Web Scraping & Text Mining

Paulo Serôdio

Postdoctoral Researcher
School of Economics
Universitat de Barcelona

May 15, 2018

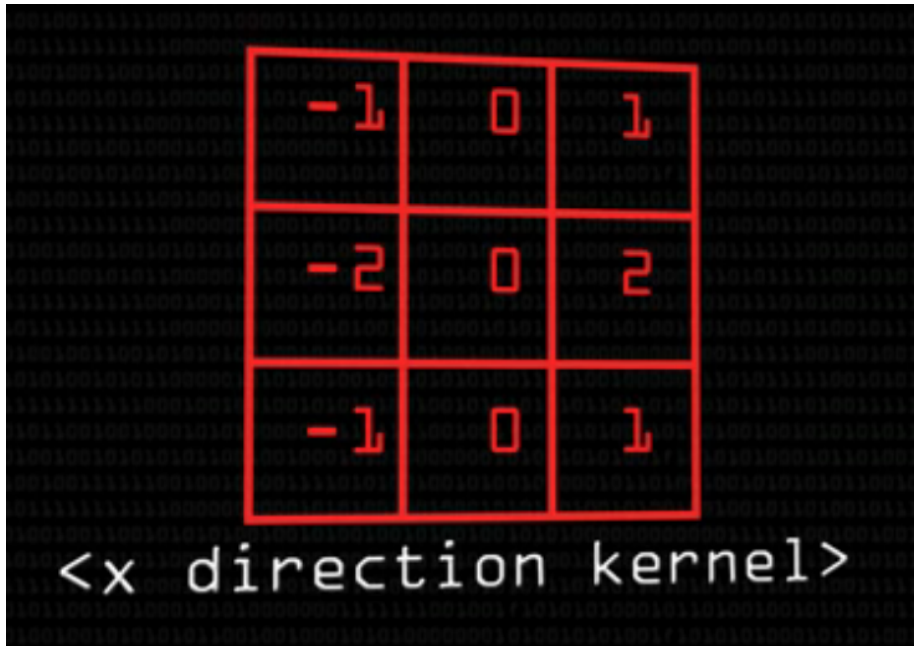
Course Materials & Structure

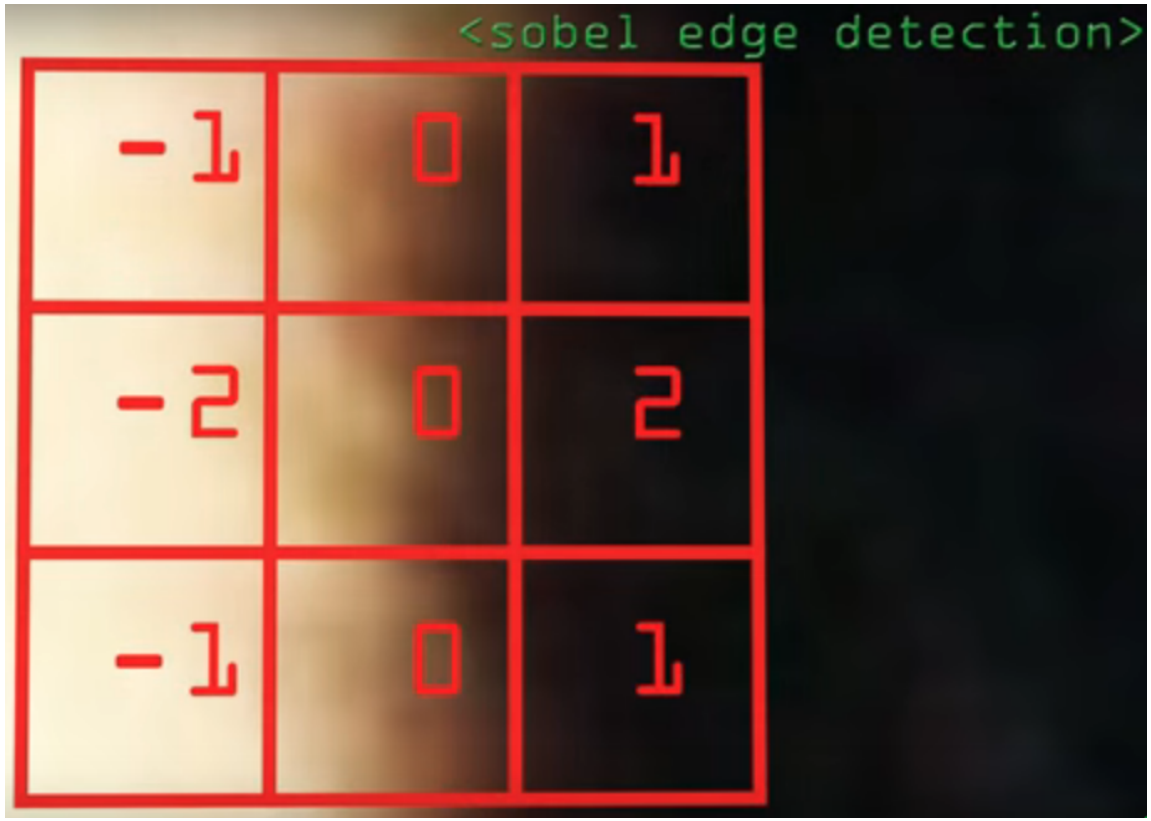
http://www.pauloserodio.com/eui2018

WEB SCRAPING II

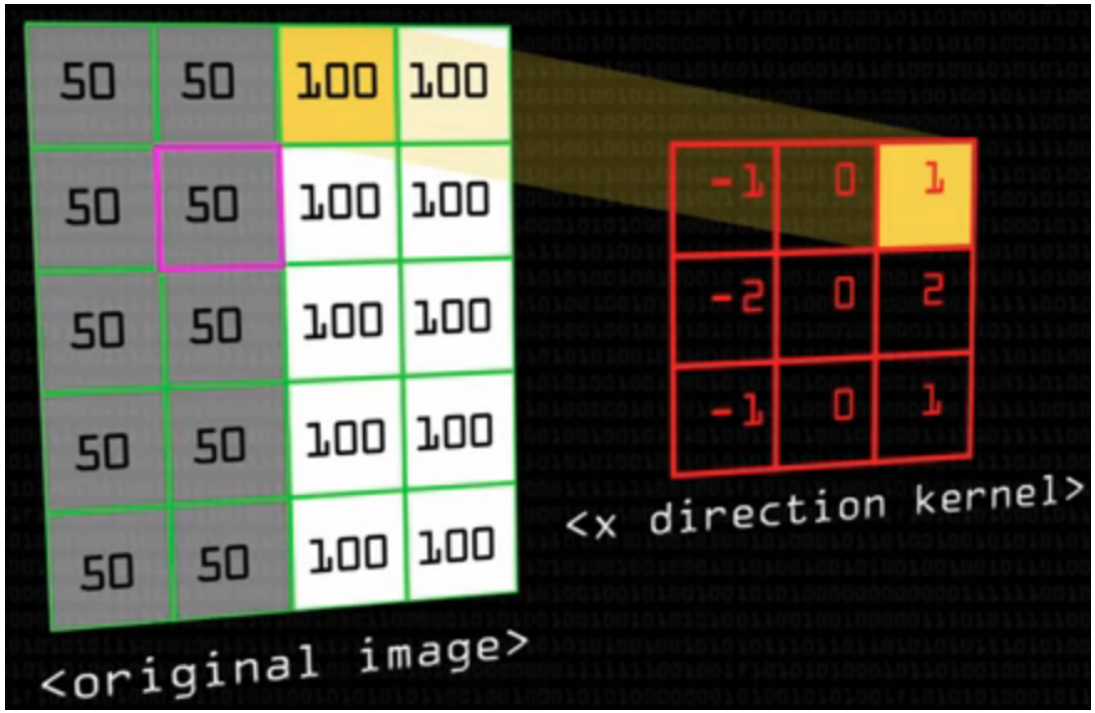
Finding tabular data in images

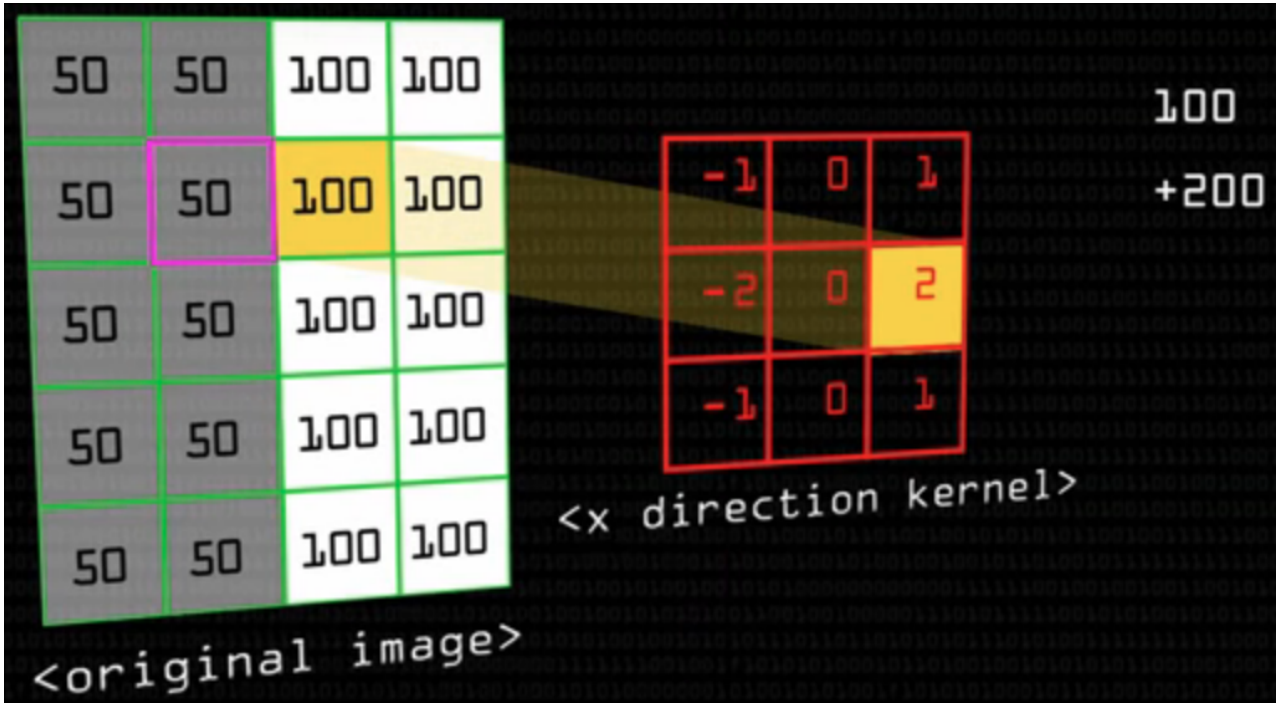
1. Color to gray scale first;
2. Gaussian filter to get rid of high intensity pixels in the image; allows to create an image where we only see the big discrepancyes/discontinuities in the image.
3. Sobel edge detector (derivative of an image; find X Y gradients, and yield total gradient;
4. Canny Edge detector
   - 4.1 Thinning Edges
   - 4.2 Hystereses thresholding (removing weak edges)
5. Hough Transform (voting procedure to identify lines)

