Using Conventional Edge Detectors And Post-Smoothing For Segmentation Of Spotted Microarray Images

Peihua Qiu[1] and Jingran Sun[2]

[1]School of Statistics, University of Minnesota, Minneapolis, MN 55455

[2]Portfolio Management, Countrywide Financial Corporation, Woodland Hills, CA 91367

Abstract

Segmentation of spotted microarray images is important in generating gene expression data. It aims to distinguish foreground pixels from background pixels for a given spot of a microarray image. Edge detection in the image processing literature is a closely related research area, because spot boundary curves separating foregrounds from backgrounds in a microarray image can be treated as edges. However, for generating gene expression data, segmentation methods for handling spotted microarray images are required to classify each pixel as either a foreground or a background pixel; most conventional edge detectors in the image processing literature do not have this classification property, because their detected edge pixels are often scattered in the whole design space and consequently the foreground or background pixels are not defined. In this paper, we propose a general post-smoothing procedure for estimating spot boundary curves from the detected edge pixels of conventional edge detectors, such that these conventional edge detectors together with the proposed post-smoothing procedure can be used for segmentation of spotted microarray images. Numerical studies show that this proposal works well in applications.

Key Words: Background; Boundary curves; Derivatives; Edge detection; Foreground; Gene expression data; Image segmentation; Jump location curves; Jump regression analysis; Local polynomial kernel smoothing; Nonparametric regression.

1 Introduction

When generating gene expression data from spotted microarray images, the foreground of a given spot should be segmented from the background using an image segmentation procedure. Then, averaged image intensities (or, some more robust measures) of foreground pixels, computed from a pair of fluorescence images (e.g., the red-fluorescent Cy5 and green-fluorescent Cy3 images), are used for computing gene expression data, after appropriate background corrections and normal-

izations (cf., e.g., Lashkari *et al.* 1997, Chu *et al.* 1998, Ferea *et al.* 1999, Yang *et al.* 2002). Therefore, appropriate segmentation of spotted microarray images is important to the reliability of gene expression data and all subsequent statistical analysis.

Edge detection methods in the image processing literature should be helpful for segmentation of spotted microarray images because their major purpose is to detect outlines of objects in a given image, including spot boundaries of a spotted microarray image. Most conventional edge detectors are based on estimation of the first-order derivatives (c.f., e.g., Canny 1986, Gijbels *et al.* 2006, Gonzalez and Woods 2002, Hall *et al.* 2001, Hall, Qiu and Rau 2008, Hall and Rau 2000, 2002, Pratt 1991, Qiu 1997, 2002, Qiu and Bhandarkar 1996, Qiu and Yandell 1997, Rosenfeld and Kak 1982, Sun and Qiu 2007), or the second-order derivatives (c.f., e.g., Haralick 1984, Huertas and Medioni 1986, Marr and Hildreth 1980, Nalwa and Binford 1986, Torre and Poggio 1986) of the underlying image intensity function, since these derivatives carry useful information about edge locations. For a systematic discussion about this topic, read Chapter 6 of Qiu (2005).

However, detected edge pixels by most existing edge detectors in the image processing literature are scattered in the entire design space and they may not form closed curves. Consequently, it is still unknown whether a given pixel is a foreground pixel of a spot even after edge pixels are detected by them. Therefore, edge detectors are helpful for segmentation of a spotted microarray image; but the latter problem can not be solved using such edge detectors alone. In the bioinformatics literature, the segmentation problem of spotted microarray images has received much attention recently, and several segmentation procedures have been proposed specifically for solving this problem. For instance, Eisen (1999) provided a fixed circle segmentation procedure in the software *ScanAlyze*, which fits circles with a constant diameter to all spots in an image. The software *GenePix* (1999) provided an adaptive circle segmentation procedure, which fitted a circle to a spot with its diameter estimated separately for each spot. The seeded region growing (SRG) procedure suggested by Adams and Bischof (1994) works more flexibly, by sequentially classifying each pixel to either the foreground or the background region. Another commonly used segmentation procedure was suggested by Chen *et al.* (1997), which segments the foreground from the background of a spot by thresholding the histogram of all intensities whose pixel locations are within a target mask. More recent segmentation procedures include clustering algorithms (e.g., Bozinov and Rahnenführer 2002, Glasbey and Ghazal 2003), segmentation based on Gaussian density estimation (Steinfath *et al.* 2001), segmentation using mathematical morphology (Angulo and Serra 2003), segmentation by

local smoothing change-point estimation (Qiu and Sun 2007), and so forth.

As noted above, a major reason why most edge detectors in the image processing literature can not completely solve the segmentation problem of spotted microarray images is that they do not have the property of *classifying* a given pixel as either a foreground or a background pixel. This limitation, however, can be lifted by the post-smoothing procedure proposed in this paper. More specifically, in the case when a given spot has a single boundary curve, we suggest fitting a closed curve through the detected edge pixels of a typical edge detector, using local linear kernel smoothing; the fitted curve would be a good estimator of the spot boundary curve. Thus, foreground and background pixels are well defined after applying this post-smoothing procedure. The case when the given spot has two boundary curves (i.e., the case of "donut" spots) can also be handled, although an appropriate modification is needed. We will show that, besides "donut" spots, the proposed method is flexible enough to handle rotated elliptical spots, D-shaped spots, spots with scratches, spots with bright speckles, and so forth. Its segmentation results are compatible with those obtained by some commonly used segmentation procedures, and in some cases the former performs much better. Therefore, by the proposed post-smoothing procedure, it becomes possible to make use of the vast literature on edge detection, for handling the segmentation problem of spotted microarray images.

The rest part of the article is organized as follows. In Section 2, a general version of edge detectors based on the first-order derivatives of the image intensity function is described, based on which the proposed post-smoothing procedure is introduced. Section 3 presents numerical studies for evaluating the numerical performance of the proposed procedure. Some concluding remarks are given in Section 4.

2 Proposed Method

The proposed method is described in three parts. In Subsection 2.1, a general edge detector based on the first-order derivatives of the image intensity function is introduced. Then, a post-smoothing procedure is proposed in Subsection 2.2, for estimating spot boundary curves. Finally, a data-driven parameter selection procedure is discussed in Subsection 2.3.

2.1 A general edge detector

Before segmentation of a spotted microarray image, borders and centers of its grid cells, each of which contains a spot in the middle, can be roughly specified by the arrayer in the image addressing stage. See Bergemann *et al.* (2004) for such an image addressing method, which is also used in the numerical examples in Section 3. Therefore, image segmentation can be performed separately for individual grid cells, which has become a common practice in analyzing microarray images. For this reason, our discussion below is for handling a single grid cell only.

For a given grid cell, let f be the image intensity function, f_x and f_y be its two partial derivatives in the x- and y-axis directions, respectively, $\{(x_i, y_j),\ i, j = 1, 2, \ldots, n\}$ be $N = n^2$ equally spaced pixels in the design space $[-1/2, 1/2] \times [-1/2, 1/2]$, and $\{Z_{ij}, i, j = 1, 2, \ldots, n\}$ be the corresponding observed image intensities. It should be pointed out that the assumption that the local grid cell is a square with the same number of rows and columns is just for simplicity of presentation. Our proposed method should also work well for rectangular grid cells.

