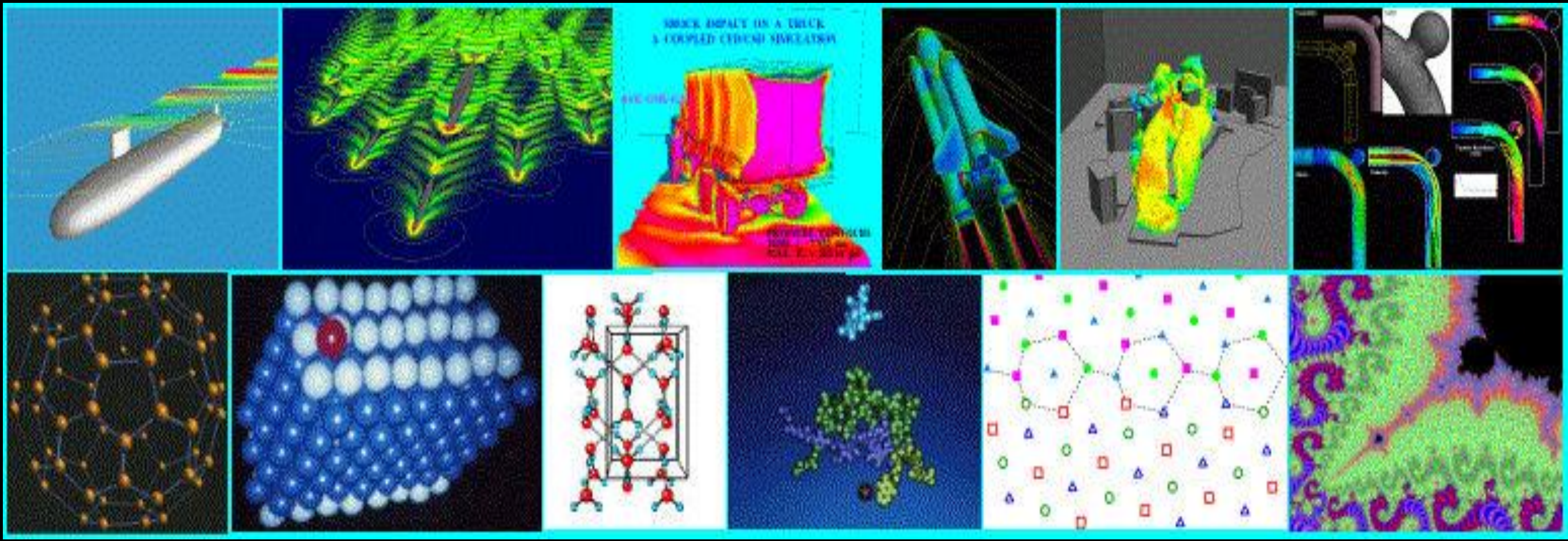
# Computing for Scientists
# Visualization (VIS)
# (Nov. 12, 2013 – Dec. 05, 2013)



Jie Zhang



CDS 130 - 001
Fall, 2013

# Where are we now?

## The Tool: MATLAB (remaining chapters)

- **CHAPTER 8:** Matrices (Arrays) and Matrix Operations
- **CHAPTER 9:** Iteration II: Double Nested FOR Loops (DNFL)
- **CHAPTER 10:** Conditionals: IF Statements

**Section 0: Introduction & Syllabus**

**Section 1. Computer Fundamentals**

**Section 2. Scientific Simulation**

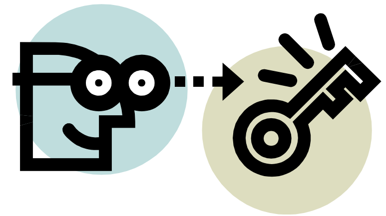
**Section 3. Visualization**

**We are here!**

**Section 4: Data Analysis (not covered)**

**Section 5: Ethics**

# Visualization (VIS)



1. Introduction

2. Image as a 2-D Array

3. Color and Colormap


# Section 3
# Chapter 1

# Introduction

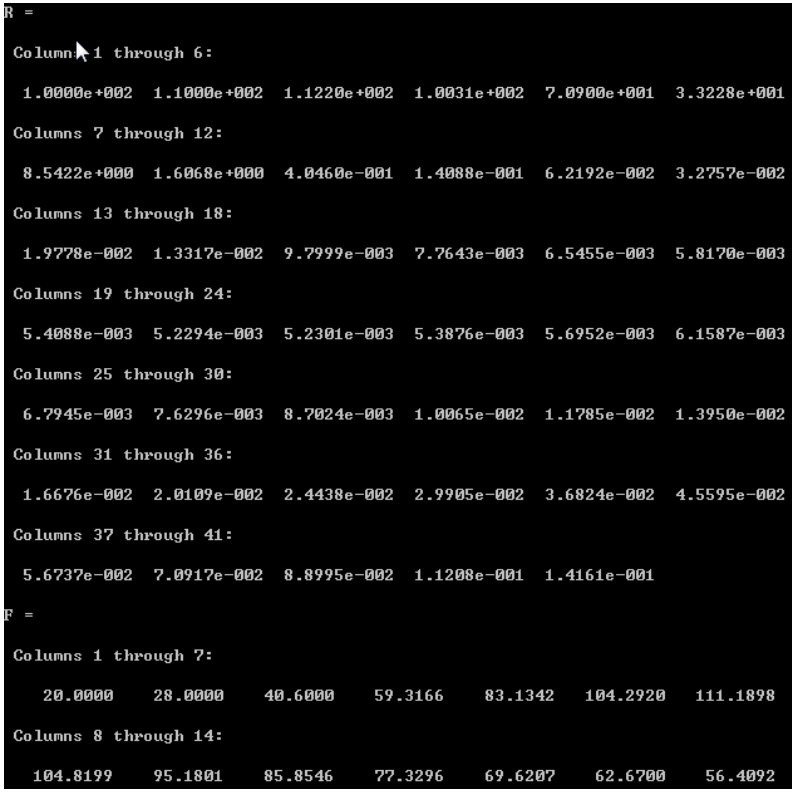
# (Nov. 12, 2013)