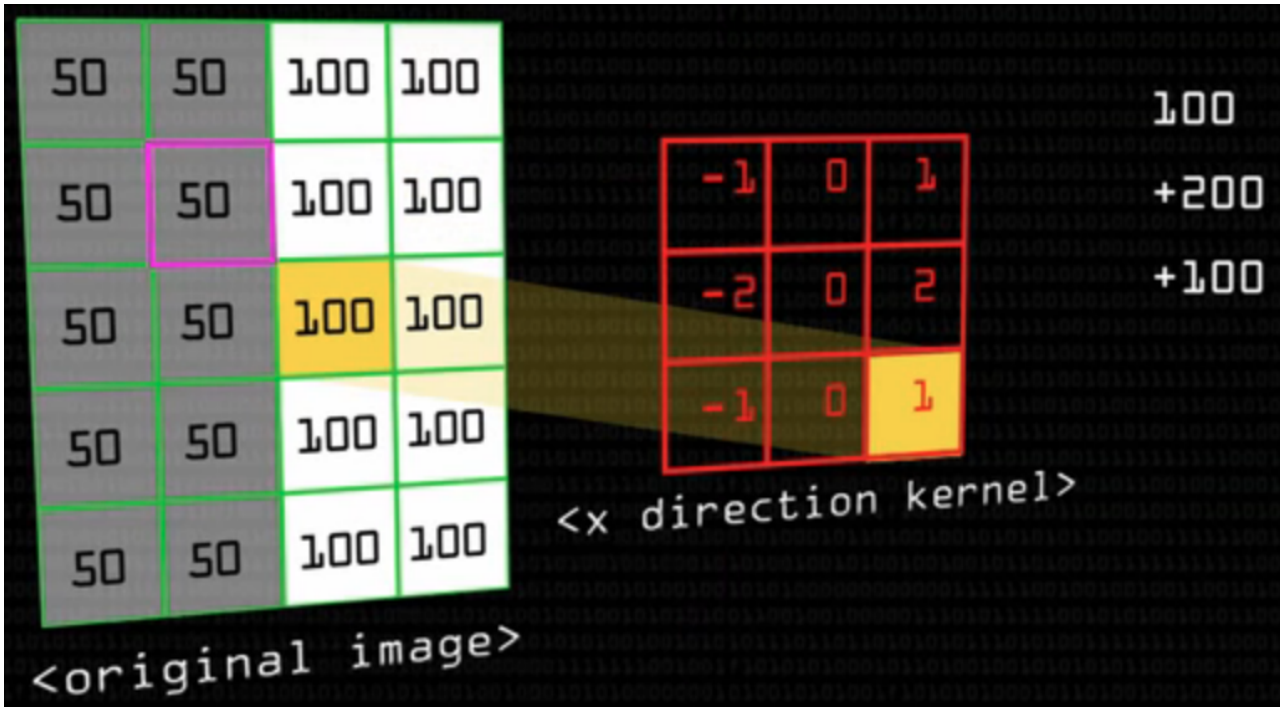

-1 0 1
-2 0 2
-1 0 1
<x direction kernel>

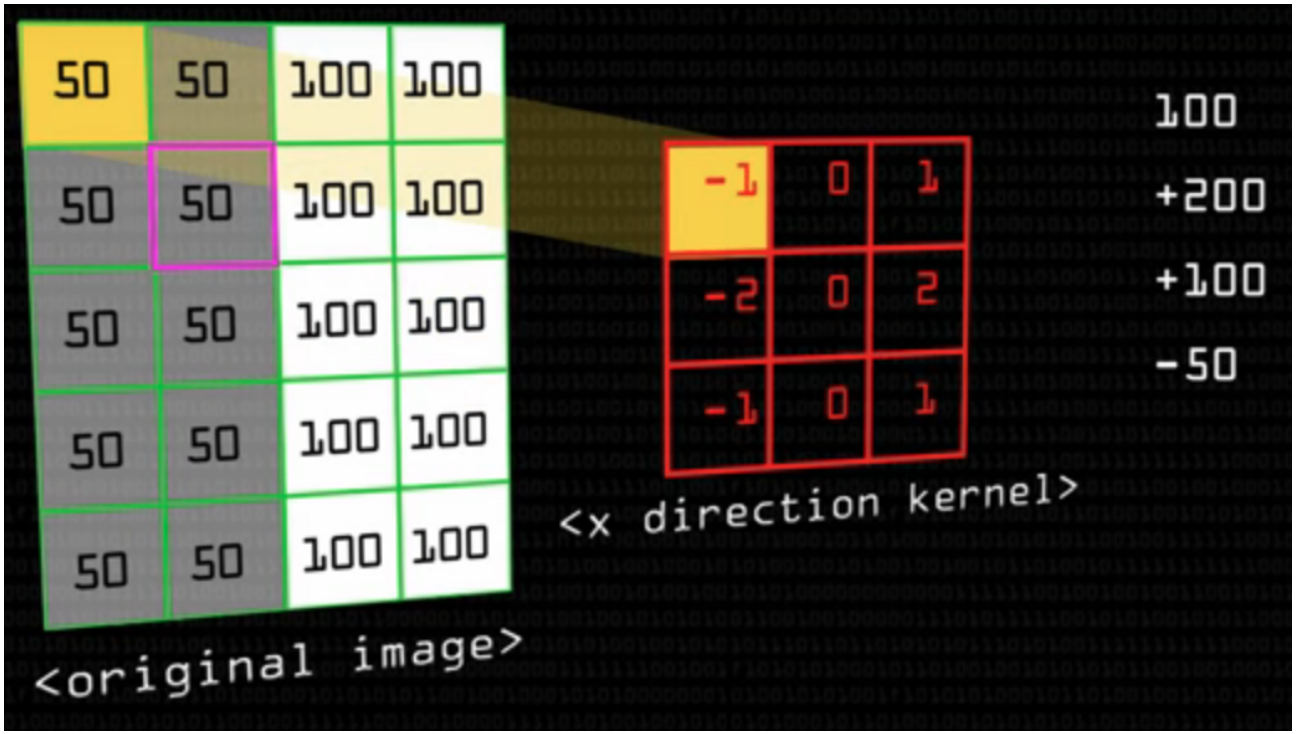

<sobel edge detection>
-1 0 1
-2 0 2