As in most image processing references, we assume that observed image intensities follow the following model:

$$Z_{ij} = f(x_i, y_j) + \varepsilon_{ij}, \text{ for } i, j = 1, 2, \ldots, n, \tag{1}$$

where $\{\varepsilon_{ij}, i, j = 1, 2, \ldots, n\}$ are independent and identically distributed (i.i.d.) random errors with mean 0 and unknown variance σ^2. Then, most existing edge detectors based on the first-order derivatives would label the pixel (x, y) as an edge pixel if

$$M_f(x, y) = \sqrt{\widehat{f}_x^{\,2}(x, y) + \widehat{f}_y^{\,2}(x, y)} \geq u_N, \tag{2}$$

and as a non-edge pixel otherwise, where $\widehat{f}_x$ and $\widehat{f}_y$ are some estimators of f_x and f_y, and u_N is a threshold parameter. In the image processing literature, several masks or operators have been suggested for constructing estimators $\widehat{f}_x$ and $\widehat{f}_y$, which include the 2×2 Roberts operators, 3×3 Prewitt, Sobel, or Frei-Chen masks, 7×7 truncated pyramid masks, derivatives of Gaussian (DoG) operators, and so forth (cf., e.g., Qiu 2005, Section 6.2).

The masks or operators mentioned above were proposed by intuition. Most of them have fixed small sizes and consequently their ability to remove noise is limited. To overcome these limitations,

we suggest using the following local linear kernel smoothing procedure:

$$\min_{a,b,c\in R} \sum_{(x_i,y_j)\in O_N(x,y)} \{Z_{ij} - [a + b(x_i - x) + c(y_j - y)]\}^2 K\left(\frac{x_i - x}{h_N}, \frac{y_j - y}{h_N}\right), \tag{3}$$

where $O_N(x,y) = \{(u,v) : \sqrt{(u-x)^2 + (v-y)^2} \le h_N\}$ is a circular neighborhood of the point (x,y) with bandwidth $h_N > 0$, and K is a radially symmetric, bivariate density kernel function with support $\{(x,y) : x^2 + y^2 \le 1\}$. Then, the solution to (b,c) of the minimization problem (3) can be used as an estimator of the gradient vector $G(x,y) = (f_x(x,y), f_y(x,y))$. Namely, $\widehat{f_x}(x,y) = \widehat{b}(x,y)$ and $\widehat{f_y}(x,y) = \widehat{c}(x,y)$.

In procedure (3), a plane is fitted locally in the neighborhood $O_N(x,y)$ so that the weighted residual sum of squares reaches the minimum, where the weights are controlled by the kernel function K. In reality, K is often chosen to be a monotone decreasing function of the radius. Thus, pixels closer to (x,y) would receive more weight. It can be easily checked that both $\widehat{b}(x,y)$ and $\widehat{c}(x,y)$ are weighted averages of the observed image intensities in $O_N(x,y)$. Thus, they can be regarded as a general version of many existing edge detection masks, such as those mentioned above. In the literature, it has been well shown that local linear kernel estimators have some good statistical properties, including easy computation, automatic boundary correction, optimal convergence rates, and so forth (cf., e.g., Fan and Gijbels 1996). Compared to most existing edge detection masks, it has the flexibility in adjusting the bandwidth parameter h_N in a data-driven way (cf., discussion in Subsection 2.3), such that its bias and variance are well balanced. For these reasons, this method is used in all numerical examples in this paper for edge detection.

2.2 Local linear kernel post-smoothing

Suppose that the detected edge pixels by the edge detector described in the previous subsection are $\{(x^*_\ell, y^*_\ell), \ell = 1,2,\ldots,m\}$, which is a subset of $\{(x_i,y_j), i,j = 1,2,\ldots,n\}$. Then, $\{(x^*_\ell, y^*_\ell), \ell = 1,2,\ldots,m\}$ can also be expressed in the polar coordinate system with respect to the center of the grid, by $\{(r_\ell, \theta_\ell), \ell = 1,2,\ldots,m\}$. See Fig. 1 for an illustration.

Next, we discuss estimation of spot boundary curves from detected edge pixels, using local linear kernel post-smoothing. We first discuss the case when the given grid cell has a single boundary curve Γ which has the radius function $r(\theta) > 0$, for $\theta \in [0, 2\pi]$. Since the boundary curve Γ can be reasonably assumed to be a closed curve, we assume that $r(0) = r(2\pi)$; for convenience, we further

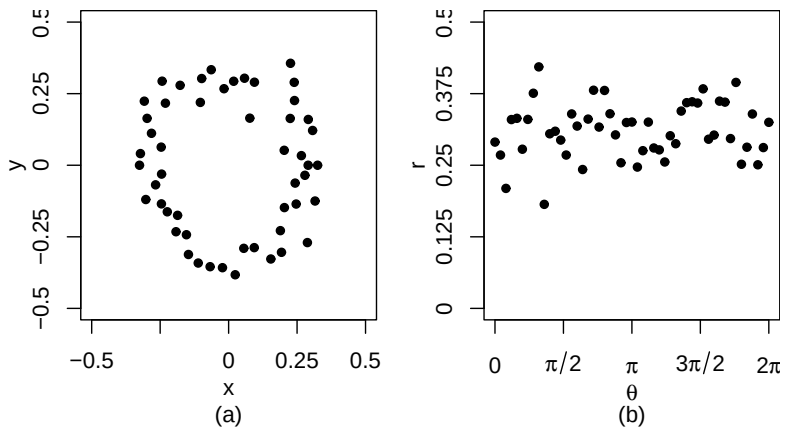


Figure 1: (a) Detected edge pixels $\{(x_\ell^*, y_\ell^*), \ell = 1, 2, \ldots, m\}$. (b) Detected edge pixels in the polar coordinate system.

assume that $r(\theta)$ is a periodic function on the number line R with period 2π. Estimation of $r(\theta)$, for any $\theta \in [0, 2\pi]$, can be accomplished by the following local linear kernel smoothing procedure:

$$\min_{a,b \in R} \sum_{\theta_\ell \in [\theta - h_\theta, \theta + h_\theta]} \{r_\ell - [a + b(\theta_\ell - \theta)]\}^2 K_\theta \left(\frac{\theta_\ell - \theta}{h_\theta} \right), \tag{4}$$

where $h_\theta > 0$ is a bandwidth parameter, and K_θ is a 1-D density kernel function with support $[-1, 1]$. Then, the solution to a of procedure (4) is the local linear kernel estimator of $r(\theta)$, denoted as $\widehat{r}(\theta)$. Note that, in procedure (4), observations $\{(r_\ell, \theta_\ell), \ell = 1, 2, \ldots, m\}$ should be extended periodically from $\theta \in [0, 2\pi]$ to $\theta \in [-h_\theta, 2\pi + h_\theta]$ beforehand, using the relationship $r(\theta_\ell \pm 2\pi) = r_\ell$, for $\ell = 1, 2, \ldots, m$. Therefore, this procedure does not suffer the notorious "boundary problem" of conventional local smoothing procedures in the current segmentation problem, because observations in the entire neighborhood $[\theta - h_\theta, \theta + h_\theta]$ are well defined when θ is in the boundary regions $[0, h_\theta)$ and $(2\pi - h_\theta, 2\pi]$ of $[0, 2\pi]$. As a side note, in the literature, there is much discussion about estimation of periodic curves. For instance, Thomas-Agnan (1990) suggested a nice regularization method for this problem, based on smoothing spline estimation.

