CS4640 Image Processing 9 Jan 2018

* Fill out assessment
* ~~Fight~~ Honor code form
* Web page
→ look at assignment : Keep these in mind when reviewing material
* present intro to images
* present geometric transformations
* my notes
* problems

CS4640 Intro to Images [put pdf's on lectures]

* Slide Set : Ross Whitaker & Tolga Tasdizen
* 4 : a mapping (usually discrete)
* 5 - 28
  * 8 → error : 3-D plot not rhsystem
* ~~33~~ - ~~39~~ arithmetic operations
* ~~40~~ - 53 grids & adjacency
  * 48 connected components : bwlabel

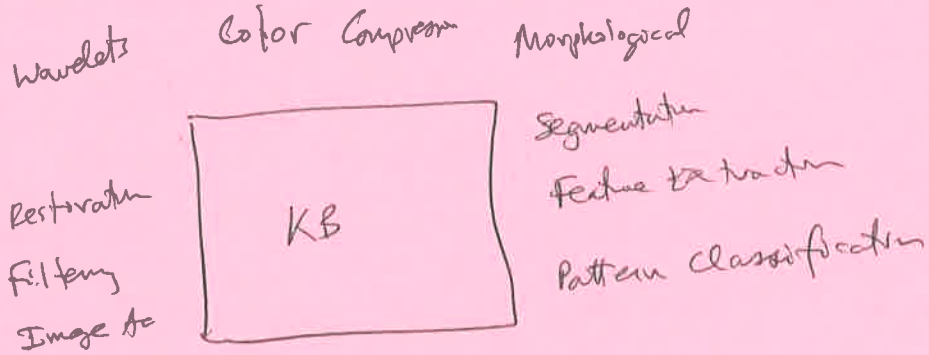
---

Geometric Transformations

* Slide Set: Sridhuman Ramalingam, Ross, J. Chai
* 2 - 6
  * 7 this is view when things are continuous when discrete; may need to interpolate (slide 11)
  * 21-33 : Rigid → Treat as 4 parameters Use this transform for registration

# Chapter 1

p. 25




## Proposed Application Domain : Traffic Analysis

---

pp. 45-47 Image Formation

illum reflectance

$$f(x,y) = i(x,y)\, r(x,y)$$

↓ e constant illum

e.g.;


---

## Image Sampling and Quantization

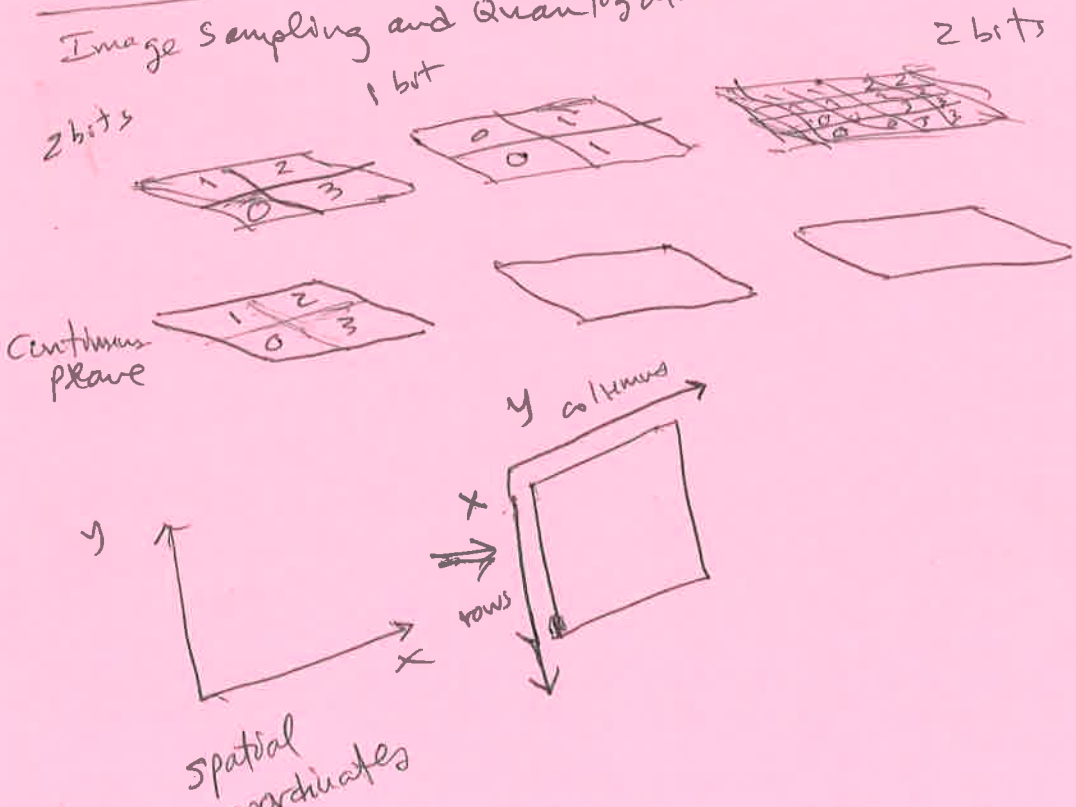
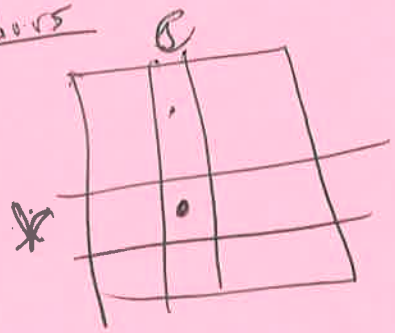


