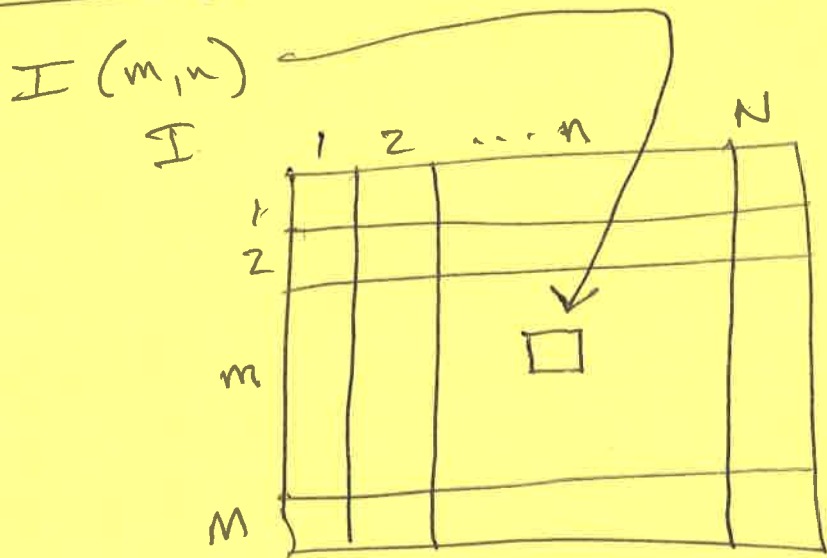CS4640   Aug. 20

Representation : chapter 1
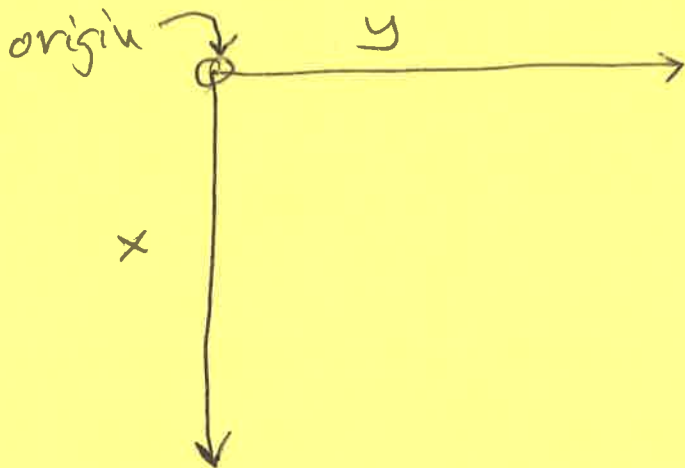
digital image (2D) : M × N array of values

$$I(m,n)$$

I

if viewed as a continuous patch of the plane
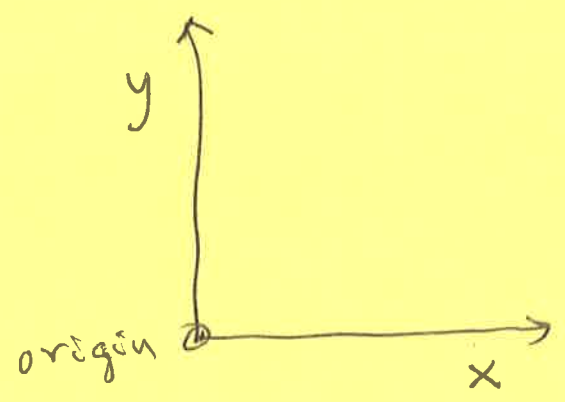
$$I(x,y)$$