-1 0 1

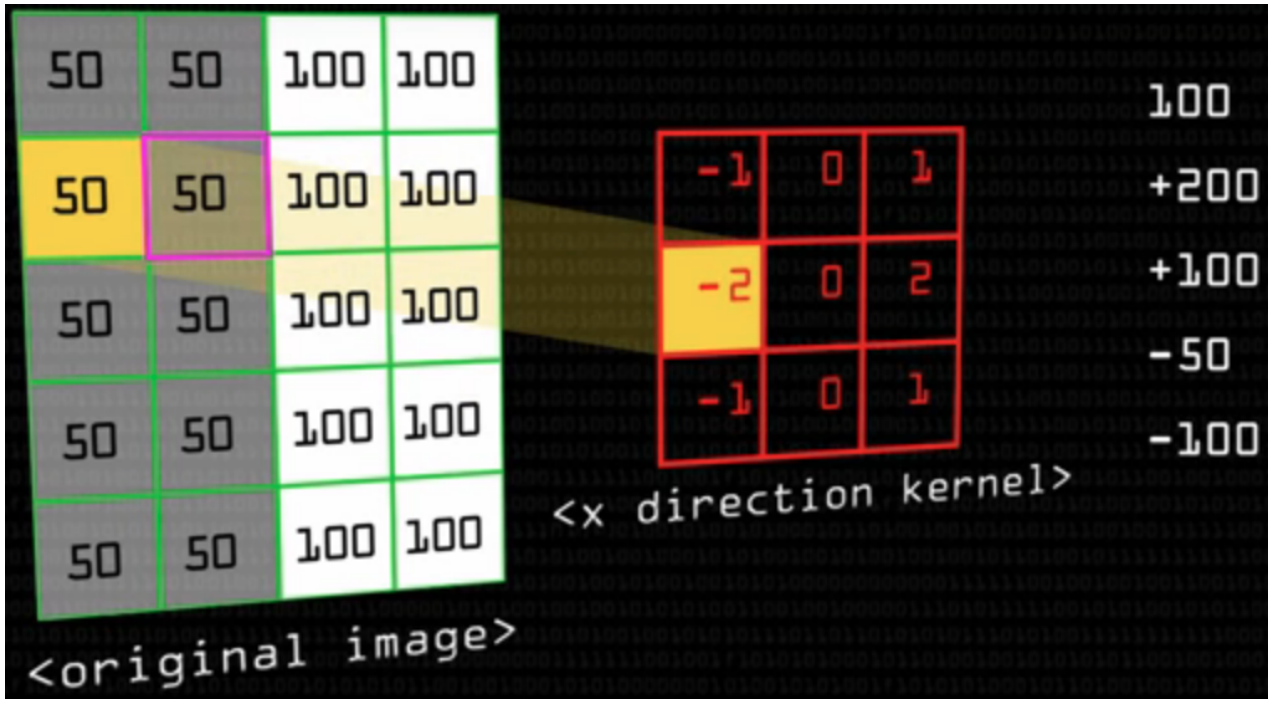

<sobel edge detection>
50 50 100 100
50 50 100 100
50 50 100 100
50 50 100 100
50 50 100 100
<original image>

