Grow-Cut Based Automatic cDNA Microarray Image


Stamos Katsigiannis, Eleni Zacharia, and Dimitris Maroulis, Member, IEEE

Abstract— Complementary DNA (cDNA) microarray is a well-established tool for simultaneously studying the expression level of thousands of genes. Segmentation of microarray images is one of the main stages in a microarray experiment. However, it remains an arduous and challenging task due to the poor quality of images. Images suffer from noise, artifacts, and uneven background, while spots depicted on images can be poorly contrasted and deformed. In this paper, an original approach for the segmentation of cDNA microarray images is proposed. First, a preprocessing stage is applied in order to reduce the noise levels of the microarray image. Then, the grow-cut algorithm is applied separately to each spot location, employing an automated seed selection procedure, in order to locate the pixels belonging to spots. Application on datasets containing synthetic and real microarray images shows that the proposed algorithm performs better than other previously proposed methods. Moreover, in order to exploit the independence of the segmentation task for each separate spot location, both a multithreaded CPU and a graphics processing unit (GPU) implementation were evaluated.

Index Terms— cDNA microarrays, spot segmentation, grow-cut, image analysis, CUDA

I. INTRODUCTION

The use of cDNA microarray technology has been proven to be a powerful tool in the hands of biomedical researchers. Microarrays are being increasingly utilized in numerous fields of biomedical research, assisting research
on cancer and pharmacology, as well as diagnosis and treatment of infectious diseases. Their most advantageous attribute is that they offer a means to simultaneously monitor the expression levels of thousands of genes over different samples [1][2]. In order to conduct a microarray experiment [3], first a test and a reference mRNA sample are isolated. These two mRNA samples are reversely transcribed into complementary DNA (cDNA), amplified using polymerase chain reaction, and then labeled. Labeling is usually performed by means of two distinct fluorescent dyes such as the red Cy5 and the green Cy3. Then, the two labeled samples are hybridized on a DNA array which is a microarray device consisting of a solid substrate where single-stranded chains of known DNA sequences are attached. Subsequently, the DNA array is scanned at the wavelength of each dye, producing a high resolution greyscale digital image for each initial mRNA sample. Each digital image depicts spots, ideally arranged in a matrix of blocks, with each block containing spots in a 2D grid structure. Spots correspond to known DNA sequences. The grey level intensity of each spot indicates the hybridization level of the cDNA sample to the single-stranded chain of the known DNA sequence and thus indicates the expression level of the corresponding gene. The straightforward approach for analyzing DNA microarray images consists of three main stages [4]. First, a grid is constructed for each block of the image so that each spot is isolated inside a distinct quadrilateral (gridding). Second, the pixels of each quadrilateral are distinguished between pixels belonging to spots and to background (spot segmentation). Third, the intensity level of each spot is quantified in order to assess the expression level of the corresponding gene.

Microarray image analysis can be a challenging and computationally intensive task due to the low quality of the examined images, which are usually contaminated with noise and artifacts [5]. Moreover, as a result of imperfect sample preparation and hybridization processes, spots may differ from the ideal ones, varying in size, shape and position. Additionally, the spots position may deviate from the aforementioned 2D grid structure resulting to misalignments, rotations and local deformations of the ideal rectangular grid.

A number of software packages has been developed and is available to biomedical laboratories. Some well-known software packages are ScanAlyze [6], Dapple [7], ImaGene [8], SpotFinder [9] and QuantArray [10]. Moreover, many methods have been proposed in the literature for all stages of microarray image analysis. For the segmentation stage, methods can be classified into two categories according to [4]: a) shape-segmentation approaches and b) shape-independent approaches. Hybrid approaches, utilizing prior knowledge for both shape and intensity, have also been proposed.
Shape-segmentation approaches operate on the basis that a spot has a certain shape. The fixed circle segmentation algorithm implemented by the ScanAlyze software program [6], as well as the adaptive circle segmentation algorithm of Buhler et al. [7], attempt to match circular templates to spots. Sarder et al. [11] proposed the use of a parametric circle with one or two elliptical center holes in order to segment noisy microarray spots. The main drawback of the aforementioned methods is their inability to efficiently segment non-circular spots.