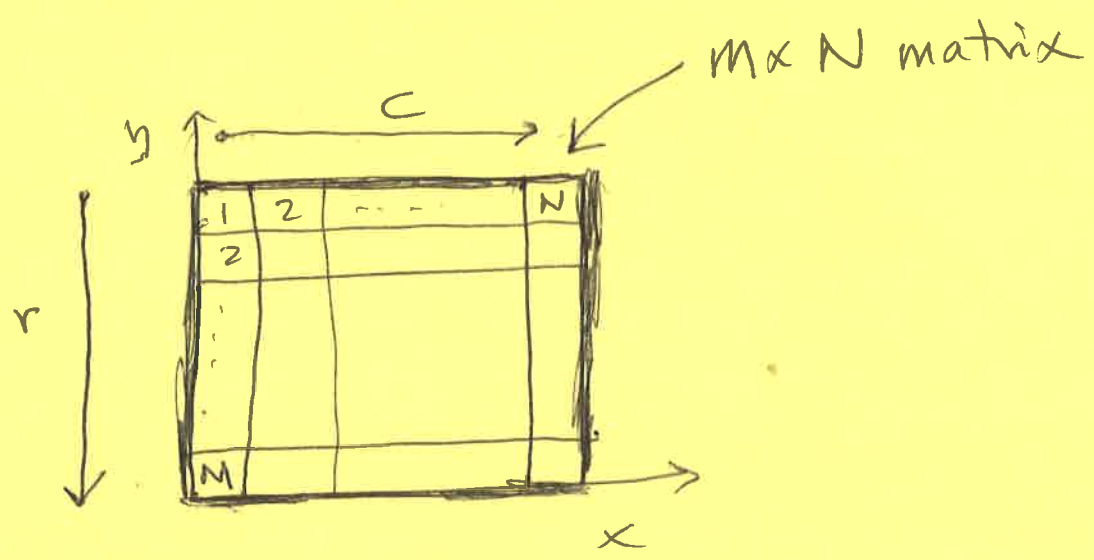then   Cartesian coordinates

origin

y

x

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
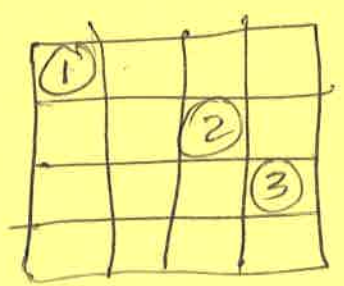
$M \times N$ matrix

$y = M - r + 1$
$x = c$

$r = M - y + 1$
$c = x$

	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

$$\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}, \frac{\partial I}{\partial r}, \frac{\partial I}{\partial c}$$
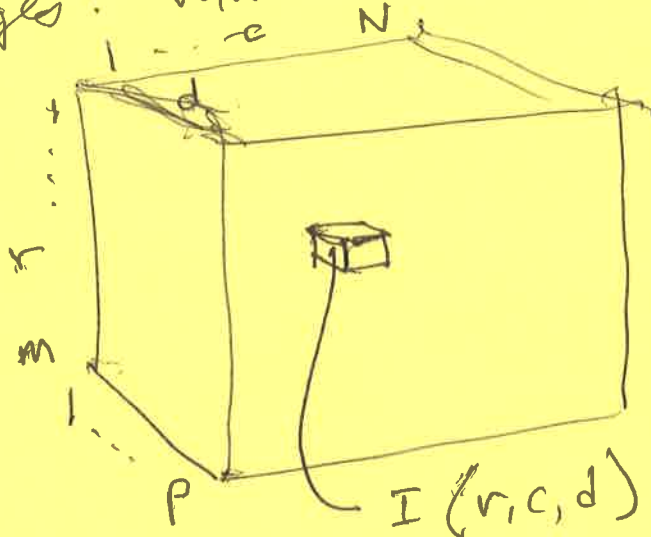
what values an image can have