In the case when the given grid cell includes a "donut" spot, the spot has two boundary curves Γ_1 and Γ_2 with radius functions $r_1(\theta)$ and $r_2(\theta)$, respectively, where $r_1(\theta) < r_2(\theta)$, for all $\theta \in [0, 2\pi)$. In such cases, we suggest estimating $r_1(\theta)$ and $r_2(\theta)$ by the following procedure, after observations

$\{(r_\ell, \theta_\ell), \ell = 1, 2, \ldots, m\}$ are extended periodically beforehand, as in (4):

$$\min_{r\in[\min r_\ell, \max r_\ell]} \min_{a_1,b_1,a_2,b_2\in R} \left\{ \sum_{\theta_\ell\in[\theta-h_\theta,\theta+h_\theta];r_\ell\le r} \{r_\ell - [a_1 + b_1(\theta_\ell - \theta)]\}^2 K_\theta\left(\frac{\theta_\ell - \theta}{h_\theta}\right) + \right.$$
$$\left. \sum_{\theta_\ell\in[\theta-h_\theta,\theta+h_\theta];r_\ell> r} \{r_\ell - [a_2 + b_2(\theta_\ell - \theta)]\}^2 K_\theta\left(\frac{\theta_\ell - \theta}{h_\theta}\right) \right\}. \qquad (5)$$

The solutions to a_1 and a_2 of procedure (5) are then defined as estimators of $r_1(\theta)$ and $r_2(\theta)$, respectively, denoted as $\widehat{r_1}(\theta)$ and $\widehat{r_2}(\theta)$. In (5), it is possible that the solution to r is not unique. For instance, if the detected edge pixels in the neighborhood $[\theta - h_\theta, \theta + h_\theta]$ have the property that about half of their r values are below R_1 and another half above R_2, where $R_1 < R_2$ are two positive constants, then all r values in $[R_1, R_2]$ could be possible solutions to r. In such cases, estimators $\widehat{r_1}(\theta)$ and $\widehat{r_2}(\theta)$ corresponding to any such r can be used.

In applications, if it is clear, based on our visual impression, whether there are "donut" spots in a microarray image, then we can simply choose between procedures (4) and (5). If, however, it is difficult to make such a judgment based on our visual impression alone, we suggest using the following data-driven decision rules. For a given grid cell, we compute two sets of boundary curve(s) using procedures (4) and (5), respectively, and let (I_f, I_b) and (I_f^*, I_b^*) be pairs of averaged foreground intensity and averaged background intensity in the two setups. Then, we conclude that the grid cell does not contain a "donut" spot if

$$I_f - I_b > I_f^* - I_b^*, \qquad (6)$$

and the opposite decision is made otherwise. It should be pointed out that more robust summary statistics, such as the trimmed means or medians, can be used in (6) in places of I_f, I_b, I_f^* and I_b^*. Also, when using the proposed segmentation procedure, step (6) is actually accommodated in the selection of procedure parameters introduced below. Therefore, it does not add much extra complexity to the procedure.

2.3 Selection of procedure parameters

In the local linear kernel edge detector described in Subsection 2.1, there are two parameters h_N and u_N. The post-smoothing procedure discussed in the previous subsection has another bandwidth parameter h_θ. These parameters should be chosen properly by a data-driven procedure because

they affect the final segmentation results. To this end, we first suggest the following performance measure of the segmentation results:

$$C(h_N, u_N, h_\theta) = \max\left\{I_f - I_b, I_f^* - I_b^*\right\}, \tag{7}$$

where I_f, I_b, I_f^* and I_b^* are defined in expression (6). Obviously, $C(h_N, u_N, h_\theta)$ is defined as the *contrast* between the estimated foreground and background. Intuitively, a good segmentation would lead to a large value of such a contrast measure. Theoretically, it can be proved that this measure reaches the maximum when the estimated foreground matches the true one, under some regularity conditions. Then, h_N, u_N, and h_θ can be chosen by

$$\max_{h_N}\left\{\max_{u_N}\left[\max_{h_\theta} C(h_N, u_N, h_\theta)\right]\right\}. \tag{8}$$

For simplicity in notation, the chosen parameter values are still denoted by h_N, u_N, and h_θ. In (8), when h_N is given, the edge detection criterion M_f (cf., expressions (2) and (3)) can be computed at each pixel. Then, procedures $\max_{u_N}$ and $\max_{h_\theta}$ are just two nested 1-D searches, in which values of M_f only need to be computed once. Therefore, computation involved in (8) is actually quite fast.

3 Numerical Studies

In this section, we first present some simulation results regarding the numerical performance of the proposed segmentation method, and then demonstrate the method by applying it to some real microarray images.

For simplicity, our simulation is for detecting boundary curve(s) of a single grid cell of a spotted microarray image, which is appropriate for reasons explained in Subsection 2.1. The design space of the grid cell is assumed to be $[-1/2, 1/2] \times [-1/2, 1/2]$ and the following seven cases are considered.

Case (i): The true image intensity function is $f(x, y) = 1500[1 - .5(x^2 + y^2)/.3^2]$ if $\frac{x^2}{.3^2} + \frac{y^2}{.3^2} \le 1$; and $f(x, y) = 500[1 - .5x^2 - .5y^2]$ otherwise.

Case (ii): $f(x, y) = 1500[1 - .5(\tilde{x}^2 + \tilde{y}^2)/.3^2]$ if $\frac{\tilde{x}^2}{.3^2} + \frac{\tilde{y}^2}{.2^2} \le 1$; and $f(x, y) = 500[1 - .5\tilde{x}^2 - .5\tilde{y}^2]$ otherwise, where $(\tilde{x}, \tilde{y})$ is obtained by rotating (x, y) 45-degree counterclockwise.

Case (iii): $f(x, y) = 1500[1 - .5(x^2 + y^2)/.3^2]$ if $\frac{x^2}{.12^2} + \frac{y^2}{.12^2} > 1$ and $\frac{x^2}{.3^2} + \frac{y^2}{.3^2} \le 1$; and $f(x, y) = 500[1 - .5x^2 - .5y^2]$ otherwise.

Case (iv): $f(x,y) = 1500[1 - .5(x^2+y^2)/.3^2]$ if $\frac{x^2}{.3^2} + \frac{y^2}{.3^2} \leq 1$ and $x \geq .36$; and $f(x,y) = 500[1 - .5x^2 - .5y^2]$ otherwise.

Case (v): $f(x,y)$ is the same as the one in case (ii), except that $f(x,y) = 1000$ when $-.04 \leq y \leq .04$.

Case (vi): $f(x,y)$ is the same as the one in case (ii), except that $f(x,y) = 1500$ when $(x,y) \in [-0.3,-0.28] \times [-0.3,-0.28]$ or $(x,y) \in [0.28,0.3] \times [0.28,0.3]$.

Case (vii): $f(x,y) = 1500[1 - .5(\tilde{x}^2+\tilde{y}^2)/.3^2]$ if $\frac{\tilde{x}^2}{.3^2} + \frac{\tilde{y}^2}{.2^2} \leq 1$; and $f(x,y) = 1000[1 - .5\tilde{x}^2 - .5\tilde{y}^2]$ otherwise, where $(\tilde{x},\tilde{y})$ is obtained by rotating (x,y) 45-degree counterclockwise.

It can be seen that, in case (i), the spot has a circular boundary curve with radius .3. Cases (ii)–(vii) simulate scenarios when the given spot is rotated elliptical, "donut"-shaped, D-shaped, rotated elliptical with a scratch, rotated elliptical with two bright speckles, and rotated elliptical with a high background, respectively. The true values of averaged foreground image intensity (AFII) in cases (i)–(vii) are respectively 1102, 1200, 1030, 1136, 1200, 1200, and 1200. In all cases, observed image intensities are generated by model (1) with i.i.d. random errors from normal distribution with mean 0. In cases (i)–(iv) and (vi)–(vii), noise variance is 500^2 in the foreground, and 200^2 in the background. In case (v), noise variance is 100^2 in the elliptical foreground and in the scratch region, and 50^2 in the remaining background. Noise levels are intentionally lower in this case, compared to the other cases, to make the scratch more visible. One realization of observed image intensities in cases (i)–(vii) is presented by the seven plots in the first column of Fig. 2, with whiter pixels denoting larger intensity levels.

