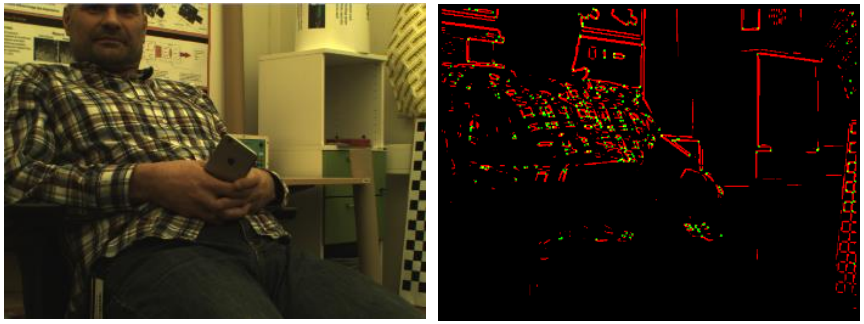
Edges and Lines

- What is
  - an edge?
  - a line?



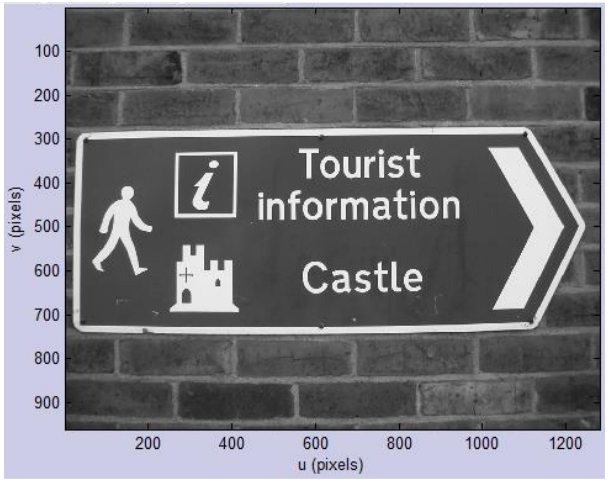
Harris

Canny

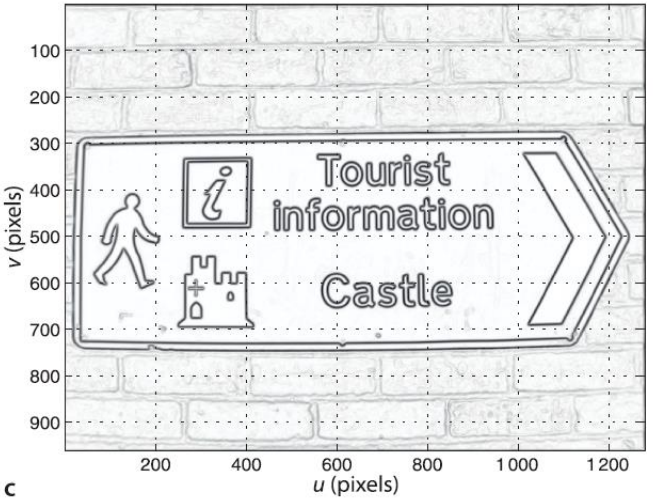
$$y = ax + b$$

Parametric form

Convolution – Edges



Gradient magnitude



Derivata:
SOBEL kernel

$$D = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Transpose for vertical gradient

$$I_u = \frac{\partial I}{\partial u} = \nabla_u I = D \otimes I$$

$$I_v = \frac{\partial I}{\partial v} = \nabla_v I = D^T \otimes I$$

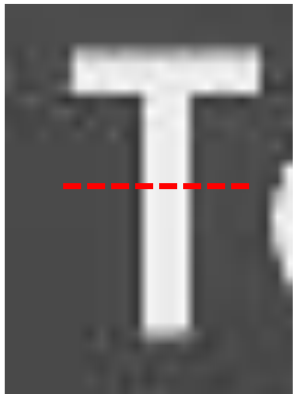
$$Magnitude = \sqrt{I_v^{\,2} + I_u^{\,2}}$$

