2\1

CS6640 Week 2

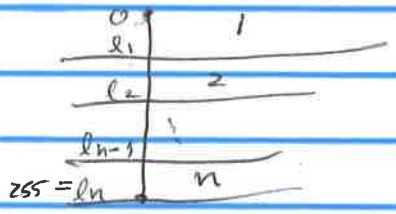
* Assessment sheets — ipcam on CADE machines
* Preprocessing (Chap 3 Gonzalez)

* Transform Pixel Values (assume 0 - 255)

* $s = c \log(1+r)$ darkl
show edge images

* $s = c r^{\gamma}$ gamma1 p. 142 Gonzalez

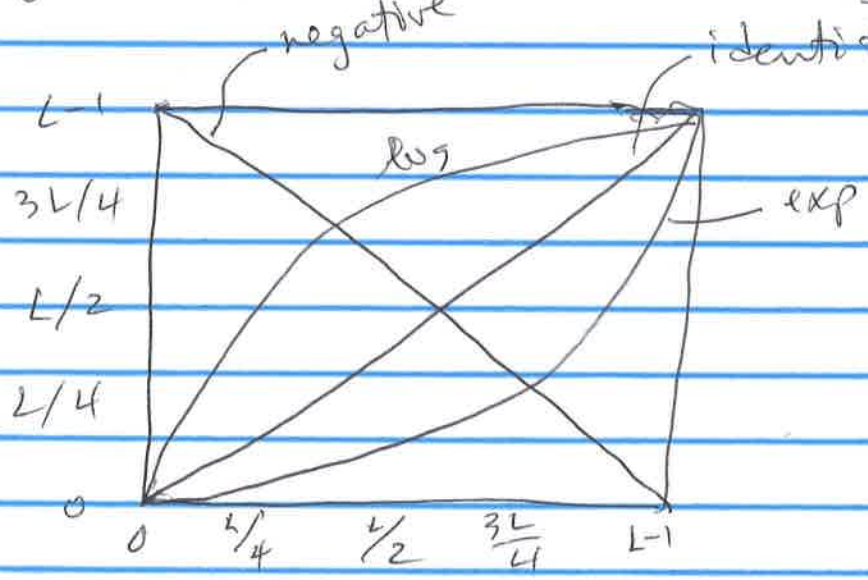
* level slicing:



imgs =

* histogram equalization
imghe then heq1, — heq4 plot heq4 heh4

Chart:



* slides 2, p. 1 : use snippet to get images
show heq on them

Eliminating noise : $f(x) + n(x)$

- averaging
- median, medfilt2

generally random + moves away from actual value

Spatial filtering Chap 3 Gonzalez

Viewing image as a function, there are many mathematical operations that can be expressed in terms of computations on the neighbors of points

E.g., consider Taylor series at a point b

$$f(x) = f(b) + f'(b)(x-b) + \frac{f''(b)(x-b^2)}{2!} \cdots$$

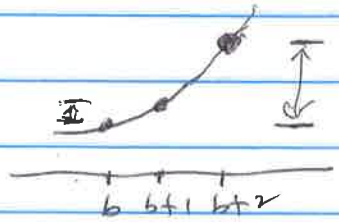
Truncate and rearrange to get estimate of derivative at b:

$$f'(b) = \frac{f(x) - f(b)}{(x-b)}$$

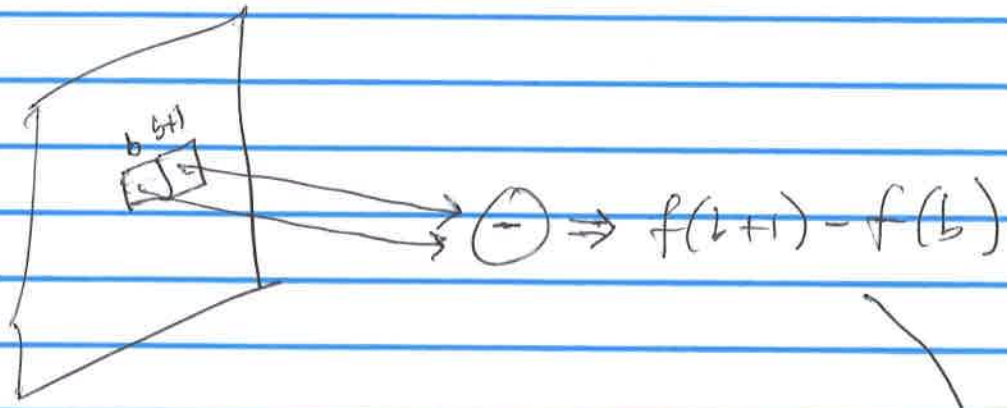
For an image, this means

$$f_1'(b) = \frac{f(b+1) - f(b)}{1}$$

$$f_2'(b) = \frac{f(b+2) - f(b)}{2}$$



but $f_1'(b)$ can be computed at each pixel

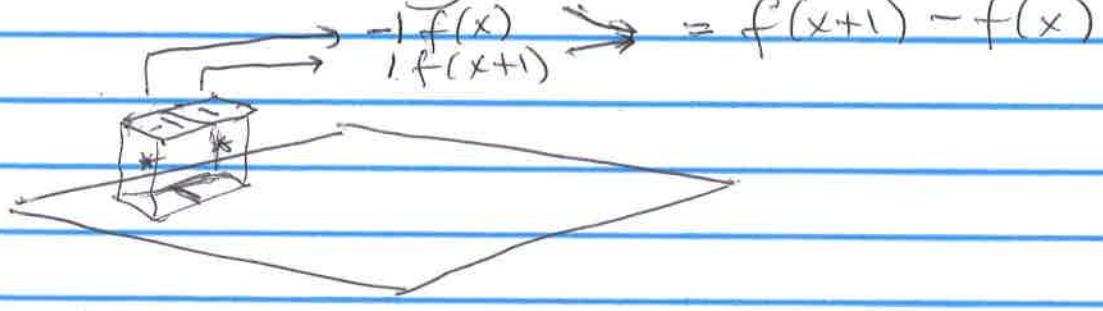


Consider equation in slides 4:

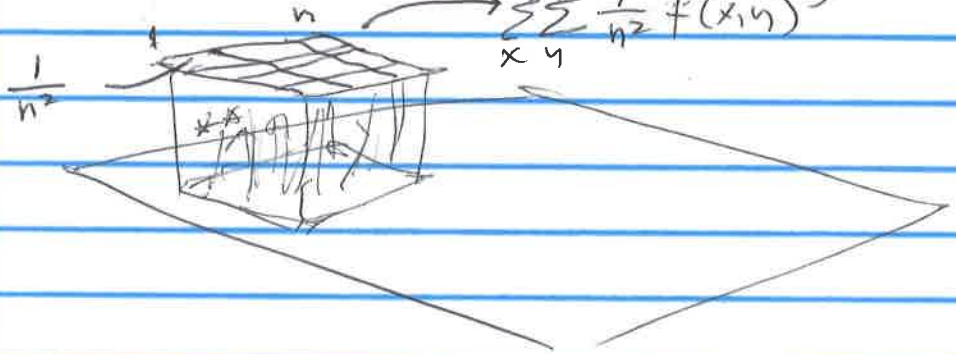
$$g(x) = \sum_{s=-a}^{a} w(s) f(x+s) = \sum_{s=b}^{b+1} w(s) f(x+s)$$

where $w = [-1, 1]$

View as moving window

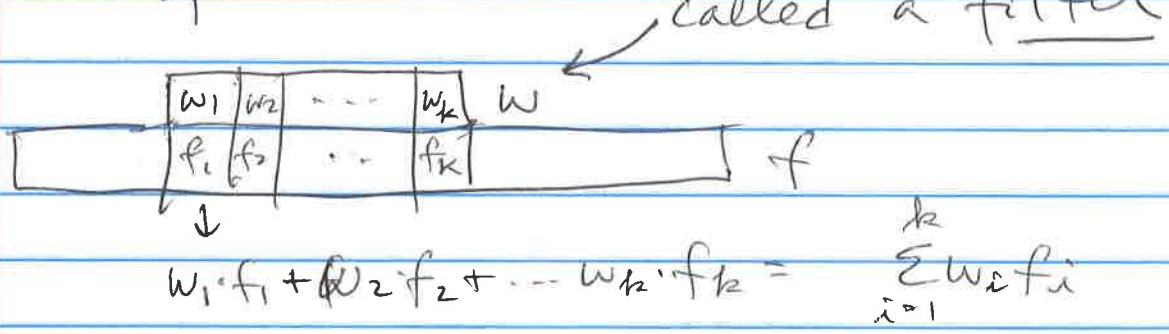


Another example is averaging:



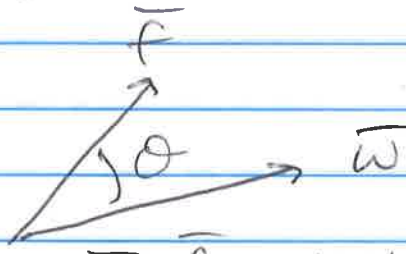
Correlation computes result of moving window across image to every location

* boundaries
* what is correlation as mathematical operation



$$w_1 \cdot f_1 + w_2 f_2 + \cdots w_k \cdot f_k = \sum_{i=1}^{k} w_i f_i$$

dot product
i.e., scaled version of $\cos\theta$



$$\bar{w} \cdot \bar{f} = |\bar{w}|\,|\bar{f}|\cos(\theta)$$

Show imfilter using [-1, 1] on shapes
" 11x11 / 121 " " noise
imfilter(, Gaussian , 15, 3
fspecial);
ven Gaussian on shapes sigma=

Pattern Matching in an image

a filter responds with high value when image "is like it" p.16 of slides 4

but when do 2 images "match"

* same gray values
* small variation in gray values
* small scale shift in gray values

Match detection : filter & subwindow

* $\sum_{x,y} |w(x,y) - f(x,y)| / size\_of\_window$

res = ...

* correlation : matches sky
  res_correl
* correlation coeff
* dot product normalized

* show example p. 22 of slides 4

res = conv2(match_tg,

convolution: $g(x,y) = \sum\sum w(s,t) f(x-s, y-t)$

flipud(fliplr(f))