Shape-independent approaches attempt to address the aforementioned problem. The Spot software program [12] utilizes a seeded region growing algorithm in order to segment spots of irregular shapes. The utilized algorithm efficiently addresses this problem but is significantly sensitive to the appropriate selection of seeds. Other approaches utilize clustering algorithms such as K-means and hybrid K-means in order to distinguish pixels belonging to spots from pixels belonging to the background [13]-[16]. Nevertheless, these methods underperform in the case of poorly contrasted spots and in the case that spots are in very close proximity. Another method based on the clustering of pixels' values introduces a removal technique that rejects small disconnected clusters [17]. Small clusters are assumed to be artifacts. However the main disadvantage is that the minimum size of a cluster has to be manually decided. Other proposed shape-independent segmentation methods are based on active contours and multiple snakes [18]-[20], which are very effective in segmenting typical microarray spots but their performance may be affected in the presence of noise and artifacts.

Hybrid approaches include methods based on Markov random fields (MRF) like [21] and [22] which combine neighboring and intensity information based on an MRF modeling of the quadrilaterals of the microarray image. Gottardo et al.'s method [23] is also based on MRFs, where intensities for both foreground and background are represented using an uncorrelated bivariate $t$-distribution. A spot segmentation method based on the correlation statistics of spot pixels has been proposed by Nagarajan et al. [24]. This method is efficient in detecting low intensity spots but underperforms in the case of low quality images. Zacharia et al. [25] proposed the use of a combination of 3D spot modeling and genetic algorithm optimization. This approach provided enhanced results, even in the presence of noise and artifacts, but at the cost of great computational complexity due to the genetic algorithm.

In this work, the authors propose and evaluate a novel approach for cDNA microarray image segmentation. A preprocessing step is initially performed in order to reduce the effects of noise. Then, the grow-cut algorithm [26] is applied separately in each quadrilateral in order to perform the segmentation. Grow-cut is a region-growing
algorithm based on *Cellular Automata* [27], where pixels are modeled as *cells* that are in a specific state. The state of each *cell* changes depending on the states of its neighbors as determined by a local update rule. The grow-cut algorithm utilizes continuous state cellular automata and user-defined seeds in order to label images [28] and has been successfully utilised in its original and expanded forms for medical image segmentation [29]-[32]. Contrary to the original grow-cut approach, in this work the authors propose a method for automated seed selection in order to avoid user intervention.

The rest of this paper is organized in three sections. Section II describes the proposed microarray image segmentation algorithm. A detailed evaluation and discussion of the proposed methodology is provided in Section III, whereas conclusions are drawn in Section IV.

II. METHODOLOGY

This work focuses on spot segmentation. Given that spot-segmentation is the second stage in the process of microarray image analysis, gridding is necessary to be performed in advance in order to obtain the grid structure of the image and thus the location of quadrilaterals that might contain spots. Several methods for microarray image gridding have been proposed in the literature and several software packages are available in laboratories. In this work, the Zacharia *et al.* [33] method is used in order to perform the gridding process. This gridding method utilizes a genetic algorithm approach in order to determine the line-segments constituting the grid of the microarray image and was selected due to its robustness against noise, artifacts, rotations and other grid deformations. Nevertheless, any gridding method can be utilized since the proposed segmentation methodology is independent from the gridding algorithm. The proposed segmentation method is divided into two steps. A preprocessing step is initially applied in order to reduce the effects of noise. Then, the grow-cut algorithm [26] is applied separately on each quadrilateral of the grid in order to distinguish between pixels belonging to spots and to background.

A. Preprocessing

A simple median filter is applied on a 3x3 pixel neighborhood in order to eliminate random noise while preserving the edges of the spots. The intensity value of each pixel is replaced by the median intensity value of the 3x3 neighborhood around the pixel. Neighborhoods larger in size can introduce significant distortion to the image due to the close proximity of spots and thus are avoided. Examples of filtering are shown in Fig. 1 and 2 for regions of a
real and a synthetic microarray image respectively.

