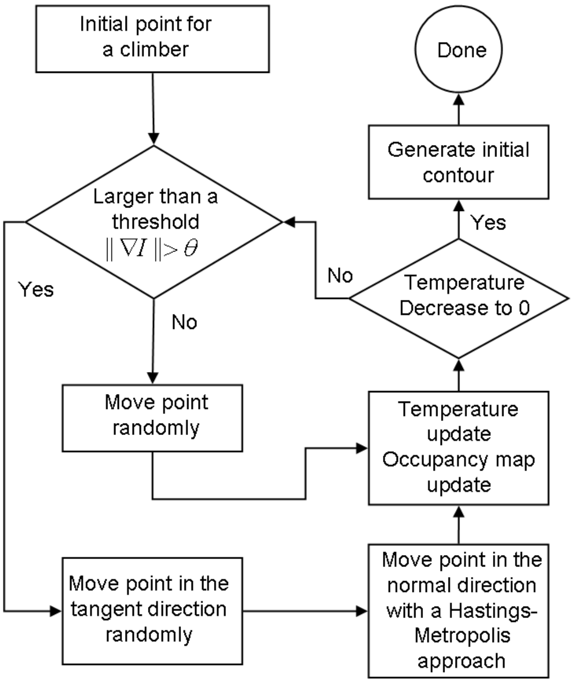
Initial point for a climber
Larger than a threshold $\| \nabla I \| > \theta$
Yes
No
Move point randomly
Move point in the tangent direction randomly
Move point in the normal direction with a Hastings-Metropolis approach
Temperature update Occupancy map update
Temperature Decrease to 0
No
Yes
Generate initial contour
Done

Fig. 1. Block diagram of the Climber algorithm.

A. Initial Contour Detection

The snake-balloon approach tries to solve the initial contour problem by adding an external force to the snake model [28]. We adopt a different approach based on the Climber algorithm [22], [23], which is easy to implement and remarkably robust against noise. The algorithm was originally proposed to stochastically estimate contours of ridge points in a time-frequency plane by randomly placing a large number of independent climbers in a time-frequency plane. Each climber moves with equal probability in the time direction; however, its movement in the frequency direction is restricted to climbing the peaks of a local energy function by a Hastings-Metropolis penalization and a temperature schedule similar to that in the simulated annealing algorithm. Thus, as the temperature approaches zero, the climber stops and settles on a suitable ridge contour. By assuming that most movements can stop close to a ridge, the points of concentrated occupation form ridges on the plane. The ridges of the plane are then detected by the occupation measure of the algorithm, which counts the number of times the climbers stop at each point in the time-frequency plane.

Since our goal is to identify the contour of an image, instead of obtaining the smooth horizontal ridges in a time-frequency plane, we need to modify the algorithm. Our modification is based on the observation that the contours of an image are mostly concentrated on the edge points, which can be defined as the local maxima of the magnitude along the gradient direction of an image. Thus, the movement of our climber in the gradient direction is restricted by the unit vector $\overrightarrow{n} = \frac{\nabla I}{\|\nabla I\|}$. However, each climber can move freely in the tangent direction, defined by the unit vector $\overrightarrow{t}$, which is perpendicular to the gradient direction. The flowchart of the algorithm is given in Figure 1. The steps are as follows:

Step 0. Let M be the number of climbers, and $C : \mathcal{R}^2 \to \mathcal{R}$ be the occupation measurement, with an initial value of zero at any point.

Step 1. Let $t = 1$ and $T_t = 1$ be the initial temperature. A climber randomly selects an initial point $p(T_t) = (i, j)$ in image I.

Step 2. Let θ to be a threshold. For a climber at $p(T_t) = (i, j)$ with temperature T_t, if $\|\nabla I\| < \theta$, the movement of the climber is determined by randomly selecting one of the 8-neighbor pixels of $p(T_t)$ that have equal probability. Otherwise, the movement of the climber is determined as follows:

A. Free Movement
Move (i, j) to $(i, j)_t = (i, j) + u_t(i, j)$ in the tangent direction, where $u_t(i, j) \in \{+\overrightarrow{t}(i, j), -\overrightarrow{t}(i, j)\}$ with equal probability.

B. Restricted Movement
Move $(i, j)_t$ to $(i, j)_n = (i, j)_t + u_n(i, j)$ in the normal direction, where $u_n(i, j) \in \{0, \overrightarrow{n}(i, j), -\overrightarrow{n}(i, j)\}$. Let $(i, j)'_n = (i, j)_t + v_n(i, j)$, where $v_n(i, j) \in \{\overrightarrow{n}(i, j), -\overrightarrow{n}(i, j)\}$, $Z = \|\nabla I\|^2$, and $\delta Z = Z((i, j)'_n) - Z((i, j)_t)$. The climber moves to $(i, j)_n = (i, j)'_n$ when the value of the function Z increases, *i.e.*, $\delta Z > 0$; otherwise, if $\delta Z \leq 0$, the transition is made with probability $exp(\frac{\delta Z}{T_t})$. Thus, in the case where the climber does not move, i.e., $(i, j)_n = (i, j)_t$, the probability is $1 - exp(\frac{\delta Z}{T_t})$.

C. Update the temperature $t = t + 1$, $T_t = \frac{1}{t}$, and set $p(T_t) = (i, j)_n$.

Step 3. Stop the the climber when the temperature approaches zero, and go to **Step 4**. If the temperature is lower than a threshold T_ϵ, we record the movement of the climber at $p(T_t)$ by letting $C(p(T_t)) \leftarrow C(p(T_t)) + 1$. Go to **Step 2**.

Step 4. $M = M - 1$. If $M \neq 0$, go to **Step 1**.

Having obtained the occupation measurement C, we associate it with closed contours by setting a threshold for it and retaining the positions that accumulate larger values than the threshold in C. The resultant image is denoted as $\hat{C}$. We then use the linking procedure in [29] to find the contours in $\hat{C}$. Initially, we "unmark" all the points in $\hat{C}$ and apply the following procedure several times to find the contours.

- We select one "unmarked" point, and calculate its tangent.
- Then, along that tangent direction we mark the best "unmarked" neighbor point that has the largest occupation value.
- We then iterate the process from the newly marked points until only "marked" neighbor points exist, or the contour is closed.

The result is a series of contours. We discard contours that are not closed, or whose length is too small. Since our goal is to segment the foreground of a spot in an microarray image, the t-test, according to [3], is a good criterion for measuring the result. Thus, we apply the t-test to the remaining contours, and retain the one with the largest t-test result as the initial contour. Note that we can retain more than one contour as an initial contour. Figure 2 illustrates the results of applying the Climber algorithm to a heart-shaped image. One can observe the evolution of the contour as the temperature approaches zero. Figure 3 shows the initial contour results derived by applying different numbers of climbers to a synthesized noisy image. As illustrated in the figure, the initial contour does not change much because the number of climbers is large enough. Figure 4 shows a noisy synthesized image with different intensity levels. There are eight closed contours in the image. The climber algorithm can obtain the approximation of all the closed contours, as shown in the bottom subfigure.

