Segmentation of DNA microarray images using an adaptive graph based method

N. Karimi\textsuperscript{1}, S. Samavi\textsuperscript{1,2}, S. Shirani\textsuperscript{2}, P. Behnamfar\textsuperscript{3}

\textsuperscript{1}Department of Electrical and Computer Engineering, Isfahan University of Technology, Iran
\textsuperscript{2}Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada
\textsuperscript{3}Department of Electrical Engineering, University of British Columbia, Vancouver, Canada

Abstract: Microarray is a powerful tool for simultaneous study of the behavior of thousands of genes through analysis of produced images. The correct segmentation of each “spot” of the microarray image is a critical step in the analysis of the results of an experiment. In this paper a graph based method is proposed which automatically performs the segmentation. Performance of the algorithm is tested both on real and simulated images. The proposed algorithm successfully detected spots of different sizes and shapes under the presence of variable noise levels. The simulation results proved that the suggested approach has high segmentation accuracy.

1 Introduction

Microarray technology is a means which has enabled the researchers to measure simultaneously the expression levels of thousands of genes. This technology, which uses complementary DNA (cDNA) arrays, involves 2-channel experiments. These experiments employ two different cDNA probes that are prepared independently for the analysis. The first probe contains the control single strand cDNA and the second probe contains single strand cDNA originated from the treated cells. One of these probes is labeled with Cy3 (green) dye and the other probe is labeled with Cy5 (red). Both single strand probes are hybridized together on the same microarray substrate. Scanning the microarray slide to obtain Cy3 labels once and Cy5 the next time would generate two grayscale images. The relative intensities of the dyes in individual spots measure the relative amounts of specific transcripts in each sample. Therefore, differential gene expression can be analyzed \cite{1, 2}

Careful analysis of the obtained images is needed to obtain the information on the gene expression level. The image analysis for cDNA arrays consists of three steps: (1) the gridding step which places a grid on the image dividing the image into blocks where each block may contain a spot and a part of the background; (2) the segmentation step which partitions the set of pixels into foreground and background; and (3) the intensity ratio estimation step which uses the ratio of the extracted red (R) and green (G) intensities \cite{3}. Segmentation of microarray images is a challenging task due to many varying qualities of these images. For example, image contrast may change from one experiment to the other. Also, high background noise and image artifacts may be present and spots’ shapes and sizes differ within an image. Therefore, robustness of the segmentation method to perform under varying conditions is important \cite{4, 5}. Furthermore, it has been shown that the segmentation accuracy has direct effect on the outcome of an experiment \cite{6}.
A number of microarray image segmentation and analysis methods are currently available, some of which have been incorporated into commercial products and some are for research-only purposes. A simple segmentation method is based on the assumption that spots are circular with fixed, predefined radius. Then circles with a constant diameter can be fit to all spots in the image. One example of application of this algorithm is in the ScanAlyze [7] software tool. This method has also been used as an option in most microarray analysis softwares [8]. Adaptive circle segmentation assumes that the radius of each spot is not constant and is adjusted for every spot separately. GenePix [9] is a popular commercial package which employs the adaptive circle segmentation algorithm. GenePix has also the capability of segmenting spots that are non-circular with irregular shapes. Seeded region growing [10] (SRG) is another segmentation technique, which is used by SPOT [3] software package. This algorithm extracts foreground and background intensities with good accuracy. This method can adapt to various shapes of spots and requires the selection of initial seeds [8]. Mann-Whitney (MW) segmentation employs an iterative algorithm for computing the threshold between foreground and background using the image histogram [11]. Clustering-based segmentation has been applied to microarray images, which uses k-means clustering and partitioning around medoids (PAM). This would generate binary partition of the pixels based on the distribution of their intensities [12]. Segmentation of spots in microarray images has also been performed by Markov random field (MRF) method which models spot foreground and background intensities as 2D markov distributions. In addition to the intensity information, the method takes the spatial information into account by modeling the neighborhood pixel labeling with MRF [4]. Model based approaches perform the segmentation based on the distribution of pixel intensities in the foreground and background. Researchers have proposed Gaussian [13], t-distribution [14], and gamma-t mixture models [15] as the basis for the distribution of intensities in the foreground and background of images. Gaussian mixture model (GMM) [13] is a flexible method that has also been used for segmentation of the microarray images.