B. Grow-cut

Introduced by Vezhnevets and Konouchine in [26], the grow-cut algorithm is a region growing algorithm that can segment a greyscale or multichannel image into regions belonging to multiple classes (i.e. \(K\) classes).

Each pixel \(p\) of the image is characterized by a triplet \((l_p, \theta_p, C_p)\). \(l_p\) is a label which represents the class that pixel \(p\) belongs to. \(\theta_p\) denotes the “strength” of pixel \(p\), i.e. a measure of certainty that \(p\) should be labeled as \(l_p\). \(\theta_p\) takes values between 0 and 1. A pixel \(p\) with “strength” equal to 1 indicates a pixel whose label cannot change, whereas a pixel with “strength”<1 indicates a pixel whose label may change during the progress of the algorithm. \(C_p\) denotes a feature vector defined by the image. In the case of multichannel images, \(C_p\) is a multidimensional vector containing the intensity of pixel \(p\) at each channel, whereas in the case of greyscale images \(C_p\) is an integral value equal to the intensity of pixel \(p\).

The goal of the grow-cut algorithm is to assign to each pixel one of the \(K\) possible labels depicting the class they belong to. Initially, all the pixels labels are set to “Undefined” and their “strength” to 0. In order to start the segmentation, the user provides the initial seeds by setting their label and “strength” accordingly. Then, the following procedure is iterated until all pixels have been labeled or a termination criterion has been met: For each pixel \(p\) and its adjacent neighbors \(q_i\) (\(i=1,...,8\)) the quantity \(g(q_i)\) is computed. \(g(q_i)\) is a monotonic decreasing function bounded to \([0,1]\) which is defined as:

\[
g(q_i) = 1 - \frac{\|C_p - C_{q_i}\|_2}{\max\|C\|_2} \tag{1}
\]

In the case of greyscale images, \(\max\|C\|_2\) is equal to \(2^L-1\), with \(L\) being the bit-depth of the image and \(\|C_p - C_{q_i}\|_2\) is equal to \(|I_p - I_{q_i}|\), with \(I_p\) and \(I_{q_i}\) being the intensity values of pixels \(p\) and \(q_i\), respectively.

Afterwards, for all the pixels \(q_i\) whose label is not “Undefined”, the quantity \(\lambda(q_i)\) is computed as:

\[
\lambda(q_i) = g(q_i) \cdot \theta(q_i) \tag{2}
\]

In the case \(\lambda(q_i) > \theta_p\), pixel \(p\) receives the label of \(q_i\) \((l_p=l_{q_i})\) and a “strength” equal to \(\lambda(q_i)\) \((\theta_p=\lambda(q_i))\), otherwise it keeps its own label and “strength”. After a number of iterations the algorithm finishes when it reaches a state of
convergence where all pixels are labeled and pixel labels cease to change.

C. Application of grow-cut to microarray images

In this work, the grow-cut algorithm is applied separately at each quadrilateral of the image provided by the gridding process. Pixels can be characterized by only two labels: (1) “Background” and (2) “Spot”, and since the input image is greyscale, $\max\|C\|_2$ is set as $2^L-1$ and $\|C_p-C_{qi}\|_2$ as $|I_p-I_{qi}|$. The initial seeds are given a “strength” value equal to 1 and are determined automatically as follows: Let $p_1, p_2, p_3$ and $p_4$ be the pixels located at the four points of intersection of the line-segments that form the examined quadrilateral, as shown in Fig. 3. The four pixels $p_{B1}, p_{B2}, p_{B3}$ and $p_{B4}$ located at the coordinates $(i_{p_1}+1, j_{p_1}+1), (i_{p_2}+1, j_{p_2}-1), (i_{p_3}-1, j_{p_3}+1), \text{ and } (i_{p_4}-1, j_{p_4}-1)$ respectively are labeled as “Background”. In case any of these pixels is located outside the quadrilateral, then its nearest neighbor located inside the quadrilateral is used instead. Moreover, the pixel $p_c$ located at the point of intersection of the two diagonal line-segments $p_1p_4$ and $p_2p_3$ is labeled as “Spot”. Since the spots are of circular shape and occupy the majority of space inside a quadrilateral, the probabilities of the pixel $p_c$ to belong to a spot as well as of the pixels $p_{B1}, p_{B2}, p_{B3}$ and $p_{B4}$ to belong to the background are very high and thus these pixels can be characterized as sufficient seeds for the problem at hand. Fig. 4 depicts an example of seed selection and segmentation result for a region of a synthetic microarray image.