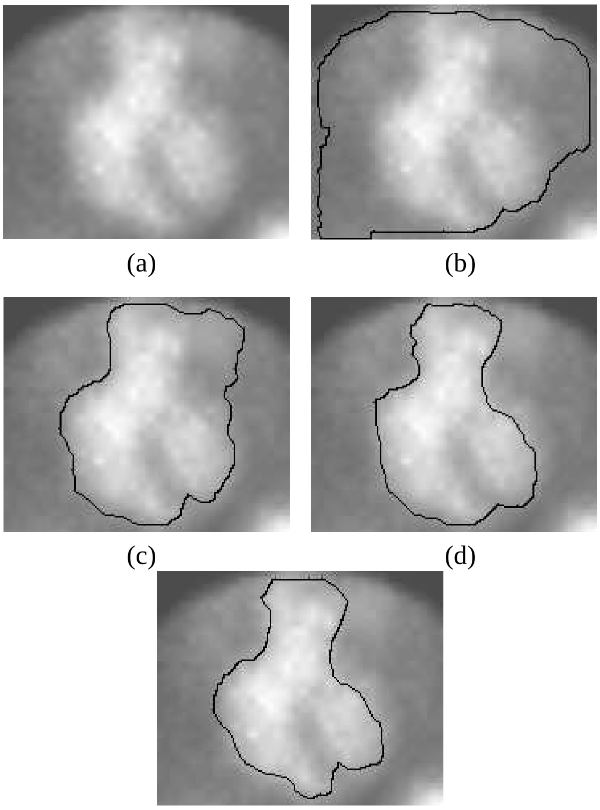

Fig. 2. The steps in the evolution of a climber's contour: (a) The input image; (b) The contour is superimposed on (a) to detect when the temperature reaches $T_0/2$, where T_0 is the initial temperature; (c) when the temperature reaches $T_0/4$; (d) when the temperature reaches $T_0/8$; and (e) the final result.

B. Parameter Estimation

After obtaining the initial contour, we need to determine the values of the parameters. A contour is the solution of our model if we can find the values of the parameters such that the contour and the values satisfy the Euler equation. For the case where there are no suitable values, we estimate the parameters by minimizing the mean-square-error (MSE) of the Euler equation with respect to the contour.

To estimate (α, γ) of a closed curve, we first select the sample pixels Γ_1. Let $\Gamma_1 = \{(x(i), y(i))| \ i = 1, \cdots, s\}$ be the sample points of the given contour, and $K(i) = [(I(x(i), y(i)) - M_1)^2 - (I(x(i), y(i)) - M_2)^2]$. The MSE $e_1^2(\Gamma_1)$ and $e_2^2(\Gamma_1)$ of (11) and (12) are, respectively,

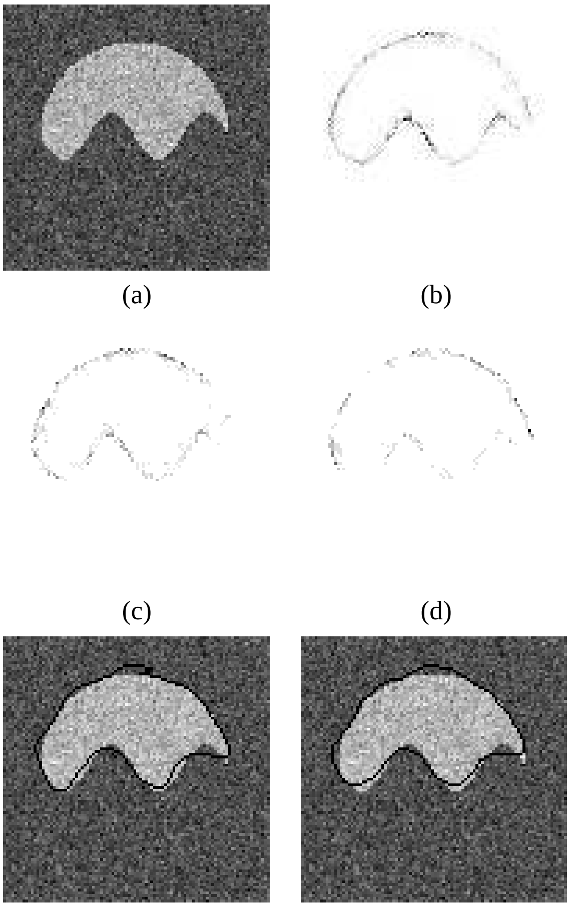

Fig. 3. (a) A noisy image of size 96 × 96 pixels; (b), (c), and (d) are the occupation measurements obtained by using 576, 196 and 100 climbers, respectively; (e) and (f) are the initial contours derived from measurements (b) and (c), respectively. Note that a closed contour is not derived from measurement (d).

$$e_1^2(\Gamma_1) = \frac{1}{s}\sum_i \Big[-\alpha x_{ss}(i) - \frac{\partial \| \bigtriangledown I \|^2}{\partial x(i)} + \gamma K(i) y_s(i) \Big]^2,$$

$$e_2^2(\Gamma_1) = \frac{1}{s}\sum_i \Big[-\alpha y_{ss}(i) - \frac{\partial \| \bigtriangledown I \|^2}{\partial y(i)} - \gamma K(i) x_s(i) \Big]^2.$$

The (α^*, γ^*) that minimizes the MSE satisfies $\frac{\partial e_1^2}{\partial \alpha} = 0$, $\frac{\partial e_1^2}{\partial \gamma} = 0$, $\frac{\partial e_2^2}{\partial \alpha} = 0$, and $\frac{\partial e_2^2}{\partial \gamma} = 0$. We then obtain the following linear system:

$$\alpha(\sum_i x_{ss}(i)) - \gamma(\sum_i K(i) y_s(i)) + \sum_i \frac{\partial \| \bigtriangledown I \|^2}{\partial x(i)} = 0,$$

$$\alpha(\sum_i y_{ss}(i)) + \gamma(\sum_i K(i) x_s(i)) + \sum_i \frac{\partial \| \bigtriangledown I \|^2}{\partial y(i)} = 0.$$

If the denominator of the above system is not zero, then the analytical solution of the parameters $\alpha^*(\Gamma_1)$ and $\gamma^*(\Gamma_1)$ can be derived easily.

