BRC_Imaging@cornell·edu

BME 6260 LAB #2: IMAGE ANALYSIS WITH IMAGEJ

Exercise Instructions:

1. Download FIJI:
http://fiji.sc/Fiji
(The FIJI distribution of ImageJ has more functionality, more scripting support and the capacity for automated updates, but it loads slower than ImageJ, which can be annoying.)

2. Download files from the BlackBoard site needed for this exercise:
FuzzyBlobs.Tif
FlashingCells.AVI
Ovary.AVI – Quirk lab data from the nano-CT.
BallMotion.AVI
splitStack_nd2_.ijm -- *A complicated macro for reference should you ever want to continue on your own.*

3. Write-up – as you work through the exercises below, enter (usually a cut and paste operation) the information /data requested from each section (A to E).

A. Using the ROI manager: Find the largest "object" in the image and encircle it, displaying the value of the perimenter.
1) Open FuzzyBlobs.Tif
2) Image -> Adjust -> Threshold
Be sure to select the objects and not the background!
3) Analyze->Set Measurements…
Make sure perimeter is selected.
4) Analyze -> Analyze Particles…
Make sure you Show: Outlines and have Display Results and Add to Manager checked. You may also want to only select particles that are 10um^2 and bigger to get rid of some of the regions due to noise.
5) In the results box, find the region-of-interest (ROI) with the biggest perimeter. In order to view that region, reopen FuzzyBlobs.Tif. In the ROI manager, click on that ROI to see this region in the original image.
6) To paste the image into your write-up document, overlay the analysis data (Image->Overlay->Flatten) and then either save the image or use Edit->Copy To System (this is the easiest way).
7) What is the perimeter of the largest object? Note that the perimeter values you obtain will have units dictated by the scaling in Image->Properties. This value may or may not read in correctly from whatever instrument produced the image and you often have to set it. (If there is no scaling and units information contained in the image header, it will be in pixels with x = y= z = 1.)

B. Analyzing a time series: Extract average intensity values from at least 5 cells and plot them over time.
1) Open FlashingCells.AVI (using Convert to Grayscale).
2) You can play through it using the scroll bar or the play symbol at the bottom of the image.
3) Analyze-> Set Measurements . . .
Select Mean Gray Value.
4) To analyze more than one cell at a time, use the ROI Manager. Analyze -> Tools -> ROI Manager . . .
5) Select a region using one of the standard drawing tools in the taskbar. (Note that if you want to use the magic wand tool, double click on it and set the tolerance to ~15. When you click in the image, it will automatically select an ROI for you, but this can be tricky.)
6) Each time you select a region, add it to the ROI manager by clicking the Add button.

For a better way to select all the cells, go to the first image, select all and copy and paste it to a new black and white image. Threshold this image as before, but now change it into a binary file by pressing the Apply button in the thresholding dialog box. This is now a binary mask. You can operate on it with the functions in the Process->Binary menu. Watershed this mask and Analyze -> Analyze Particles… to get all the ROI's into the manager. Once you select the image series they will be active on that.

7) Select all the ROI's.
8) To measure a property of those ROI's throughout the stack, choose More>>-> Multi Measure in the ROI Manager window. The output should be columns of data corresponding to the average intensity from each ROI.
9) Copy and paste data to a graphing program of your choice (or save it as an Excel file). X axis info - for this series, images were acquired every 10 sec. Paste the plot into your write-up.

C. Visualizing a Z-series: Display your favorite view and paste into your lab write-up.
1) Open Ovary.AVI
2) Run through it to check out what the data looks like.
3) Image->Stacks->3D Project… and use default values. You get a standard rotating projection.
4) Save your favorite view by saving it as a jpg or use edit->Copy To System.
5) Note also Analyze->3D Objects Counter and Plugins->3D Object Viewer. These can be a little funky.