It is worth pointing out that in the case that no spot exists inside the quadrilateral the aforementioned seeds fail to provide successful segmentation results. In this case the algorithm labels the area near $p_c$ as “Spot” resulting to the emergence of spurious spots at the segmentation results. In order to address this limitation, a seed elimination criterion is applied as follows: After determining the initial seeds, the second maximum intensity $I_{BG_{max2}}$ of the background seeds is determined. The maximum intensity of the background seeds is not utilized in order to reduce the odds of it being the result of noise. If the intensity of the spot seed pixel $p_c$ is equal or less than $I_{BG_{max2}}$ then $p_c$ is labeled as “Undefined” and its “strength” $\theta_{pc}$ is set to 0. Applying the grow-cut algorithm without a “Spot” seed results to the whole quadrilateral being labeled as “Background” and thus eliminates the problem of spurious spots emerging due to incorrect initialization of the grow-cut algorithm. An example of the effect of the seed elimination criterion is shown in Fig. 5.
D. Implementation

Since the segmentation of each quadrilateral is an independent task, quadrilateral segmentation can be performed in parallel by utilising a parallel computing architecture. Two different implementations were evaluated. The first one is a CPU implementation that utilises the OpenMP API [34] for multithreading. Each quadrilateral of the microarray image is assigned to a separate thread, with the maximum number of threads being the number of CPU cores. The multithreaded CPU approach provides enhanced performance compared to a single-threaded approach but is not able to fully exploit the complete independence of the segmentation tasks. The second implementation utilises the NVIDIA CUDA architecture [35] in order to exploit the vast parallel computational resources of modern graphics processing units (GPUs). Each quadrilateral is assigned to a separate CUDA thread and the segmentation of all quadrilaterals is performed in parallel, ideally for all the quadrilaterals simultaneously. Since the CUDA architecture organizes the execution of threads in blocks of threads, the number of allocated CUDA blocks should be determined. In order to calculate this number, the number of quadrilaterals in the microarray image $N_{quad}$ is first determined and then $N_{CB}$ CUDA blocks are used to perform the segmentation process. $N_{CB}$ is defined as:

$$N_{CB} = \left\lceil \frac{N_{quad}}{N_{TPB}} \right\rceil$$  \hspace{1cm} (3)$$

where $N_{TPB}$ the maximum number of threads per CUDA block supported by the utilised GPU.

III. RESULTS AND DISCUSSION

Several experiments on both synthetic and real cDNA microarray images were conducted in order to evaluate the performance of the proposed methodology.

A. Experiments on synthetic microarray images

In order to provide an objective and quantifiable comparison, the proposed methodology was evaluated against other methods proposed in the literature using a publicly available microarray image dataset which consists of 100 synthetic microarray images and their ground truth information, i.e. each pixel is pre-assigned either to spot or background class. The dataset is divided into two sets, with one set containing 50 “good” quality images and the other set containing 50 “low” quality images. All images have been created using the microarray simulator by Nykter at al. [36] that has the ability to produce microarray images with realistic characteristics. Each image contains 1000 spots and has a size of 750x330 pixels. Images belonging to the “good” quality set contain spots with low variability
in size and shape and their noise level is considerably low. On the contrary, images belonging to the “low” quality set exhibit significant variations in spot size and shape, as well as high levels of noise.