C. Alternative Refinement Algorithm

Figure 5 shows the flowchart of the proposed algorithm used to find the solution of our model. After initial contour detection, the algorithm estimates the optimal parameters that minimize the MSE of the contour, and then solves the Euler equation by using the contour and the parameters to obtain a new contour. The statistics of the region partitioned by the new contour are then updated, after which the parameters are updated. Based on the obtained contour, and the updated statistics and parameters, the algorithm solves the Euler equation at each iteration and generates a new contour. The process is repeated until a certain stopping criterion is reached.

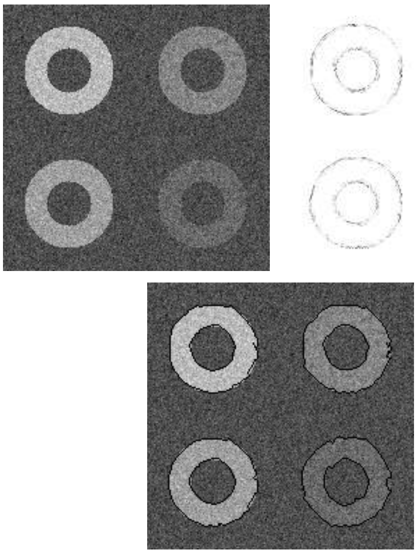

Fig. 4. Top left: A 192×192 image with different intensity levels and eight closed contours. The SNR of the image is $5dB$ and the noise is additive white Gaussian noise. Top right: The occupation measurement of the Climber algorithm. Bottom: The contours obtained by the occupation measurement in the top right subfigure.

There are many ways to define the stopping criterion of our algorithm. For example, we can define it as the point when the change in the model's parameters in consecutive iterations is smaller than a given threshold. We use the t-test of the interior and exterior regions separated by the contour as the criterion for stopping. The algorithm stops if the t-test value of the current contour is smaller than that of the previous contour. We use t-test because it is a simple way to assess whether the means of the interior and exterior regions are statistically different. Also, the test provides a good result when it is used as a criterion for segmenting spots in a microarray image. Figure 6 illustrates the results of applying our algorithm to a noisy synthesized image.

IV. Performance Evaluation

We conduct experiments to evaluate the performance of our algorithm on several images, including synthesized data, medical images, and the microarray images of different manufacturing techniques. The experiment parameter for θ is 15% and the threshold for obtaining $\hat{C}$ is the top 10% of the occupation measure in C. We use the following numerical approximations: $x_s(i) \simeq x(i+1) - x(i)$ and $x_{ss}(i) \simeq x(i+1) - 2x(i) + x(i-1)$. We sample at every four points along a contour and use the sampled points to calculate the iterative algorithm numerically. The upper and lower bounds

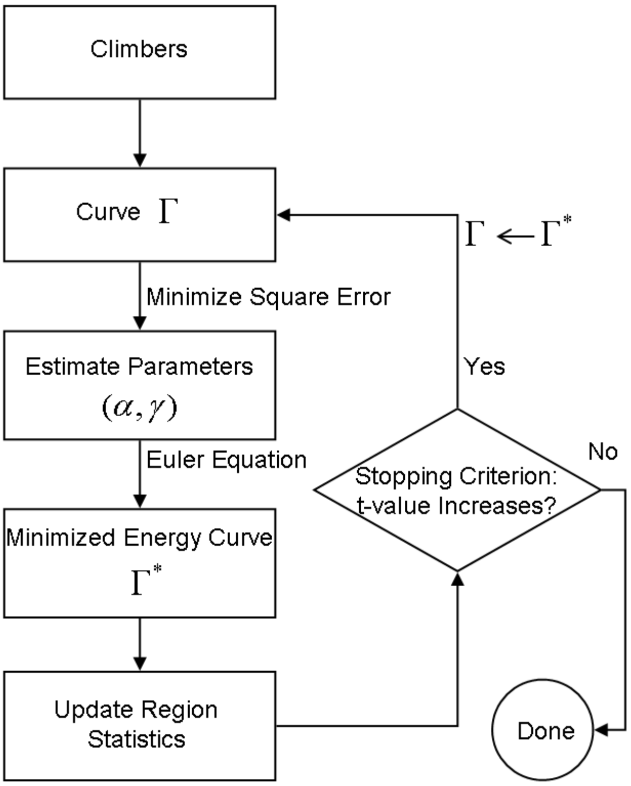
Climbers
Curve Γ
Minimize Square Error
Estimate Parameters (α,γ)
Euler Equation
Minimized Energy Curve Γ^*
Update Region Statistics
$\Gamma \leftarrow \Gamma^*$
Yes
Stopping Criterion: t-value Increases?
No
Done

Fig. 5. Block diagram of our approach. We use the Climber algorithm to find the initial contour. The parameters are then generated, and the contour's energy is minimized. The process is repeated after the statistics of the regions have been modified in the minimizing energy step.

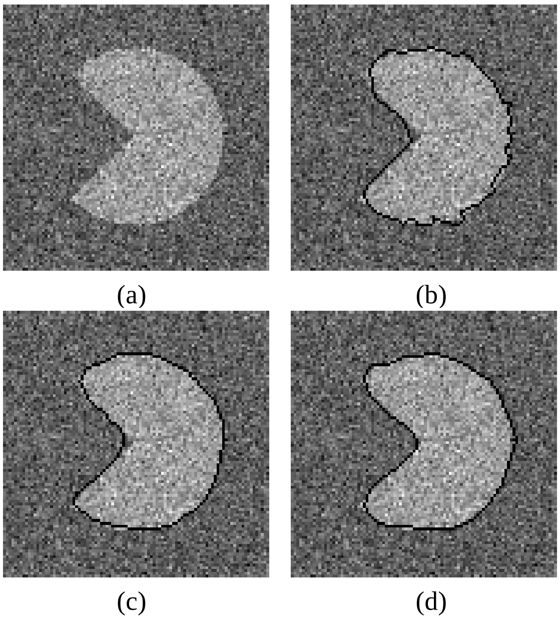

Fig. 6. (a) Noisy data of a 96×96 pixel image with $SNR = 5dB$. (b) The initial contour obtained by the Climber algorithm, where $(M_1, M_2) = (250.344, 2.0756)$ and the t-test value is 90.829. (c) The contour obtained after the first iteration, where $(\alpha, \gamma, M_1, M_2) = (3.0, 0.8746, 249.5908, 1.9105)$ and the t-test value is 90.8930. (d) The result of the second iteration, where $(\alpha, \gamma, M_1, M_2) = (0.8, 1.2, 249.2892, 1.7105)$ and the t-test value is 91.1054. Our algorithm stops after two iterations.

for the values of the estimated parameters γ^* and α^* are set such that their values are truncated to the bound if they are out of range.