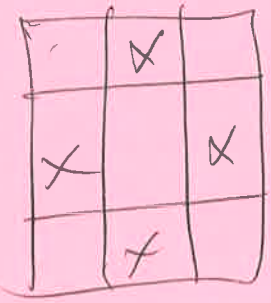
image interpolation

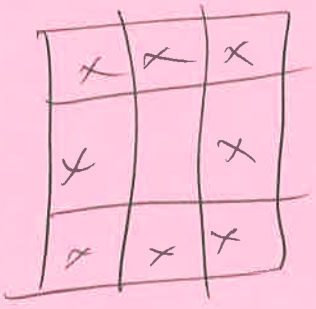
pp. 61-62

Matlab: imresize

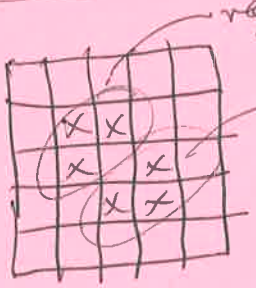
Neighbors





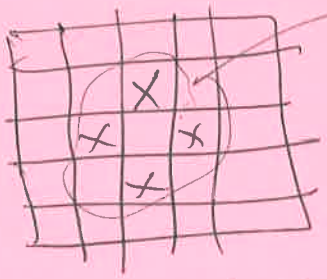




have to be careful




4-neighbors

Two distinct regions separate the plane into two parts - but they are not connected!



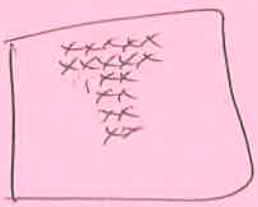
8-neighbor

1 connected region do not disconnect plane.

Jordan Curve Theorem: Any connected curve in the plane cuts the plane into a least 2 pieces.

Issue: Find connected components in an image.

E.g.:



Letter T

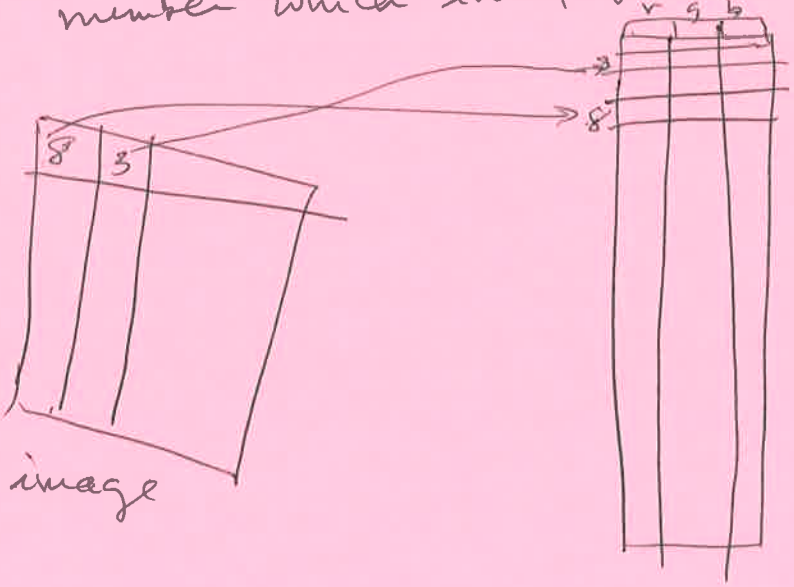
**boundary** of a connected component
pixels that have at least one background pixel as a neighbor

## Mathematical Tools

An image is a matrix

Matlab has various representations:

* regular matrix: each pixel is regular float (double)
* uint8: each element is a byte
* rgb: 3 image planes for red, green & blue
* indexed: ~~image has~~ each element is a whole number which indexes into a color table




## Matlab functions:

mat2gray
rgb2gray
⋮

## Matrix operations

pp. 67 –

Element-wise operations (Matlab):

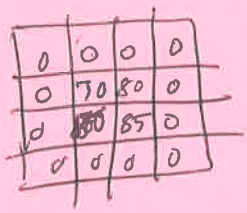
`im > 0.5` : every pixel > 0.5 becomes 1, else 0

↑ a Boolean operator : produces true or false

let mask = 

and im = 

| 50 | 60 | 70 | 50 |
|---|---|---|---|
| 50 | 70 | 80 | 90 |
| 70 | 80 | 85 | 95 |
| 70 | 70 | 75 | 90 |

then `im .* mask` = 

| 0 | 0 | 0 | 0 |
|---|---|---|---|
| 0 | 70 | 80 | 0 |
| 0 | 80 | 85 | 0 |
| 0 | 0 | 0 | 0 |

## arithmetic operations

$\frac{im1 + im2}{2}$ is average of im1, im2 (same size)

## adding noise to an image

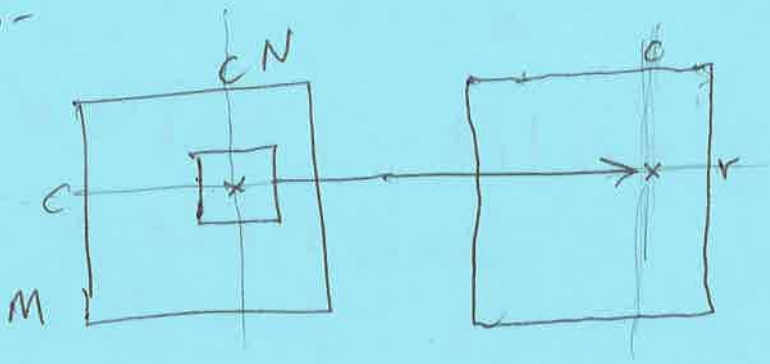
```
[m,n] = size(im);
imn = im + rand(m,n);
```

this may or may not work

→ if im is in range [0,1], then adding a random value from [0,1] may make the result out of range

## Neighborhood operations

pp. 83-




e.g., average, max

## Geometric Transformations

- produces (globally) a new image

Matlab: imrotate : about center of image & fits in new image

## Image Registration

pp. 88-91


pick 4 or more points in each image that correspond

Then transform new image as:

$$x = c_1 v + c_2 w + c_3 vw + c_4$$
$$y = c_5 v + c_6 w + c_7 vw + c_8$$

$$\bar{b} = A \bar{x}$$

$$\begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ x_3 \\ y_3 \\ x_4 \\ y_4 \end{bmatrix} = \begin{bmatrix} v_1 & w_1 & v_1w_1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & v_1 & w_1 & v_1w_1 & 1 \\ v_2 & w_2 & v_2w_2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & v_2 & w_2 & v_2w_2 & 1 \\ v_3 & w_3 & v_3w_3 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & v_3 & w_3 & v_3w_3 & 1 \\ v_4 & w_4 & v_4w_4 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & v_4 & w_4 & v_4w_4 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \\ c_8 \end{bmatrix}$$

pts

$$\begin{bmatrix} x_1 & y_1 & v_1 & w_1 \\ x_2 & y_2 & v_2 & w_2 \\ \vdots & & & \\ x_n & y_n & v_n & w_n \end{bmatrix}$$

$$\bar{x} = A \backslash b$$

## Image Transforms pp. 93-95

pair
$$\begin{cases} T(u,v) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\, r(x,y,u,v) & \text{forward} \\ f(x,y) = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} T(u,v)\, s(x,y,u,v) & \text{inverse} \end{cases}$$

Fourier transform:
$$r(x,y,u,v) = e^{-j2\pi(ux/M + vy/N)}$$
$$s(x,y,u,v) = \frac{1}{MN} e^{j2\pi(ux/M + vy/N)}$$