The above mentioned methods have diverse levels of complexity in terms of implementation, but complexity is not always translated into effectiveness and precision. A segmentation method is faced with a number challenges such as irregularities in shapes and sizes of spots, increasing density of spots on a single substrate, and artifacts caused by contamination of microarray substrates. These challenges have created a demand for progress in the precision and robustness of microarray image segmentation and analysis.

Computational power of modern computers has created an opportunity for researchers to apply graph theory to many image processing applications such as face recognition in biometrics, and segmentation of medical images [16]. In image processing algorithms that are based on graph theory the input image is usually converted to a weighted graph. Every pixel of the image is mapped to a vertex. If two vertices are neighbors then there is an edge between them. The weight assigned to an edge is the dissimilarity of the two pixels corresponding to two vertices of that edge. When graph theory is used for image segmentation a minimum spanning tree is usually formed [17-19]. Elimination of an edge with the largest weight in the minimum spanning tree would divide the tree into two subtrees. By eliminating another edge, in one of the subtrees, we can further divide that subtree into two components. Resemblance of a node in a subtree with its neighboring nodes is higher than the similarity of that node with its neighbors in other subtrees. Hence, by eliminating only one edge from the minimum spanning tree, pixels of the original image are segmented into two regions.

This method produces good segmentation results if there are smooth regions and high contrast exists between adjacent regions [18]. These conditions normally are not present in microarray images and hence customization of the graph based segmentation algorithms is required to obtain acceptable results.
In this paper a new method is suggested based on minimum spanning tree for segmentation of microarray images. To evaluate our method we compared it with a number of well known publicly available software packages such as ScanAlalyze and SPOT. Also, GenePix, as a powerful commercial software, and GMM, as a model based segmentation algorithm, were used for comparison purposes. Two modes of GenePix package, namely adaptive-circle (circular) and adaptive irregular shape (irregular), were used. These comparisons were performed on both real and simulated microarray images in terms of reliability of the extracted gene expression levels (which is the ultimate goal of microarray image processing).

We show by using a number of known criteria that the effectiveness of the proposed method is high as compared to other methods in the comparison pool. The proposed method operates independent of the shape and size of the spots and uses the spatial correlation that exists among the pixels of a spot. The remainder of the paper is organized as follows. Section 2 contains the proposed segmentation algorithm. Experimental results are offered in Section 3 along with the comparison of our method with the mentioned group of algorithms. Concluding remarks are offered in Section 4.

2 Proposed method

Before the proposed Microarray Graph based Segmentation algorithm (MiGS) is applied, the image is preprocessed and image gridding is performed. In order to achieve a better precision with the algorithm a 3×3 median filter is applied to the image for smoothing purposes and diminishing dark points inside the spot. Then a grayscale morphological opening operation, using a 2×2 structuring element, is performed [20]. Therefore, beside the reduction of noise, the inside region of spots are smoothed with minimal distortion of the boundaries of spots. These preprocessing steps are performed on a copy of the image to produce the segmentation mask and the original image remains preserved. Therefore, a copy of image is processed and altered to produce a segmentation mask. This mask eventually will be superimposed on the original unaltered image to segment the spots.

The next required step is to divide the image into rectangular blocks through a gridding process. At most one spot would fall within a grid block. The grid lines are obtained from the horizontal and the vertical profiles of the image. Also, coordinates of the estimated center of each potential spot is found in this process [21, 22]. The segmentation algorithm is then applied to each block. The outcome of the segmentation stage is a binary mask where each pixel of a spot is represented by a “1” and each background pixel would correspond to a “0”.

To explain our graph based algorithm let us define \( P \), according to equation (1), as the set of all pixels of a block with the height and width of \( H_B \) and \( W_B \).

\[
P = \{(i, j) | i, j \in \mathbb{N}, i \leq W_B, j \leq H_B\}
\]  

Let \( C \) be the set of possible grayscale values that a pixel could have as shown in equation (2).

\[
C = \{0, 1, 2, \ldots, 2^L - 1\}
\]  

Also, let \( I_B \), as defined in (3), be a function that assigns grayscale values to elements of \( P \).

\[
I_B : P \rightarrow C
\]  

Let us consider an undirected weighted graph \( G_B(V, E) \) corresponding to a block. Vertices and edges of this graph are defined in (4) and (5) respectively. An edge exists between two vertices of this graph if their corresponding pixels are neighbors where 4-neighborhood is considered.

