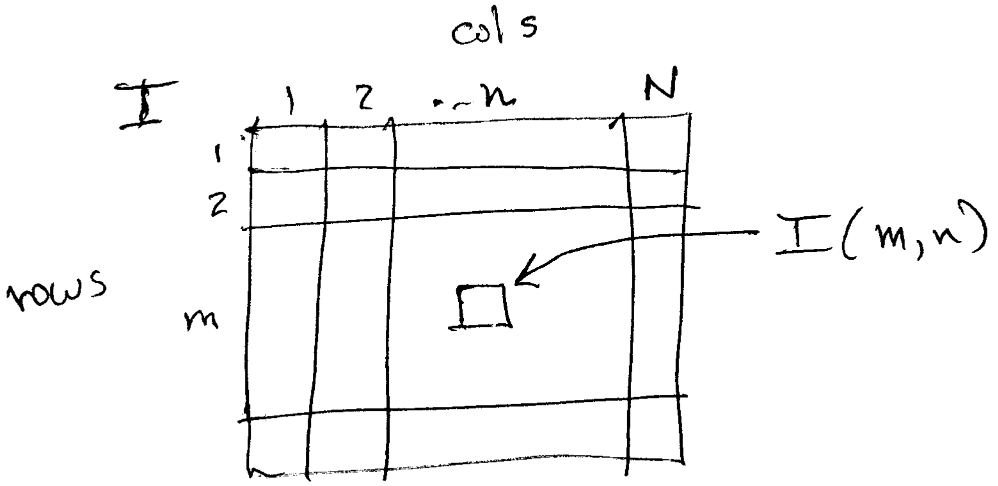
Image Processing Lecture Notes

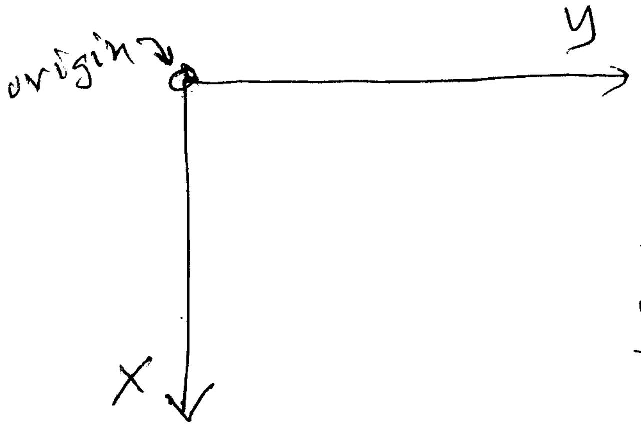
Representation : chapter 1

digital image (2D): $M \times N$ array of values



if viewed as a continuous patch of the plane

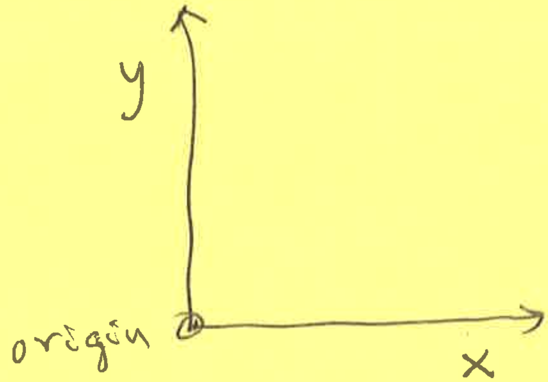
$\underbrace{I(x,y)}$ Cartesian coordinates



x matches row

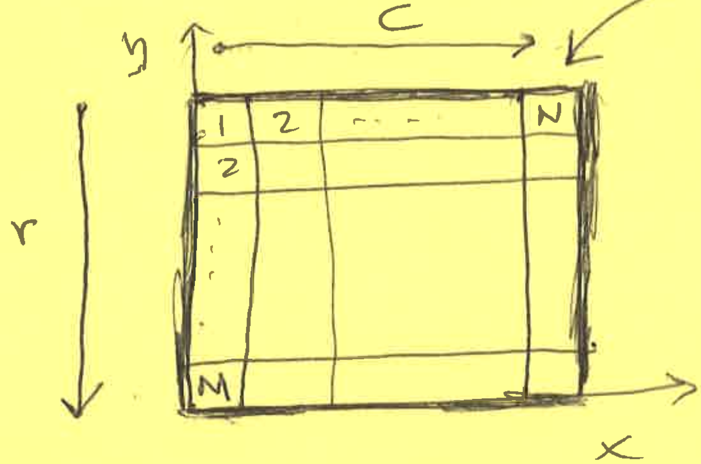
y matches col

However, some put origin at lower left:



Given an image, sometimes need to convert between this coordinate frame + row, col indexes

M x N matrix

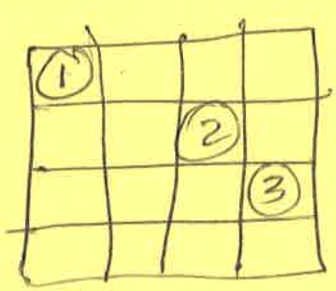


$y = M - r + 1$

$x = c$

$r = M - y + 1$

$c = x$



	r	c	x	y
1	1	1	1	4
2	2	3	3	3
3	3	4	4	2

Note, this becomes important when differentiating

$$\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}, \frac{\partial I}{\partial r}, \frac{\partial I}{\partial c}$$

---

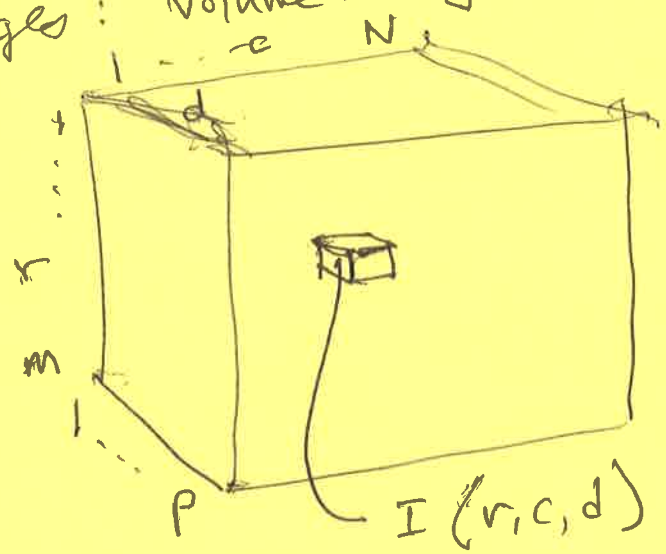
what values an image can have

binary: $I(m,n) \in \{0,1\}$

gray level: $I(m,n) \in \{0,1,\dots,255\}$

floating point: $I(m,n) \in$ floats

3D images: volume image



e.g., density in CAT scan

1\4

Color

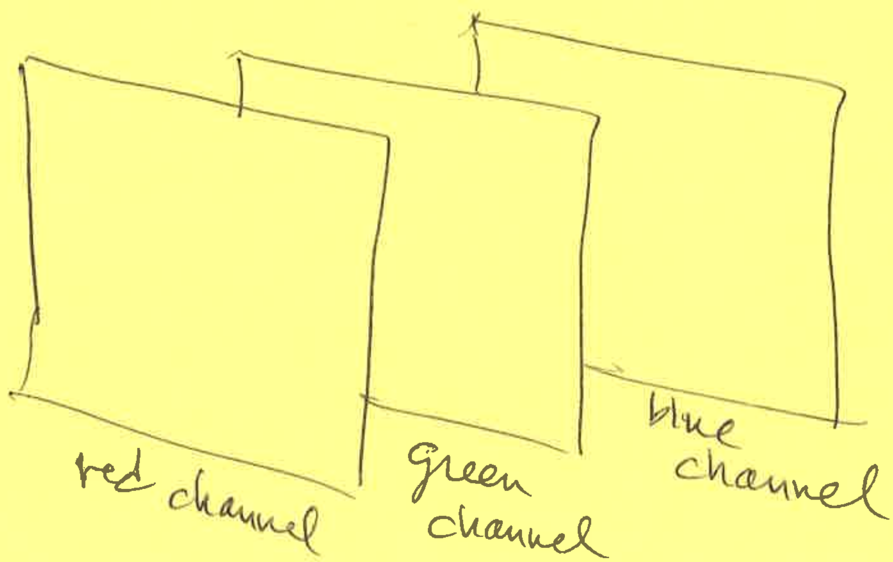
false color map: index gray level into a color map, e.g.:

1	2	4	5	3
1	2	4	5	3
1	2	4	5	3
1	2	4	5	3

black column, red column, blue column, white column, green column

$$\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{array}{l} \text{black} \\ \text{red} \\ \text{green} \\ \text{blue} \\ \text{white} \end{array}$$

true color: RGB



HSV : hue, saturation, value

hue ↑ color; saturation ↑ purity; value ↑ gray level

Resolution : spatial layout of image
temporal layout of image
pixel value layout (quantization)

spatial : # rows, # cols, # planes

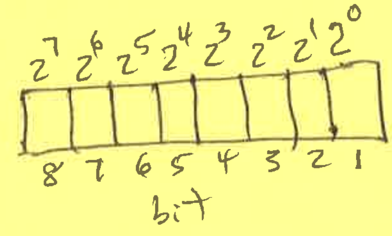
temporal : frames per second

bit resolution : # bits for each value & layout
e.g., binary, gray-level, float

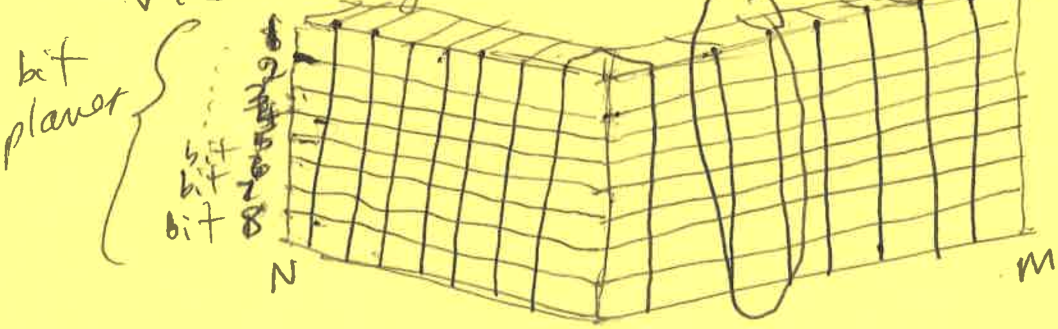
---

bit-plane splicing

Given gray level image : 256 gray levels
⇒ 8 bits ≡ 1 byte per value
can pack this into memory



View image as :



display only 1 bit plane at a time

CS4640_slices where is the noise?

Image Formats (see p. 6 of text)

GIF
JPEG
BMP
PNG
TIF/F

Image Compression

E.g., only keep bit planes 3 to 8

saves: 2 * M * N bits

assume M & N are even, then saves M * N bytes

Color revisited

To convert RGB to gray level:

$$g = \alpha r + \beta g + \gamma b$$

$\gamma = 0.1140$

$\beta = 0.5870$

$\alpha = 0.2989$

Test this out in Matlab

1\7

Matlab
Basic statements

<statements> ≡ <statement>;
<statement>;
⋮
<statement>;

Assignment

A = 3;

---

if <cond>
<statements>
end

---

if <cond>
<statements>
else
<statements>
end

---

for k = 1:n
<statements>
end

---

while <cond>
<statements>
end

Matlab data structures

variables

A
b27
min_dist

vectors

row: e.g., $[2, 4, 6, 7, 9]$ is 1×5

col: e.g., $\begin{bmatrix} 4 \\ 8 \\ 7 \end{bmatrix}$ is 3×1

arrays (matrices)

2D $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ is 3×3 identity

$\begin{bmatrix} 1 & 8 & 7 & 4 \\ 3 & 2 & 1 & -1 \end{bmatrix}$ is 2×4

3D $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ is a $2 \times 2 \times 3$