On [37] and [25] the aforementioned dataset has been utilized for the comparison of various microarray image segmentation techniques proposed in the literature (ten in total). In order to achieve a direct comparison to the experimental results of these techniques, the proposed method has been evaluated using the same statistical analysis procedure and the same two measures:

1) The probability of error (PE). PE measures only the mis-segmented pixels and is defined by the following formula:

\[ PE = P(S) \cdot P(B|S) + P(B) \cdot P(S|B) \]  \hspace{1cm} (4)

\( P(S|B) \) is the probability of background pixels being classified as spot pixels and \( P(B|S) \) is the probability of spot pixels being classified as background pixels. \( P(S) \) and \( P(B) \) are the \textit{a priori} probabilities of spot and background pixels respectively.

2) The discrepancy distance (DD). DD provides different weights for mis-segmented pixels based on their spatial distance from the nearest correctly segmented pixel and is defined as:

\[ DD = \frac{\sum_{i=1}^{N} d(i)^2}{A} \]  \hspace{1cm} (5)

\( A \) is the total number of pixels in the image, \( N \) the number of mis-segmented pixels and \( d(i) \) the Euclidean distance from the \( i^{th} \) mis-segmented pixel to the nearest pixel that belongs to the mis-segmented class.

Table I depicts the results for the proposed method, as well as for ten previously proposed methods obtained from [37] and [25] using the same dataset of synthetic microarray images. Examining the experimental results, it is evident that the proposed approach can efficiently segment the spots of both the “good” and “low” quality images. In the case of “good” quality images, the proposed method matches the highest performance achieved by the K-means [13] and the Zacharia et al. [25] methods. However, in the case of the “low” quality images the proposed method outperforms all the other examined methods, demonstrating its efficiency. Fig. 6 and 7 depict the segmentation results for a “good” and “low” quality synthetic microarray image respectively. As one may observe, the resulting mask of the proposed method is almost similar to the ground truth.

In order to evaluate the sensitivity of the proposed method to the median filtering step, a dataset of noisy images was created by adding Gaussian noise to 10 synthetic images from the “good” quality images of the aforementioned
dataset. The PSNR of the images varied from 15 dB to 49.5 dB. The size of the median filter was set to 3x3, 5x5, and 7x7 and performance was evaluated in terms of PE and DD. Fig. 8 and 9 show the performance of the proposed method in terms of the PE and DD measures respectively. It is evident that the use of a 3x3 filter provides the best results.

In order to evaluate the seed elimination criterion, a new dataset was also created since the ground truth shows no existing missing spots. 20 blocks of “good” and 20 blocks of “low” quality images were randomly selected from the synthetic image dataset. At each block, 15 quadrilaterals containing a spot were replaced by quadrilaterals of the same size, randomly selected from the background of each image. This process lead to 15 spots missing from each block. The proposed criterion managed to detect 86.67% and 73.33% of the missing spots for the “good” and the “low” quality blocks respectively.

Except for the evaluation on pixel-level accuracy, the proposed method was also evaluated on intensity-level accuracy. The mean intensity for each spot in the synthetic image dataset was extracted based on the proposed method’s results as well as based on the ground truth. Then, the difference between the intensities was computed. The median difference for the “good” quality images was 0% and the average difference 0.8%. For the “low” quality images the median difference was also 0%, with the average difference being 3.3%.