\[
V(G_B) = P
\]
Draft: submitted to Image Processing Journal, IET, 2010

\[ E(G_B) = \left\{ \begin{array}{l}
(i = i') \text{ and } (|j - j'| = 1) \\
\text{or} \\
(j = j') \text{ and } (|j - i'| = 1)
\end{array} \right. \]

Equation (6) expresses the weight \( w(e) \) of an edge as the absolute value of the difference between the grayscale intensities of the two pixels corresponding to the vertices of that edge.

\[ w(e) = \{w|e = uv, u,v \in V(G_B), w = |I_B(u) - I_B(v)|\} \]

If \( T \) is a subtree of \( G_B \) then all of the elements in the mask corresponding to nodes in that subtree will have a value of 1, that is:

\[ Mask_T : V(G_B) \rightarrow \{0,1\}, \quad Mask_T((i,j)) \left\{ \begin{array}{l}
1 \text{ if } (i,j) \in V(T) \\
0 \text{ if } (i,j) \not\in V(T)
\end{array} \right. \]

Let \( N_{GB}(r) \) represent the neighboring vertices of a vertex \( r \) in \( G_B \). Also, the degree of a node \( r \) in \( G_B \) is called \( d_{GB}(r) \) and is the number of neighboring nodes of \( r \) or the number of elements in \( N_{GB}(r) \).

Using the above definitions we can now express the proposed segmentation algorithm as the flowchart of Figure 1. The weighted graph \( G_B \) would have small weights for smooth regions of an image block due to similarities of neighboring pixels in that region. At the boundaries between regions the dissimilarities cause larger weights in the graph.

The first step of MiGS algorithm is to form a minimum spanning tree (MST) \( T_B \) from the weighted graph of \( G_B \) (box 1 of Figure1). The goal is to choose an edge in \( T_B \) where by cutting it the minimum spanning tree is divided to two subtrees corresponding to the foreground and background regions of the image. The selection process consists of two steps. In the first step a group of candidate edges are selected. A number of tests are run on these candidate edges and only those which have certain condition will go to the second step of the selection process (boxes 2 to 13). In the second step another criteria is applied on the refined group of candidate edges (boxes 14 to 19). Eventually, from the pool of candidate edges, one edge is selected. By elimination of this selected edge the two desired subtrees are formed and the segmentation process is finished with the generation of the segmentation mask.

The first step in the selection process starts with descendingly ordering the edges of \( G_B \) based on their weights. Then the first \( k \) edges are placed in set \( Q \) (box 2). Each edge in set \( Q \) is once eliminated (in a separate and independent test) from \( T_B \) to divide \( G_B \) into two subtrees. Let us assume that \( u \) and \( v \) denote the vertices of edge \( q_i \), a member of \( Q \). As stated in box 4, by eliminating \( q_i \) two subtrees, namely \( T_1(q_i) \) and \( T_2(q_i) \) are produced. Each one of these subtrees represents a distinct region in the block. We have to decide which of these two regions corresponds to the spot and which one is the background. One of the two mentioned subtrees which has fewer pixels on the boundary of the block (\( G_B \)) is chosen as \( \text{Spot}(q_i) \). This is performed using node degrees, \( d_{GB}(r) \). As mentioned in box 5, the subtree, which has least block-border pixels, \( \text{Spot}(q_i) \) is chosen as the spot component. Then we want to use the centroid of the spot region to see if it is close to the center of the grid or not. The centroid of the region of the block denoted by \( \text{Spot}(q_i) \) is found as \((\hat{x}, \hat{y})\) (box 6). The Euclidian distance of this centroid to the estimated center of the spot, \((x_B, y_B)\), is calculated as \( \hat{d}(\text{Spot}(q_i)) \).
Figure 1 Steps of proposed MiGS algorithm
In order to distinguish a spot from possible noise specks, the \( \text{Spot}(q_i) \) component has to be larger than a certain threshold and it should also be close to the center of the block. This means that \( \tilde{d}(\text{Spot}(q_i)) \) distance should be smaller than a threshold \( d_B \) and the number of nodes of \( \text{Spot}(q_i) \) has to be larger than a threshold \( n \). Under these circumstances \( q_i \) becomes a member of \( Q' \). Threshold \( n \) indicates the minimum acceptable spot size. The first step of the selection process, as expressed in boxes 2 to 13, is performed by eliminating an edge \( q_i \) from the tree to form two subtrees. Each subtree is studied. Then the eliminated edge is placed back in the tree and the next edge from the set \( Q \) is eliminated. This is done independently for all of the edges in set \( Q \). Members of the set \( Q \) which pass the above criteria will be in set \( Q' \) and go to the second step of the selection process.

