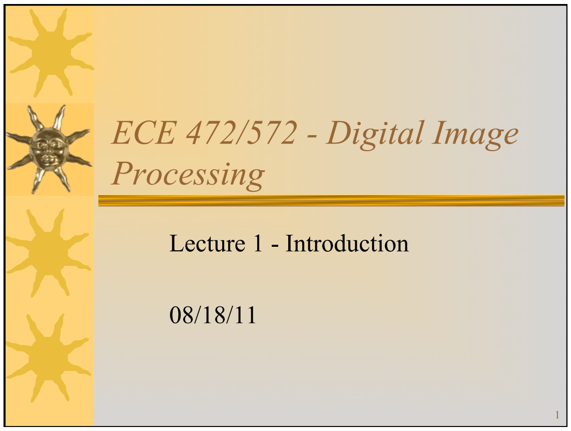
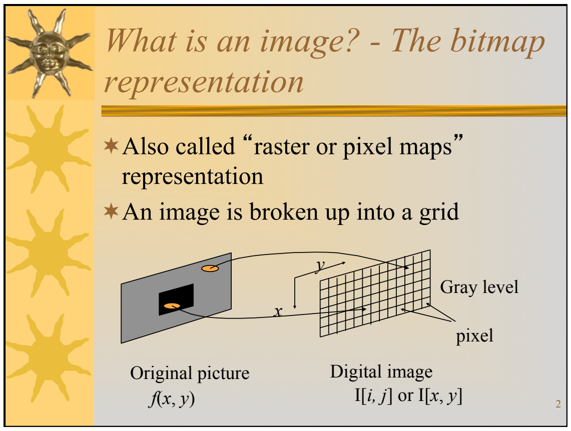


ECE 472/572 - Digital Image Processing

Lecture 1 - Introduction

08/18/11

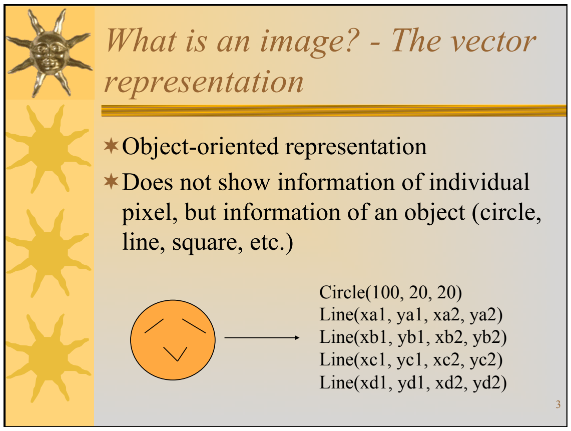


What is an image? - The bitmap representation

- Also called "raster or pixel maps" representation
- An image is broken up into a grid

Original picture
$f(x, y)$

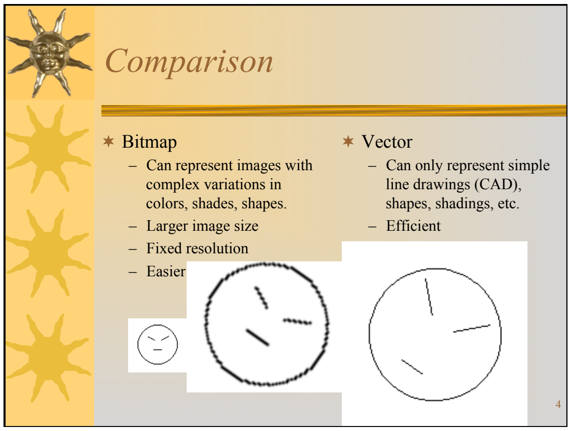
Digital image
$I[i, j]$ or $I[x, y]$



What is an image? - The vector representation

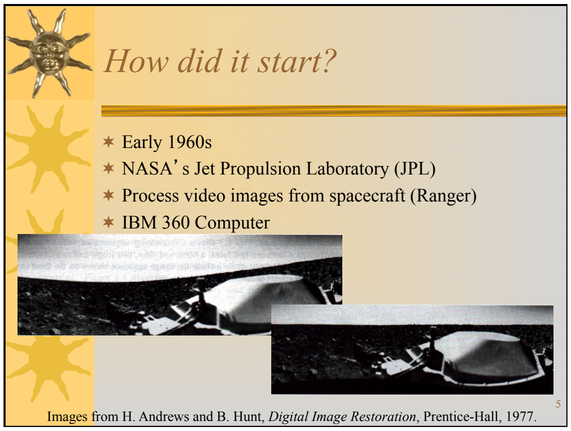
- Object-oriented representation
- Does not show information of individual pixel, but information of an object (circle, line, square, etc.)

```
Circle(100, 20, 20)
Line(xa1, ya1, xa2, ya2)
Line(xb1, yb1, xb2, yb2)
Line(xc1, yc1, xc2, yc2)
Line(xd1, yd1, xd2, yd2)
```