Figures 7 and 8 illustrate, respectively, the step-by-step application of our algorithm to a synthesized image with 5 dB of additive Gaussian noise and an image obtained from Internet. Figures 7(c) and 7(d) are the results of applying the snake algorithm (by setting $\gamma = 0$) with different α values. Figure 7(e) is the result of setting $\alpha = 0$ in our algorithm. Figure 8(d) illustrates the result of the snake algorithm, where the value of the parameter α is the same as that obtained by our algorithm. Comparing the results in Figures 8(d) and 8(e), we observe that the proposed algorithm yields a better contour for the image. Note that the Climber algorithm produces remarkably good initial results in all cases.

Finally, we evaluate and compare our spot segmentation results with those obtained by other algorithms for three sets of microarray images. Two of the sets contain some poor quality images from the Stanford Microarray Database (SMD) [30], while the third contains Agilent 60-mer oligonucleotide microarrays whose specifications are given on the related web pages [31]. The Agilent microarrays are some of the best quality oligonucleotide chips available commercially. In the experiments, we separate each spot region from adjacent regions manually, and process the segmentation algorithm inside each region. Because the spot centers between the SMD and oligonucleotide images are small (15 and 10 pixels, respectively), we cannot apply our algorithm directly to segment the foreground and background of a spot. Therefore, we enlarge each spot image by interpolation so that its size is four times that of the original spot. We then apply our algorithm to detect the contours of the enlarged spot image. Finally, the spot boundaries are obtained by down-sampling the contours detected in the enlarged image to their original size. To evaluate the performance of the proposed algorithm, we compare it with the representative image analysis methods and software in $GenePix\ Pro\ 5.0$, which detects spots by circular boundary adjustment, and $Spot\ 2.0$, which detects spot regions by seed region growing. For the different segmentation results, we calculate the two-sample t-test value between the gray level pixels in the foreground and background, and use it to assess the performance of a segmentation algorithm. The t-test assesses whether the means of two groups are statistically different; the larger the t-test value, the better the segmentation result obtained. Figures 9 and 10 show the sub-blocks, the initial contours obtained by the Climber algorithm, and the results of applying our algorithm to microarray images. Note that each spot in Figure 9 contains a single connected component, while some spots in Figure 10 may contain holes. Figures 11 and 12 compare the t-test values of our methods and those of the other methods. As shown by the figures, the distributions of the t-values of our method are statistically larger than those of the other methods, which indicates that the contours derived by our method generally yield better segmentation results.

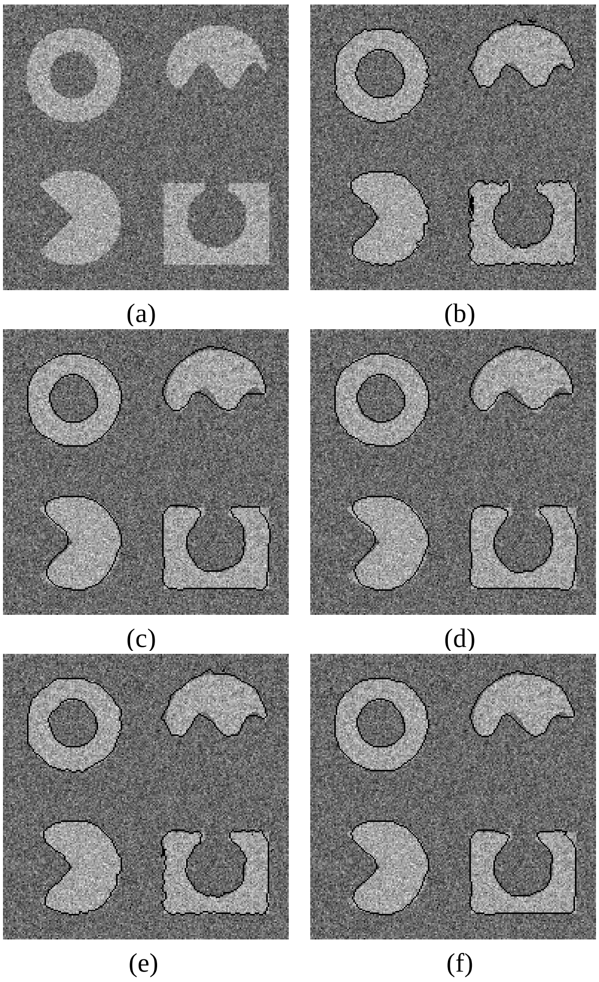

Fig. 7. (a) A 192×192 noisy image of components with different shapes. (b) The initial contours obtained by the Climber algorithm for the images in subfigures (c),(d),(e), and (f). (c) The snake algorithm's result with $\alpha = 1.5$. (d) The snake algorithm's result with $\alpha = 2.4$. (e) Our algorihm's result with $\alpha = 0$. (f) Our algorithm's result with $(\alpha, \gamma) = (0.8, 1)$.

V. Conclusion

We have integrated the snake model and the Fisher criterion to segment spots in microarray images. The initial contour is obtained by the robust Climber algorithm. The segmentation problem is then solved with an iterative algorithm, where the parameters of our model and the contour are modified alternately until the t-value of the regions cannot be improved further. We compare our results with those of the snake algorithm for noisy synthesized images and natural images. The proposed algorithm's performance is superior because it is automatic and can segment the spots of microarray images without human intervention. The experiment results on microarray data manufactured by different techniques also demonstrate that our algorithm outperforms other methods. Although we use the Fisher criterion for class separation, other discriminate analysis terms can be used in our model after appropriate modifications. An interesting research direction would be to analyze the properties and theoretical results of the Climber algorithm.

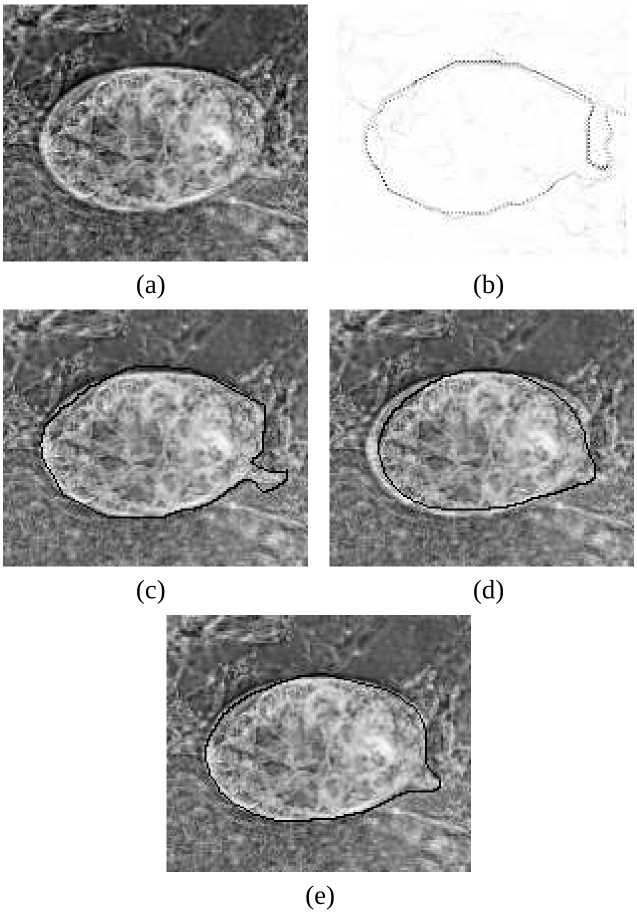

Fig. 8. (a) An egg image of size 129×153 pixels. (b) The occupation measurment C of the Climber algorithm. (c) The initial contour of (d) and (e) obtained from(b). (d) The snake algorithm's result with $\alpha = 2.97$. (e) Our algorithm's result with $(\alpha, \gamma) = (2.97, 7.02)$.