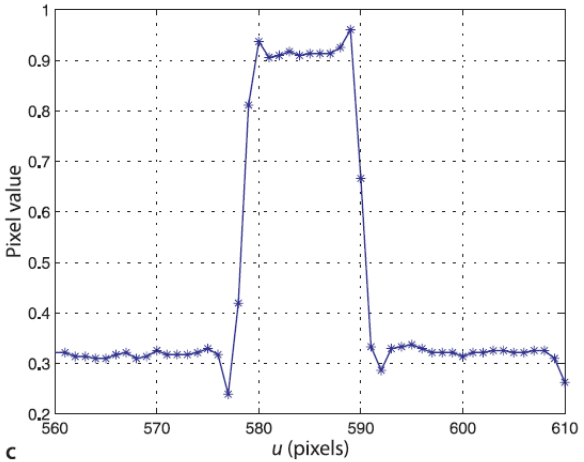
$$Direction = tg^{-1}\left(I_u / I_v\right)$$

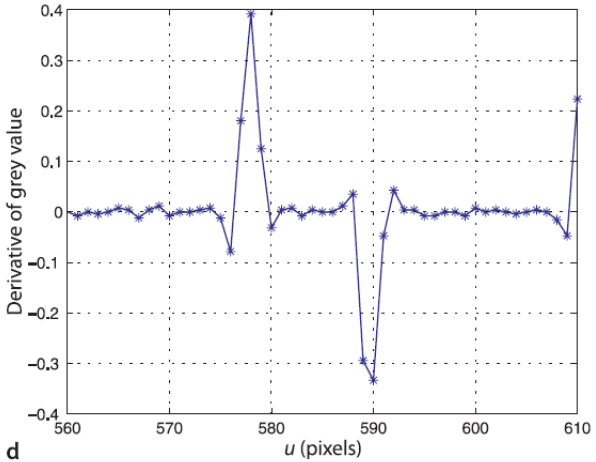
Comment: Edge direction is perpendicular to gradient direction

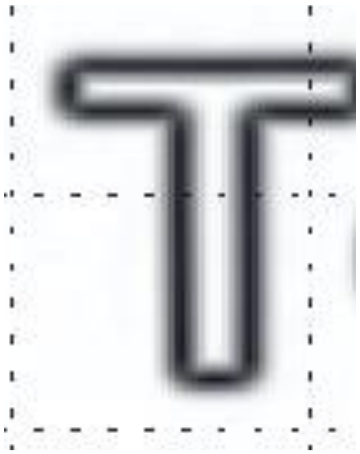
Edge detection – Sobel + G

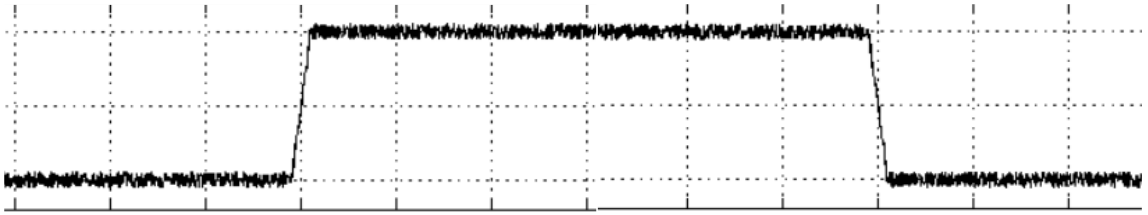
abs(DoG)