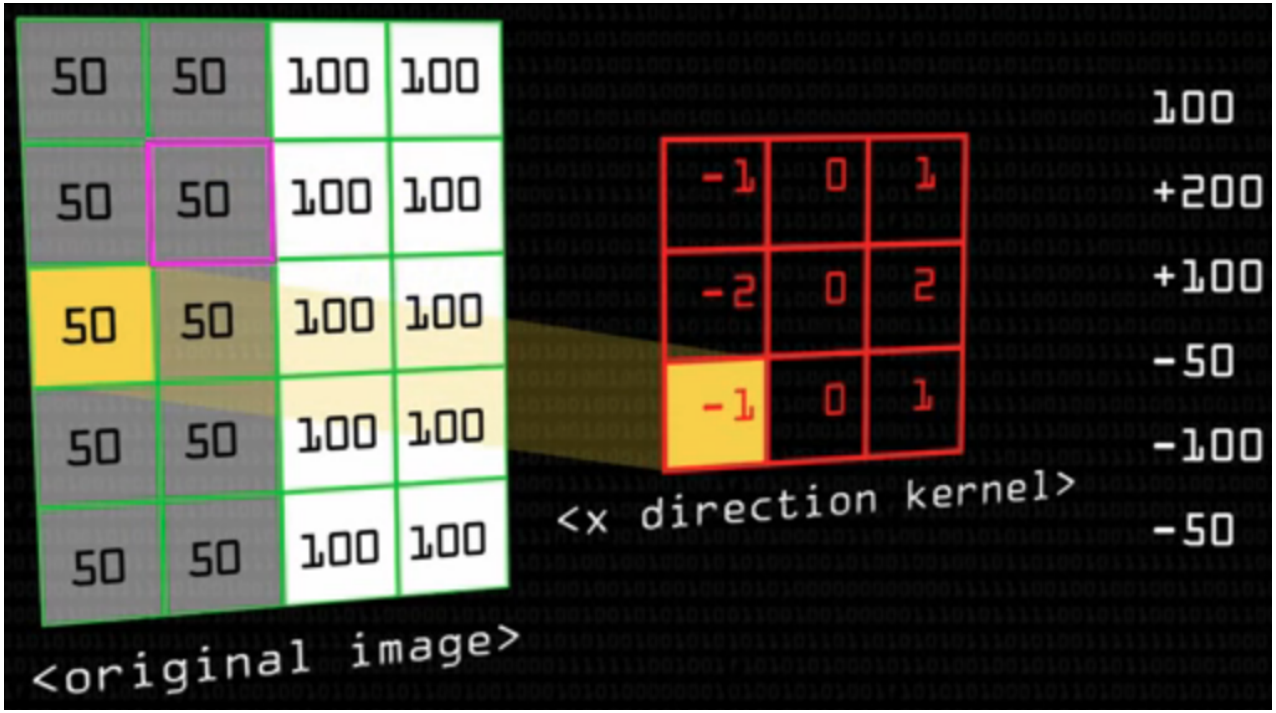

50 50 100 100
50 50 100 100
50 50 100 100
50 50 100 100
50 50 100 100
<original image>
-1 0 1
-2 0 2
-1 0 1
<x direction kernel>

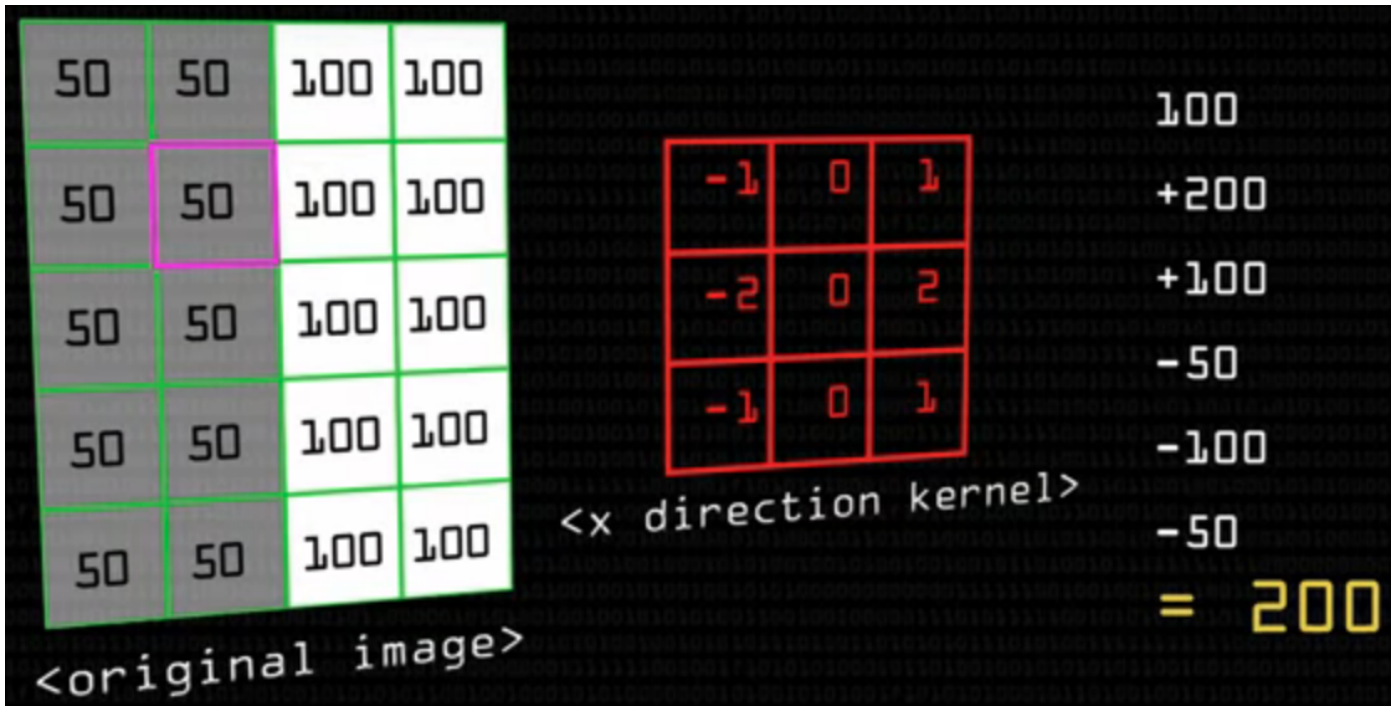

50 50 100 100
50 50 100 100
50 50 100 100
50 50 100 100
50 50 100 100
<original image>
-1 0 1
-2 0 2
-1 0 1
<x direction kernel>
100
+200


50 50 100 100
50 50 100 100
50 50 100 100
50 50 100 100
50 50 100 100
<original image>
-1 0 1
-2 0 2
-1 0 1
<x direction kernel>
100
+200
+100


50 50 100 100
50 50 100 100
50 50 100 100
50 50 100 100
50 50 100 100
<original image>
-1 0 1
-2 0 2
-1 0 1
<x direction kernel>
100
+200
+100
-50