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

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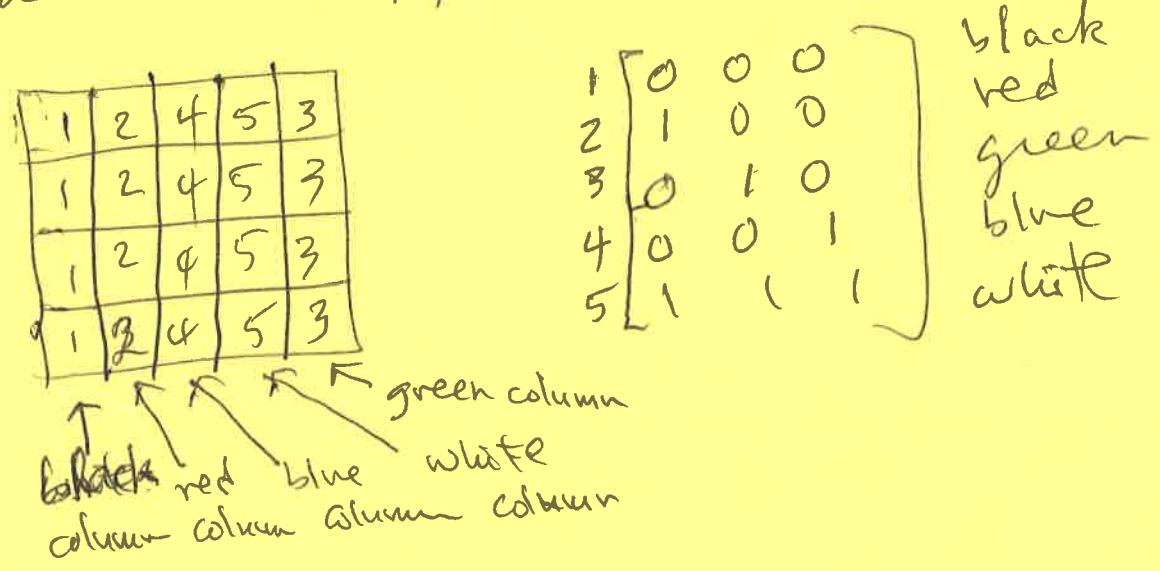
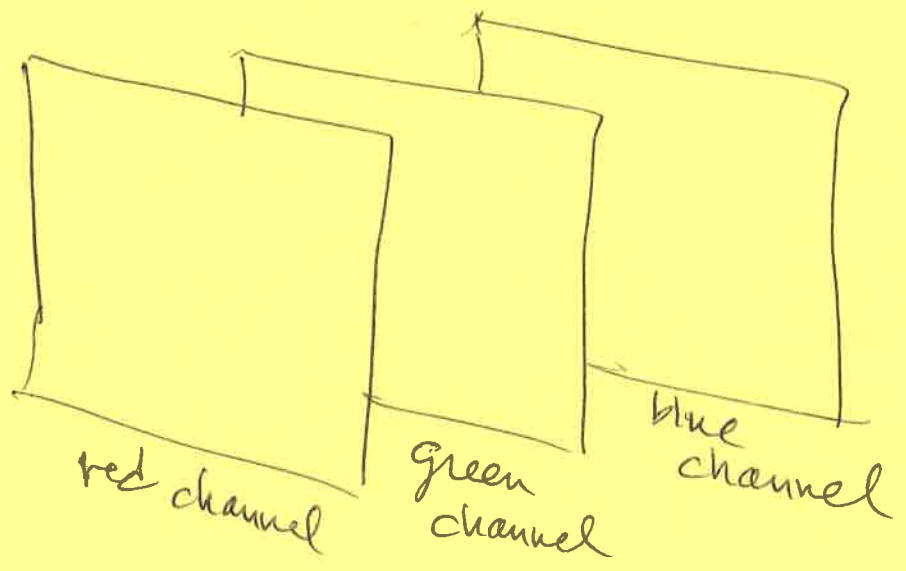floating point: $I(m,n) \in \text{floats}$