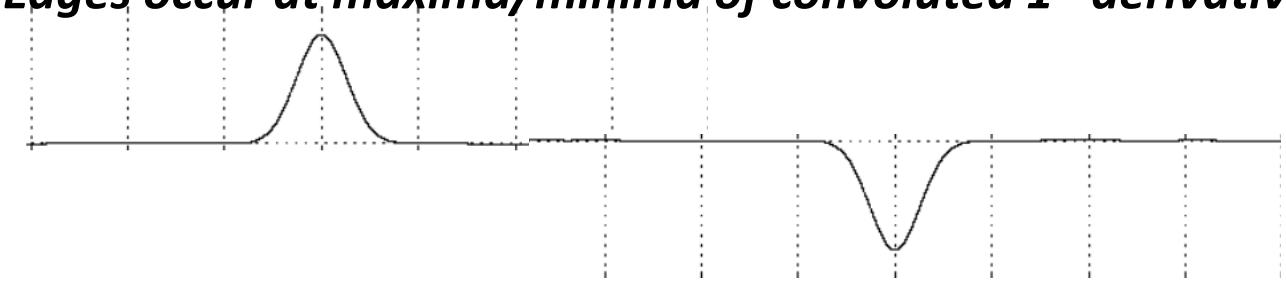








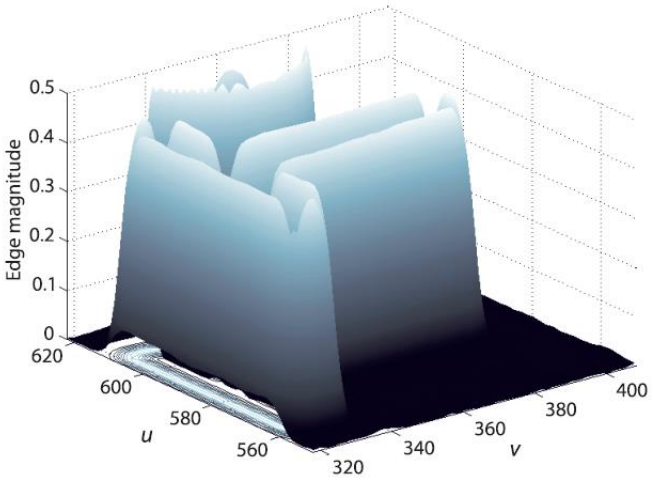
Edges occur at maxima/minima of convoluted 1st derivative



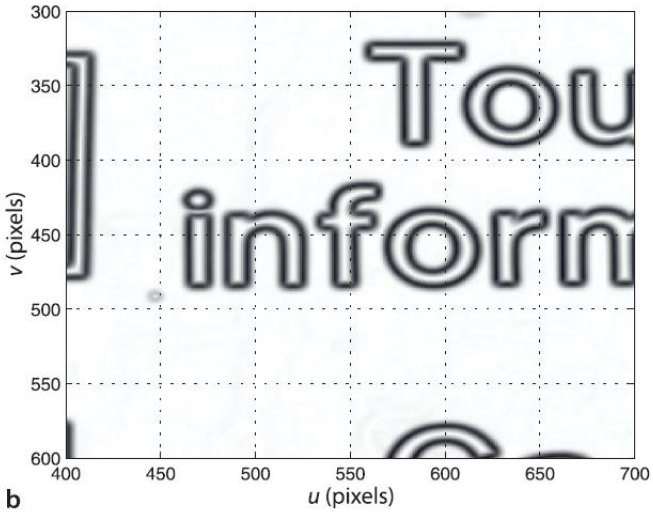
Edge identification

- So exactly where is the edge?
- Aggregating 1D edges add uncertainty in the form of thickness to the edge.
- Suppressing all but the maxima generates a true edge identificator