$\begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$

$\begin{bmatrix} 9 & 10 \\ 11 & 12 \end{bmatrix}$

functions (examples)

```
function fun1
display('Hello world!');
```

→ goes in .m file e.g., hello.m

```
function fun2(x)
x
```

→ fun2.m

```
function y = fun3(x)
y= x*x +3;
```

→ fun3.m

```
function [y1,y2]= fun4(x)
y1= x^2;
y2= sqrt(x);
end
```

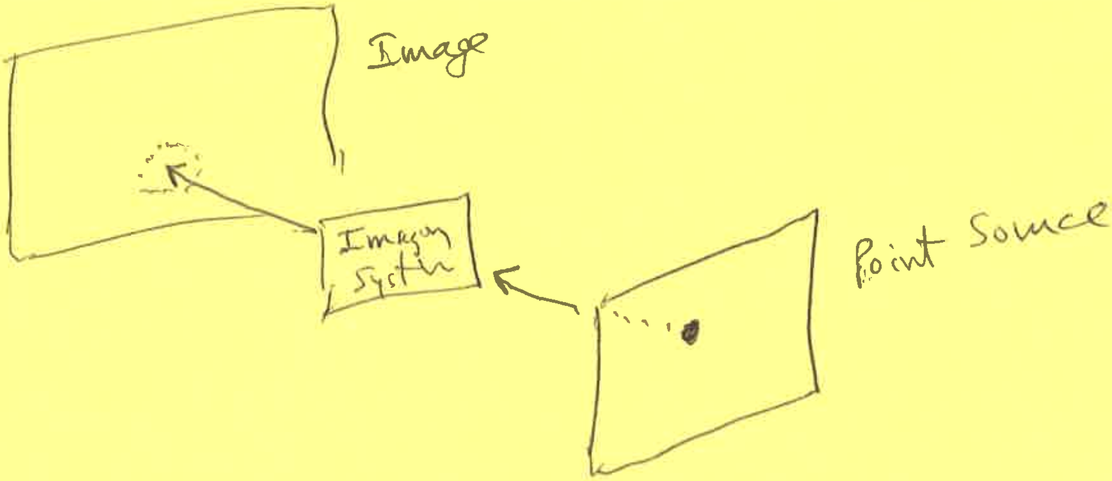
→ fun4.m

Matlab IP functions

imfinfo
imread
imwrite
imshow
imagesc
colormap
subplot
imtool (not imview)
impixelinfo
rgb2gray
mat2gray
rgb2hsv
load

2/1

Image Formation



$$\text{image} = \underbrace{PSF}_{\text{point spread function}} \; \underbrace{\circ}_{\text{convolution}} \; \underbrace{f}_{\text{scene function}} + \underbrace{n}_{\text{noise}}$$

point spread function: describes how imaging system spreads input across image

$$g(x,y) = \iint f(x',y') \underbrace{h(x,y,x',y')}_{PSF} \, dx' \, dy'$$

want to linearly add influence of each scene point at each image point

use Dirac delta function

$$\delta(x,y) = \begin{cases} \infty & x=0,\ y=0 \\ 0 & \text{else} \end{cases}$$

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \delta(x,y)\, dx\, dy = 1$$

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\, \delta(x-x_0, y-y_0)\, dx\, dy = f(x_0,y_0)$$

	<u>idealized</u>
$f(x,y) = \delta(x-x_0, y-y_0)$	point
$f(x,y) = \delta(x-x_0)$	vertical line
$f(x,y) = \delta(y-y_0)$	horizontal line
$f(x,y) = \delta(ax+by+c)$	arbitrary line

$ax+by+c=0$ & suppose $\left\| \begin{bmatrix} a \\ b \end{bmatrix} \right\| = 1$

Then $\begin{bmatrix} a \\ b \end{bmatrix}$ is unit normal to line

if $b \neq 0$, then a/b is slope of line

and $(0, -\frac{c}{b})$ is on line

Consider single intensity point as input

then

$$g(x,y) = \iint \delta(x'-x_0, y'-y_0)\, h(x,y;x',y')\,dx'dy'$$

then $g(x,y) = h(x,y;x',y')$ sifting theorem

called impulse response. h is the point spread function.

since imaging is linear: PSF describes process completely.