Appendix A
Derivations of Euler Equation

Here, we derive the Euler equation for the energy function in (6). For convenience, in the following derivations, a' and $\frac{\partial a}{\partial s}$ denote the same notation ($a' = \frac{\partial a}{\partial s}$). Using variational calculus, for any C^2 function $\eta : [0,1] \to \mathbb{R}^2$, we let

$$\tilde{E}(\epsilon) = \int_\Gamma F(s; v + \epsilon\eta, v_s + \epsilon\eta', v_{ss} + \epsilon\eta'')ds, \tag{16}$$

and perform the following approximations:

$$\begin{aligned} & F(s; v + \epsilon\eta, v_s + \epsilon\eta', v_{ss} + \epsilon\eta'') \\ & \approx F(s; v, v_s, v_{ss}) + \frac{\partial F}{\partial v}\epsilon\eta + \frac{\partial F}{\partial v_s}\epsilon\eta' + \frac{\partial F}{\partial v_{ss}}\epsilon\eta'', \quad (17) \\ & \frac{d}{d\epsilon}F(s; v + \epsilon\eta, v_s + \epsilon\eta', v_{ss} + \epsilon\eta'') \\ & \approx \frac{\partial F}{\partial v}\eta + \frac{\partial F}{\partial v_s}\eta' + \frac{\partial F}{\partial v_{ss}}\eta''. \quad (18) \end{aligned}$$

From (16), (17), and (18), we obtain

$$\begin{aligned} \tilde{E}(\epsilon) &= \int F(s; v + \epsilon\eta, v_s + \epsilon\eta', v_{ss} + \epsilon\eta'')ds \\ &= \int \left[F(s; v, v_s, v_{ss}) + \frac{\partial F}{\partial v}\epsilon\eta + \frac{\partial F}{\partial v_s}\epsilon\eta' + \frac{\partial F}{\partial v_{ss}}\epsilon\eta''\right]ds \\ \frac{d\tilde{E}(\epsilon)}{d\epsilon} &= \int \left[\frac{\partial F}{\partial v}\eta + \frac{\partial F}{\partial v_s}\eta' + \frac{\partial F}{\partial v_{ss}}\eta''\right]ds. \quad (19) \end{aligned}$$

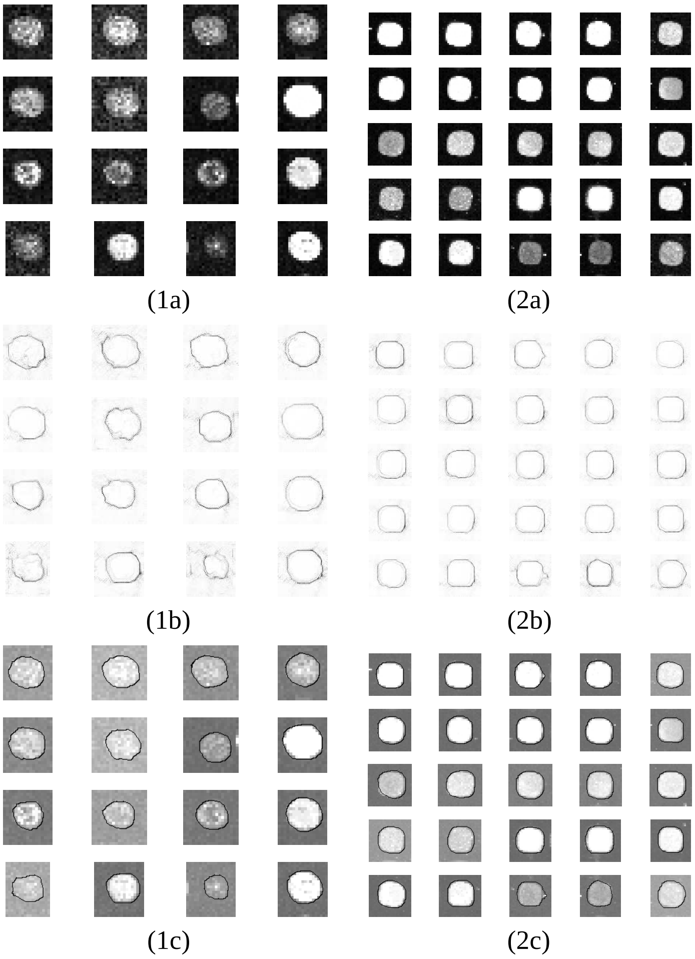

Fig. 9. (1a) the spots in a subblock of the image $LC23N085$ in the SMD; (2a) the spots in a subblock of an oligonucleotide image; (1b) and (2b) are the respective initial contours of (1a) and (2a); and (1c) and (2c) are the results of our algorithm derived by segmenting (1a) and (2a) respectively.

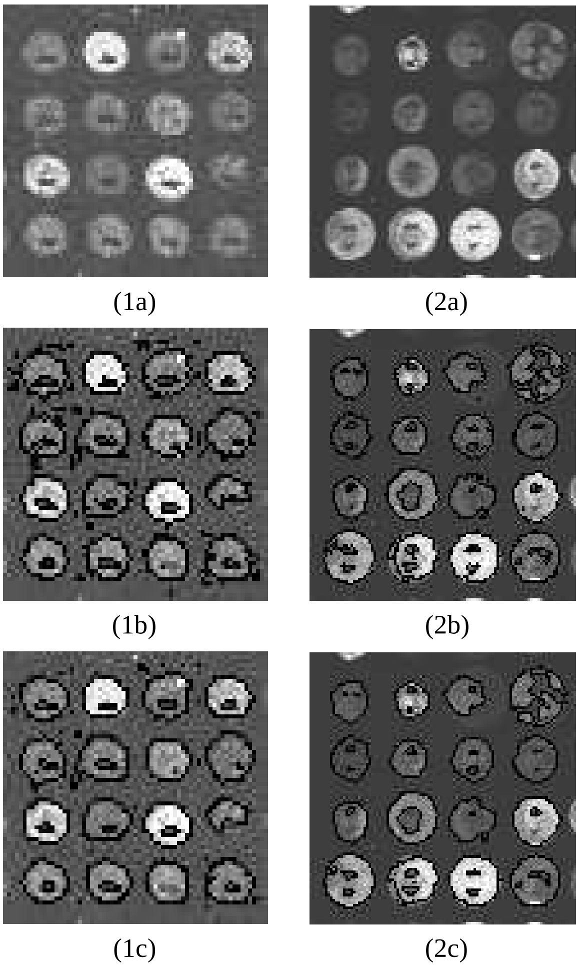

Fig. 10. (1a) the spots in a subblock of the image $LC23N085$ in the SMD; (2a) the spots in a subblock of the image $hp7004b$ in the SMD; (1b) and (2b) are the respective initial contours of (1a) and (2a); and (1c) and (2c) are the results of our algorithm derived by segmenting (1a) and (2a) respectively.