Comparison

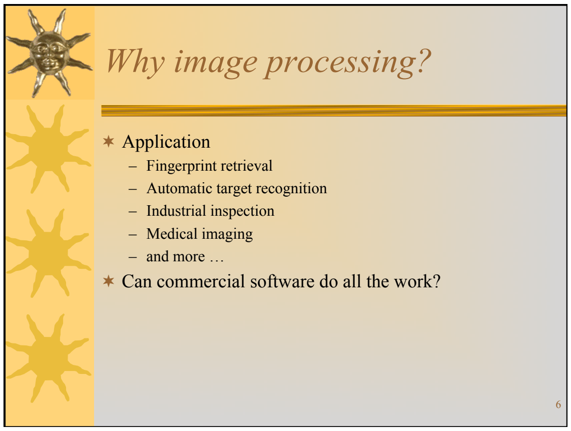
- Bitmap
  - Can represent images with complex variations in colors, shades, shapes.
  - Larger image size
  - Fixed resolution
  - Easier
- Vector
  - Can only represent simple line drawings (CAD), shapes, shadings, etc.
  - Efficient



How did it start?

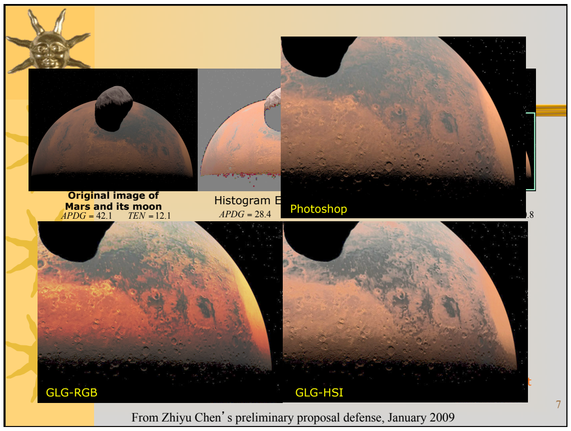
- Early 1960s
- NASA's Jet Propulsion Laboratory (JPL)
- Process video images from spacecraft (Ranger)
- IBM 360 Computer

Images from H. Andrews and B. Hunt, *Digital Image Restoration*, Prentice-Hall, 1977.

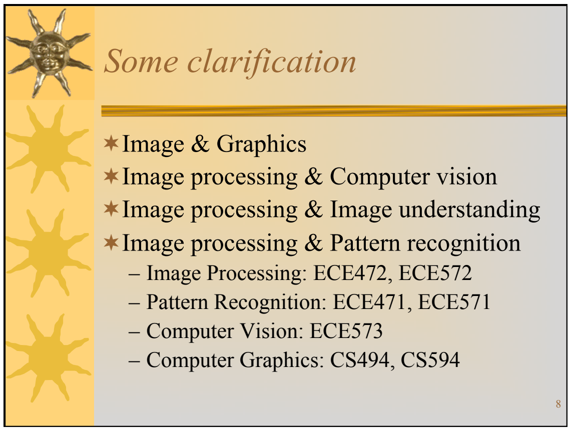


Why image processing?

- Application
  - Fingerprint retrieval
  - Automatic target recognition
  - Industrial inspection
  - Medical imaging
  - and more …
- Can commercial software do all the work?