In the second step of the selection process all of the candidate spot components are tested for compactness (box 17). From a pool of candidate spot components we intend to choose the one which has higher resemblance to a circle. This is due to the fact that microarray technology is designed to generate circular spots and the spots actually turn out to have roughly circular shape. For each \( q_i \) member of \( Q' \) the number of pixels that lay on the boundary of region represented by \( \text{Spot}(q_i) \) is computed as \( \alpha(\text{Spot}(q_i)) \). Dividing \( \alpha(*)^2 \) by the area of the region would give a measure of the compactness of that region [23]. Out of all candidate regions that may be formed by eliminating a \( q_i \) the one which has a more circular shape, based on its compactness, is chosen as \( T_3 \). By using the \( \text{Mask} \) function a binary mask is formed for the block where any coordinate that corresponds to a vertex in \( T_3 \) would be 1 and other coordinates will contain zeros. In case no edge can pass the tests performed in the first step of the selection process it means no spot exists in that block and the corresponding segmentation mask will contain all zeros.

For the implementation of the algorithm the Kruskal’s method [24] was used to generate the minimum spanning tree. We chose number of edges equal to 10 and \( d_B \) equal to half of a spot’s average radius. Threshold \( n \) is chosen to be equal quarter of an average spot’s area. The average spot radius is known for each manufacturer.

### 3 Simulation results

#### 3.1 Overview

We tested our proposed microarray image segmentation method on both simulated microarray images as well as real images. As mentioned before the preprocessed images go through a gridding routine before we applied the proposed MiGS algorithm. An approach similar to the one proposed in [22] was used for the gridding process.

Figure 2 shows some examples of the inputs and the results of the algorithm. Figure 2(a) illustrates three typical grid units with roughly circular shape spots. Figure 2(b) shows \( T_3 \) which corresponds to the spot region. Also, the tree that surrounds \( T_3 \) and represents the background is shown in Figure 2(b). Figure 2(c) shows the final segmented blocks. The boundary around each spot is produced from the corresponding binary mask.

We have used a large number of tests on simulated images where the results will be presented in the next subsection. We also have used real microarray images to evaluate our algorithm where the results
are shown in subsection 3.2. As mentioned before we compared our method with those of ScanAlyze, GenePix (Circular), GenePix (Irregular), SPOT, and GMM (which is a model based approach).

![Simulated microarray images](image)

**Figure 2** a) Three samples of grid units, b) $T_3$ and surrounding tree, c) segmented grid units

### 3.2 Simulated microarray images

Evaluating a segmentation algorithm designed for microarray images is a difficult task since it is not possible to establish any essential ground truth information [8]. In order to alleviate this difficulty we created two sets of simulated microarray images by using a simulator proposed in reference [25]. Since for these simulated images each pixel is either assigned to foreground or background we have the ground-truth information available.

Each image set has different quality characteristics and consisted of 10 images. Each simulated image consists of ten subarrays with a total of 1000 spots per image. The first image set is of normal quality, with low noise variance (0.02) and relatively round shapes and regularly distributed spot sizes. On the other hand, the second set has more disturbing noise with higher variance (0.04) and high variability in spot shapes and sizes compared to another set. Also, the position of spots varies as compared to the normal quality images. Therefore, 10,000 normal quality spots and 10,000 low quality spots are tested.
In order to assess the pixel based segmentation accuracy of MiGS, we selected two traditional measures with different perspectives [26]. The first method measures the probability of error ($P_{\text{Error}}$) [27] which reflects the number of background pixels wrongly assigned to foreground, as well as foreground pixels assigned to background. For two-class segmentation problem with foreground and background pixels, $P_{\text{Error}}$ is defined as:

$$P_{\text{Error}} = P(F) \times P(B \mid F) + P(B) \times P(F \mid B)$$  \hspace{1cm} (8)

where $P(B \mid F)$ is the probability of error in classifying foreground pixels as background pixels, $P(F \mid B)$ is the probability of error in classifying background pixels as foreground pixels, $P(F)$ and $P(B)$ are a priori probabilities of foreground and background pixels in image. The minimum value of zero occurs for $P_{\text{Error}}$ when all of the pixels of the grid unit are segmented correctly. A maximum value of 1 for $P_{\text{Error}}$ indicates a situation that all of the pixels of the background are segmented as foreground and vice-versa.