Edge detection: Canny



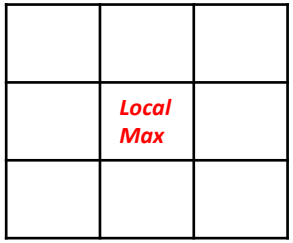
DoG



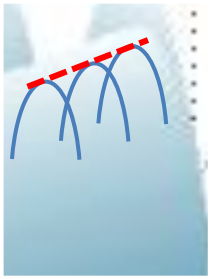
Non-maximum suppression

$$Direction = tg^{-1}\left(I_u / I_v\right)$$

$$Magnitude = \sqrt{I_v^{\,2} + I_u^{\,2}}$$



Consider the gradient <u>orthogonal</u> to the edge: <u>kill all non max</u>

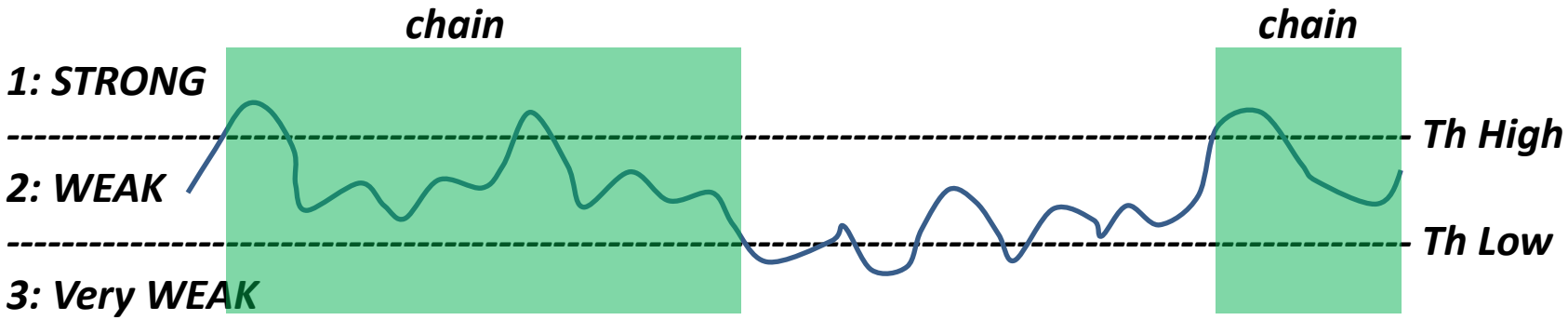


Edge detection: Canny

Hysteresis thresholding

$$Magnitude = \sqrt{I_v^{\,2} + I_u^{\,2}}$$

A week max is still part of the edge if connected to a chain



Edge detection: Canny

Sobel+G

Non-maximum suppression

Hysteresis thresholding (chains)

Sobel vs Canny

Sobel+G

Canny

Hough transform

- What is
  - an edge?
  - a line?

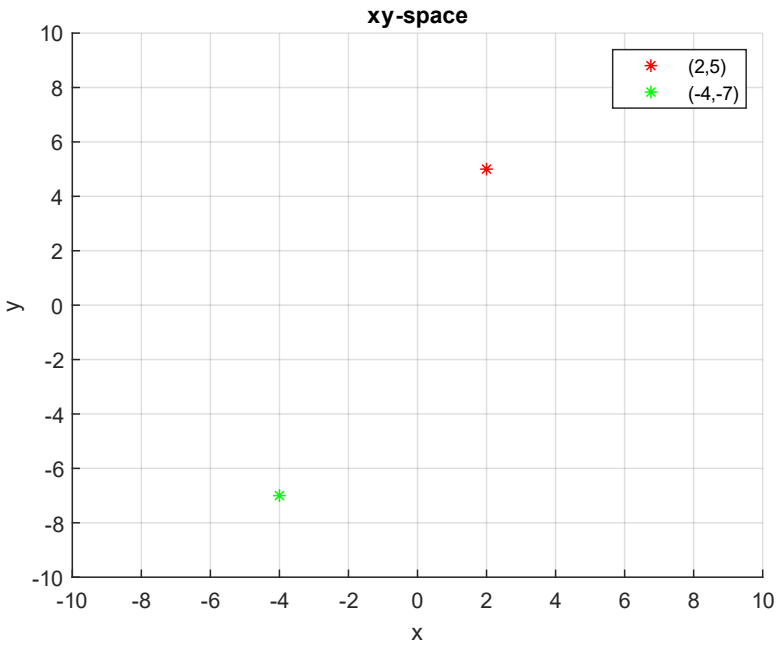
- Isolate features of specific shape
- => Hough transform
  - Lines
  - Circles
  - Ellipses
- Parametric representation
  - Robust wrt noise or partial occlusion