The second and third terms on the right-hand side of the above equation can be extended as follows. For the second term, since

$$\frac{\partial F}{\partial v_s}\eta' = \left(\frac{\partial F}{\partial v_s}\eta\right)' - \left(\frac{\partial F}{\partial v_s}\right)'\eta \Rightarrow \int \frac{\partial F}{\partial v_s}\eta' + \int \left(\frac{\partial F}{\partial v_s}\right)'\eta = 0,$$

we have

$$\int \frac{\partial F}{\partial v_s}\eta' = -\int \left(\frac{\partial F}{\partial v_s}\right)'\eta. \qquad (20)$$

For the third term, since

$$\frac{\partial F}{\partial v_{ss}}\eta'' = \left(\frac{\partial F}{\partial v_{ss}}\eta'\right)' - \left(\frac{\partial F}{\partial v_{ss}}\right)'\eta',$$

it is implied that

$$\int \frac{\partial F}{\partial v_{ss}}\eta'' + \int \left(\frac{\partial F}{\partial v_{ss}}\right)'\eta' = 0. \qquad (21)$$

The second term of (21) can be further developed as follows. According to

$$\left(\frac{\partial F}{\partial v_{ss}}\right)'\eta' = \left(\left(\frac{\partial F}{\partial v_{ss}}\right)'\eta\right)' - \left(\frac{\partial F}{\partial v_{ss}}\right)''\eta,$$

we obtain

$$\begin{aligned} &\int \left(\frac{\partial F}{\partial v_{ss}}\right)'\eta' = -\int \left(\frac{\partial F}{\partial v_{ss}}\right)''\eta \\ \Rightarrow \quad &\int \left(\frac{\partial F}{\partial v_{ss}}\right)''\eta + \int \left(\frac{\partial F}{\partial v_{ss}}\right)'\eta' = 0. \qquad (22) \end{aligned}$$

With the results of (21) and (22), we have

$$\int \left(\frac{\partial F}{\partial v_{ss}}\right)\eta'' = \int \left(\frac{\partial F}{\partial v_{ss}}\right)''\eta. \qquad (23)$$

Finally, we substitute (20) and (23) into (19) to obtain

$$\begin{aligned} \frac{d\tilde{E}(\epsilon)}{d\epsilon} &= \int \frac{\partial F}{\partial v}\eta - \int (\frac{\partial F}{\partial v_s})'\eta + \int (\frac{\partial F}{\partial v_{ss}})''\eta \\ &= \int \left[\frac{\partial F}{\partial v} - (\frac{\partial F}{\partial v_s})' + (\frac{\partial F}{\partial v_{ss}})''\right]\eta = 0. \qquad (24) \end{aligned}$$

The above equation is zero for all η; hence,

$$\frac{\partial F}{\partial v} - (\frac{\partial F}{\partial v_s})' + (\frac{\partial F}{\partial v_{ss}})'' = 0, \qquad (25)$$

where $F(s;v,v_s,v_{ss}) = \frac{\alpha}{2}|\Gamma_s|^2 + \frac{\beta}{2}|\Gamma_{ss}|^2 - \|\bigtriangledown I\|^2 +$

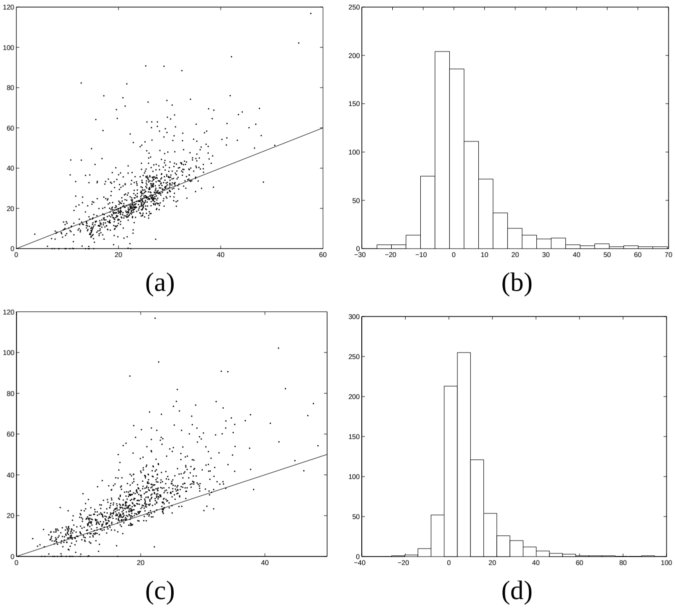

Fig. 11. Comparison of the t-test values of different methods. The data set comprises the spots of subblock $(2, 1)$ of the $LC23N085$ microarray image, which contains $784 = 28 \times 28$ spots. (a) The scattered plots of the t-test values, where the x-axis represents $Spot$, and the y-axis represents our algorithm. (b) The histogram of the t-value difference between our algorithm and that of $Spot$. The distribution is to the right of the origin, indicating that our t-value is statistically larger than that of $Spot$. The mean of the difference is 3.6 and the standard deviation is 12.2. (c) The scattered plots of the t-values of our algorithm (y-axis) and those of $GenePix\ Pro$ 5.0 (x-axis). (d) The histogram of the t-value difference between our algorithm and that of $GenePix\ Pro$ 5.0. Again the distribution is to the right of zero, and our t-values have a larger distribution. The mean of the difference is 8.7 and the standard deviation is 11.4.

