CSR Research Journal

Brian Fehrman

October 15, 2012

Part I
Computer Vision

1   Canny Edge Detector

[1]

Considered the ultimate edge detector. Clean, thing edges that are well connected to nearby edges.

Multistage edge detector:

1. Preprocessing

2. Calculating Gradients

3. Nonmaximum suppression

4. Thresholding with hysterysis.

Two key parameters given are upper and lower thresholds; upper is used to mark definite edges, lower is used for faint pixels that are part of an edge.

1.1   Preprocessing

Edge detectors prone to noise, gaussian blur filter helps. Typically need to perform this step yourself even with software packages. Typically, 5x5 Gaussian filter with standard deviation of 1.4 is used.

1.2   Calculating Gradients

Gradient is calculated at every point in image. Magnitude of gradient at point determines if it lies on an edge or not. Higher the gradient the greater the likelihood of an edge. Direction of gradient shows how image is oriented. The standard *Sobel Edge Detector* is used. Magnitude of the gradient is:

$$m = \sqrt{G_x^2 + G_y^2} \tag{1}$$

And the direction is:

$$\theta = arctan(\frac{G_y}{G_x}) \tag{2}$$

Where G_x and G_y are the derivatives for the given point. The gradients are calculated by using a convolution kernels which approximate the derivative in the X and Y directions.

1.3 Non maximum suppression

Check edges in four directions: Vertical, Horizontal, and the two diagonals. For each case, the middle pixel is checked against neighboring pixels to see if it is a maximum and is suppressed if it is not.
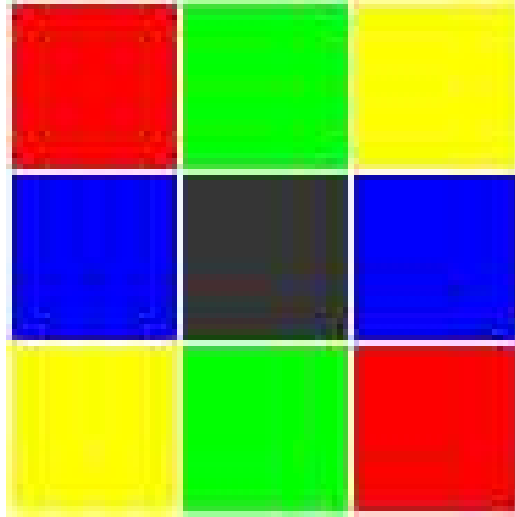

Figure 1: Possible directions of an edge

1.3.1 Gradient from 22.5 to 67.5

This case means that the edge goes from the top right to the bottom left. To check if an edge exists for this case, first need to check if gradient is maximum at this point. To do this, compare gradient magnitude of middle pixel to that of the top left and lower right pixels. If it's a maximum and its magnitude is greater than threshold than its an edge.
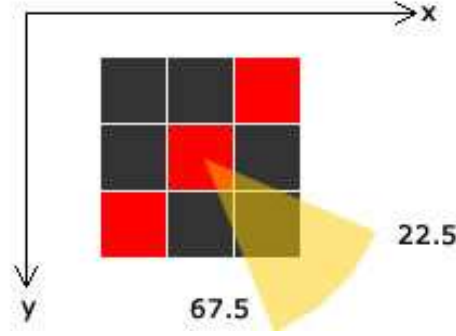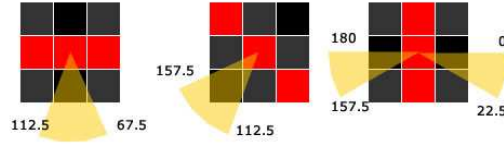
Figure 2: Diagonal south to north case

Figure 3: Remaining 3 edge cases

2 Sobel Edge Detector

[2]

Figure 4 shows an example of the intensity change of an image. There is a steep slope when the image changes from dark to light which means there is most likely an edge present. If the first derivative at each point is taken then the algorithm can search for a peak above some threshold and mark that point as an edge. A second derivative could be taken to find a zero crossing to signify the location of an edge and this technique is called the *Laplacian edge detector* (versus the *Sobel edge detector* here) and is shown in

Since images are typically 2-D, a gradient is used to calculate the derivatives. Convolution kernels are used to approximate the gradients in the X and Y directions. These kernels are shown in Figure

Can give the magnitude:

$$\sqrt{G_x^2 + G_y^2} \tag{3}$$
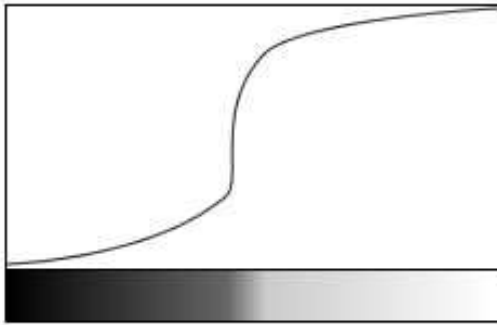
Approx strength:

4

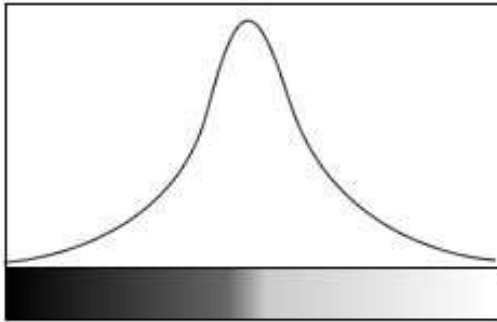Figure 4: Curve representing intensity

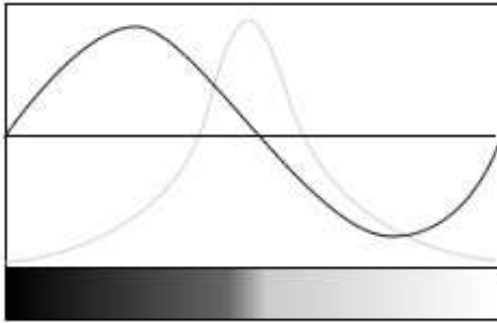Figure 5: First derivative of the curve in Figure 4

Figure 6: Second derivative of the curve in Figure 4

$$|G_x| + |G_y| \tag{4}$$

Figure 7: Convolution kernels

And the edge orientation:

$$arctan(\frac{G_y}{G_x}) \tag{5}$$

3   Scale Invariance

[3]

References

[1] AI Shack. The canny edge detector. http://www.aishack.in/2011/06/the-canny-edge-detector/#more-1999, Sept 2012. Online; accessed Sept-29-2012.

[2] AI Shack: Utkarsh. The canny edge detector. http://www.aishack.in/2011/04/the-sobel-and-laplacian-edge-detectors/, April 2011. Online; accessed Sept-29-2012.

[3] Tony Lindeberg. Principleso for automatic scale selection. *Handbook on Computer Vision and Applications*, 1998.