The second method, which is called discrepancy distance [28], is based on the position of misclassified pixels and is defined as

$$D = \frac{\sqrt{\sum_{i=1}^{N} d^2 (i)}}{A}$$  \hspace{1cm} (9)

In equation (9), $N$ is the number of misclassified pixels, $d(i)$ the Euclidean distance between the $i^{\text{th}}$ misclassified pixel and the nearest pixel of its true class and $A$ is the number of pixels in the image. When all of the pixels of the grid unit are correctly segmented a minimum value of 0 is calculated for $D$. On the other hand, the maximum value of $D$ occurs when every foreground pixel of the grid unit is segmented as background and vice-versa. The maximum value of $D$ depends on the size of the grid unit, the size, and the shape of the shape of the spot. For example for a 20x20 pixel grid unit with a round spot of radius 10 in the middle, the maximum value of $D$ is 0.3. Table 1 shows estimates for the mean probability of error and discrepancy distance for both normal and low quality image sets. These results show high segmentation accuracy of the proposed method. While there were significant variations in the quality of simulated images, the mean probability error and discrepancy distance for all of the simulated spots were very low.

<table>
<thead>
<tr>
<th>Table 1 Pixel-based accuracy measurement of the proposed method</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Probability of Error ($P_{\text{Error}}$)</strong></td>
</tr>
<tr>
<td>Normal quality</td>
</tr>
<tr>
<td>Low quality</td>
</tr>
<tr>
<td>0.0121</td>
</tr>
<tr>
<td>0.0211</td>
</tr>
<tr>
<td><strong>Discrepancy Distance ($D$)</strong></td>
</tr>
<tr>
<td>Normal quality</td>
</tr>
<tr>
<td>Low quality</td>
</tr>
<tr>
<td>0.0174</td>
</tr>
<tr>
<td>0.0287</td>
</tr>
</tbody>
</table>

Pixel-based segmentation accuracy is an objective evaluation of a segmentation algorithm [8, 29]. But publicly available packages that are in our pool of comparison do not provide segmentation masks. Therefore, to compare our proposed method to other microarray segmentation methods we calculated the pairwise differences between the extracted average spots’ intensities and the original simulated image spots’ intensities using the absolute error (AE) [29].

The AE values for the normal and low quality simulated images are respectively illustrated in Figures 3 and 4. Boxplot representation of our method and those of other methods are presented. For comparison purposes, the mean value of the AE’s of each method is also shown in Table 2 for both of the mentioned sets of images.
Figure 3 Boxplot representation of absolute error between segmented spot intensities and original spot intensities for normal quality set of images.

Figure 4 Boxplot representation of absolute error between segmented spot intensities and original spot intensities for low quality set of images.
Table 2 The mean value of the absolute errors of each of the compared methods

<table>
<thead>
<tr>
<th>Images Set</th>
<th>Proposed Method</th>
<th>ScanAlyze</th>
<th>GenePix (Circular)</th>
<th>GenePix (Irregular)</th>
<th>SPOT</th>
<th>GMM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal Quality</td>
<td>0.0086</td>
<td>0.0495</td>
<td>0.0252</td>
<td>0.0229</td>
<td>0.0148</td>
<td>0.0082</td>
</tr>
<tr>
<td>Low Quality</td>
<td>0.0094</td>
<td>0.0768</td>
<td>0.0343</td>
<td>0.0412</td>
<td>0.0273</td>
<td>0.0127</td>
</tr>
</tbody>
</table>

As shown in Figures 3 and 4 as well as Table 2, the proposed method outperformed other methods by segmenting spots and extracting intensities that are closer to those of the simulated images. At low noise levels, where normal quality images were used, the proposed method performed better than commercial packages and was comparable with GMM. But the performance of the proposed method is more pronounced for low quality images where the proposed method robustly outperforms all of the mentioned algorithms.