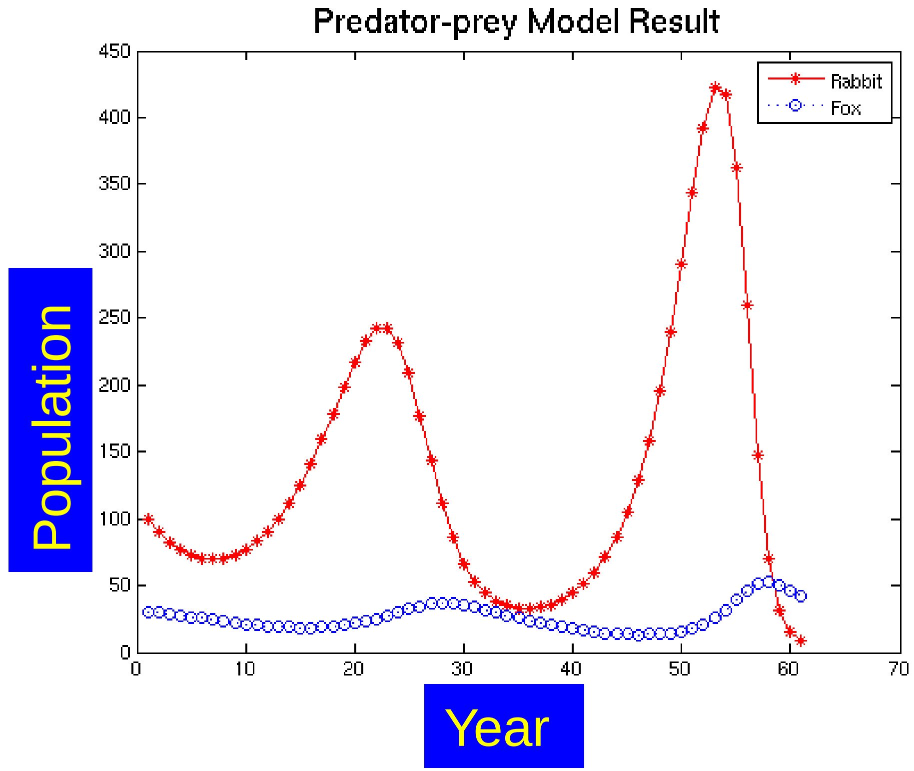
# Picture

**A Picture Is Worth a Thousand Words**

Cave Painting: the dawn of civilization

# The Line Plot



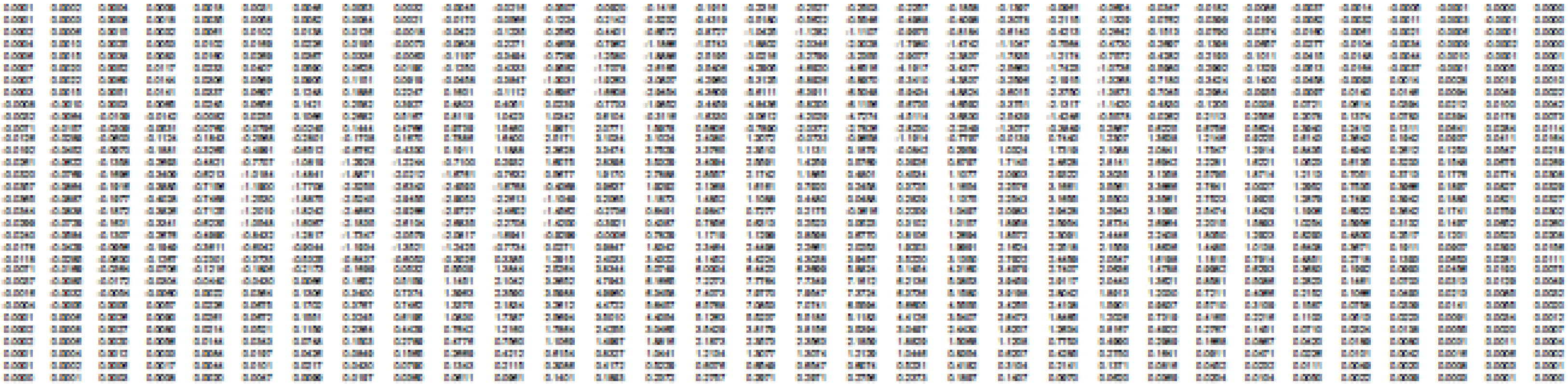




Numeric output of the Predator-Prey Model

The line plot of the Predator-Prey Model

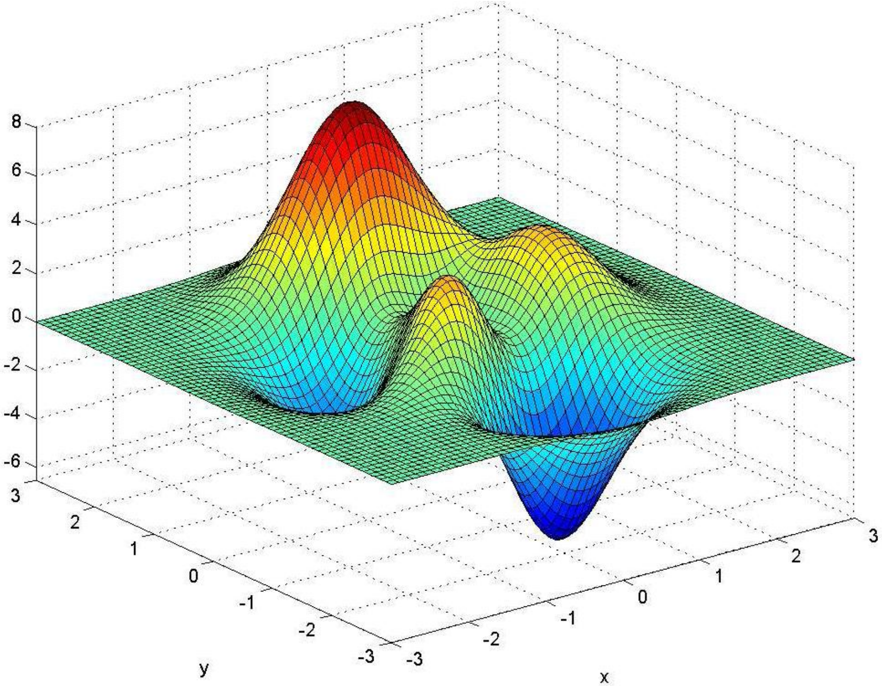
**A graph is worth a thousand numbers**

# 2-D Data and 3-D data



**A 2-Dimension data array,**

**which is the same as ……**



# Cognition Visualization

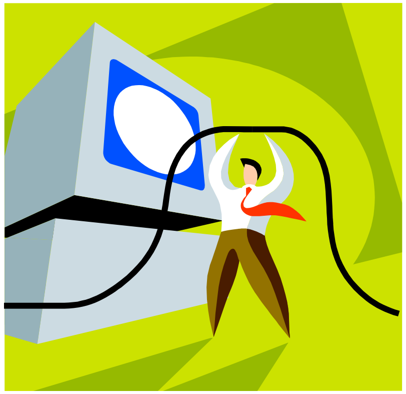
**Visualization is a cognitive process performed by humans in forming a mental image of a domain space**

Human Cognition

- **Most effective way human takes in information, digest information, making decision, and being creative.**
- **The most important among the five senses**

# What is Scientific Visualization

**Scientific visualization is an interdisciplinary branch of science primarily concerned with the visualization of three dimensional phenomena (meteorological, medical, biological etc) where the emphasis is on realistic rendering of volumes, surfaces, illumination sources with a dynamic (time) component.**