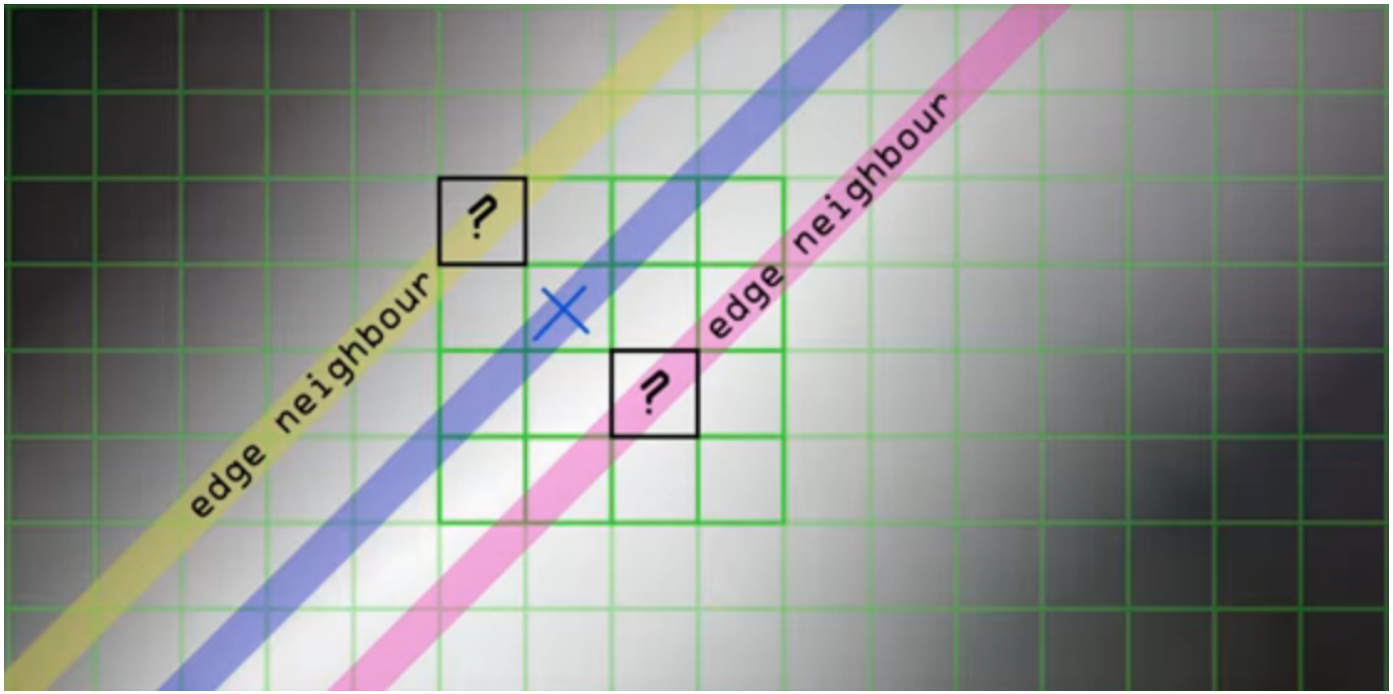

50 50 100 100
50 50 100 100
50 50 100 100
50 50 100 100
50 50 100 100
<original image>
-1 0 1
-2 0 2
-1 0 1
<x direction kernel>
100
+200
+100
-50
-100

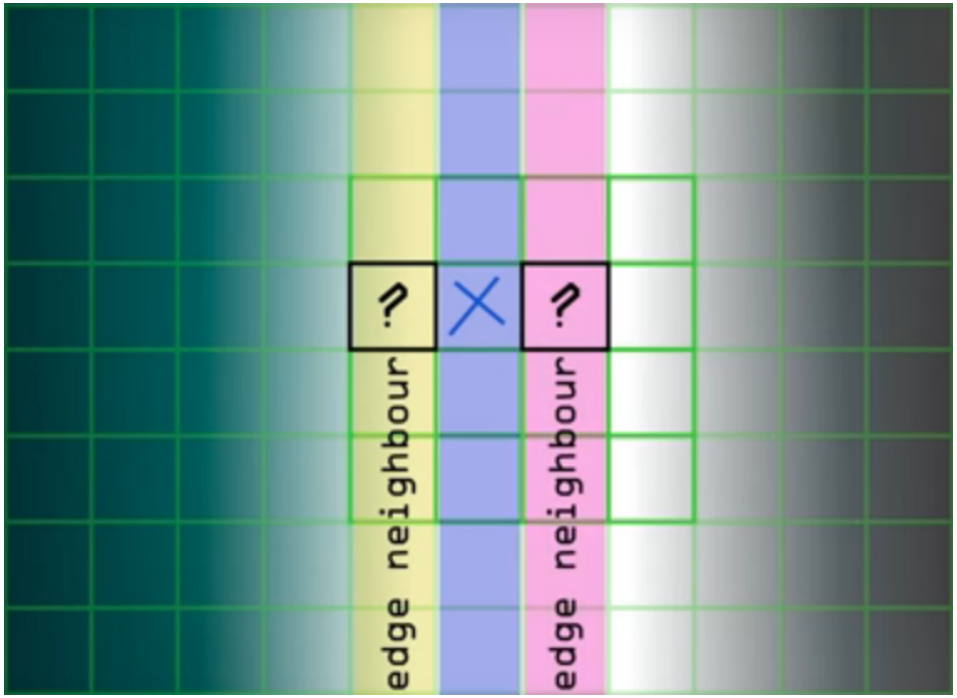

50 50 100 100
50 50 100 100
50 50 100 100
50 50 100 100
50 50 100 100
<original image>
-1 0 1
-2 0 2
-1 0 1
<x direction kernel>
100
+200
+100
-50
-100
-50