$\gamma L(v, v_s)$. We re-write the force function F as follows:

$$\begin{aligned} F(s; v, v_s, v_{ss}) &= F(s; x, y, x_s, y_s, x_{ss}, y_{ss}) \\ &= \frac{\alpha}{2}[{x_s}^2 + {y_s}^2] + \frac{\beta}{2}[{x_{ss}}^2 + {y_{ss}}^2] \\ &\quad -\| \bigtriangledown I\|^2 + \gamma L(v, v_s). \end{aligned} \tag{26}$$

Let F_1 and F_2 be defined such that $F = F_1 + F_2$ with

$$\begin{aligned} F_1(s; v, v_s, v_{ss}) &= \frac{\alpha}{2}[{x_s}^2 + {y_s}^2] + \frac{\beta}{2}[{x_{ss}}^2 + {y_{ss}}^2] \\ &\quad -\| \bigtriangledown I\|^2, \qquad (27) \\ F_2(s; v, v_s) &= \gamma L(s; v, v_s). \qquad (28) \end{aligned}$$

Now (25) can be re-written as

$$\begin{aligned} & \frac{\partial F_1}{\partial x} - \left(\frac{\partial F_1}{\partial x_s}\right)' + \left(\frac{\partial F_1}{\partial x_{ss}}\right)'' \\ + \quad & \frac{\partial F_2}{\partial x} - \left(\frac{\partial F_2}{\partial x_s}\right)' + \left(\frac{\partial F_2}{\partial x_{ss}}\right)'' = 0, \\ & \frac{\partial F_1}{\partial y} - \left(\frac{\partial F_1}{\partial y_s}\right)' + \left(\frac{\partial F_1}{\partial y_{ss}}\right)'' \\ + \quad & \frac{\partial F_2}{\partial y} - \left(\frac{\partial F_2}{\partial y_s}\right)' + \left(\frac{\partial F_2}{\partial y_s}\right)'' = 0. \end{aligned} \tag{29}$$

Then, from the definition of F_1, we have:

$$\begin{aligned} & \frac{\partial F_1}{\partial x} - \left(\frac{\partial F_1}{\partial x_s}\right)' + \left(\frac{\partial F_1}{\partial x_{ss}}\right)'' \\ = \quad & -\frac{\partial \| \bigtriangledown I\|^2}{\partial x} - \alpha x_{ss} + \beta x_{ssss}. \end{aligned} \tag{30}$$

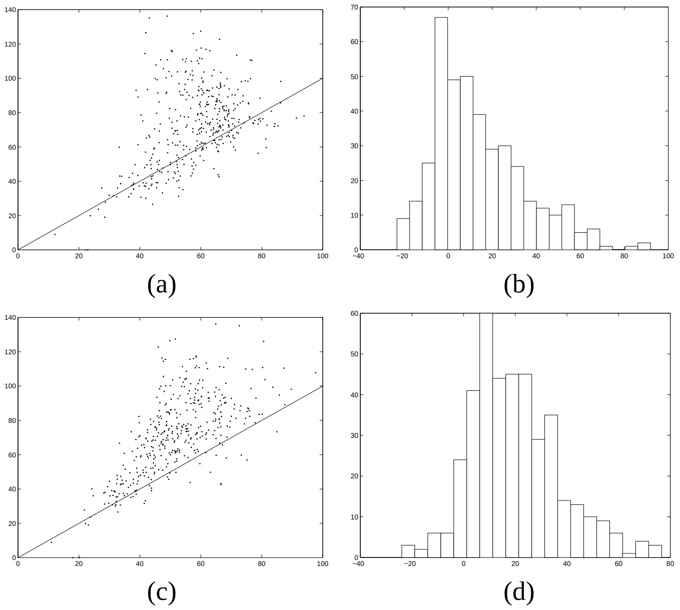

Fig. 12. Comparison of the t-test values of different methods. The data set comprises the 400 spots of a subblock of an oligonucleotide microarray image. (a) The scattered plots of the t-test value, where the x-axis represents $Spot$, and the y-axis represents our algorithm. (b) The histogram of the t-value difference between our algorithm and that of $Spot$. The distribution is to the right of the origin, indicating that our t-value is statistically larger than that of $Spot$. The mean of the difference is 14.2 and the standard deviation is 20.8. (c) The scattered plots of the t-values of our algorithm (y-axis) and those of $GenePix\ Pro$ 5.0 (x-axis). (d) The histogram of the t-value difference between our algorithm and that of $GenePix\ Pro$ 5.0. Again the distribution is to the right of zero, and our t-values have a larger distribution. The mean of the difference is 20.1 and the standard deviation is 17.3.

From (6) and (8), we can write $L(s; v, v_s) = Px_s + Qy_s$. Therefore, $F_2 = \gamma L(s; v, v_s) = \gamma[Px_s + Qy_s]$. As a consequence, we have

$$\begin{aligned} & \left(\frac{\partial F_2}{\partial x_{ss}}\right)'' = 0, \\ & \frac{\partial F_2}{\partial x} - \left(\frac{\partial F_2}{\partial x_s}\right)' = \gamma\left[\frac{\partial L}{\partial x} - \left(\frac{\partial L}{\partial x_s}\right)'\right]. \end{aligned} \tag{31}$$

According to the definition of L, we have

$$\frac{\partial L}{\partial x} = \frac{\partial}{\partial x}\Big[Px_s + Qy_s\Big] = \frac{\partial P}{\partial x}x_s + \frac{\partial Q}{\partial x}y_s, \tag{32}$$

$$\left(\frac{\partial L}{\partial x_s}\right)' = P'. \tag{33}$$

Substituting the results of (32) and (33) into (31), we have

$$\begin{aligned} \frac{\partial F_2}{\partial x} - (\frac{\partial F_2}{\partial x_s})' &= \gamma\Big[\frac{\partial L}{\partial x} - (\frac{\partial L}{\partial x_s})'\Big] \\ &= \gamma\Big[\frac{\partial P}{\partial x}x_s + \frac{\partial Q}{\partial x}y_s - P'\Big]. \end{aligned} \tag{34}$$

Since

$$P' = \frac{dP}{ds} = \frac{\partial P}{\partial x}x_s + \frac{\partial P}{\partial y}y_s,$$

from (8) we can derive

$$\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} = (I - M_1)^2 - (I - M_2)^2.$$

Therefore, (34) can be re-written as

$$\begin{aligned}\frac{\partial F_2}{\partial x} - \left(\frac{\partial F_2}{\partial x_s}\right)' &= \gamma\left[\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y}\right]y_s \\ &= \gamma[(I-M_1)^2 - (I-M_2)^2]y_s. \quad (35)\end{aligned}$$

Similarly, we apply the above calculations to the y-coordinate and obtain

$$\frac{\partial F_1}{\partial y} - (\frac{\partial F_1}{\partial y_s})' + (\frac{\partial F_1}{\partial y_{ss}})'' = -\frac{\partial \| \bigtriangledown I\|^2}{\partial y} - \alpha y_{ss} + \beta y_{ssss}, \quad (36)$$

$$\frac{\partial F_2}{\partial y} - (\frac{\partial F_2}{\partial y_s})' = -\gamma[(I-M_1)^2 - (I-M_2)^2]x_s. \quad (37)$$

Finally, we summarize (29), (30), (35), (36), and (37) to derive the final formula of the Euler equation:

$$-\frac{\partial \| \bigtriangledown I\|^2}{\partial x} - \alpha x_{ss} + \beta x_{ssss} + \gamma[(I-M_1)^2 - (I-M_2)^2]y_s = 0,$$

$$-\frac{\partial \| \bigtriangledown I\|^2}{\partial y} - \alpha y_{ss} + \beta y_{ssss} - \gamma[(I-M_1)^2 - (I-M_2)^2]x_s = 0.$$

Acknowledgment

We would like to express our gratitude to Professor Wen-Hsiung Li of the University of Chicago for his insightful suggestions.