At Information Age

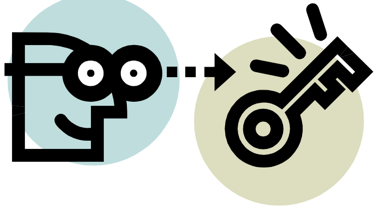
**Friendly (2008), also**

**http://en.wikipedia.org/wiki/Scientific_visualization**

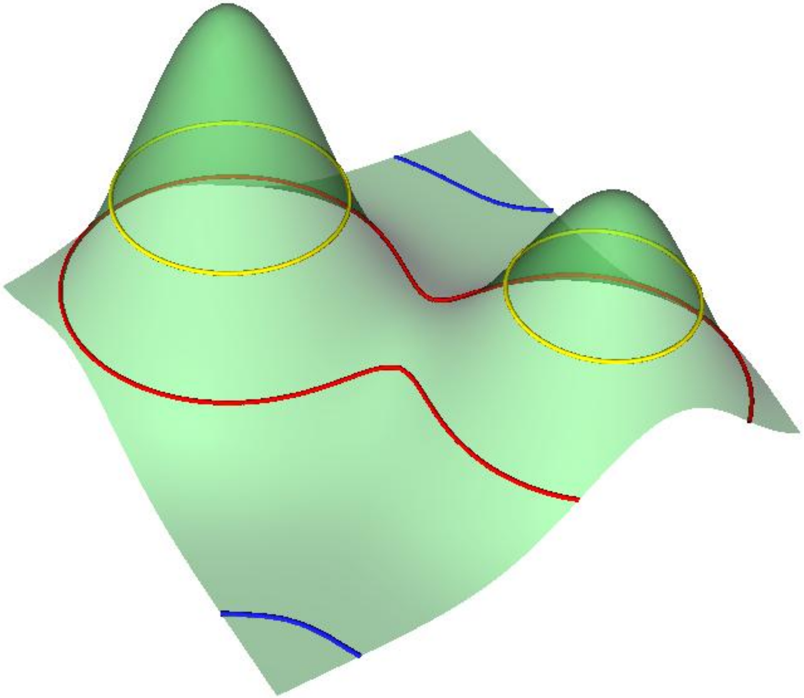
# Goals of Scientific Visualization

- Provide scientific insight

- Effective communication

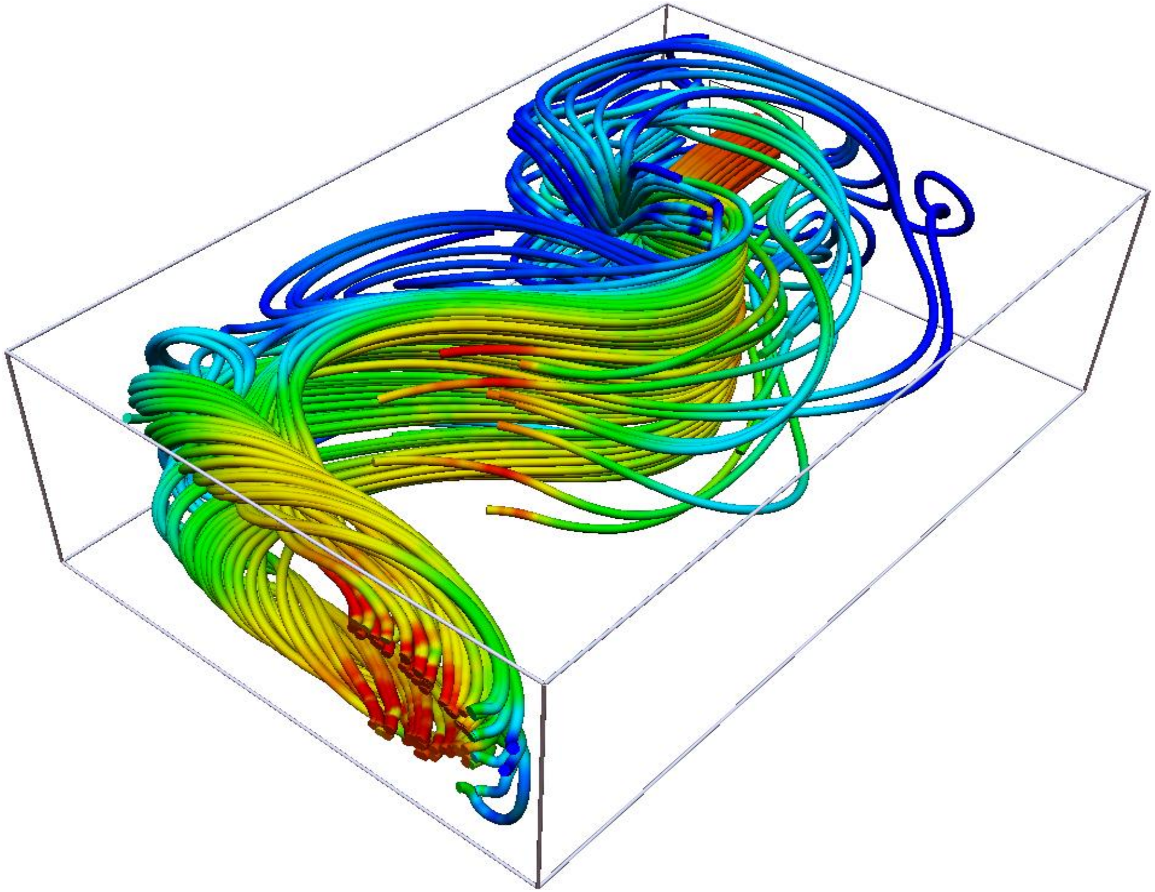


# Example: Scalar Data



•Height plot of a 2-D data, and the contour lines
•In topology, the data are of 2-D: a curved surface
•The data are mapped into 3-D geometric space in computer memory
•The data are then rendered onto a 2-D visual plane on a screen

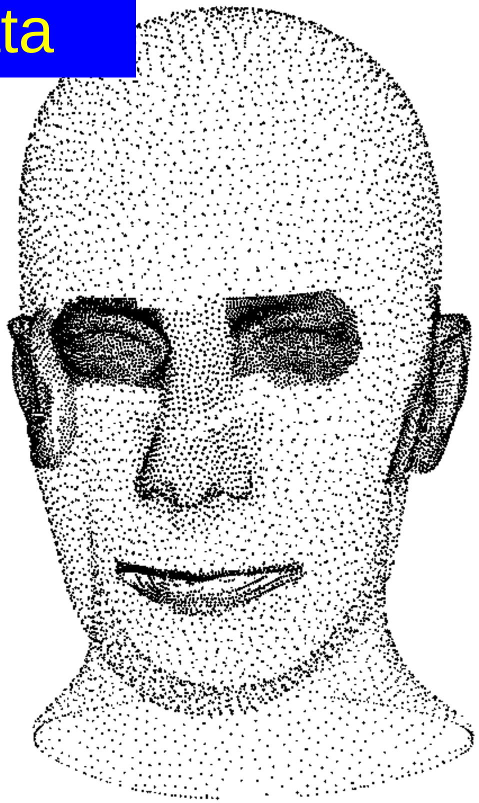
# Example: Vector Data



Stream tubes: show how water flows through a box with an inlet at the top-right and an outlet at the lower-left of the box; the data are (1) 3-D volume, (2) vector