3D images: volume image

$I(r,c,d)$

e.g., density in CAT scan

Color

<u>false color map</u>: 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

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

green column

black   red   blue   white
column column column column

true color: RGB

red channel   green channel   blue channel

HSV : hue, saturation, value

color purity gray level

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

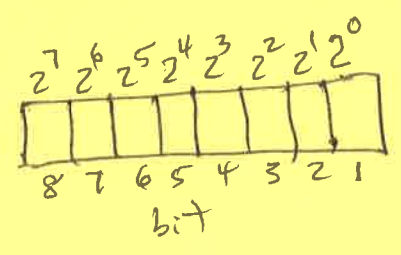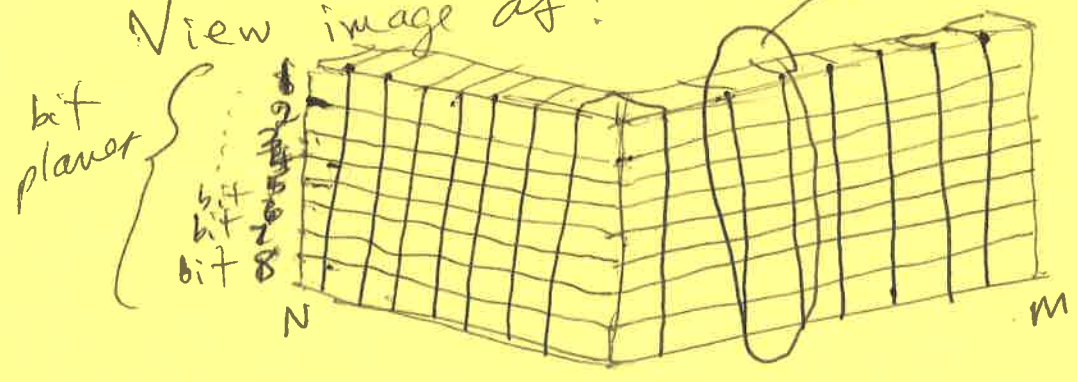spatial : # rows, # cols, # planes

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
$\Rightarrow$ 8 bits $\equiv$ 1 byte per value
can pack this into memory

$$2^7\ 2^6\ 2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0$$

8 7 6 5 4 3 2 1
bit

View image as :

bit planes {

bit 1
bit 2
bit 8

N   M

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

Matlab
Basic statements

$\langle statements \rangle \equiv$ $\langle statement \rangle;$
$\langle statement \rangle;$

$\langle statement \rangle;$

Assignment

$A = 3;$

if $\langle cond \rangle$
$\quad \langle statements \rangle$
end

if $\langle cond \rangle$
$\quad \langle statements \rangle$
else
$\quad \langle statements \rangle$
end

for $k = 1:n$
$\quad \langle statements \rangle$
end

while $\langle cond \rangle$
$\quad \langle statements \rangle$
end

Matlab data structures

variables

A
b27
min_dist

vectors

row: e.g., ~~#~~ $[2,4,6,7,9]$ is 1×5

col: e.g., $\begin{pmatrix} 4 \\ 8 \\ 7 \end{pmatrix}$ 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 $2 \times 2 \times 3$

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

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

functions (examples)