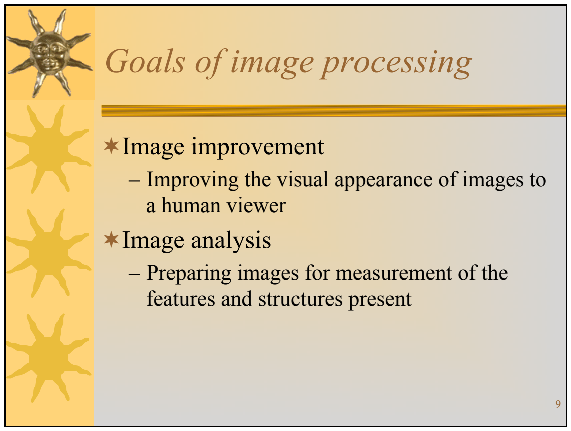


From Zhiyu Chen's preliminary proposal defense, January 2009



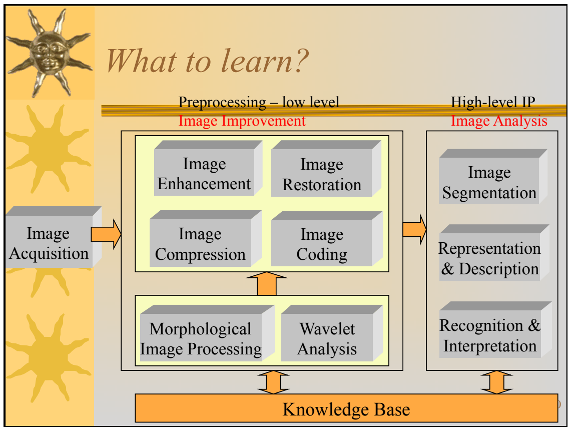
Some clarification

- Image & Graphics
- Image processing & Computer vision
- Image processing & Image understanding
- Image processing & Pattern recognition
  - Image Processing: ECE472, ECE572
  - Pattern Recognition: ECE471, ECE571
  - Computer Vision: ECE573
  - Computer Graphics: CS494, CS594



Goals of image processing

- Image improvement
  - Improving the visual appearance of images to a human viewer
- Image analysis
  - Preparing images for measurement of the features and structures present



What to learn?

Preprocessing – low level
Image Improvement

High-level IP
Image Analysis

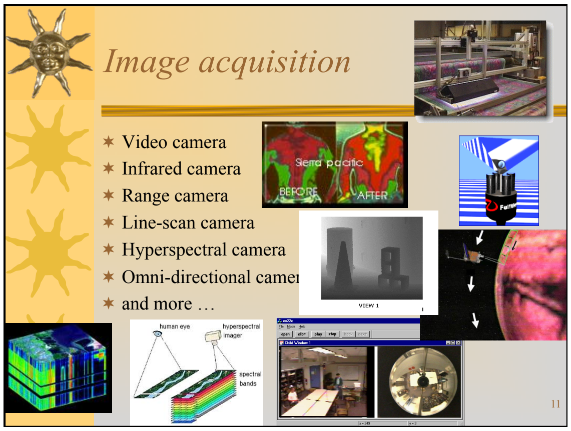
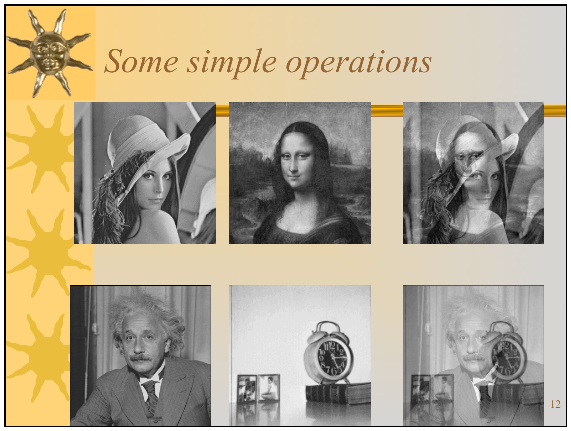


Image acquisition

- Video camera
- Infrared camera
- Range camera
- Line-scan camera
- Hyperspectral camera
- Omni-directional camer
- and more …