# Example: Unstructured Data

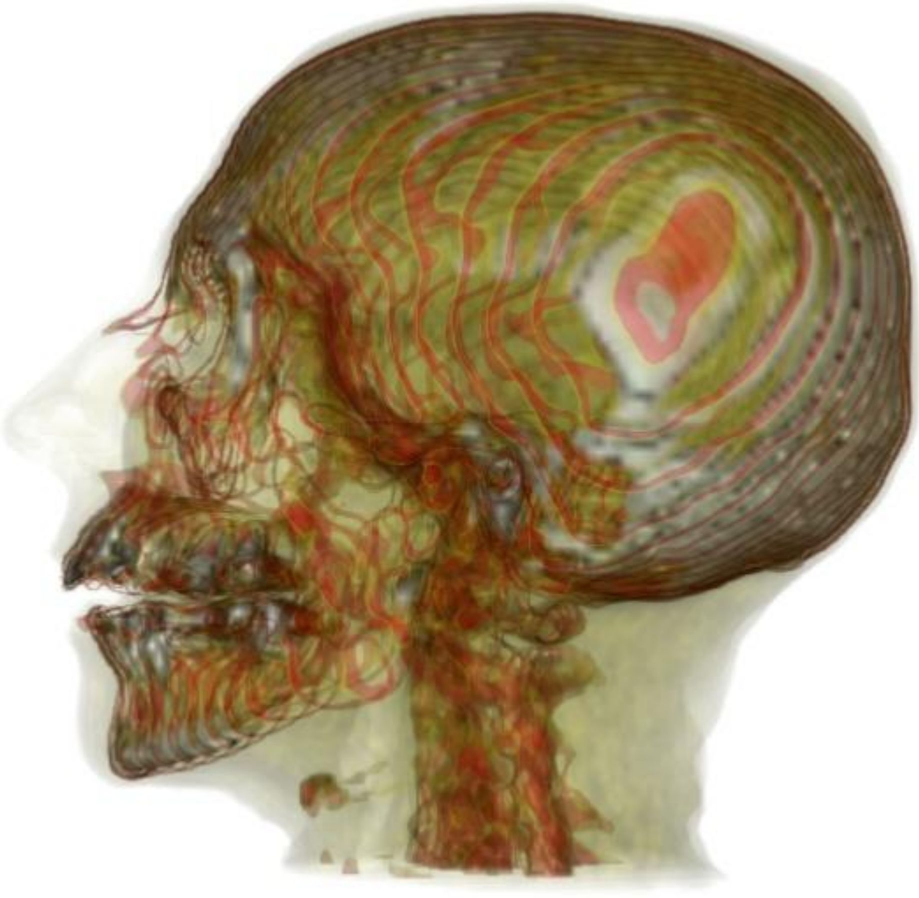
Sampled data



Rendered data

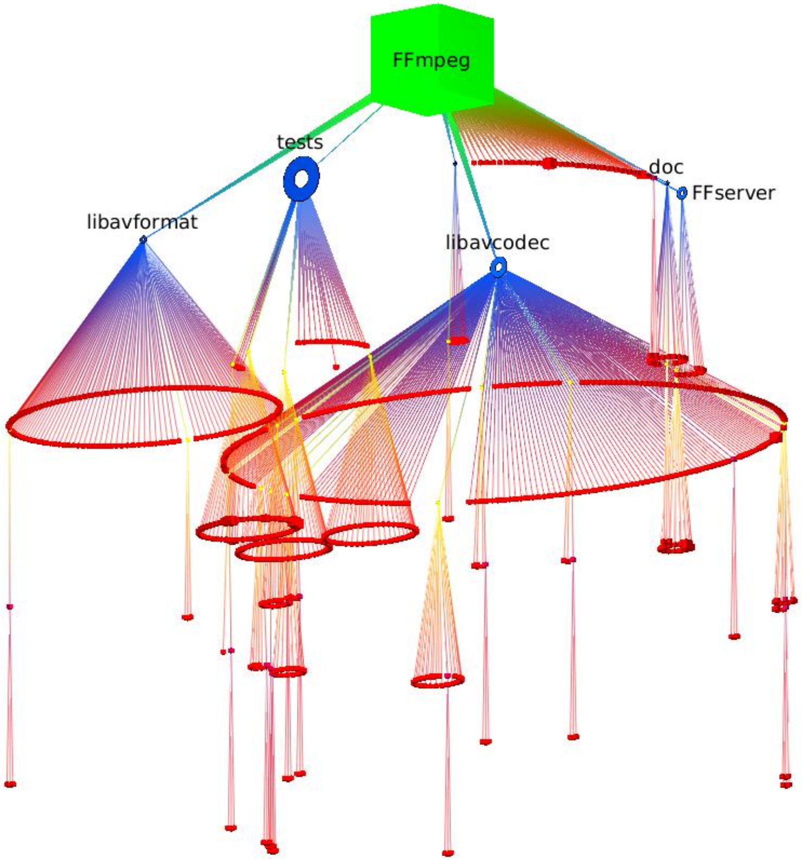
Scattered Point Cloud and Surface Reconstruction.
The data are a set of sampled data points in 3-D space, and the distribution of data is unstructured; need to reconstruct the surface from the scattered points

# Example: Volume Visualization



3-D data of human head. Instead of showing a subset (e.g, a slice as in CT images), the whole 3-D data are shown at once using the technique of Opacity Transfer Function

# Example: Information Visu.




- The file system of FFmpeg, a popular software package for encoding audio and video data into digital format
- Information, such as name and address, can not be interpolated; the visualization focuses on the relations.

An attempt at visualizing the Fourth Dimension: Take a point, stretch it into a line, curl it into a circle, twist it into a sphere, and punch through the sphere.

--Albert Einstein--

**Difficulty in visualizing the fourth dimension**

**(Nov. 12, 2013 Stops Here)**

**(Dec. 03, 2013 Starts Here)**

# Review: IF and DNFL

**Question**: What final value of `a` is printed out?

```
a = 0;
c = 0;
for m = 1:3
    for n = 1:3
        if(m <= n)
            c = c + 1;
        end
        if(c > a)
            a = a + 1;
        end
    end
end
a
```