goes in .m file
→ e.g., hello.m

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

→ fun2.m

```
function fun2(x)

x
```

→ fun3.m

```
function y = fun3(x)

y = x*x +3;
```

→ fun4.m

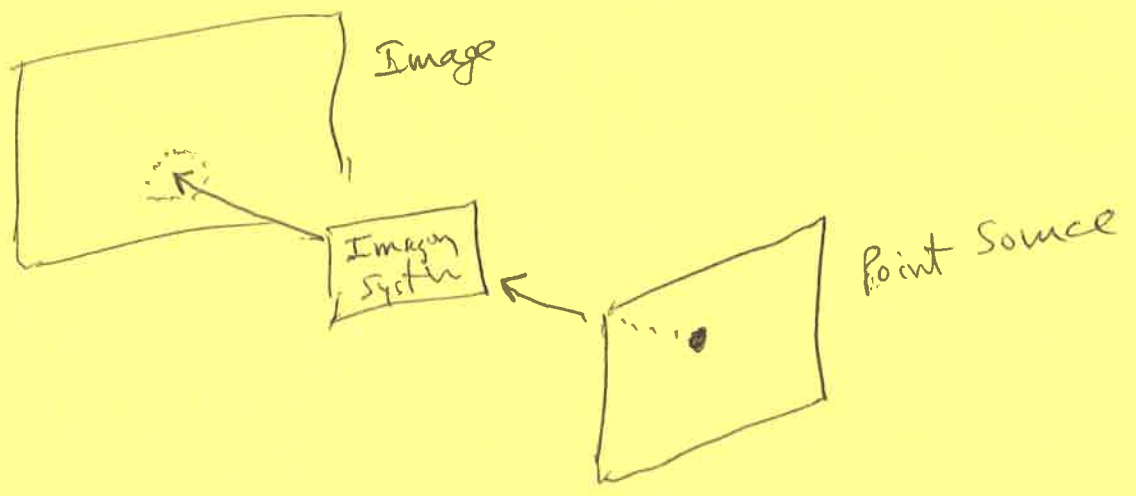```
function [y1,y2] = fun4(x)

y1 = x^2;
y2 = sqrt(x);