Some simple operations

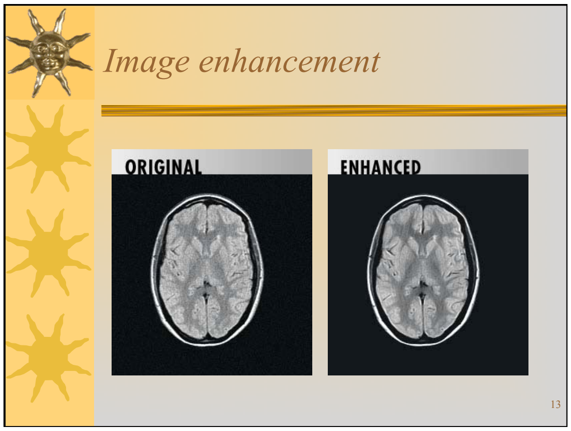


Image enhancement

13

Movie film restoration

14

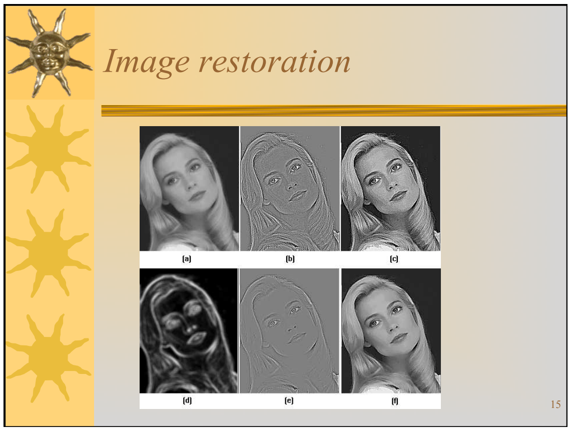


Image restoration

15

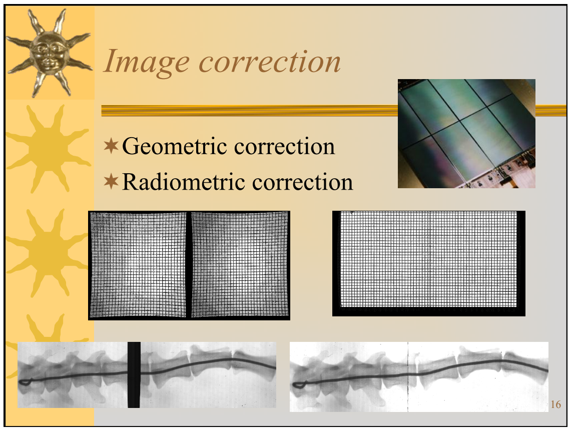


Image correction

- ✶Geometric correction
- ✶Radiometric correction

16

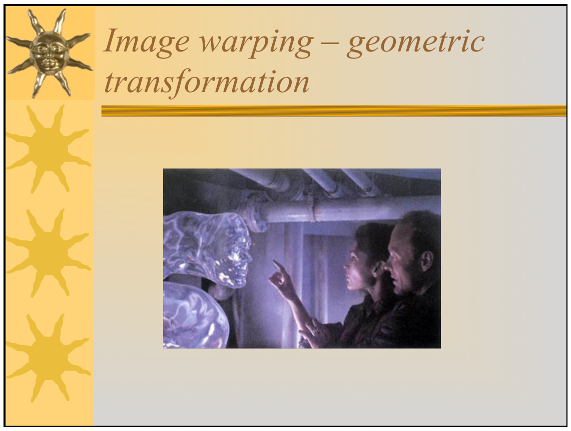


Image warping – geometric transformation

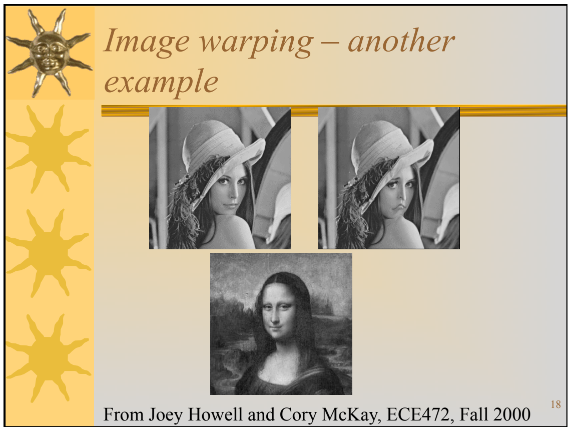


Image warping – another example

From Joey Howell and Cory McKay, ECE472, Fall 2000

18

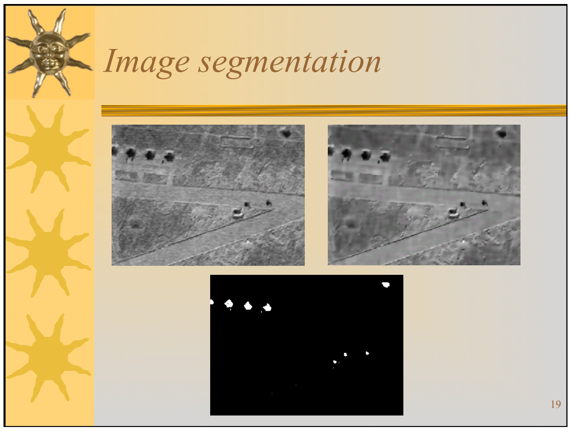


Image segmentation