B. Experiments on real microarray images

The proposed segmentation method was also evaluated on two sets of real cDNA microarray images. The first set was obtained from the publicly available Stanford Microarray Database (SMD) [38]. The SMD microarray images have been produced by comprehensively analyzing the gene expression profiles in 54 specimens of acute lymphoblastic leukemia. The SMD images exhibit high levels of noise and contain several spots whose intensity value is significantly low and difficult to distinguish from the background. It is worth mentioning that in order for the human eye to observe the spots in the selected images, the brightness should be adjusted due to the low intensity of spot pixels. From the selected images, 100 blocks were arbitrarily selected for the experiments. Each block contained 867 spots and had a size of 450x450 pixels at 16-bit grey level depth. Since the ground truth information is not available for real cDNA microarray images, performance was evaluated through visual inspection of the results. Fig. 10 and 11 illustrate a real microarray image as well as the segmentation results for one of its blocks. On this image, it is apparent that the proposed method has efficiently segmented the real microarray spots.
The second set included six replicated real microarray images and was used in order to provide a quantifiable evaluation of the proposed method. These images are the result of three experiments on breast cancer (each experiment produced a Cy3 and a Cy5 image) and are publicly available from the UCSF Cancer Research Institute [39]. The size of the images is 1024x1024 pixels and each image contains 16 blocks of 21x21 spots. For the evaluation, the “empty” spots were excluded from the analysis, as proposed in [37]. It should be noted that one of the experiments contains one row of spots less than the other two. As a result, this row was also excluded from the analysis. After applying the proposed segmentation method, the mean intensity of each spot and the mean background for each spot were extracted. Then, the background intensities were subtracted from the spot intensities and the background subtracted intensities corresponding to Cy5 were divided by the ones corresponding to Cy3 in order to calculate the gene expression ratio. Then, the pairwise absolute values of error between expression ratios for the replicated spots were calculated and the mean absolute error (MAE) between the replicated spots was obtained. The proposed method achieved a mean MAE equal to 0.3651 for the examined set of replicated real microarray images.

The importance of the use of the seed elimination criterion described in the previous section is demonstrated in Fig. 10. Fig. 10 depicts segmentation results for a region of the real microarray image depicted in Fig. 11, with and without the use of the seed elimination criterion. Annotated regions of the image indicate some of the cases of emergence of spurious spots in case the seed elimination criterion is not utilised. It is evident that the regular occurrence of spurious spots significantly degrades the quality of the segmentation results and would lead to serious mistakes in the outcome of a microarray experiment.

TABLE I
SEGMENTATION RESULTS FOR ALL THE EXAMINED METHODS

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Probability of error (PE)</th>
<th>Discrepancy distance (DD)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Good</td>
<td>Low</td>
</tr>
<tr>
<td>Fixed circle [6]</td>
<td>0.049</td>
<td>0.049</td>
</tr>
<tr>
<td>Adaptive circle [7]</td>
<td>0.019</td>
<td>0.192</td>
</tr>
<tr>
<td>Seeded region growing [4]</td>
<td>0.099</td>
<td>0.114</td>
</tr>
<tr>
<td>Mann-Whitney [40]</td>
<td>0.165</td>
<td>0.162</td>
</tr>
<tr>
<td>Hybrid K-means [14]</td>
<td>0.017</td>
<td>0.020</td>
</tr>
<tr>
<td>Markov random model [21]</td>
<td>0.154</td>
<td>0.053</td>
</tr>
<tr>
<td>Matarray [41]</td>
<td>0.004</td>
<td>0.031</td>
</tr>
<tr>
<td>Model-based segmentation [17]</td>
<td>0.094</td>
<td>0.101</td>
</tr>
<tr>
<td>K-means [13]</td>
<td>0.000</td>
<td>0.025</td>
</tr>
<tr>
<td>Zacharia et al. [25]</td>
<td>0.000</td>
<td>0.012</td>
</tr>
<tr>
<td><strong>Proposed method</strong></td>
<td>0.000</td>
<td>0.006</td>
</tr>
</tbody>
</table>

C. Performance

Computational times were measured using a computer equipped with an Intel Core i3 CPU, 4 GB of DDR3 memory and a NVIDIA GeForce GTX 650 Ti graphics card with 1024 MB of GDDR5 memory.

For the gridding process, a CUDA implementation of the Zacharia et al. [33] gridding method was utilised, achieving an average of 50 ms per generation of the genetic algorithm for the synthetic images and 84 ms per generation for the replicated real images.

