Problem D1
Finding objects in images
Due: May 2, 2006

You should use **NIH Image** or **ImageJ** public domain image analysis packages, and/or **Matlab,** for this assignment. You will also need to download file(s) from the Homework Problems web page.

Questions (40 points)

1. Open file **giantin2.tif.** This is an image from a fluorescence microscope where light regions (high pixel values) indicate the presence of fluorescence and the background is dark (low pixel values). The image is of cells that have been stained for the Golgi protein giantin. Threshold the image so that only the bright major objects are visible. Now use the Analyze Particles option to identify individual objects in the image that have a minimum size of 2 pixels and a maximum size of 10,000 pixels (check the box for Label Objects). You should obtain 7 objects if your settings are correct. (a) What threshold value did you use? (b) Print the list of objects.

2. You are given files **D1first.tif** and **D1second.tif** and told that they contain fluorescence images (of the same field) for different proteins. (a) Print an inverted image of just those organelles that contain both proteins (i.e., have pixels that are above threshold in both images.) (b) Describe how you created it.

3. Use the Analyze Particles option to find objects in the **D1first.tif** image. Save the list of object measurements to disk. Repeat for **D1second.tif**. Use a spreadsheet program such as Excel to import the object measurements. Print a scatter plot of mean intensity vs. object area for both images on the same plot, using different colors or symbols for the objects from the different images. Do not print the lists of objects!

Extra Credit Questions (10 points each)

4. Use cross-validation to train and test a classifier to distinguish objects like those in **D1first.tif** from those in **D1second.tif.** Submit your code, your results (in the form of a confusion matrix), and a brief discussion of your approach and results.

5. Design a kernel that can be used to detect diagonal edges of slope -1 (i.e., running from upper left to lower right). Apply it to the **Angles.tif** image. Can you use it in combination with thresholding to make an image that shows (in black against a white background) only the edges of those objects in the image whose slope matches? Submit a printout of the kernel, a printout of the black on white image, and a description of the operations you used to create it.

Acknowledgments

Thanks to Michael Boland for providing images.