---

imaging systems are shift invariant:

$$h(x,y;x',y') = h(x-x', y-y')$$

so,

$$\underbrace{g(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x',y')\,h(x-x', y-y')\,dx'dy'}$$

definition of convolution

book uses notation $f(x,y) ** h(x,y)$
$f(x) * h(x)$

most people use $f(x,y) \circ h(x,y)$
$f(x) \circ h(x)$

Discrete convolution (our view)

1D: suppose $f = [1\ 2\ 3\ 4\ 5]$ $h = [5\ 4\ 3\ 2\ 1]$

$$g(1) \equiv \left.\begin{array}{l} 0\ 0\ 0\ 0\ 1\ 2\ 3\ 4\ 5 \\ 1\ 2\ 3\ 4\ 5\ 0\ 0\ 0\ 0 \\ \underbrace{0+0+0+0+5}_{\text{indexed by formulas}} +0+0+0+0 \end{array}\right\} = 5$$

$$g(2) \equiv \left.\begin{array}{l} 0\ 0\ 0\ 1\ 2\ 3\ 4\ 5 \\ 1\ 2\ 3\ 4\ 5 \\ \hline 0+0+0+4+10 \end{array}\right\} = 14$$

$$g(3) = \left.\begin{array}{l} 0\ 0\ 1\ 2\ 3\ 4\ 5 \\ 1\ 2\ 3\ 4\ 5 \\ \hline 3+8+15 \end{array}\right\} = 26$$

⋮

i.e., slide reversed h along 0-padded f wherever there is overlap

$[5, 14, 26, 40, 55, 40, 26, 14, 5] = f \circ h$

2D: suppose $f = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ $h = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$

$g(1,1) \equiv$

$$\begin{matrix} 8 & 7 & 0 & \\ 6 & 5 & & \\ & 0 & 1 & 2 \\ & & 3 & 4 \end{matrix}$$

$7 \cdot 0 + 8 \cdot 0 + 6 \cdot 0 + 5 \cdot 1 = 5$

$g(1,2) \equiv$

$$\begin{matrix} 8 & 7 & \\ 6 & 5 & \\ & 1 & 2 \\ & 3 & 4 \end{matrix} = 6 + 10 = 16$$

$g(1,3) \equiv$

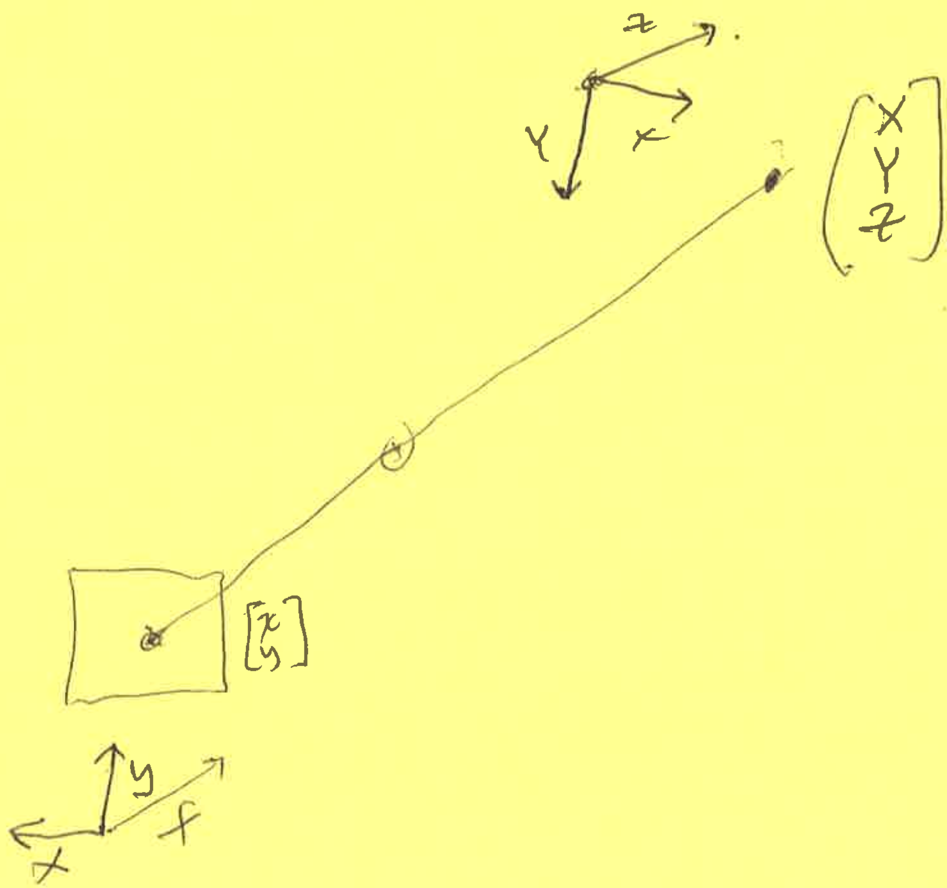
$$\begin{matrix} & 8 & 7 \\ & 6 & 5 \\ 1 & 2 & \\ 3 & 4 & \end{matrix} = 6 \cdot 2 = 12$$

$$f \circ h = \begin{bmatrix} 5 & 16 & 12 \\ 22 & 60 & 40 \\ 21 & 52 & 32 \end{bmatrix}$$