The proposed segmentation procedure (2)–(8) is then applied to the simulated data in case (i) which are shown in the (1,1)-th plot of Fig. 2. The kernel functions used in (3) and (4) are chosen to be the truncated, bivariate and univariate, Gaussian density functions with supports $\{(x,y) : x^2+y^2 \leq 1\}$ and $[-1,1]$, respectively. By procedure (7)–(8), h_N, u_N, and h_θ are chosen to be 0.08, 142, and 0.50, respectively. The corresponding detected edge pixels are shown in the (1,2)-th plot of Fig. 2 by the white pixels; they are shown in the (1,3)-th plot of Fig. 2 as a scatterplot of $\{(r_\ell, \theta_\ell, \ell = 1,2,\ldots,m\}$ in the polar coordinate system, together with the estimated radius function $\widehat{r}(\theta)$. The estimated boundary curve is shown in the (1,4)-th plot of Fig. 2 by white pixels. Corresponding segmentation results in the other six cases are shown in the remaining six rows of Fig. 2. It can be seen that related spots are segmented reasonably well in all cases.

Next, we compare the proposed image segmentation procedure (denoted as "Post-Smoothing")

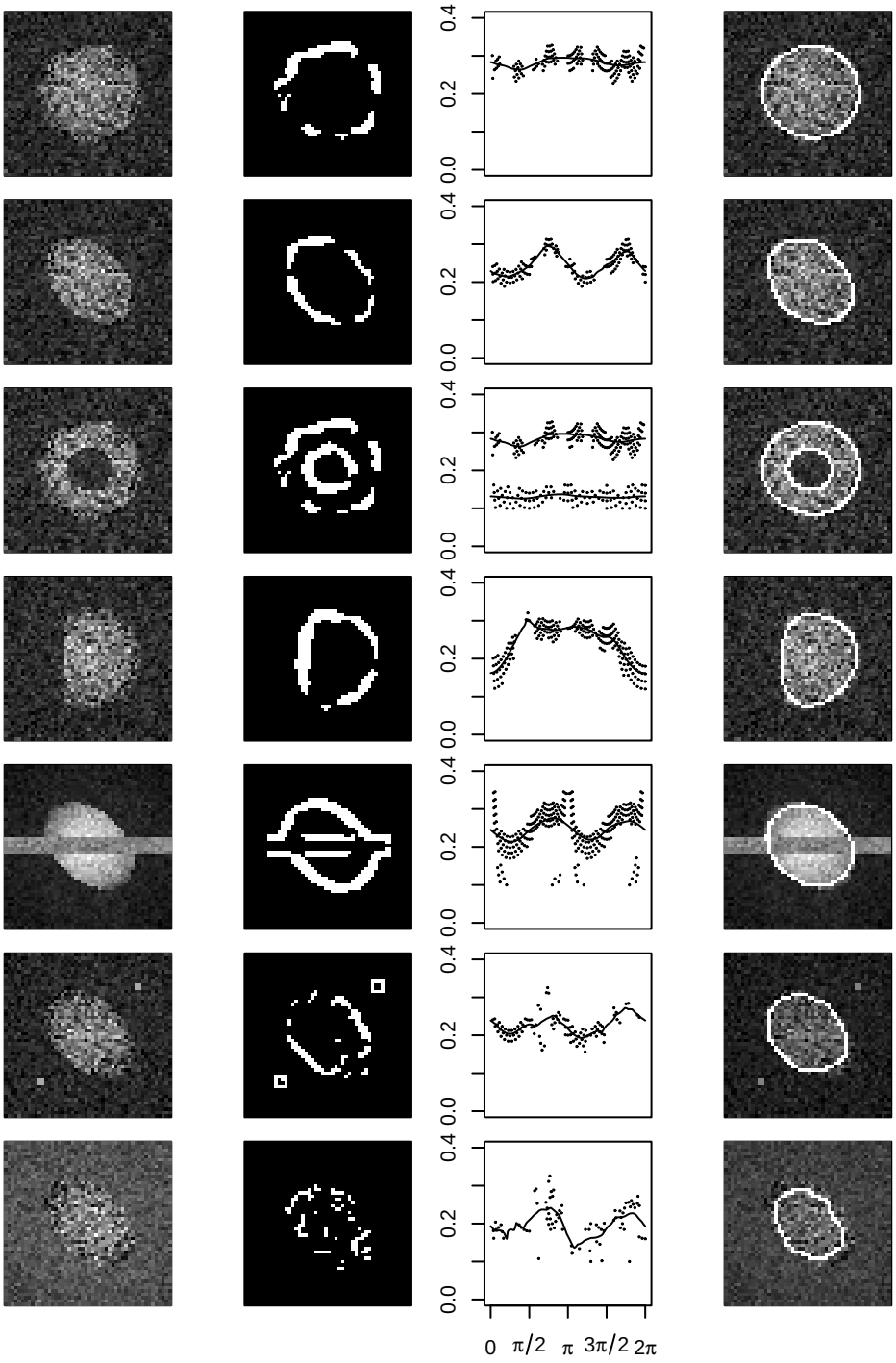


Figure 2: The four plots in the first row show, respectively, an observed grid cell with a circular spot, detected edge pixels from the observed grid cell, detected edge pixels shown in the polar coordinate system together with the estimated radius function, and the estimated boundary curve. Plots in the remaining six rows show corresponding results when the spot is rotated elliptical, "donut"-shaped, D-shaped, rotated elliptical with a scratch, rotated elliptical with two bright speckles, rotated elliptical with a high background, respectively.

with the following four existing image segmentation procedures: (i) SRG, (ii) segmentation by thresholding the histogram of image intensities (denoted as "Histogram"), (iii) adaptive circle procedure (denoted as "Adaptive-Circle"), and (iv) the recent segmentation procedure suggested by Qiu and Sun (2007) based on change-point estimation (denoted as "Change-Point"). See Section 1 for a brief introduction about these procedures. In procedure Post-Smoothing, the kernel functions and bandwidth parameters are chosen as before. In procedure SRG, pixels located at the border of the grid cell are used as background seeds and pixels in a square of size $.1 \times .1$ centered at the origin are used as foreground seeds. In procedure Histogram, its circular mask is centered at the origin with radius .4 in all cases, and its threshold value is determined by the Mann-Whitney statistic with significance level .05%, as used by Chen *et al.* (1997). In procedure Adaptive-Circle, the radius of the circle is searched by the approach used in the software package *Dapple* (2000), which first generates a Laplacian image from the original image, using a standard four-neighbor Laplacian mask (cf., Qiu 2005, Section 6.2), and then chooses the radius as the maximizer of function $\psi(r)$, defined as the average of all pixels in the Laplacian image whose Euclidean distances from the center are r. In procedure Change-Point, the version based on gradient estimation is used here and its procedure parameters are all chosen by cross-validation.