19

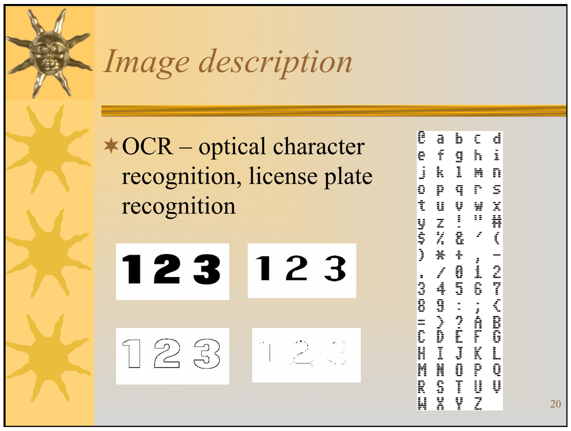
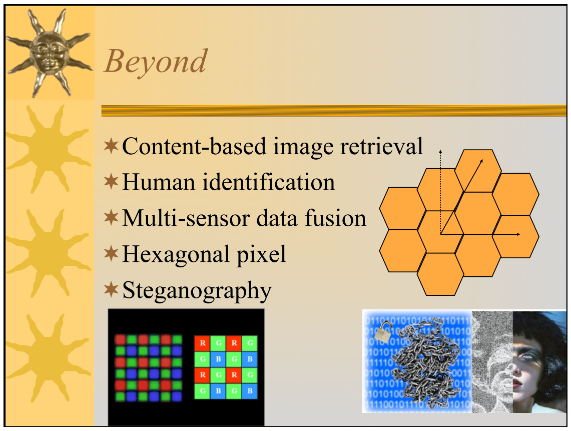


Image description

- OCR – optical character recognition, license plate recognition

1 2 3
1 2 3
1 2 3
1 2 3
@ a b c d
e f g h i
j k l m n
o p q r s
t u v w x
y z ! " #
$ % & ' (
) * + , -
. / 0 1 2
3 4 5 6 7
8 9 : ; <
= > ? A B
C D E F G
H I J K L
M N O P Q
R S T U V
W X Y Z

20



Beyond

- Content-based image retrieval
- Human identification
- Multi-sensor data fusion
- Hexagonal pixel
- Steganography

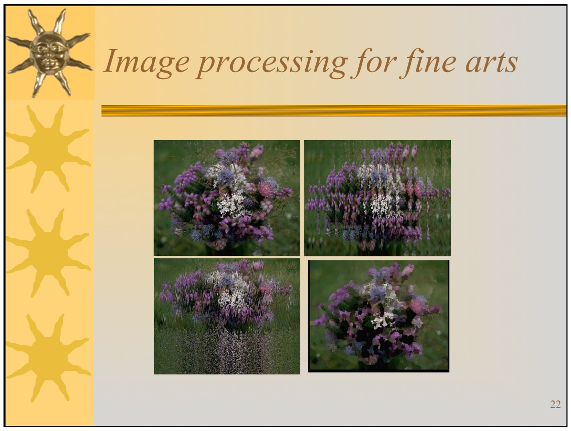
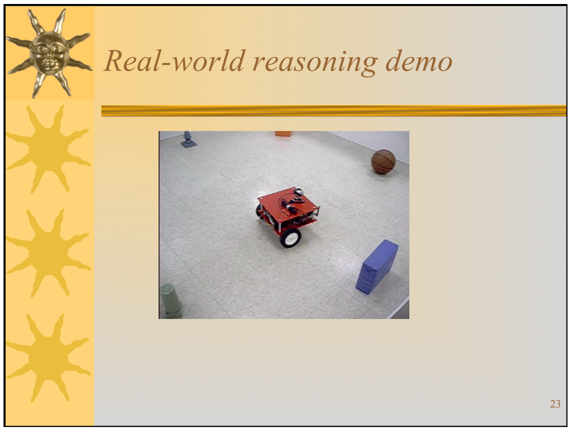