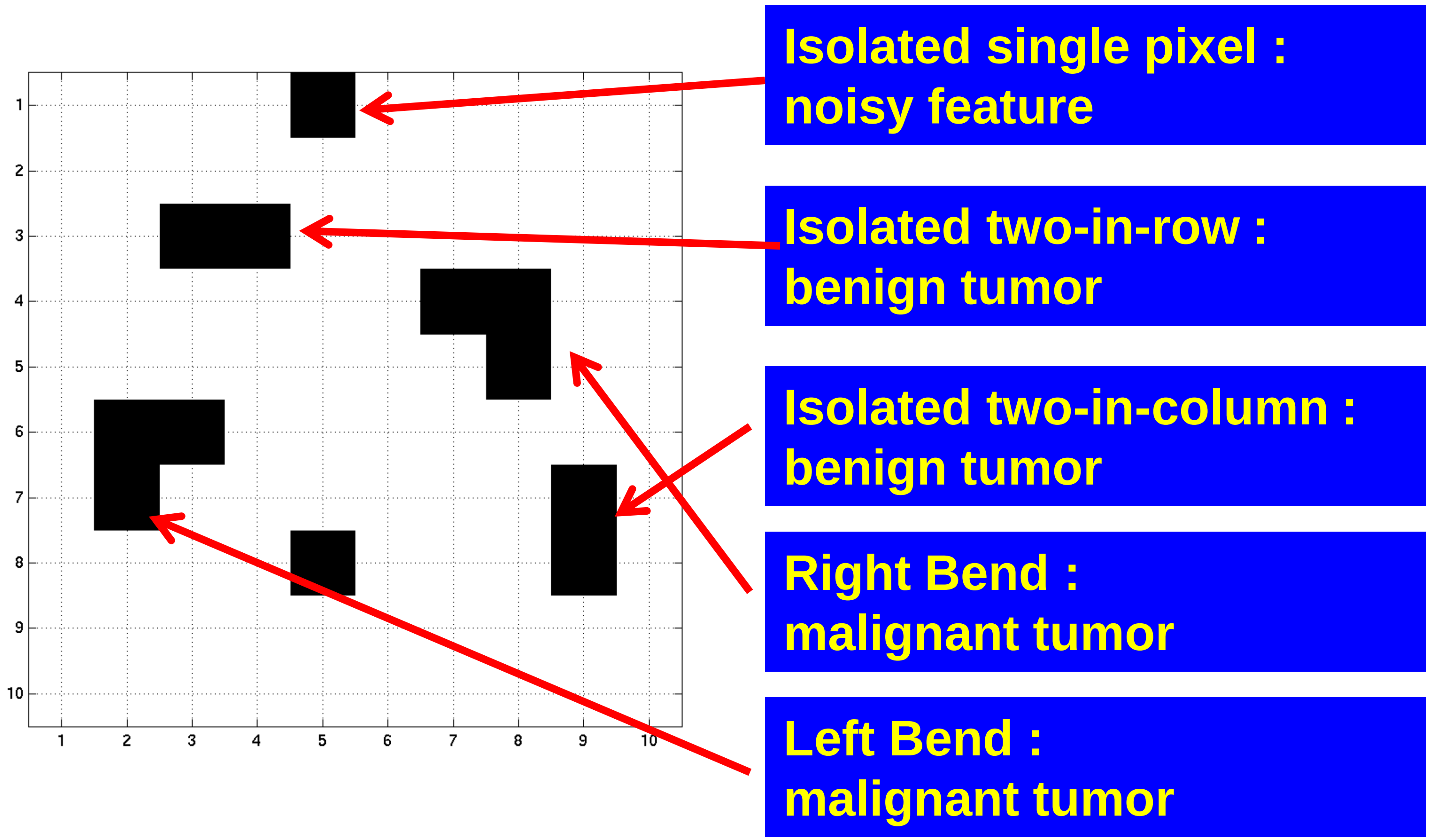

# Section 3
# Chapter 2

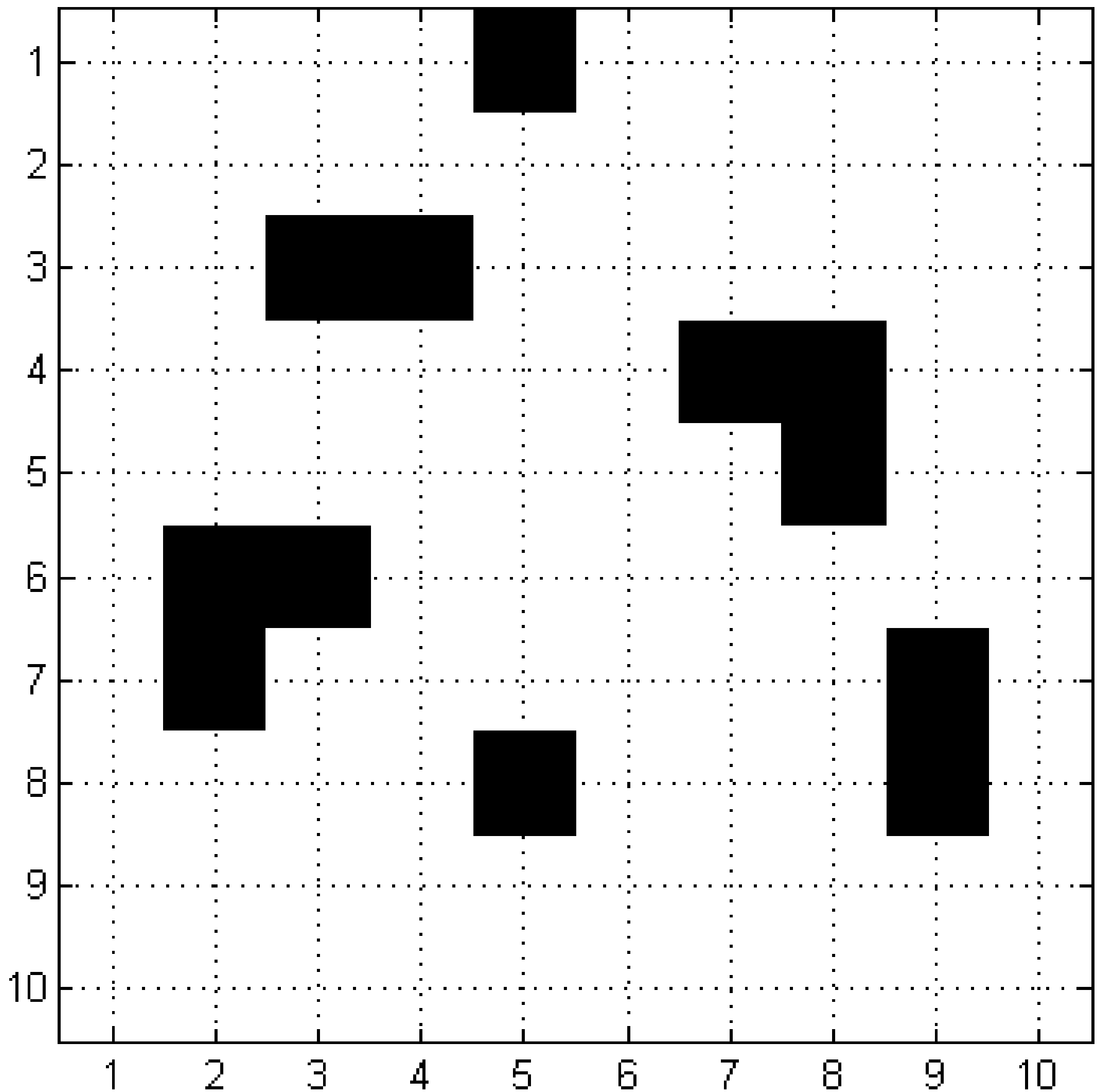
# Image As a 2-D Array

**(Dec. 03, 2013)**

# Exercise: Tumor Finder

- **Implement a MATLAB program to find tumors in an X-ray image, a 10 X 10 2-D array, or 100 pixels.**
- **A positive pixel has a value of 1, indicated by black**
- **A negative pixel has a value of 0, indicated by white**




