Efficacy of Edge Detection on Fluorescent Cell Images

Jack Wearden
Jack Jacques

1 Aim

This investigation aims to the efficacy of 6 edge detection mechanisms in finding edges of cells in images obtained from a fluorescent microscope. The mechanisms being tested are Simple Gradient, Sobel Filter, Roberts Filter, First order Gaussian, Laplacian, and Laplacian of Gaussian.

The efficacy of each mechanism will be evaluated by comparing images produced from fluorescing cells to a manually edge-detected set of images of the same source.

2 Method

From the three provided cell images, grayscale copies of the fluorescing cells were created, in order to allow simpler manipulation based on brightness at each point of the image, without having to consider colour. These were produced using the `rgb2gray` matlab function. The grayscale images were then convolved by the filters being tested using the `conv2` function.

This function was applied with the `valid` convolution behaviour - this means the outermost pixels in the image are disregarded, as opposed to having the image padded with 0.

As most of these filters are applied independently in the horizontal and vertical planes, it was necessary to combine these convolved results. To do this, a function to compute the combined magnitude was needed. The function used was as follows:

```
function m = magnitude(x,y)
m = sqrt(x.*x + y.*y);
```

This computes a pythagorian value for the corresponding elements of the two matrices to create the vector for an ideal edge. This step was required for every filter with the exception of Laplacian (and Laplacian of Gaussian) as Laplacian works on both the horizontal and vertical planes.

In order to account for the earlier cropping of the cell image, it was necessary to also crop the manually edge detected images before any comparison could occur. Variants were created which cropped 1, 2 or 3 elements from each edge of the image according to the size of the filter or filters used. This is trivial to achieve in matlab - `matrix(2:end-1, 2:end-1)`, for example, crops one line of the outermost pixels from each side of the image.

With the convolved images and correctly adjusted, the next step was to calculate the specificity and sensitivity of each sensor. To do this, a function was implemented for each, which takes a thresholded, edge detected image and the manually edge detected image and calculates the ratio.

To fairly compare the sensitivity & specificity ratios for each detector, however, it was necessary to find the optimal threshold for each sensor. To achieve this, a batch function was created, as shown below. This function attempts a sequence of thresholds and shows the sensitivities and specificities of the detector on each image.

```
function b = batch( start, step, stop , m1, m2, m3, p1, p2, p3)
  b = [];
  for t=start:step:stop
      sens = [
          sensitivity(m1 > t, p1)
          sensitivity(m2 > t, p2)
          sensitivity(m3 > t, p3)
          ];
      specs = [
          specificity(m1 > t, p1)
          specificity(m2 > t, p2)
          specificity(m3 > t, p3)
      ];
      b = [b; sens specs ];
  end
end
```

Using these results, it is then possible to find the optimal threshold for each value, and from that use the mean specificities and sensitivities for each detector.

3 Results

Sensor	Optimal Threshold	Image	Sensitivity	Specificity
Simple Gradient	10	1	0.9047	0.8048
		2	0.9563	0.7674
		3	0.5824	0.9091
		Average	0.8145	0.8271
Roberts Filter	7	1	0.9122	0.8001
		2	0.9600	0.7571
		3	0.6070	0.9015
		Average	0.8264	0.8196
Sobel Filter	25	1	0.8504	0.7919
		2	0.8983	0.7259
		3	0.6541	0.8633
		Average	0.8009	0.7937
1st order Gaussian	381	1	0.8406	0.8526
		2	0.9484	0.8421
		3	0.3616	0.9730
		Average	0.7169	0.8892
Laplacian	-1	1	0.4467	0.482
		2	0.5186	0.4766
		3	0.5341	0.4414
		Average	0.4998	0.4667
Laplacian of Gaussian	18	1	0.7323	0.7409
		2	0.8086	0.7192
		3	0.6578	0.7538
		Average	0.7329	0.7380

Full data available from http://notbobthebuilder.github.io/cell-edge-detection

4 Conclusions

The results of the experiment show that the first three filters - Simple, Roberts and Sobel, all perform roughly as well as each other. Gaussian Filter has a significantly higher specificity than the rest, but a much lower sensitivity. This looks as though it's due to an anomaly in the 3rd image - further experimentation would be required to see what feature of this image which causes the Gaussian filter to behave suboptimally.

Laplacian is by far the worst of the filters, actually coming worse below the line of no discrimination in ROC space. This is likely due to its sensitivity to noise, and the increased performance of the Gaussian-Laplacian filter would imply it can perform better on a noise-reduced image.

Future investigations could determine how filters perform when presented with already-processed images from other filters in the sequence, and if there is an optimal ordering - for example, a sensitive filter working on the output of a filter with a small grid size.