Hough transform: straight lines

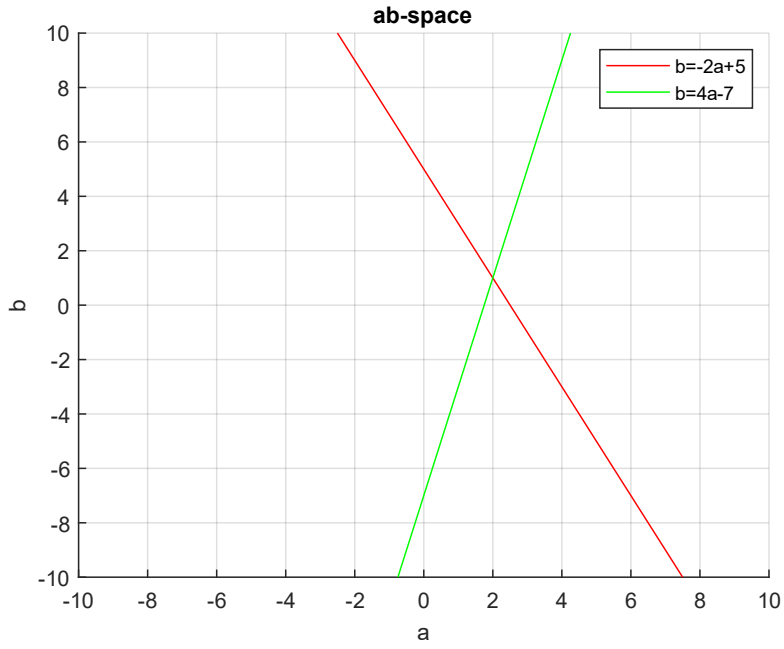
- The line equation
  - $y = ax + b$
  - Every point (x_i,y_i) on the line satisfies the condition
  - Parameters: a, b
- This can also be written as
  - $b = -xa + y$
  - In the ab-space the parameters are x, y
  - Conversely: all lines (a_i,b_i) that pass through a specific point
  - Parameters: x, y

Hough transform: straight lines

$y = ax + b$



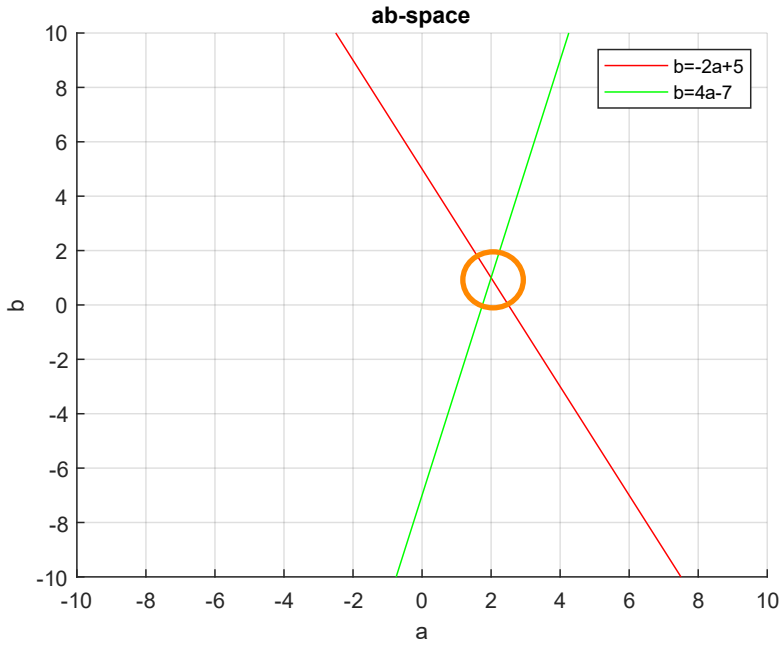
$b = -xa + y$



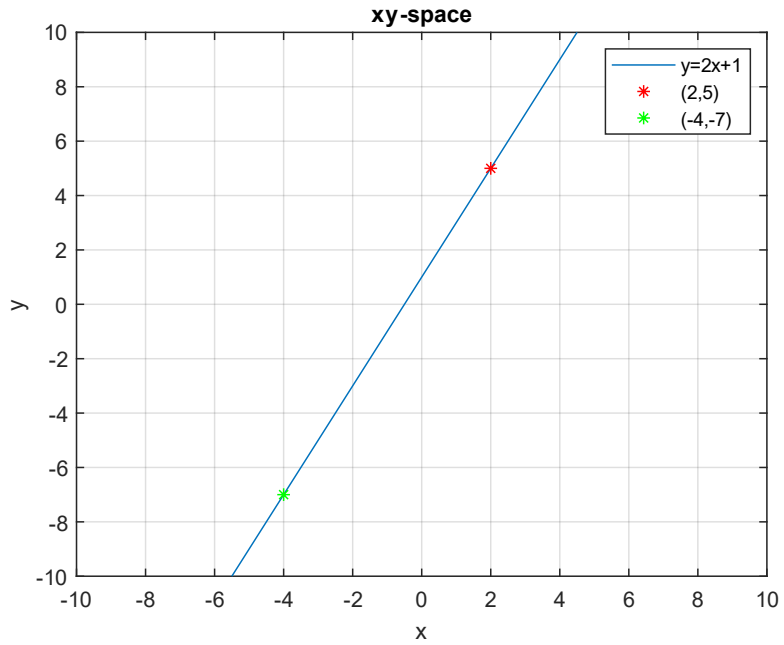
- A point in the xy-plane is a line in the parameter space