1
2
3
4
5
6
7
8
9
10
1
2
3
4
5
6
7
8
9
10

# Exercise: Tumor Finder

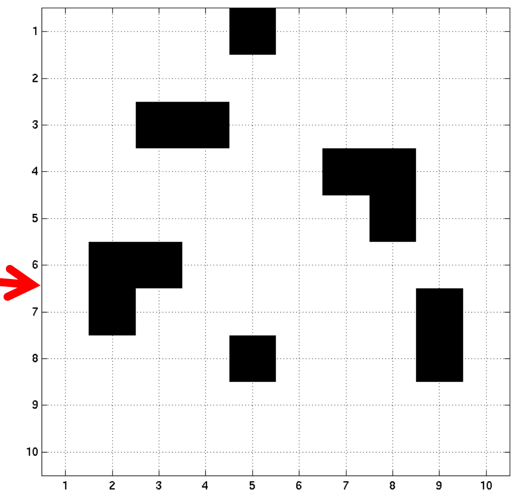
- **How the image is created ("tumor1.m")**

```
tum(10,10)=0  %initiate the tumor variable
tum(1,5)=1    %define a noise feature
tum(8,5)=1    %define another noise featuer
tum(3,3)=1    %define a benign tumor
tum(3,4)=1
tum(7,9)=1    %define a begin tumor
tum(8,9)=1
tum(6,2)=1   %define a malignant tumor (left bend)
tum(6,3)=1
tum(7,2)=1
tum(4,7)=1   %define another malignant tumor (right bend)
tum(4,8)=1
tum(5,8)=1

mycolormap=[1,1,1;0,0,0]   %define a color map
colormap(mycolormap)       %call my color map
imagesc(tum)               %show the image of the 2-D data
axis square                %square axis
grid on                    %show data grid
set(gca,'XTick',[1:10],'YTick',[1:10])
```

# Exercise: Tumor Finder

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |




**2-D data**

**Image**

**In real applications, the doctor takes the digital image, and analyze it.**

# Exercise: Tumor Finder

Question:
Use a MATLAB program to find how many positive pixels (with value of 1) in the image?

**To save your time**

(1) Download the “tumor1.m” file from class website at
(2) Open this file in MATLAB, and add additional program code to it

Algorithm: (scan the image, count all positive pixels)

For all (i,j)
  count image(i,j) == 1

# Exercise: Tumor Finder

Answer:

-------------------------------------------

```
%count how many positive pixels in the image (2-D array)
    count_pos=0;
    for i=[1:10]
        for j=[1:10]
            if (tum(i,j) == 1)
                count_pos=count_pos+1;
            end
        end
    end

    disp('Number of Positive Pixels=')
    count_pos
```

-------------------------------------------------

# Exercise: Tumor Finder

Question:

Use a MATLAB program to find how many two-in-row in the image?

Algorithm:

For all (i,j)

count image(i,j) == 1 and image(i,j+1) ==1

# Exercise: Tumor Finder

Answer:

```
----------------------------------------------
%count how many two-in-row
    count_two_row=0;
    for i=[1:10]
        for j=[1:9]
            if (tum(i,j) == 1 && tum(i,j+1)==1)
                count_two_row=count_two_row+1;
            end
        end
    end

    disp('Number of two-in-row=')
    count_two_row
----------------------------------------------
```

# Exercise: Tumor Finder

Question:

Use a MATLAB program to find how many two-in-column in the image?

Algorithm:

For all (i,j)

count image(i,j) == 1 and image(i+1,j) ==1

# Exercise: Tumor Finder

Answer:

---------------------------------------------

```
%count how many two-in-column
     count_two_col=0;
     for i=[1:9]
         for j=[1:10]
            if (tum(i,j) == 1 && tum(i+1,j)==1)
                count_two_col=count_two_col+1;
            end
         end
     end

     disp('Number of two-in-col=')
     count_two_col
```

---------------------------------------------------

# Exercise: Tumor Finder

Question:
Use a MATLAB program to find how many left bend in the image?

Algorithm:
For all (i,j)
count image(i,j) == 1 and image(i,j+1) ==1 and image(i+1,j) == 1

# Exercise: Tumor Finder

Answer:

-----------------------------------------------

```
%count how many left bend
     count_left_bend=0;
     for i=[1:9]
         for j=[1:9]
            if (tum(i,j) == 1 && tum(i,j+1)==1 && tum(i+1,j)==1)
                count_left_bend=count_left_bend+1;
            end
         end
     end

     disp('Number of left_bend=')
     count_left_bend
```

-----------------------------------------------------

# Exercise: Tumor Finder

Question:
Use a MATLAB program to find how many right bend in the image?

Algorithm:

For all (i,j)
count image(i,j) == 1 and image(i,j+1) ==1 and image(i+1,j+1) == 1

# Exercise: Tumor Finder

Answer:

```
-------------------------------------------------
%count how many right bend
     count_right_bend=0;
     for i=[1:9]
         for j=[1:9]
            if (tum(i,j) == 1 && tum(i,j+1)==1 && tum(i+1,j+1)==1)
                count_right_bend=count_right_bend+1;
            end
         end
     end

     disp('Number of right_bend=')
     count_right_bend
-------------------------------------------------
```

# Exercise: Tumor Finder

Question:
Use a MATLAB program to find how many malignant tumors in the image?

Algorithm:

Number of Left Bend + Number of Right Bend

# Exercise: Tumor Finder

Answer:

---------------------------------------------

```
%find the number of malignant tumors
    count_malig=count_left_bend + count_right_bend;
    disp('Number of malignant tumors=')
    count_malig
```

---------------------------------------------------

# Exercise: Tumor Finder

Question:
Use a MATLAB program to find how many benign tumors in the image?

Algorithm:

Number of Two-in-row + Number of Two-in-column – Number of Malignant Tumor X 2

# Exercise: Tumor Finder

Answer:

```
-------------------------------------------
count_benign=count_two_row + count_two_col -
    count_malig*2;

    disp('Number of benign tumors=')
    count_benign
-------------------------------------------------
```