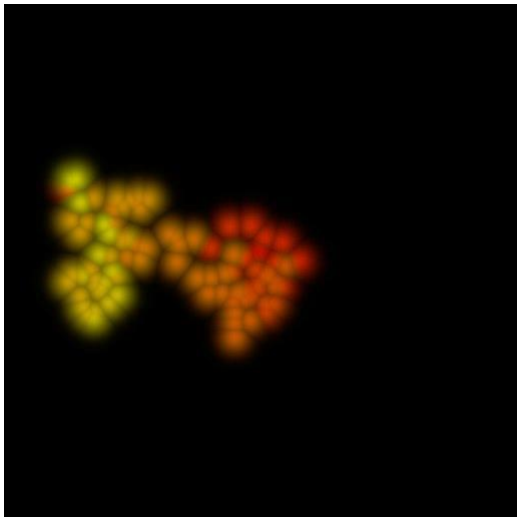
D. ImageJ scripting: Get X and Y coordinates out of the BallMotion.AVI and find the frame at which the ball is the farthest from its initial location.
1) Open BallMotion.AVI (using Convert to Grayscale).
2) Analyze->Set Measurements. . .
Select centroid.
3) Identifying the ball in each frame using menu items would be relatively tedious to do manually for each frame. So lets try scripting!
4) File -> New -> Script for FIJI or File -> New -> Text Window for ImageJ
5) Copy the following into the text box.

```
//Lines preceded by // are comments. They are ignored by the interpreter, so write whatever you want here!
//nSlices is a protected variable giving the number of images in a stack.
//setSlice is a function. Info on macro commands can be found by selecting Help->Macro Functions
//The wait statement makes the program pause for 10 msec for each incremented i value, enabling you to view
//what is going on. This can be removed at any time.

for (i = 1; i <= nSlices; i++){
setSlice(i);
wait(10);
}
```

6) Try running this. It should merely display all the images in the series. For FIJI, you will have to select the language first (Language->ImageJ Macro). For ImageJ, select Macro-> Run Macro.
7) To figure out what else needs to go between the curly braces, a great feature to use is Plugins->Macros->Record. There are many ways to identify the position of the ball. Try this one. Process-> Find Maxima . . . using Point Selection followed by Analyze-> Measure. Copy these lines into the script immediately after the setSlice command and run the program. Scripting gives you the flexibility to do whatever you want with images.
8) In which frame is the ball farthest away from the original location in frame 1? Iinsert your final imageJ script into your write-up.

E. ImageJ Scripting (harder): Make an image that is color-coded in time (green to yellow rather than the displayed red to yellow). Display that image and your imageJ script in your writeup.



1) Use the current image for the red stack. *Note this exercise will be much harder if you have not converted this stack to grayscale on opening!*
2) Make a new image for the green stack (width X height X nSlices).
3) Copy and paste slice-by-slice through the stack.
4) Multiply the green slices appropriately (Process->Math).
5) Project both stacks (Image->Stacks->Z Project…).
6) Merge colors (Image->Color->Merge Channels…).
7) You will probably need the following commands (or figure out a better way to do it!):

```
newImage("Green", "8-bit Black", 512, 512, nSlices);
selectWindow("BallMotion.Avi");
run("Copy");
selectWindow("Green");
run("Paste");
scale=i/nSlices;
run("Multiply...", "value="+scale+" slice");
```

General tips for ImageJ Scripting:

- **Menu items to Macro:** Plugins->Macros->Record…
- **Built-in macro functions:** Help->Macro Functions…
- **Scripts are case sensitive!** nSlices ≠ nslices
- **\'s need to be escaped.** infile="C:\\Dir1\\Dir2\\File.txt";
- **Still want more?** Check out this book that covers Image Processing with ImageJ and Java:
- Burger and Burge (2008) Digital Image Processing

Lab Write-up summary:

A. FuzzyBlobs: show the image with the objects encircled and note the perimeter value of the largest blob.
B. FlashingCells: Plot of intensity values vs time from at least 5 cells.
C. Ovary: Image of a single view from a 3D projection.
D. Ball Motion: What is the image frame at which the ball is farthest from its initial position? Show your ImageJ macro.
E. Ball Motion: Show the ball motion in 2D using color-coding to display time (green-to-yellow rather than red to yellow). Show your ImageJ macro.