References

[1] M. B. Eisen, P. O. Brown, "DNA Arrays for Analysis of Gene Expression", *Methods Enzymol 303,* 179-205 (1999).

[2] M. Schena, D. Shalon, R. W. Davis, P. O. Brown, "Quantitative Motoring of Gene Expression Patterns with a Complementary DNA Microarray", *Science,* Oct 20;270(5235):467-70, 1995.

[3] Y. H. Yang, M. J. Buckley, S. Dudoit, T. P. Speed, "Comparison of Methods for Image Analysis on cDNA Microarray Data", *Journal of Comuptational and Grahical Statistics,* 11:108-136, 2002.

[4] J. Buhler, T. Ideker, D. Haynor "Dapple: Improved Techniques for Finding Spots on DNA Microarray", *UW CSE Technical Report UWTR 2000-08005*, 2000. http://www.cs.wustl.edu/ jbuhler/research/dapple/.

[5] J. Ho, W. L. Hwang, H. H. S. Lu, D. T. Lee, "Gridding Spot Centers of Smoothly Distored Microarray Images", *IEEE Trans. on Image Processing*, 15, no. 2, pp. 342-353, February 2006.

[6] M. Katzer, F. Kummert, G. Sagerer, "Methods for Automatic Microarray Image Segmentation", *IEEE Transactions on Nano-Bioscience*, 2(4):202-214, 2003.

[7] R. Nagarajan, "Intensity Based Segmentation of Microarray Images", *IEEE Trans. Med. Imaging*, 22(7), pp. 882-889, 2003.

[8] GenePix Pro, http://www.axon.com/gn_GenePixSoftware.html.

[9] Y. Chen, E. R. Dougherty, M. L. Bittner, "Ratio-Based Decisions and the Quantitative Analysis of cDNA Microarray Images", *Journal of Biomedical Optics*, 2, 364-374, October 1997.

[10] G.K. Smyth, Y.H. Yang, T. Speed, "Statistical Issues in cDNA Microarray Data Analysis", *Methods Mol Biol.*, 2003;224:111-36.

[11] D. Mumford, J. Shah, "Optimal Approximations by Piecewise Smooth Functions and Associated Variational Problems", *Comm. Pure Appl. Math.*, 42, pp 577-684, 1989.

[12] Y. Chen, E. R. Dougherty, M. L. Bittner, "Deformable Boundary Finding in Medical Images by Integrating Gradient and Region Information", *IEEE Trans. on Medical Imaging*, 15, no. 6, 859-870, December 1996. 1997.

[13] Y.G. Leclerc, "Constructing Simple Stable Descriptions for Image Partitioning", *International Journal of Computer Vision*, 3, pp 73-102, 1989.

[14] C. Sagiv, N. A. Sochel, Y. Y. Zeevi, "Integrated Active Contour for Texture Segmentation", *IEEE Trans. on Image Processing*, 15, no. 6, pp. 1633-1646, June 2006.

[15] S. C. Zhu, A. L. Yuille, "Region Competition: Unifying Snake/balloon, Region Growing and Bayes/MDL/Energy for multi-band Image Segmentation", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18, no.9, pp.884-900, Sept. 1996.

[16] N. Paragios, R.Deriche, "Geodesic Active Regions: A new Framework to Deal with Frame Partition Problems in Computer Vision", *International Journal of Computer Vision*, 46, pp. 223-247, 2002.

[17] M. Kass, A. Witkin, D. Terzopoulos, "Snakes: Active Countour Models", *International Journal of Computer Vision*, 1988, pp.321-331.

[18] K. Fukunaga, "Introduction to statistical pattern recognition", *Academic Press, New York*, 1972.

[19] M. A. T. Figueiredo, J. M. N. Leitao, A. K. Jain, "Unsupervised Contour Representation and Estimation Using B-Splines and a Minimum Description Length Criterion", *IEEE Trans. on Image Processing*, 9, no. 6, June 2000, pp. 1075-1087.

[20] J. Park, J. M. Keller, "Snake on the Watershed", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23, no. 10, pp. 1201-1205, October 2001.

[21] B. Shahraray, "Optimal Estimation of Contour Properties by Cross-Validated Regularization", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11, no. 6, pp. 600-610, June 1989.

[22] R. Carmona, W.L. Hwang, B. Torr'esani, "Multiridge Detection and Time-Frequency Reconstruction", *IEEE Trans. on Signal Processing*, 47, no. 2, pp. 480-492, February 1999.

[23] R. Carmona, W.L. Hwang, B. Torr'esani, "Practical time-frequency analysis: Gabor and wavelet transforms with an implementation in S ", *Academic Press, San Diego*, 1998.

[24] D. Cremers, M. Rousson "Review of Statistical Approaches to Level Set Segmentation: Integrating Color, Texture, Motion, and Shape", *International Journal of Computer Vision,* 2006, To appear.

[25] S. J. Osher, J. A. Sethian, "Fronts Propagation with Curvature Dependent Speed: Algorithms based on Hamilton-Jacobi Formulations", *Journal of Computational Physics*, 79, pp. 12-49, 1988.

[26] G. Aubert, P. Kornprobst, "Mathematical Problems in Image Processing", *Springer*, 2002.

[27] V. Caselles, R. Kimmel, G. Sapiro "Geodesic Active Contours", *International Journal of Computer Vision*, 22(1), 61-19, 1997.

[28] L. D. Cohen, "On Active Contour Models and Balloons", *CVGIP: Image Understanding*, 53, no. 2, pp. 211-218, March 1991.

[29] D. H. Ballard, C. M. Brown, "Computer Vision", *Prentice Hall,* 1982.

[30] J. Gollub, C. A. Ball, G. Binkley, K. Demeter, D. B. Finkelstein, J. M. Hebert, T. Hernandez-Boussard, H. Jin, M. Kaplper, JC Matese, M. Schroeder, PO Brown, D. Botstein, and G. Sherlock, "The Stanford Microarray Database: data access and quality assessment tools", *Nucleic Acids Res.*, 31(1):94-96, January 1, 2003.

[31] Agilent Technologies, http://www.chem.agilent.com/scripts/ generic.asp?lpage= 10692&prodcol=Y.