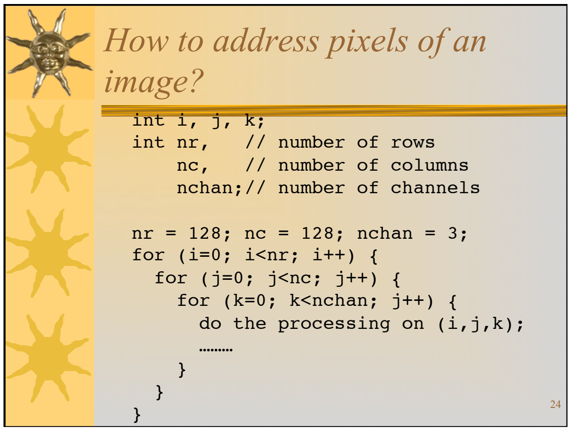
Image processing for fine arts

22



Real-world reasoning demo

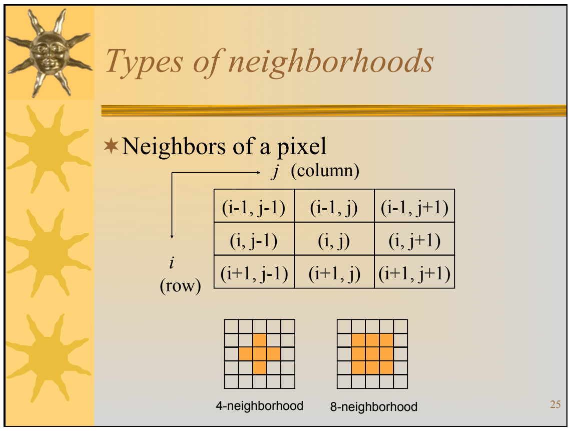
23



How to address pixels of an image?

```
int i, j, k;
int nr,    // number of rows
    nc,    // number of columns
    nchan;// number of channels

nr = 128; nc = 128; nchan = 3;
for (i=0; i<nr; i++) {
  for (j=0; j<nc; j++) {
    for (k=0; k<nchan; j++) {
      do the processing on (i,j,k);
      ………
    }
  }
}
```

24



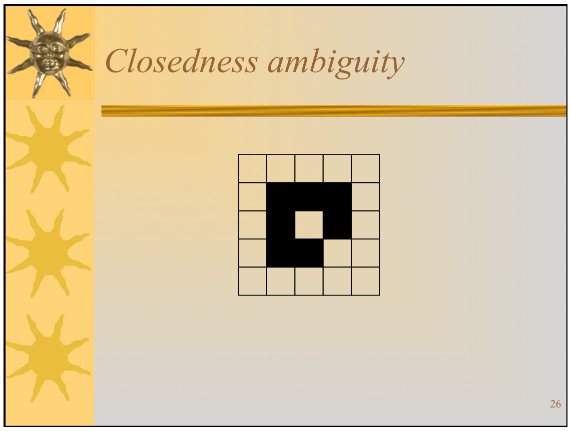
Types of neighborhoods

- Neighbors of a pixel

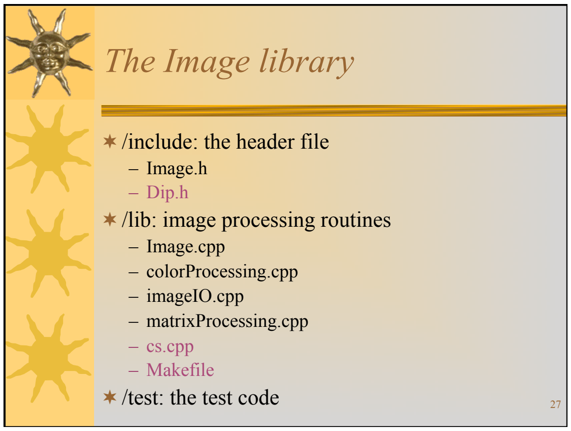
j (column)

i (row)

(i-1, j-1)	(i-1, j)	(i-1, j+1)
(i, j-1)	(i, j)	(i, j+1)
(i+1, j-1)	(i+1, j)	(i+1, j+1)