For the proposed method, the average computational time for the segmentation of the synthetic microarray images using the CUDA implementation was 281 ms per image and 405 ms per image using the CPU implementation. Taking into account the time needed for loading the input image, computing each quadrilateral’s coordinates from the grid structure and creating the output image, the average computational time reached 390 ms per image for the
CUDA implementation and 514 ms per image for the CPU implementation. For the real replicated microarray images used, the average computational time for the segmentation was 358 ms and 1981 ms per image for the CUDA and the CPU implementations, respectively. Taking into consideration all the aforementioned operations, the computational times reached an average of 508 ms and 2131 ms per image for the CUDA and CPU implementations respectively.

Another advantage of the proposed algorithm is that it is simpler than the Zacharia et al. [25] segmentation method which achieved the second best results. That method employs a genetic algorithm approach which introduces great computational complexity. A genetic algorithm is used consecutively on each spot in order to compute a 3D model representing the spot. Then, the segmentation of each spot is conducted by drawing the contours of their 3D spot model. Using the CPU implementation provided by the authors, the method achieved an average of 3.3 min per spot for the segmentation of the real replicated microarray images, compared to 1981 ms for the whole image using the proposed method’s CPU implementation and 358 ms for the CUDA implementation. It is evident that the proposed method clearly outperforms the Zacharia et al. [25] method in terms of execution times.

It must be noted that when the image size and the total number of spots increases, the CUDA implementation is expected to yield faster computational times due to its higher parallel computation capacity, as in the case of the examined replicated images. Since the number of iterations needed to compute the grow-cut algorithm for each spot is not constant, the CUDA implementation benefits from the large number of concurrent executed threads due to less waiting for thread termination, compared to the limited number of threads available to the CPU implementation. Moreover, the graphics card utilised for the experiments is a mid-range consumer grade graphics card, with each computational core slower than the cores of the utilised CPU. A faster GPU or a GPU dedicated to general computations (e.g. NVIDIA Tesla series) is expected to provide significantly faster performance. Furthermore, performance could also benefit from further optimization of the implementation, since the utilised approach is straightforward and simple.

IV. CONCLUSIONS

In this work, the authors propose a novel approach for cDNA microarray image segmentation. Experimental results showed that the proposed method outperforms ten state of the art algorithms. The automatic selection of the initial seeds for the grow-cut algorithm along the seed elimination criterion leads to enhanced segmentation efficiency and minimizes the occurrence of spurious spots. By avoiding user intervention for seed selection, the algorithm becomes
more robust and the results can be replicated for comparison and validation purposes. Moreover, the median filtering step enhances the ability of the proposed method to cope with noisy images and reduces the effects of noise on both the seed selection and the final segmentation stages. Additionally, the simplicity of the proposed algorithm allows for straightforward and efficient implementations. Finally, the parallel computation of the independent segmentation tasks using the CUDA architecture leads to significantly low computational times. The evaluation of the proposed methodology demonstrated its efficiency for segmenting cDNA microarray images, providing a fast and robust solution for the second stage of cDNA microarray analysis.

REFERENCES


FIGURE CAPTIONS

Fig. 1. Example of median filtering for a 152x121 pixels region of a real microarray image of size 1865x5500 pixels. (a) Original region, (b) filtered region.

Fig. 2. Example of median filtering for a 142x121 pixels region of a synthetic microarray image of size 330x750 pixels. (a) Original region, (b) filtered region.

Fig. 3. The points of intersection of the line-segments that form a quadrilateral \( p_1, p_2, p_3 \) and \( p_4 \) are used to determine the “Background” seeds \( p_{B1}, p_{B2}, p_{B3} \) and \( p_{B4} \). \( p_c \) refers to the “Spot” seed. Each rectangle in this image represents a pixel.

Fig. 4. Example of seed selection and segmentation result for a region of a synthetic microarray image. (a) Magnified region of the preprocessed original image. Black-filled rectangles correspond to “Background” seeds. White-filled rectangles correspond to “Spot” seeds. Each rectangle corresponds to a single pixel of the original image. Dashed lines refer to the line-segments of the grid. (b) Segmentation mask of (a). Images have been upscaled.

