

COSC 4393/6380
Digital Image Processing
Department of Computer Science
University of Houston
Assignment #1
Microarray Analysis
Due: 10/7/09

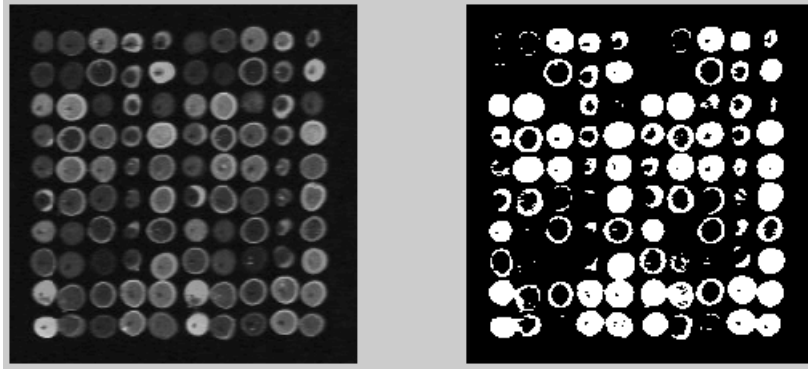
Background: Over the last several years there has been an explosion of microarray technology in the biosciences, medical sciences, biotechnology, and pharmaceutical industry. The technology has centered on providing a platform for determining the gene expression profiles of hundreds to tens of thousands of genes (or transcript levels of RNA species) in tissue, tumors, cells, or biological fluids in a single experiment. The rapid and global adoption of this technology has been predicated on its simplicity and success in providing large amounts of highly relevant data.

In their most generic form, microarrays are ordered sets of DNA molecules attached to a solid surface. The DNA molecules are typically either oligonucleotide (ranging from 35 base pairs to several hundred) or cDNAs. The matrix to which the DNA molecule is attached is usually glass, silicon, or nylon. The DNA is attached to the matrix in an ordered pattern such that hundreds to tens of thousands of DNA molecules may be spotted. Before the DNA is attached, the matrix is typically coated with a material such as poly-lysine, or silane, to make the matrix reactive. In most cases, the DNA is attached to the matrix using UV radiation or covalent coupling to permanently link the DNA to the surface. In this form a cDNA or oligonucleotide, corresponding to a specific sequence of a gene, can be spotted onto the solid surface. This can be repeated for hundreds to tens of thousands of other genes. With this ordered set of DNA segments attached to the surface, RNA from a specimen (e.g. tissue, cell line, tumor) can be either directly or indirectly labeled (usually with a fluorescent nucleotide) and hybridized to the array of genes. The amount of fluorescence at each DNA spot corresponds to the transcript level of that particular gene. Therefore, the expression of thousands of genes can be analyzed in a single specimen.

Image Processing Task: In the maturation process of microarray technology, there are several kinds of challenges. One of them is the extraction of the massive amount of information associated with this technology so that the results can yield insight into the genomic functions in biological systems. The key image processing challenge is to process microarray images to obtain the quantified gene expression values from the arrays. In this assignment, you will design algorithms and the processing pipeline to analyze small microarray images. In doing so, the following tasks are to be accomplished:

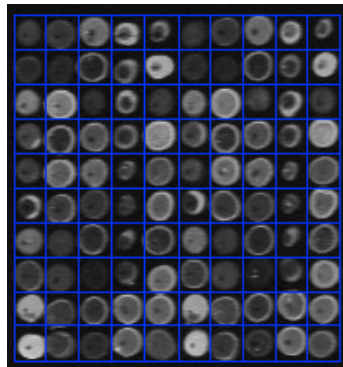
1. Write a program to binarize a gray-level image based on the assumption that the image has a bimodal histogram. You are to implement a method that assumes that both the foreground and background regions of the image represent a Gaussian distribution and the optimal threshold to binarize the image is the average of the

means of the two Gaussians. Your code should report both the binarized image and the optimal threshold value. Assume that foreground takes a value of 1 and the background a value of 0 in the binary image. Also assume that foreground objects are brighter than background objects in the input gray-level image.



Example input image and the corresponding binarized result.

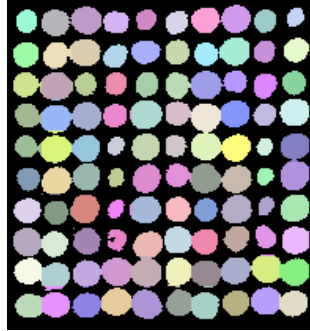
2. Design an algorithm and write program to identify the number of rows and columns of spots in the input microarray image. Assume that the microarray fabrication process is based on a regularized grid resulting in a matrix of spots. Nonetheless, images are contaminated with noise and hybridization artifacts that could distort the grid in local regions of the image. In addition, spots may be missing at specific grid locations. Under these assumptions, your algorithm should be able to report the total number of rows and columns of the spot grid and delineate a bounding box around each spot.



Example result of grid dimension detection and spot delineation.

3. Write to program to perform morphological smoothing. Include both Open-Close and Close-Open smoothing operations using a windowing function that is a 3 x 3 window and a 5 x 5 window. The input to your code would be a binary image and the output would be a smoothed version that eliminates small regions and fills holes of larger regions.

4. Write a program to perform blobcoloring based on both a 4-connected and a 8-connected neighborhood (3 x 3 window). The input to your code should be a binary image from the previous step and the output should be a count of total objects in the image as well as the labeled image where each object is color-coded starting with the value of 1 and the background taking a value of 0. Assume that objects in the binary image take a value of 1 while the background takes a value of 0. In addition, your code should also report the area and centroid of each object in the binary image. Also report the Average Optical Density of each spot as calculated from the original gray-level image.



Example output of labeled spots.

5. Write a program to extract the bounding contour of each spot. Please only use binary operations to perform this task.
6. Write a main program to read in the supplied gray-level image ('Array-1.tif') and threshold the image to generate a binary image using the function developed in part 1. Next, calculate the grid dimensions to estimate the total number of spots and delineate each potential spot location with a bounding box. Use morphological operations to eliminate small regions and fill holes for each spot. Use the blobcoloring function developed in part 4 to count the number of spots in the image and report on their statistics. Ignore spots smaller than 15 pixels in area and generate a report of the remaining spots (Spot Number, Area, Location, AOD). Delineate the contour of image spot using the function developed in part 5. Finally, display the original input image, the binary image, the gridded image, smoothed image, labeled image, and the final contour image. The final image should use a distinct color to represent each uniquely labeled spot.