Hough transform: straight lines

$b = -xa + y$



$y = ax + b$



- An intersection in the parameter space is a line in the xy-plane
- (a, b) = (2, 1) => $y = 2x+1$

Hough transform: straight lines

The algorithm

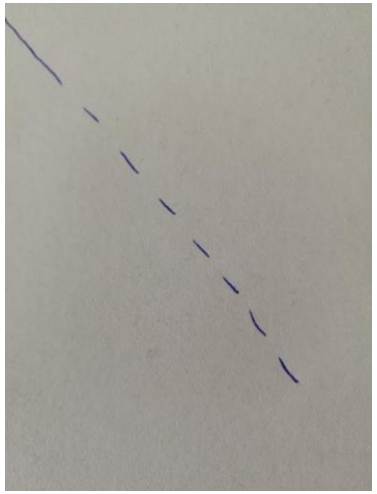
- Create an accumulator in the parameter space
  - Here two-dimensions, a and b, and quantize (resolution)
- Initialize the accumulator to 0
- For each pixel (x_i,y_i) that lies on an edge increment all elements of the accumulator that satisfy $b = -xa + y$
- Search the accumulator for large values
  - This is where lines intersect
  - And correspond to a line in the original image

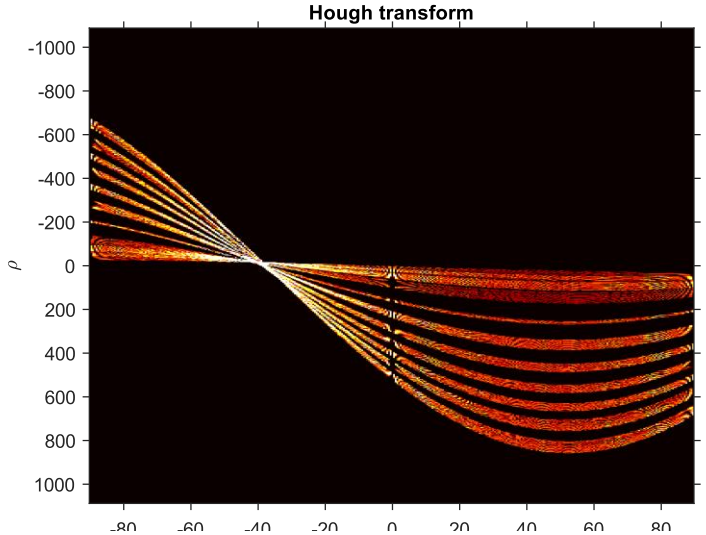
Hough transform: straight lines

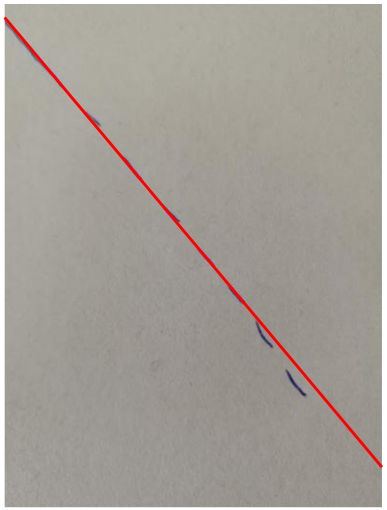
Problem

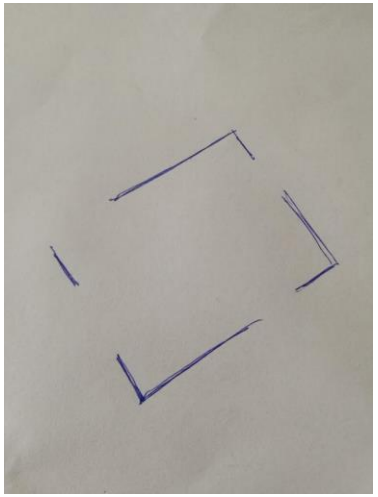
- How to represent vertical lines? x => inf
  - $r = x \cos \theta + y \sin \theta$
  - Rho and Theta, still two variables
  - but straight lines are represented by curves in the parameter space
  - These still intersect