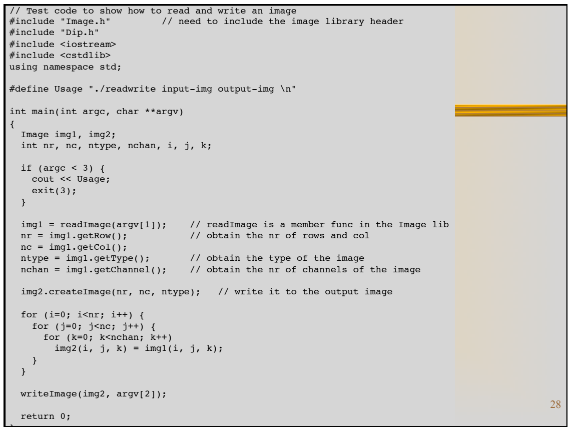


Closedness ambiguity

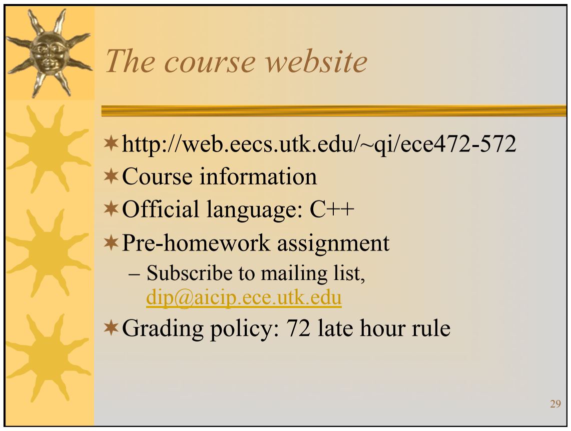


The Image library

- /include: the header file
  - Image.h
  - Dip.h
- /lib: image processing routines
  - Image.cpp
  - colorProcessing.cpp
  - imageIO.cpp
  - matrixProcessing.cpp
  - cs.cpp
  - Makefile
- /test: the test code

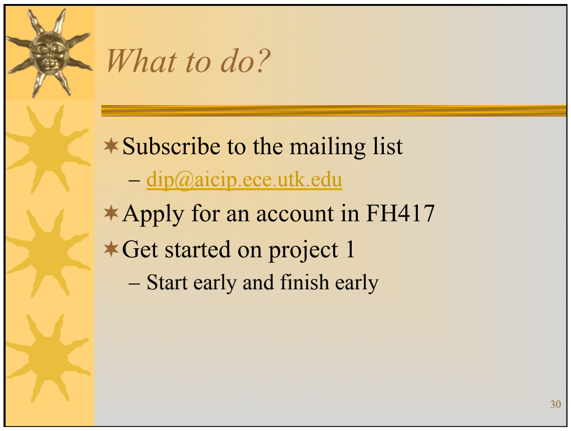


```
// Test code to show how to read and write an image
#include "Image.h"         // need to include the image library header
#include "Dip.h"
#include <iostream>
#include <cstdlib>
using namespace std;

#define Usage "./readwrite input-img output-img \n"

int main(int argc, char **argv)
{
  Image img1, img2;
  int nr, nc, ntype, nchan, i, j, k;

  if (argc < 3) {
    cout << Usage;
    exit(3);
  }

  img1 = readImage(argv[1]);    // readImage is a member func in the Image lib
  nr = img1.getRow();           // obtain the nr of rows and col
  nc = img1.getCol();
  ntype = img1.getType();       // obtain the type of the image
  nchan = img1.getChannel();    // obtain the nr of channels of the image

  img2.createImage(nr, nc, ntype);   // write it to the output image

  for (i=0; i<nr; i++) {
    for (j=0; j<nc; j++) {
      for (k=0; k<nchan; k++)
        img2(i, j, k) = img1(i, j, k);
    }
  }

  writeImage(img2, argv[2]);

  return 0;
```



The course website

- http://web.eecs.utk.edu/~qi/ece472-572
- Course information
- Official language: C++
- Pre-homework assignment
  - Subscribe to mailing list, dip@aicip.ece.utk.edu
- Grading policy: 72 late hour rule



What to do?

- Subscribe to the mailing list
  - dip@aicip.ece.utk.edu
- Apply for an account in FH417
- Get started on project 1
  - Start early and finish early