Fig. 5. Example of the effect of the seed elimination criterion for a region of a real microarray image. (a) Region of the original image. (b) Median filtered image. (c1) Seeds without using the seed elimination criterion. (c2) Seeds after applying the seed elimination criterion. Black pixels correspond to “Background” seeds. White pixels correspond to “Spot” seeds and grey pixels correspond to pixels with “Undefined” label. (d1) Segmentation result using the (c1) seeds. (d2) Segmentation result using the (c2) seeds. All images have been upscaled.

Fig. 6. Segmentation results for a block of a “good” quality synthetic microarray image. (a) Original image, (b) selected block, (c) segmentation mask of (b), (d) ground truth of (b).

Fig. 7. Segmentation results for a block of a “low” quality synthetic microarray image. (a) Original image, (b) selected block, (c) segmentation mask of (b), (d) ground truth of (b).

Fig. 8. Probability of error achieved by the proposed method for various PSNR values using median filters of size 3x3, 5x5, and 7x7.

Fig. 9. Discrepancy distance achieved by the proposed method for various PSNR values using median filters of size 3x3, 5x5, and 7x7.

Fig. 10. Segmentation results for a region of a real microarray image with and without seed elimination. (a) Region of the original image, (b) segmentation mask using the seed elimination criterion, (c) segmentation mask without using the seed elimination criterion.

Fig. 11. Segmentation results for a region of a real microarray image. (a) A magnified block of the original image. (b) Segmentation mask of (a).
Fig. 1. Example of median filtering for a 152x121 pixels region of a real microarray image of size 1865x5500 pixels. (a) Original region, (b) filtered region.
Fig. 2. Example of median filtering for a 142x121 pixels region of a synthetic microarray image of size 330x750 pixels. (a) Original region, (b) filtered region.
Fig. 3. The points of intersection of the line-segments that form a quadrilateral $p_1, p_2, p_3$ and $p_4$ are used to determine the “Background” seeds $p_{B1}, p_{B2}, p_{B3}$ and $p_{B4}$. $p_c$ refers to the “Spot” seed. Each rectangle in this image represents a pixel.
Fig. 4. Example of seed selection and segmentation result for a region of a synthetic microarray image. (a) Magnified region of the preprocessed original image. Black-filled rectangles correspond to “Background” seeds. White-filled rectangles correspond to “Spot” seeds. Each rectangle corresponds to a single pixel of the original image. Dashed lines refer to the line-segments of the grid. (b) Segmentation mask of (a). Images have been upscaled.
Fig. 5. Example of the effect of the seed elimination criterion for a region of a real microarray image. (a) Region of the original image. (b) Median filtered image. (c1) Seeds without using the seed elimination criterion. (c2) Seeds after applying the seed elimination criterion. Black pixels correspond to “Background” seeds. White pixels correspond to “Spot” seeds and grey pixels correspond to pixels with “Undefined” label. (d1) Segmentation result using the (c1) seeds. (d2) Segmentation result using the (c2) seeds. All images have been upscaled.
Fig. 6. Segmentation results for a block of a “good” quality synthetic microarray image. (a) Original image, (b) selected block, (c) segmentation mask of (b), (d) ground truth of (b).
Fig. 7. Segmentation results for a block of a “low” quality synthetic microarray image. (a) Original image, (b) selected block, (c) segmentation mask of (b), (d) ground truth of (b).
Fig. 8. Probability of error achieved by the proposed method for various PSNR values using median filters of size 3x3, 5x5, and 7x7.
Fig. 9. Discrepancy distance achieved by the proposed method for various PSNR values using median filters of size 3x3, 5x5, and 7x7.
Fig. 10. Segmentation results for a region of a real microarray image with and without seed elimination. (a) Region of the original image, (b) segmentation mask using the seed elimination criterion, (c) segmentation mask without using the seed elimination criterion.
Fig. 11. Segmentation results for a region of a real microarray image. (a) A magnified block of the original image. (b) Segmentation mask of (a).