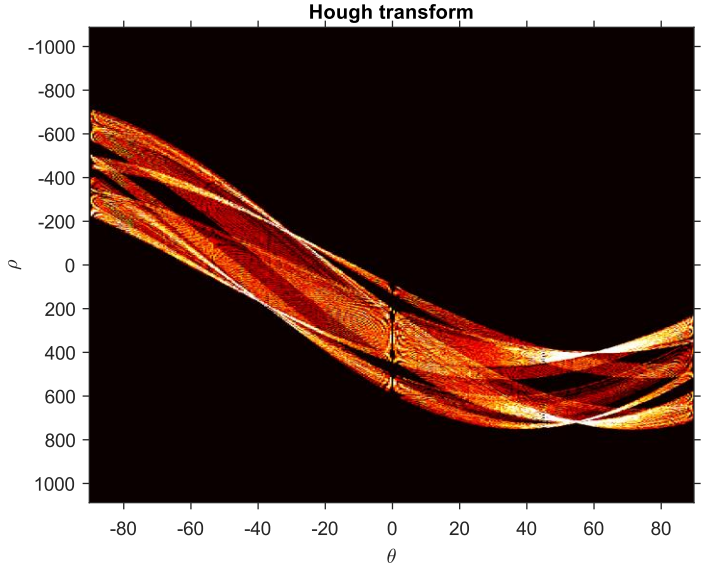
Hough transform: straight lines

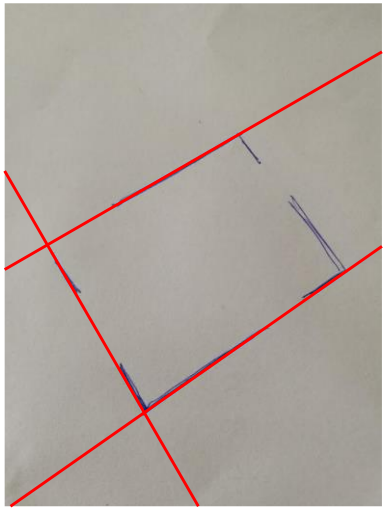




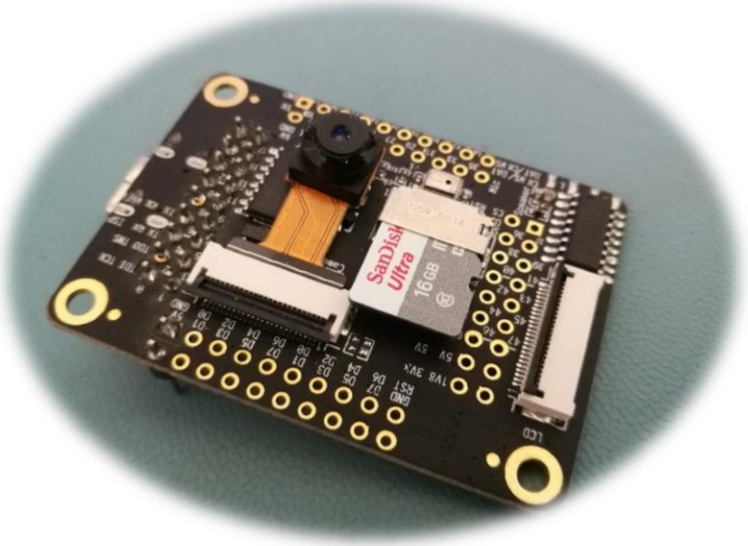


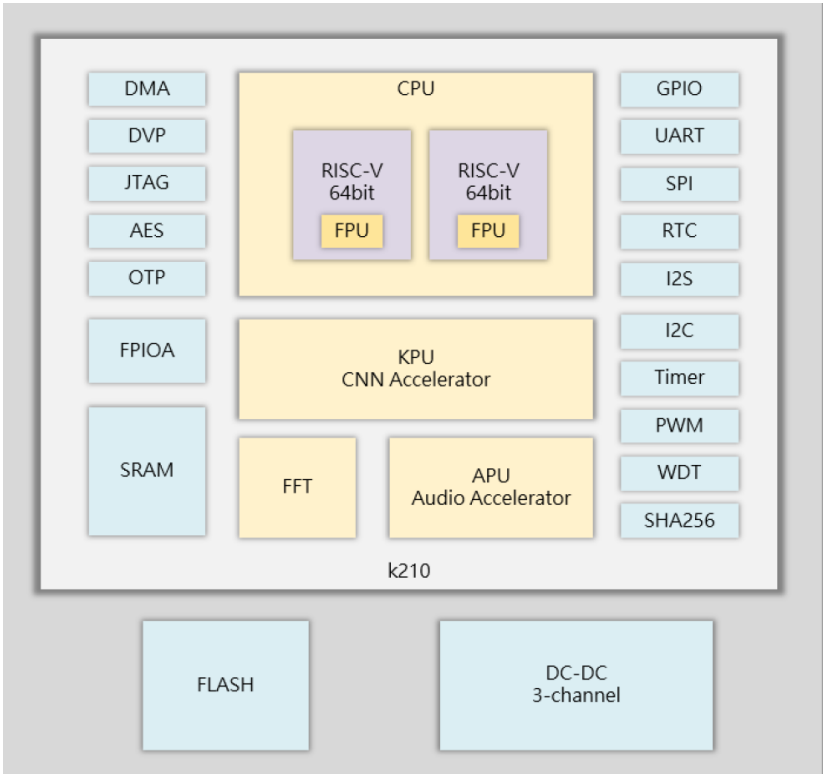


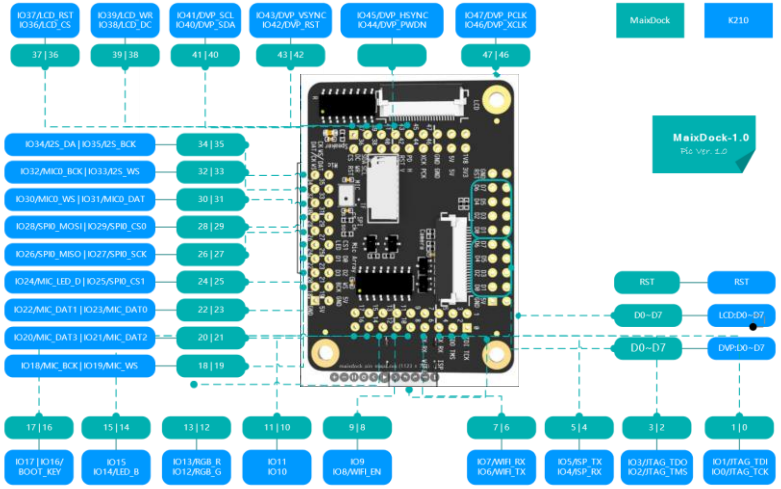




Maix Dock M1


SanDisk
Ultra
16GB


DMA
DVP
JTAG
AES
OTP
FPIOA
SRAM
CPU
RISC-V
64bit
FPU
RISC-V
64bit
FPU
KPU
CNN Accelerator
FFT
APU
Audio Accelerator
k210
GPIO
UART
SPI
RTC
I2S
I2C
Timer
PWM
WDT
SHA256
FLASH
DC-DC
3-channel


MaixDock
K210
MaixDock-1.0

MaixPy

- MaixPy -> Machine Vision -> Image module
- https://wiki.sipeed.com/soft/maixpy/zh/api_reference/machine_vision/image/image.html

  - Histogram
  - Threshold
  - get_regression
  - find_lines
    - Uses Hough
  - find_edges
    - image.EDGE_CANNY