## Probability pp. 96 -

More later

Histogram p. 107 : count number of occurences of an outcome

for an image: counts gray levels

## Uniform distribution p. 112

$$p(z) = \begin{cases} \frac{1}{b-a} & a \le z \le b \\ 0 & \text{otherwise} \end{cases}$$

mean: $\frac{a+b}{2}$

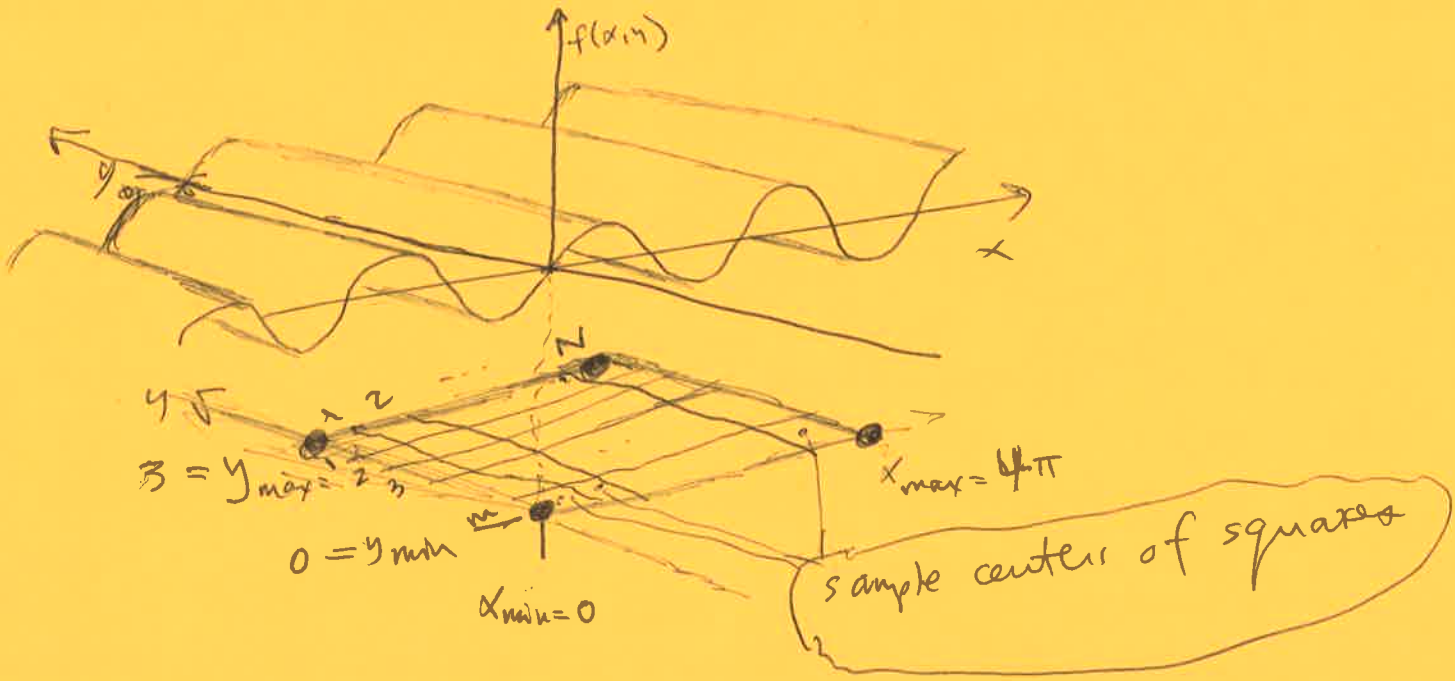
variance $= \frac{1}{12}(b-a)^2$

## Gaussian distribution p. 112

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(z-\bar{z})^2}{2\sigma^2}}$$

Problem 1

Create an image : e.g., sinusoid $f(x,y) = \sin(x)$




Idea: Get points from $x_{min}$ to $x_{max}$ N steps (equispaced)
$y_{min}$ to $y_{max}$ M steps

Then find midpoints between each pair

Use sample points to get f values

Get complete set first

Scale $f_{min}$ to $f_{max}$ into Q integers (0 to Q-1)

## Problem FT

Check that $FT_i(FT(im)) = im$

Can also check that $FT(im) = fft2(im)$

use helper function for addition

## Problem noise

make sure image is not noisy if low noise requested

## Problem pixel dist

just create (at each pixel) a vector for each image and compute norm

[M,N,P] = size(im): allows images with different channels

## Problem Register

* use any number of correspondences
* use rigid, 4-variable transform:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11}x + a_{12}y + a_{13} \\ -a_{12}x + a_{11}y + a_{23} \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ -a_{12} & a_{11} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

gives form of problem set up

$$\begin{bmatrix} x_1' \\ y_1' \\ \vdots \\ x_n' \\ y_n' \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & 1 & 0 \\ y_1 & -x_1 & 0 & 1 \\ \vdots & & & \\ x_n & y_n & 1 & 0 \\ y_n & -x_n & 0 & 1 \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{23} \end{bmatrix}$$