end
```

## Matlab IP functions

imfinfo

imread

imwrite

imshow

imagesc

colormap

subplot

imtool     (not imview)

impixelinfo

rgb2gray

mat2gray

rgb2hsv

load

# Image Formation



Image

Imaging System

Point Source

$$\text{image} = PSF \circ f + n$$

point spread function

convolution

scene function

noise

describes how imaging system spreads input across image

$$g(x,y) = \iint f(x',y')\, h(x,y,x',y')\, dx'\, dy'$$

PSF

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

use Dirac delta function

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

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

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

idealized

$$f(x,y) = \delta(x-x_0, y-y_0) \qquad \cdot \text{point}$$
$$f(x,y) = \delta(x-x_0) \qquad \text{vertical line}$$
$$f(x,y) = \delta(y-y_0) \qquad \text{horizontal line}$$
$$f(x,y) = \delta(ax+by+c) \qquad \text{arbitrary line}$$

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

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,

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

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

$$g(1) = \begin{array}{cccccccccc} 0 & 0 & 0 & 0 & 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 & 0 & 0 & 0 & 0 \\ " & " & " & " & " \\ 0+0+0+0+5, & +0 & +0 & +0 & +0 \end{array} \Bigg\} = 5$$

$\underbrace{\phantom{xxxxxxxxxxxx}}$ indexed by formulas

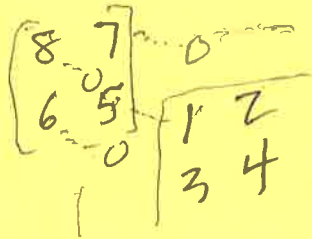$$g(2) \equiv \begin{array}{ccccccc} 0 & 0 & 0 & 1 & 2 & 3 & 4\ 5 \\ 1 & 2 & 3 & 4 & 5 \\ \hline 0 +0+0 +4 + 10 \end{array} \Bigg\} = 14$$