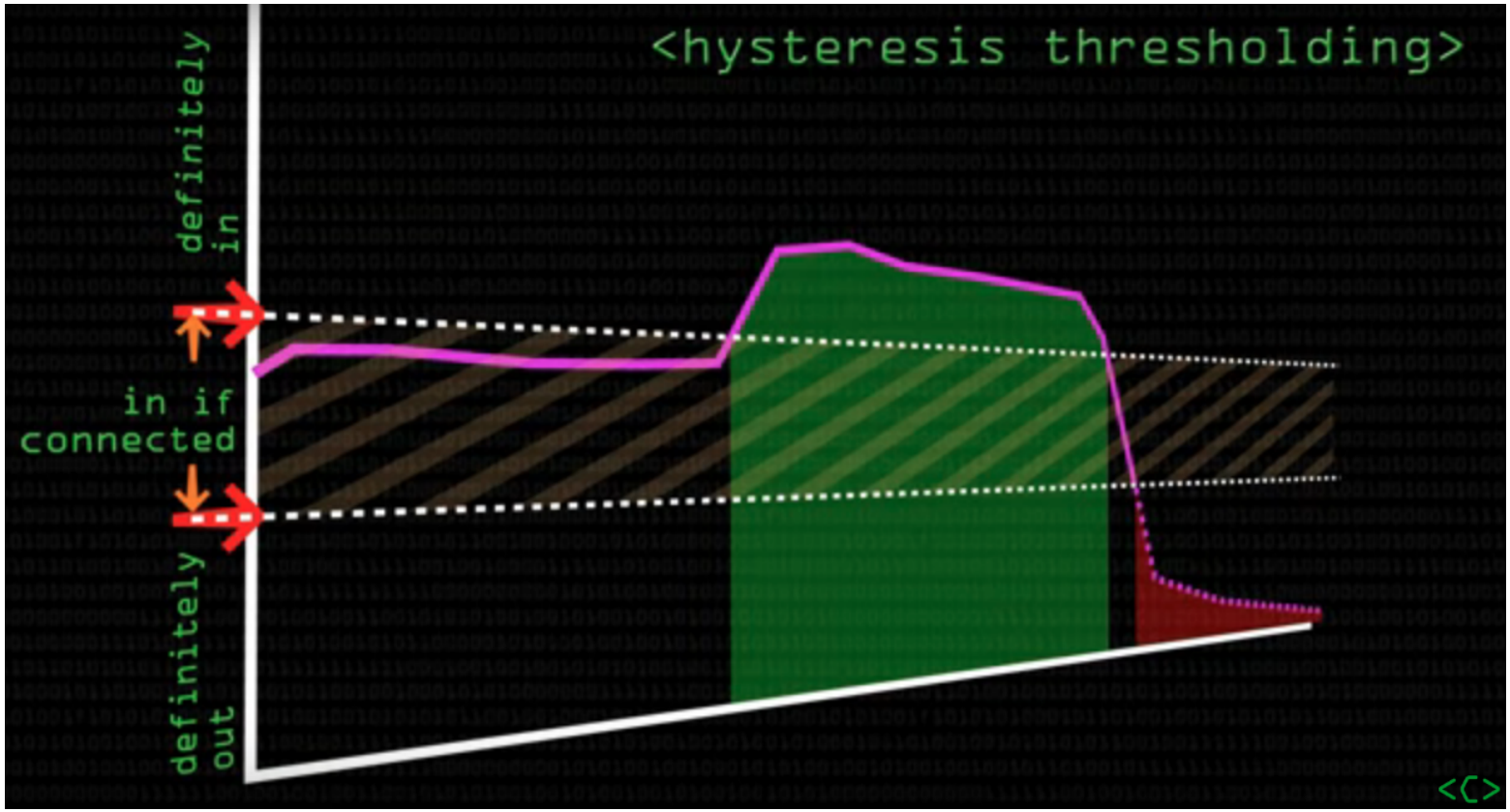

50 50 100 100
50 50 100 100
50 50 100 100
50 50 100 100
50 50 100 100
<original image>
-1 0 1
-2 0 2
-1 0 1
<x direction kernel>
100
+200
+100
-50
-100
-50
= 200


<original image>


<product of sobel edge detector>

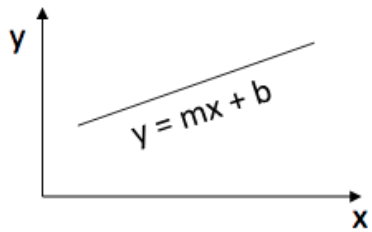

?
?
edge neighbour
edge neighbour


?
?
edge neighbour
edge neighbour


<hysteresis thresholding>
definitely in
in if connected
definitely out
<C>

Hough Transform (continued)

- A line has two parameters (m,b)