For each method, its performance is evaluated by

$$d_{\Omega^{(1)},\widehat{\Omega}^{(1)}} = \frac{\left|\left(\Omega^{(1)} \backslash \widehat{\Omega}^{(1)}\right) \bigcup \left(\widehat{\Omega}^{(1)} \backslash \Omega^{(1)}\right)\right|}{\left|\Omega^{(1)}\right|},$$

where $|A|$ denotes the number of pixels in pointset A, $\Omega^{(1)}$ is the true foreground, and $\widehat{\Omega}^{(1)}$ is its estimator. Obviously, $d_{\Omega^{(1)},\widehat{\Omega}^{(1)}}$ is a ratio of the total number of false foreground pixels and false background pixels to the number of true foreground pixels. Besides this measure, we also compute

$$d^{(+)}_{\Omega^{(1)},\widehat{\Omega}^{(1)}} = \frac{|\widehat{\Omega}^{(1)} \backslash \Omega^{(1)}|}{\Omega^{(1)}}, \text{ and } d^{(-)}_{\Omega^{(1)},\widehat{\Omega}^{(1)}} = \frac{|\Omega^{(1)} \backslash \widehat{\Omega}^{(1)}|}{\Omega^{(1)}}$$

to measure the amounts of false foreground pixels and false background pixels, respectively. With the estimated foreground, the estimated AFII, denoted as $\widehat{\text{AFII}}$, is also computed. The averaged values of $d_{\Omega^{(1)},\widehat{\Omega}^{(1)}}$, $d^{(+)}_{\Omega^{(1)},\widehat{\Omega}^{(1)}}$, $d^{(-)}_{\Omega^{(1)},\widehat{\Omega}^{(1)}}$, and $\widehat{\text{AFII}}$ from 100 replications along with the true values of AFII are presented in Table 1.

From Table 1, it can be seen that the proposed procedure Post-Smoothing performs the best among all procedures in all cases with regard to $d_{\Omega^{(1)},\widehat{\Omega}^{(1)}}$. It is the best or close to the best in all cases with regard to $d^{(+)}_{\Omega^{(1)},\widehat{\Omega}^{(1)}}$, $d^{(-)}_{\Omega^{(1)},\widehat{\Omega}^{(1)}}$ and $\widehat{\text{AFII}}$. Please note that, in case (iii) when the spot

Table 1: This table presents the averaged values of $d_{\Omega^{(1)},\widehat{\Omega}^{(1)}}$, $d^{(+)}_{\Omega^{(1)},\widehat{\Omega}^{(1)}}$, $d^{(-)}_{\Omega^{(1)},\widehat{\Omega}^{(1)}}$, and $\widehat{\text{AFII}}$ from 100 replications, along with the true values of AFII.

Case	Method	$d_{\Omega^{(1)},\widehat{\Omega}^{(1)}}$	$d^{(+)}_{\Omega^{(1)},\widehat{\Omega}^{(1)}}$	$d^{(-)}_{\Omega^{(1)},\widehat{\Omega}^{(1)}}$	$\widehat{\text{AFII}}$	AFII
(i)	Post-Smoothing	.143	.003	.140	1166	1102
	SRG	.331	.005	.326	1278	
	Histogram	.343	.065	.278	1317	
	Adaptive-Circle	.548	0	.548	1332	
	Change-Point	.183	.005	.178	1176	
(ii)	Post-Smoothing	.120	.024	.096	1228	1200
	SRG	.278	.010	.268	1325	
	Histogram	.360	.141	.219	1323	
	Adaptive-Circle	.686	.010	.676	1416	
	Change-Point	.124	.026	.098	1226	
(iii)	Post-Smoothing	.141	.017	.124	1066	1030
	SRG	.756	.221	.535	996	
	Histogram	.419	.103	.316	1251	
	Adaptive-Circle	1.158	.158	1.0	473	
	Change-Point	.206	.080	.126	1033	
(iv)	Post-Smoothing	.175	.025	.150	1185	1136
	SRG	.318	.007	.311	1300	
	Histogram	.367	.108	.259	1314	
	Adaptive-Circle	.718	.044	.674	1370	
	Change-Point	.181	.012	.169	1203	
(v)	Post-Smoothing	.138	.053	.085	1128	1200
	SRG	.287	.230	.057	1096	
	Histogram	.435	.286	.149	1096	
	Adaptive-Circle	.347	.002	.345	1198	
	Change-Point	.206	.166	.040	1084	
(vi)	Post-Smoothing	.120	.026	.094	1223	1200
	SRG	.266	.009	.257	1330	
	Histogram	.353	.125	.228	1341	
	Adaptive-Circle	.658	.019	.639	1395	
	Change-Point	.134	.027	.107	1232	
(vii)	Post-Smoothing	.223	.073	.150	1239	1200
	SRG	.519	.019	.500	1509	
	Histogram	.689	.187	.502	1551	
	Adaptive-Circle	.701	.011	.690	1424	
	Change-Point	.424	.006	.418	1349	

is "donut"-shaped, the estimated boundary curve by the adaptive circle procedure is completely inside the "hole" of the "donut", making its $d^{(-)}_{\Omega^{(1)},\widehat{\Omega}^{(1)}}$ value to be 1 and its $d_{\Omega^{(1)},\widehat{\Omega}^{(1)}}$ value larger than 1.

Next, we apply the related procedures to some real microarray images, which are from a study by van't Wout *et al.* (2003) about the biochemical changes that occur during HIV-1 infection. In the study, expression levels of 4,608 cellular RNA transcripts were assessed in $CD4^+$-T-Cell lines, after infection with HIV virus type 1 strain BRU (HIV-1_{BRU}), using DNA microarrays. Four replications of the experiment were performed. For ease of presentation, here we only present results from four replications of one subarray consisting of $32 \times 12 = 384$ genes. Results from other subarrays are similar.

For each replication of the experiment, a pair of red-fluorescent dye and green-fluorescent dye images is obtained, and image segmentation is performed on the combined image. Before segmentation, the automatic image addressing procedure described in Bergemann *et al.* (2004) is used for specifying the centers and borders of grid cells of the image. By this approach, peak positions of column/row totals of image intensities are used for specifying the grid cell centers, after the column/row totals are pre-smoothed by a smoothing operator. Similarly, the borders of grid cells are estimated by the valley positions of the smoothed column/row totals. The size of a typical grid cell of the image is about 30×30 pixels. Then, the five segmentation procedures are applied to each grid cell. In the proposed procedure, the kernel functions and three parameters are chosen in the same way as in Table 1. In the SRG procedure, pixels located at the border of the grid cell are used as background seeds and pixels in a square of size 5×5 pixels centered at the origin are used as foreground seeds. In the Histogram procedure, the circular mask is centered at the origin and has radius of 12 pixels; its threshold is chosen in the same way as we did in Table 1. In the adaptive circle procedure, the circle radius is chosen in the same way as in Table 1. In the Change-Point procedure, the foreground noise variance is estimated in the region within a circle of radius 4 pixels centered at the origin, and the background noise variance is estimated in the region outside a circle of radius 12 pixels centered at the origin.