### 3.3 Real microarray images

Although real microarray experiments do not allow direct performance measurements for segmentation results, indirect measures provide possibility for comparing different algorithms [8]. As we mentioned before it is not possible to obtain direct ground-truth information about real microarray images. For example we do not know the exact gene expression levels or the possible background noise of a certain experiment when an image was produced. To alleviate this problem researchers have been using images from replicated experiments. Of course, the main goal of the replicated experiments is the reduction of experimental variation in the extraction of genetic information. There are two possible methods of replication. One method uses different microarray substrates for the same experiment [30]. Another option is to use spot-replicated microarray images where one spot is replicated a number of times on a single substrate. Replicate spots which are side-by-side are likely to be very highly correlated since they are not only printed with the same gene but are also spatially close together and therefore likely to share many common causes including local effects on the array surfaces as well as hybridization and labeling effects [31].

Application of different algorithms on images from replicated spot experiment could result in suitable comparison between these algorithms. Hence, to compare MiGS’s performance with those of other segmentation methods we used the available spot-replicated real microarray images. The real data set used in this study was the results of three experiments on the breast cancers which produced 6 images. Every experiment produces one Cy3 and one Cy5 image. Each image contains 16 blocks where there are 21x21 spots per block. It should be mentioned that one of the experiments has one row of spots less than the other two. One of the features of these real images is that some bright contamination spots exist irregularly and are scattered over the array. To guarantee reliability of the experiment each spot is triplicated. The replicated spots are placed next to each other in a row. Hence close to 7000 triple spots were used in our experiments. These images are available at the UCSF Cancer Research Institute [32].

The segmentation process using our proposed method was performed and the gene expression ratio was calculated for each spot by subtracting the background intensity from the average intensity of the spot. This is done both for the Cy3 and Cy5. The obtained quantity for each spot in Cy5 is divided by the corresponding Cy3 quantity. For every replicated group of three spots the same ratio was calculated and then, the pairwise absolute values of errors between expression ratios for replicated group of spot are calculated and the mean absolute error (MAE) is then obtained [8, 29]. MAE of spots using other segmentation and/or microarray analysis software tools was calculated similarly.

A number of algorithms segment pixels into a spot, while in some algorithms complete absence of a spot is possible. Absence of a spot may result when the low pixels intensities create such circumstances
or when the algorithm is not capable of identifying a spot. When a spot is missing the calculation of expression ratio makes no sense and such spots should not be in the analysis. On the other hand if an algorithm ignores low intensity spots and only considers high quality spots then our performance evaluation would fail. Therefore, in our analysis we only selected those spots that all of the algorithms were able to segment all three replicas of them. By doing so, we ignored 1604 sets of triple spots from the image pool and considered 5340 sets in our analysis.

Figure 5 shows the boxplot of MAE’s obtained from different segmentation algorithms. Table 3 illustrates the mean values of the boxplots of Figure 5. Lower MAE values correspond to higher segmentation performance which in turn shows higher accuracy in the extraction of gene expression levels. Again, as shown in Table 3, the proposed method achieved better results than the algorithms that are compared with.

![Boxplot of MAE’s obtained from different segmentation algorithms for real set of images.](image)

**Figure 5** Boxplot of MAE’s obtained from different segmentation algorithms for real set of images.

<table>
<thead>
<tr>
<th>Proposed Method</th>
<th>ScanAlyze</th>
<th>GenePix (Circular)</th>
<th>GenePix (Irregular)</th>
<th>SPOT</th>
<th>GMM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mean values of boxplots of Figure 5</strong></td>
<td>0.2340</td>
<td>0.3940</td>
<td>0.3414</td>
<td>0.3365</td>
<td>0.3026</td>
</tr>
</tbody>
</table>

MiGS successfully detected spots of different sizes and shapes under the presence of variable noise levels. Figure 6(a) shows a section of a low quality real microarray image. The segmentation results from the application of MiGS are shown in Figure 6(b). It is shown that despite the presence of noise the algorithm was capable of segmenting spots with regular as well as irregular shapes in almost all instances. MiGS was implemented by Matlab 7.0.4 software tool. The results are produces in average of 0.21 seconds per spot on a Pentium 4 (2.5 GHz) with 2GB of RAM. It should be noted that the execution time that is mentioned is just indicative and the codes can be optimized for faster operations.
4 Conclusion

In this paper we proposed a new segmentation method for microarray images based on graph theory. Using spanning trees we incorporated intensity characteristics of the image into the algorithm. Also, the shape information of the spots is exploited by embedding the compactness criterion into the routine. The whole process is autonomous and requires no supervision which is one of the advantages of the proposed algorithm. Both synthetically produced microarray images and real images were used. The accuracy of the algorithm was higher than the publicly available software packages.

5 References