$$g(3) = \begin{array}{ccccccc} 0 & 0 & 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \\ \hline 3 + 8 + 15 \end{array} \Bigg\} = 26$$

$\vdots$

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

$[5, 14, 26, 40, 55, 40, 26, 14, 5] = f \circ h$
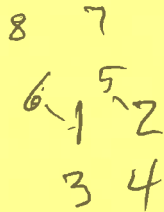
2D: suppose $f = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$  $h = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$
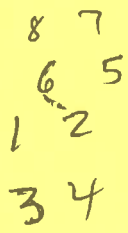
$g(1,1) \equiv$

$$\begin{bmatrix} 8 & 7 \\ 6 & 5 \end{bmatrix} \begin{array}{|cc} 0 & \\ \hline 1 & 2 \\ 3 & 4 \end{array}$$

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

$g(1,2) \equiv$

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

$g(1,3) =$

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

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

# Camera Model

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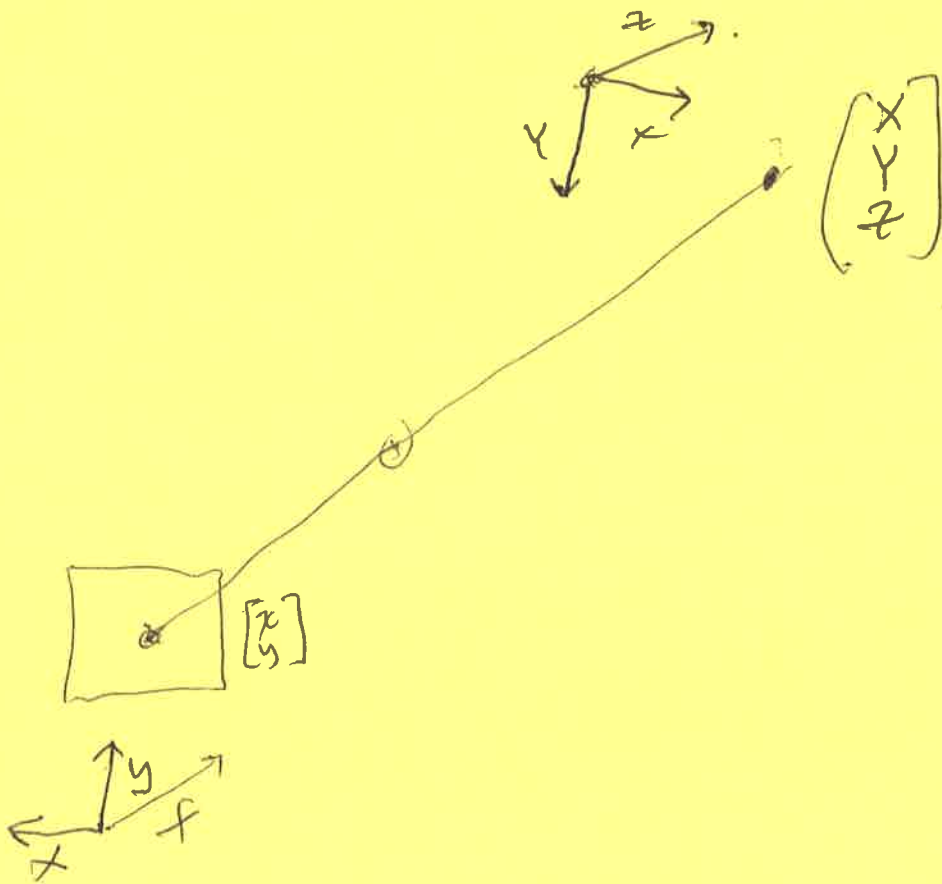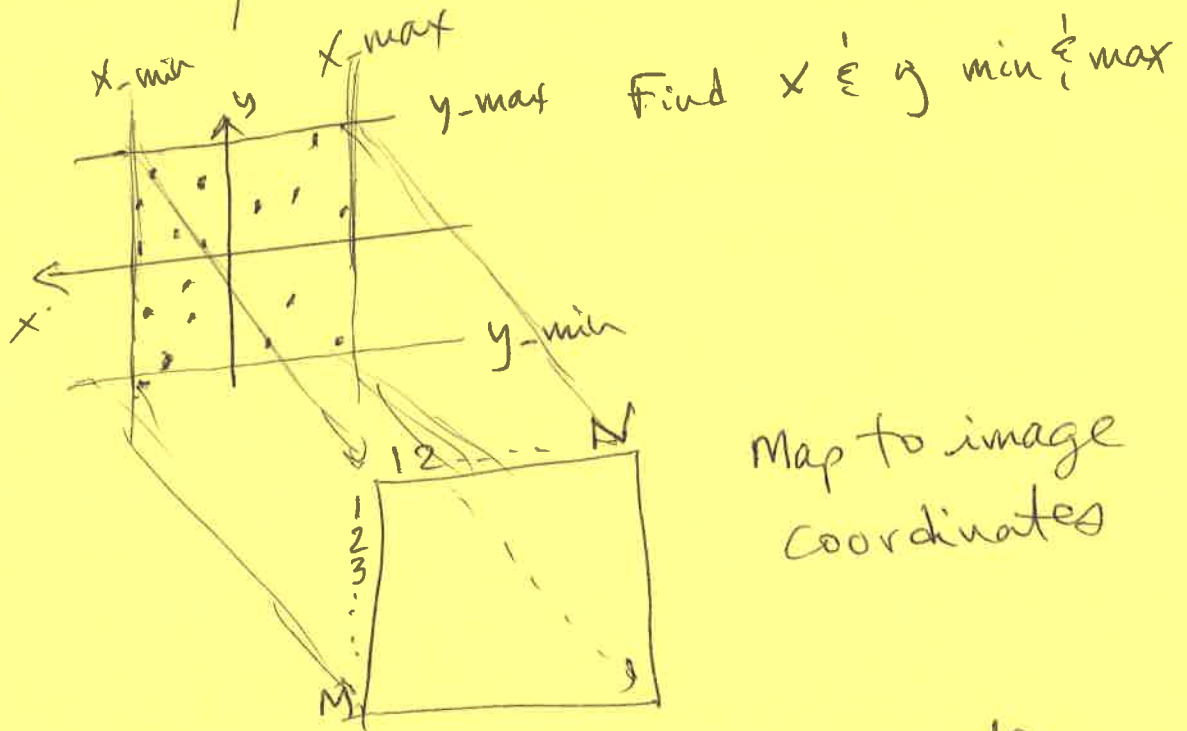
$$x = f \frac{X}{Z} \qquad\qquad y = f \frac{Y}{Z}$$

# Image Formation Issues

(from figure 2.15)

Find $x$ & $y$ min & max
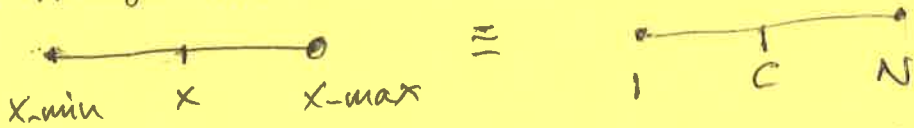
Map to image coordinates

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 $w \sim N(\mu, \sigma^2)$

mean   variance

$w$ is a value sampled from → Normal distribution



most probable is $\mu$
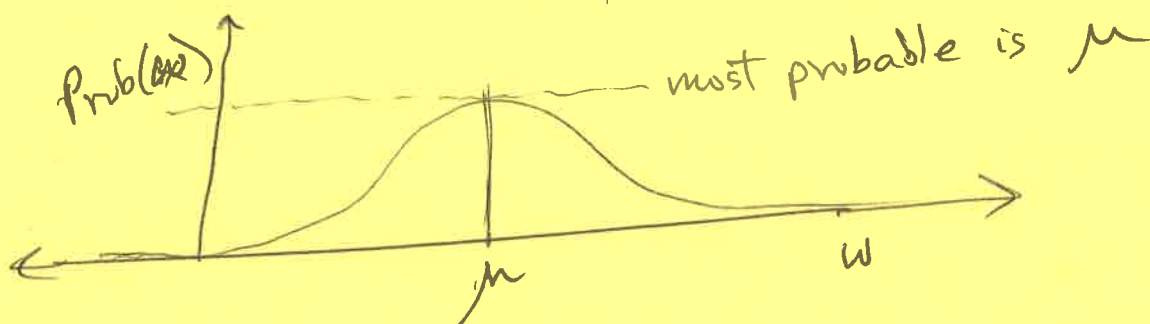
If you take a bunch of samples, the histogram will be:



Salt & Pepper: pixels turned to 0 or 1 randomly