Segmentation results of the five procedures for the first replicated pair of images are shown in Fig. 3, along with the combined image. In plots (b)–(f), white pixels denote detected foreground pixels for the Histogram procedure; they denote detected boundary curves for the other four procedures. It can be seen that the proposed procedure detects most boundary curves, including the ones of some "donut" spots (cf., e.g., the (5,12)-th spot), reasonably well. The SRG procedure can not handle "donut" spots well, and its detected boundary curves are quite noisy in some grid cells (cf., e.g., the (1,11)-th spot). The Histogram procedure can handle some "donut" spots well, but

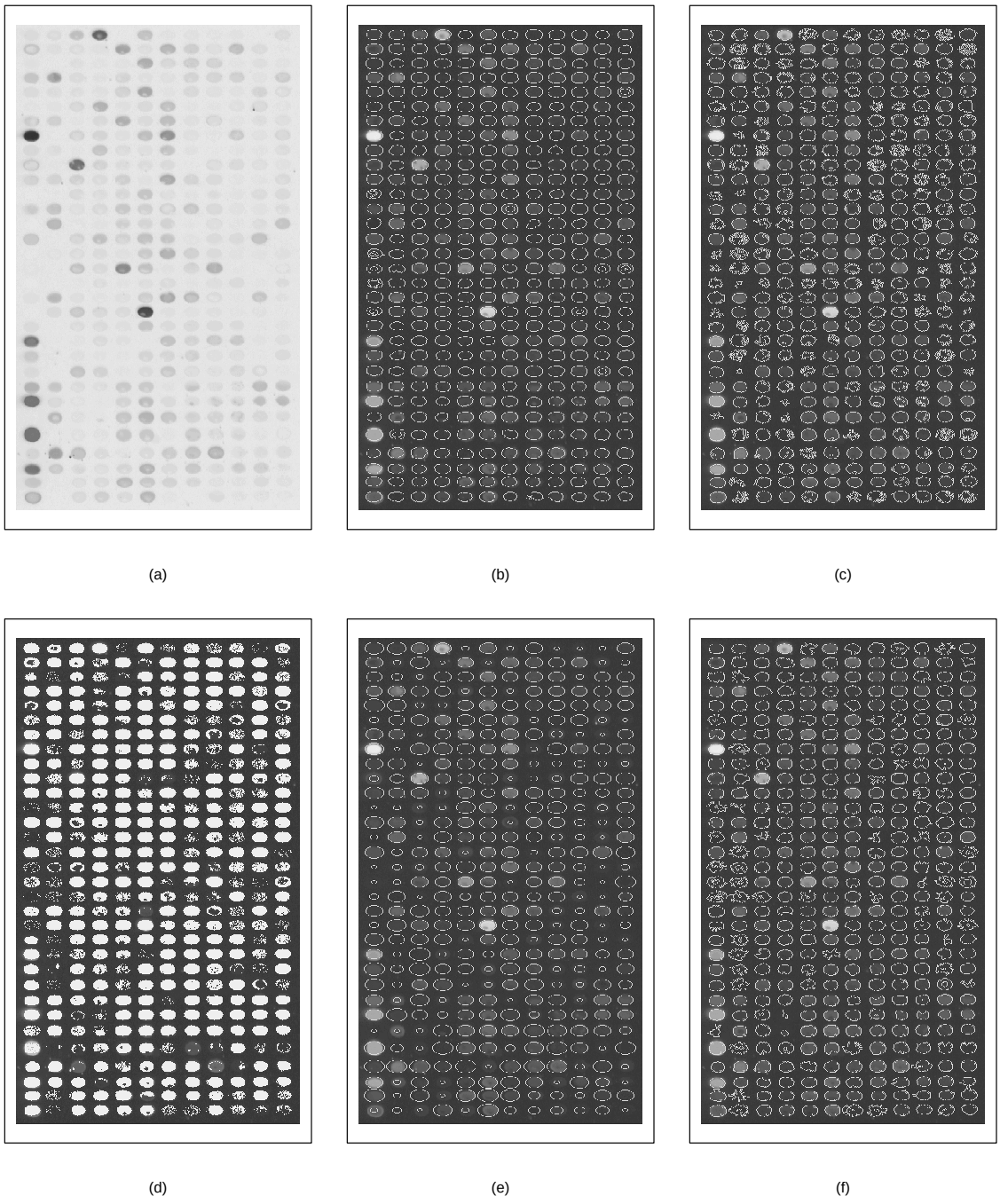


Figure 3: (a) Combined image of the first replicated pair of microarray images. (b) Detected boundary curves by the proposed procedure. (c) Detected boundary curves by the SRG procedure. (d) Detected foreground pixels by the histogram thresholding procedure. (e) Detected boundary curves by the adaptive circle procedure. (f) Detected boundary curves by the change-point procedure.

its detected foreground pixels do not form connected regions in some cases (cf., e.g., the (5,10)-th spot), due to the fact that it does not make use of any spatial information of the image; it misses many foreground pixels when the signal is weak (cf., e.g., the (3,5)-th spot). The adaptive circle procedure can not handle "donut" spots at all (cf., e.g., the (24,11)-th spot), and its results are affected much if the grid cell center is not specified well (cf., e.g., the (8,1)-th spot). The Change-Point procedure can detect most boundary curves well, but some of its detect boundary curves are quite noisy (cf., e.g., the (1,11)-th spot).

One may ask whether differences in image segmentation by these procedures would have an impact on downstream analyses. Generally speaking, this question is difficult to answer because substantial impact on one analysis may not imply similar impact on the other, and it is even impossible to list related downstream analyses inclusively. With the real microarray images in this example, next we demonstrate that image segmentation could have a real impact on certain downstream analyses. To this end, based on the segmentation results of each procedure considered, we compute the gene expression data for each replicated pair of the observed microarray images. For a given spot, its gene expression level is computed by the formula

$$\log_2 \left(\frac{R_f - R_b}{G_f - G_b} \right),$$

where R_f and R_b denote the averaged image intensities of the foreground and background of that spot in the red-fluorescent dye image, and G_f and G_b denote the averaged image intensities of the foreground and background of the same spot in the green-fluorescent dye image. Then, for a specific spot, the sample mean (denoted as $\bar{x}$) and sample standard deviation (denoted as s_x) of the gene expression levels across four replications are computed. For simplicity, let us assume that we are interested in testing whether the mean gene expression level of a given spot is significantly different from zero, and that its observed gene expression level follows a Normal distribution. Then, the spot is flagged to have a significantly non-zero mean gene expression level if

$$\left| \frac{\bar{x}}{s_x/2} \right| > t_{0.975}(3)$$

where $t_{0.975}(3) = 3.182$ is the 0.975 quantile of the t distribution with 3 degrees of freedom. Testing results based on the segmentation of the five procedures are summarized in Table 2, which is a combination of four contingency tables. In the table, "1" denotes spots with significantly non-zero mean gene expression levels and "0" denotes other spots. Table 2 shows that testing results based on the segmentation of one procedure agree on most spots with testing results based on the

Table 2: Contingency table summarizing the results of testing the null hypothesis that the mean gene expression level of a spot is zero versus the alternative hypothesis that it is non-zero, based on segmentation of the five segmentation procedures. In the table, "1" denotes spots with significantly non-zero mean gene expression levels and "0" denotes other spots.

		SRG		Histogram		Adaptive-Circle		Change-Point	
		1	0	1	0	1	0	1	0
Post-Smoothing	1	115	32	113	34	122	25	123	24
	0	37	200	39	198	49	188	49	188

segmentation of another procedure; but they also disagree on a substantial number of spots. For instance, with procedures Post-Smoothing and Change-Point, their corresponding testing results agree on $123 + 188 = 311$ spots. But they also disagree on $49 + 24 = 73$ spots, which is about 19% of the total number of spots in this study. It should be pointed out that the normality assumption used in the above t-test may not be valid in certain applications. In such cases, the permutation test might be more reasonable to use, since it does not require any knowledge of the distribution of the observed gene expression levels. See Huang *et al.* (2006) for a recent discussion on this topic. For readers' reference, the first row of Fig. 4 shows spotwise means of the gene expression levels based on segmentation of the proposed procedure Post-Smoothing versus the spotwise means based on segmentation of alternative procedures. From the plots, it seems that the range of spotwise means based on Post-Smoothing is narrower than those based on alternative segmentation procedures. The second and third rows of Fig. 4 present corresponding results regarding spotwise standard deviations and spotwise mean-standard deviation ratios, respectively. We can see that different segmentation procedures generate similar results for most spots, and they differ quite significantly for a number of spots, as demonstrated in Table 2.