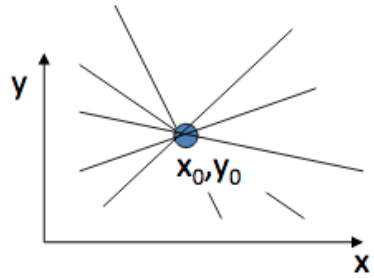
- Given a point (x_0,y_0), the lines that could pass through this point are all (m,b) satisfying

$$y_0 = m\, x_0 + b$$

- Or

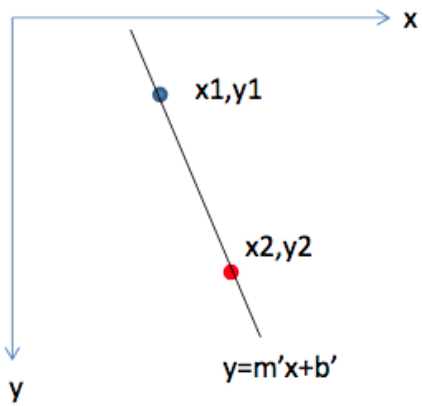
$$b = -x0\ m + y_0$$

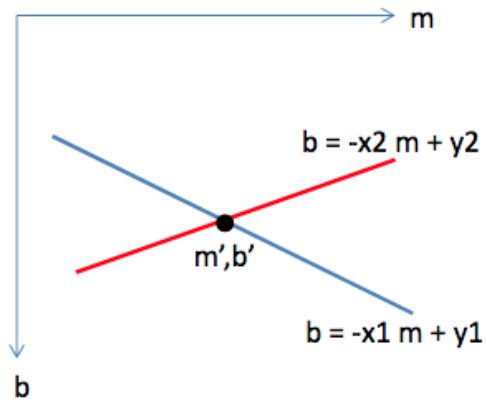
The equation b = -x0 m + y0 is a line in (m,b) space



Hough Transform (continued)

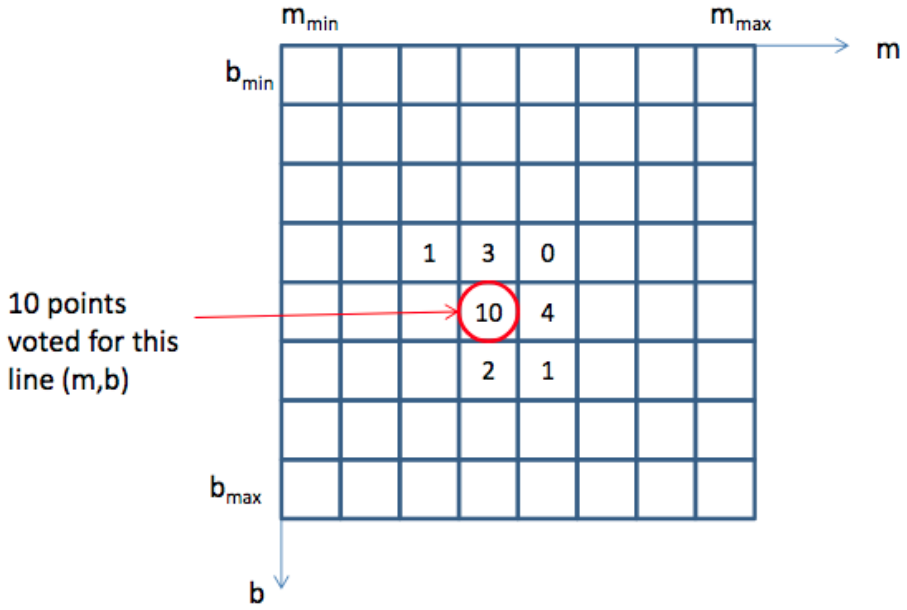
- All points on a line in image space, yield lines in parameter space which intersect at a common point
  - This point is the (m,b) of the line in image space





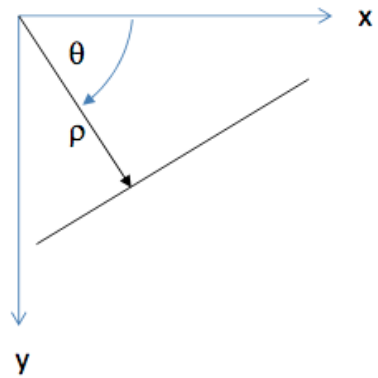
Hough Transform Algorithm

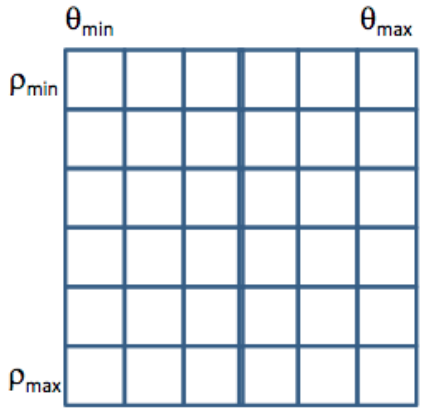
- Initialize an accumulator array A(m,b) to zero
- For each edge element (x,y), increment all cells that satisfy b = -x m + y
- Local maxima in A(m,b) correspond to lines



Polar Coordinate Representation of Line

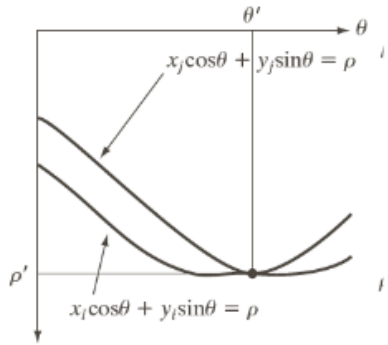
- $\rho = x \cos\theta + y \sin\theta$
  - Avoids infinite slope
  - Constant resolution





A(ρ,θ)

The parameter space transform of a point is a sinusoidal curve



- Recommended method to collect tweets
- Firehose: real-time feed of all public tweets (400M tweets/day = 1 TB/day), but expensive.
- Spritzer: random 1% of all public tweets (4.5K tweets/minute = 8 GB/day), implemented in streamR as `sampleStream`
- Filter: public tweets filtered by keywords, geographic regions, or users, implemented as `filterStream`.
- Issues:
  - Filter streams have same rate limit as spritzer (1% of all tweets)
  - Stream connections tend to die spontaneously. Restart regularly.
  - Lots of invalid content in stream. If it can't be parsed, drop it.