Camera Model

Perspective projection equations

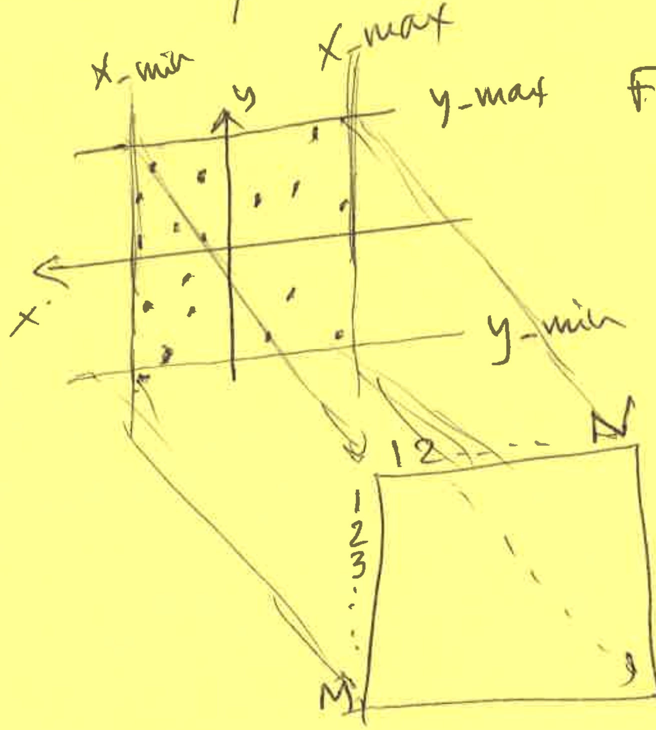
$$x = f\frac{X}{Z} \qquad y = f\frac{Y}{Z}$$



2/7

Image Formation Issues

(from figure 2.15)



Find x & y min & max

Map to image coordinates

Issues: * x, y pts are not integer coordinates
* intensity related to distance
* should apply Gaussian filter
* image inverted

Note:

$$\frac{x - x\_min}{x\_max - x\_min} = \frac{c-1}{N-1}$$ } can solve for c (or x)

Noise

$im = f \circ h + \underline{\underline{n}}$

Usually:

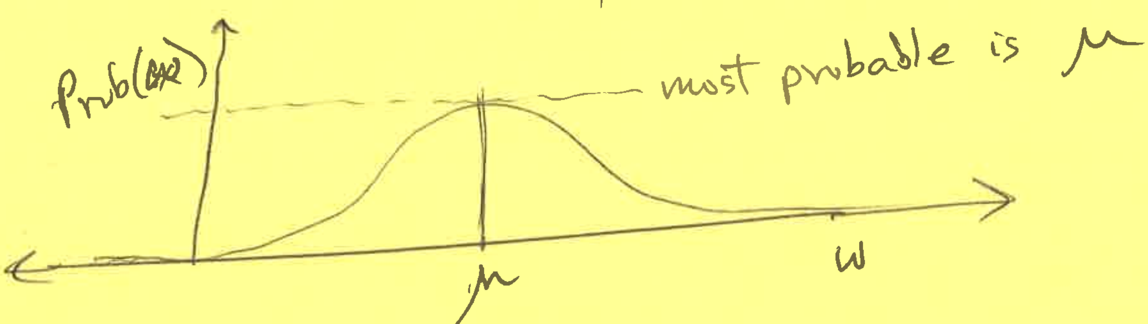
Gaussian:

$im = f \circ h + w$

where $w \sim \mathcal{N}(\mu, \sigma^2)$

mean variance

w is a value sampled from Normal distribution



If you take a bunch of samples, the histogram will be:

Salt & Pepper: pixels turned to 0 or 1 randomly