To make a more specific comparison of the impact of different segmentation procedures on the downstream analysis considered above, we consider a test dataset shown in Fig. 5(a). This test image has 100 spots; the 50 genes in the first 5 columns are known to be non-differentially expressed and the 50 genes in the remaining 5 columns are known to be differentially expressed. The true image intensity functions of the first five columns are the same as those of cases (i)–(iv) and (vi) considered in Figure 2, respectively, except that the constant 500 is changed to 1000 in the definition of backgrounds, to make image segmentation more challenging (cf., definitions of some image intensity functions in the second paragraph of this section). So, the largest possible true intensity level is 1500 in the 50 spots of the first 5 columns. The true image intensity functions

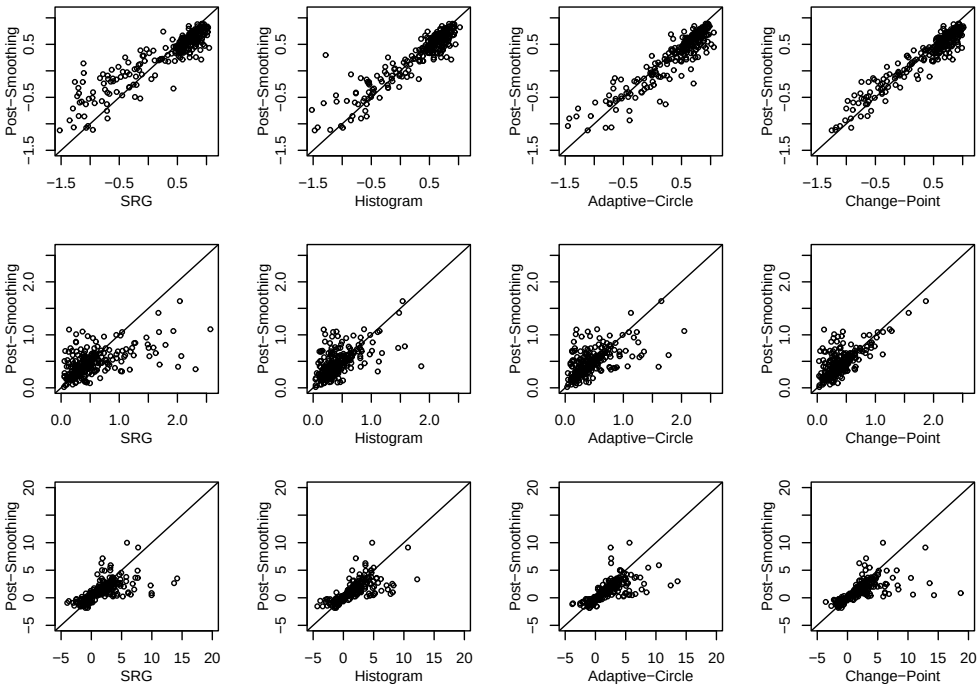


Figure 4: The first, second, and third rows present spotwise means, spotwise standard deviations, and spotwise mean-standard deviation ratios, respectively, of the gene expression levels based on segmentation of the proposed procedure Post-Smoothing versus the corresponding results based on segmentation of alternative procedures.

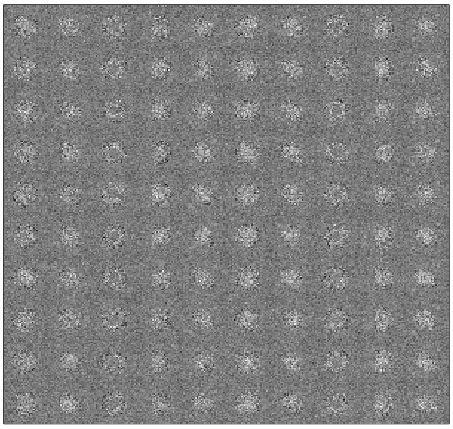
of the last five columns are the same as those in the first five columns, except that the largest possible true intensity level of the last five columns is set to be 1600. For all spots, observed image intensities are generated from model (1) with i.i.d. random errors from normal distribution with mean 0 and variance 600^2 in the foreground and 300^2 in the background. Fig. 5(a) can be regarded as a red-fluorescent dye image. The paired green-fluorescent dye image is generated in the same way, except that the largest possible true intensity level is set to be 1500 for all 100 spots. Ten replicated pairs of images are used in this example.

Based on a given segmentation procedure, a gene is flagged to be differentially expressed if

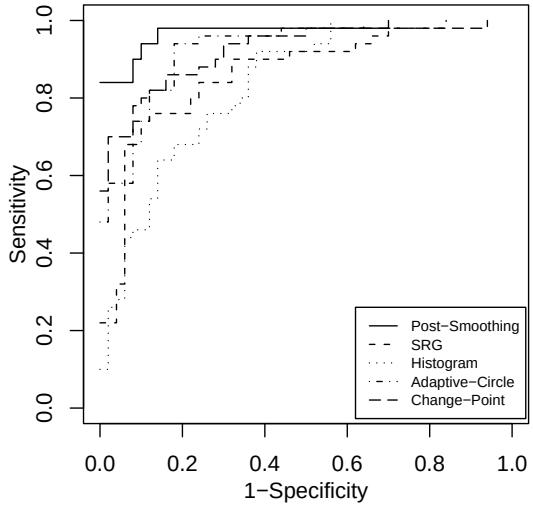
$$\left| \frac{\bar{x}}{s_x/2} \right| > u, \tag{9}$$

where $\bar{x}$ and s_x are the spotwise mean and standard deviation of gene expression levels of a given spot defined in the previous example, and u is a threshold parameter. In the literature, *sensitivity* of a testing procedure is defined to be the probability that a truly differentially expressed gene is flagged to be differentially expressed, and *specificity* is defined to be the probability that a truly non-differentially expressed gene is not flagged by the testing procedure. In practice, both

sensitivity and specificity are estimated by relative freqencies of related events. The curve of the pair (1-specificity, sensitivity) generated by changing the value of u is called the *receiver operating characteristic (ROC) curve* in the literature (cf., e.g., Qiu and Le 2001), and is often used for evaluating the performance of a testing procedure. For instance, if the ROC curve of one testing procedure is uniformly above the ROC curve of another testing procedure, then we can conclude that the former performs uniformly better than the latter. In the current example, the ROC curves of the testing procedure (9) based on segmentation of the five segmentation procedures are shown in Fig. 5(b). From the plot, we can see that the ROC curve based on segmentation of Post-Smoothing is above the other four ROC curves in the entire range, except that they overlap a little bit in their right tails, which demonstrates that the proposed segmentation procedure would improve downstream analysis in this example.



(a)



(b)

Figure 5: (a) A red-fluorescent dye test image. (b) ROC curves of the testing procedure (9) based on segmentation of various procedures.

4 Concluding Remarks

We have presented a general version of edge detectors based on first-order derivatives of the image intensity function, and a post-smoothing procedure for estimating spot boundary curves of spotted microarray images. By using the proposed post-smoothing procedure, theoretically speaking, almost all existing edge detectors in the image processing literature can be used for solving the

segmentation problem of spotted microarray images. It has been shown that segmentation results by the proposed edge detector together with the proposed post-smoothing procedure are reasonably good, compared to those by several routinely used segmentation procedures. We also suggested a data-driven method for choosing appropriate values of the procedure parameters.

The proposed segmentation method can be easily computed because both the edge detection and the post-smoothing steps are based on local smoothing procedures which are generally fast to compute. For instance, with the 4 replicated pairs of red-fluorescent dye and green-fluorescent dye images (each image has 384 spots) in the real-image example discussed in Section 3, the entire process to generate gene expression data from the observed images, which includes image addressing, image segmentation, and background corrections and normalizations, takes about 20 seconds of CPU time on our 1.2-GHz Pentium III PC running a Linux Operating system. We would also want to point out that, although each image has only 384 spots in this example, which is for demonstration purposes only, the proposed segmentation procedure has no difficulty to handle larger microarray images, due to the parallel nature of the segmentation problem that individual spots can be segmented separately after image addressing. The 4 replicated pairs of red-fluorescent dye and green-fluorescent dye images from the HIV-1 infection study, which are used in Section 3, are available from the JCGS website, along with some computer programs used in our numerical studies.

Acknowledgments: We thank the editor, the associate editor and two referees for many constructive comments and suggestions which greatly improved the quality of the paper. This work was supported in part by an NSA grant and an NSF grant.

References

R. Adams and L. Bischof (1994), "Seeded region growing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 641–647.

J. Angulo and J. Serra (2003), "Automatic analysis of DNA microarray images using mathematical morphology," *Bioinformatics*, vol. 19, pp. 553–562.

T.L. Bergemann, R.J. Laws, F. Quiaoit and L.P. Zhao (2004), "A statistically driven approach for

image segmentation and signal extraction in cDNA microarrays," *Journal of Computational Biology*, vol. 11, pp. 695–713.

D. Bozinov and J. Rahnenführer (2002), "Unsupervised technique for robust target separation and analysis of DNA microarray spots through adaptive pixel clustering," *Bioinformatics*, vol. 18, pp. 747–756.

J. Canny (1986), "A Computational Approach to Edge Detection," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 679–698.

Y. Chen, E.R. Dougherty and M.L. Bitter (1997), "Ratio-based decisions and the quantitative analysis of cDNA microarray images," *Journal of Biomedical Optics*, vol. 2, pp. 364-374.

S. Chu, J. DeRisi, M. Eisen, J. Mulholland, D. Botstein, P.O. Brown and I. Herskowitz (1998), "The transcriptional program of sporulation in budding yeast," *Science*, vol. 282, pp. 699–705.

Dapple (2000), *Dapple: Image Analysis Software for DNA Microarrays,* http://www.cs.wustl.edu/ jbuhler/research/dapple/.

M.B. Eisen (1999), *ScanAlyze,* http://rana.Stanford.EDU/software/.

J. Fan and I. Gijbels (1996), *Local Polynomial Modelling and Its Applications,* Chapman & Hall: London.

T.L. Ferea, D. Botstein, P.O. Brown and R.F. Rosenzweig (1999), "Systematic changes in gene expression patterns following adaptive evolution in yeast," *Proceedings of the national Academy of Science*, vol. 96, pp. 9721–9726.

GenePix (1999), *GenePix 4000A Users' Guide,* Axon Instruments, Inc..

I. Gijbels, A. Lambert and P. Qiu (2006), "Edge-preserving image denoising and estimation of discontinuous surfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 1075–1087.

C.A. Glasbey and P. Ghazal (2003), "Combinatorial image analysis of DNA microarray features," *Bioinformatics*, vol. 19, pp. 194–203.

R.C. Gonzalez and R.E. Woods (2002), *Digital Image Processing (2nd ed.)*, New York: Prentice Hall.

P. Hall, L. Peng and C. Rau (2001), "Local likelihood tracking of fault lines and boundaries," *Journal of the Royal Statistical Society - B*, vol. 63, pp. 569–582.

P. Hall, P. Qiu and C. Rau (2008), "Edge, corners and vertex estimation for images and regression surfaces," *Scandinavian Journal of Statistics*, vol. 35, pp. 1–17.

P. Hall and C. Rau (2000), "Tracking a smooth fault line in a response surface", *The Annals of Statistics*, vol. 28, pp. 713–733.

P. Hall and C. Rau (2002), "Likelihood-based confidence bands for fault lines in response surfaces," *Probability Theory and Related Fields*, vol. 124, pp. 26–49.

R.M. Haralick (1984), "Digital step edges from zero crossing of second directional derivatives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 58–68.

Y. Huang, H. Xu, V. Calian and J.C. Hsu (2006), "To permute or not to permute," *Bioinformatics*, vol. 22, pp. 2244–2248.

A. Huertas and G. Medioni (1986), "Detection of intensity changes using Laplacian-Gaussian masks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 651–664.

D.A. Lashkari, J.L. DeRisi, J.H. McCusker, A.F. Namath, C. Gentile, S.Y. Hwang, P.O. Brown and R.W. Davis (1997), "Yeast microarrays for genome wide parallel genetic and gene expression analysis," *Proceedings of the National Academy of Science*, vol. 94, pp. 13057–13062.

D. Marr and E. Hildreth (1980), "Theory of edge detection," *Proceedings of the Royal Society in London*, vol. B207, pp. 187–217.

V.S. Nalwa and T.O. Binford (1986), "On detecting edges," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 699–714.

W.K. Pratt (1991), *Digital Image Processing (2nd ed.),* New York: John Wiley & Sons.

P. Qiu (1997), "Nonparametric estimation of the jump regression surface," *Sankhya (Series A)*, vol. 59, pp. 268–294.

P. Qiu (2002), "A nonparametric procedure to detect jumps in regression surfaces," *Journal of Computational and Graphical Statistics*, vol. 11, pp. 799–822.

P. Qiu (2005), *Image Processing and Jump Regression Analysis,* New York: John Wiley & Sons.

P. Qiu and S.M. Bhandarkar (1996), "An edge detection technique using local smoothing and statistical hypothesis testing," *Pattern Recognition Letters*, vol. 17, pp. 849–872.

P. Qiu and C. Le (2001), "ROC curve estimation based on local smoothing," *Journal of Statistical Computation and Simulation*, vol. 70, pp. 55–69.

P. Qiu and J. Sun (2007), "Local smoothing image segmentation for spotted microarray images," *Journal of the American Statistical Association*, vol. 102, pp. 1129–1144.

P. Qiu and B. Yandell (1997), "Jump detection in regression surfaces," *Journal of Computational and Graphical Statistics*, vol. 6, pp. 332–354.

A. Rosenfeld and A.C. Kak (1982), *Digital Picture Processing (2nd ed.),* New York: Academic Press.

M. Steinfath, W. Wruck, H. Seidel, H. Lehrach, U. Radelof and J. O'Brien (2001), "Automated image analysis for array hybridization experiments," *Bioinformatics*, vol. 17, pp. 634–641.

J. Sun and P. Qiu (2007), "Jump detection in regression surfaces using both first-order and second-order derivatives," *Journal of Computational and Graphical Statistics*, vol. 16, pp. 289–311.

C. Thomas-Agnan (1990), "Smoothing periodic curves by a method of regularization," *SIAM Journal on Scientific and Statistical Computing*, vol. 11, 482–502.

V. Torre and T. Poggio (1986), "On edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 147–163.

A.B. van't Wout, G.K. Lehrman, S.A. Mikheeva, G.C. O'Keeffe, M.G. Katze, R.E. Bumgarner, G.K. Geiss and J.I. Mullins (2003), "Cellular gene expression upon human immunodeficiency virus type 1 infection of CD4(+)-T-cell lines," *Journal of Virology*, vol. 77, pp. 1392–1402.

Y.H. Yang, M.J. Buckley, S. Dudoit and T. Speed (2002), "Comparison of Methods for Image Analysis on cDNA Microarray Data," *Journal of Computational and Graphical Statistics*, vol. 11, pp. 108–136.