# Exercise: Tumor Finder

**Refer to “tumor2.m” for the final program**

# The End of Chapter 2

**(December 03, 2013 Stops Here)**

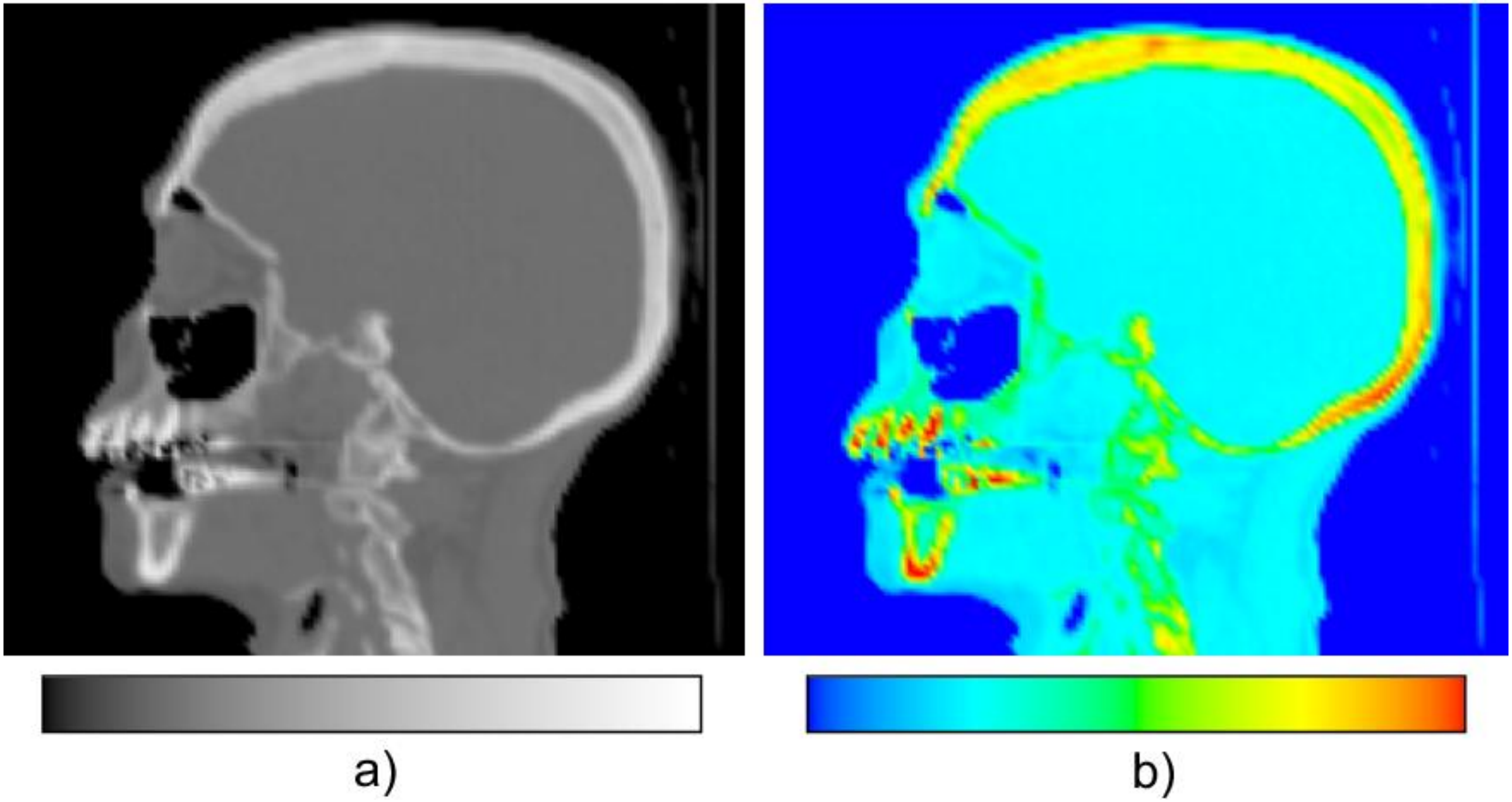
**(December 05, 2013 Starts Here)**


# Section 3
# Chapter 3

# Color and Colormap

**(Dec. 05, 2013)**

# Colormap



Gray-scale Colormap

Rainbow Colormap

# Color

**Question:**

**Do scientific data have intrinsic colors?**

**For example, hourly temperature reading at Fairfax?**

**For example, wind speed in a tornado?**

# Color

**Answer:**

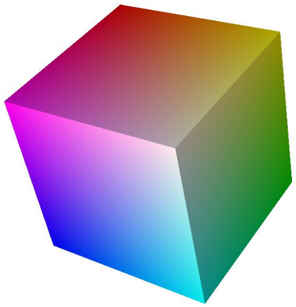
- **Most scientific data concern about value, strength, or intensity. They have no color**

- **Color in scientific images are called pseudo-color, which use the color in the visualization as a cue to indicate the value of the data**

- **For example, red for high temperature, and blue for low temperature**

# Computer Color

- A special type of vector with dimension c=3
  - The value of a color = (R, G, B)



- RGB system: convenient for computer
  - R: red
  - G: green
  - B: blue

- HSV system: intuitive for human
  - H: Hue (the color, from red to blue)
  - S: Saturation (purity of the color, or how much is mixed with white
  - V: Value (the brightness)

# Computer Color

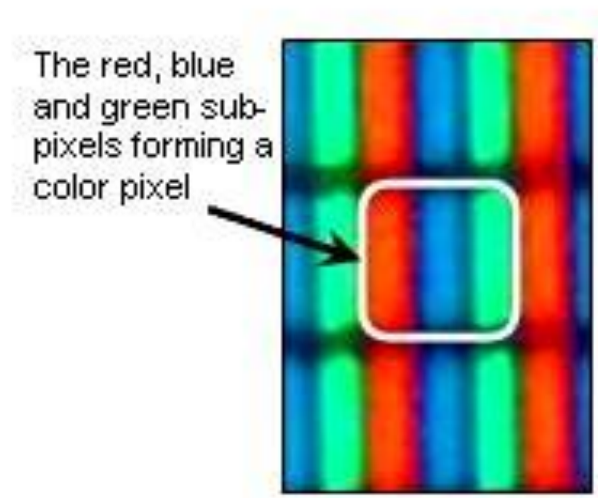
- A pixel of a LCD monitor contains three sub-pixels: R, G, B
- Each sub-pixel displays the intensity of one primary color
- Modern LCD monitor uses 1 Byte or 8 bits to determine the intensity of the primary color
    - Each primary color has 255 intensity levels
- The number of colors of a LCD monitor

= 255 (red) x 255 (green) x 255 (blue)

= 16,581,375

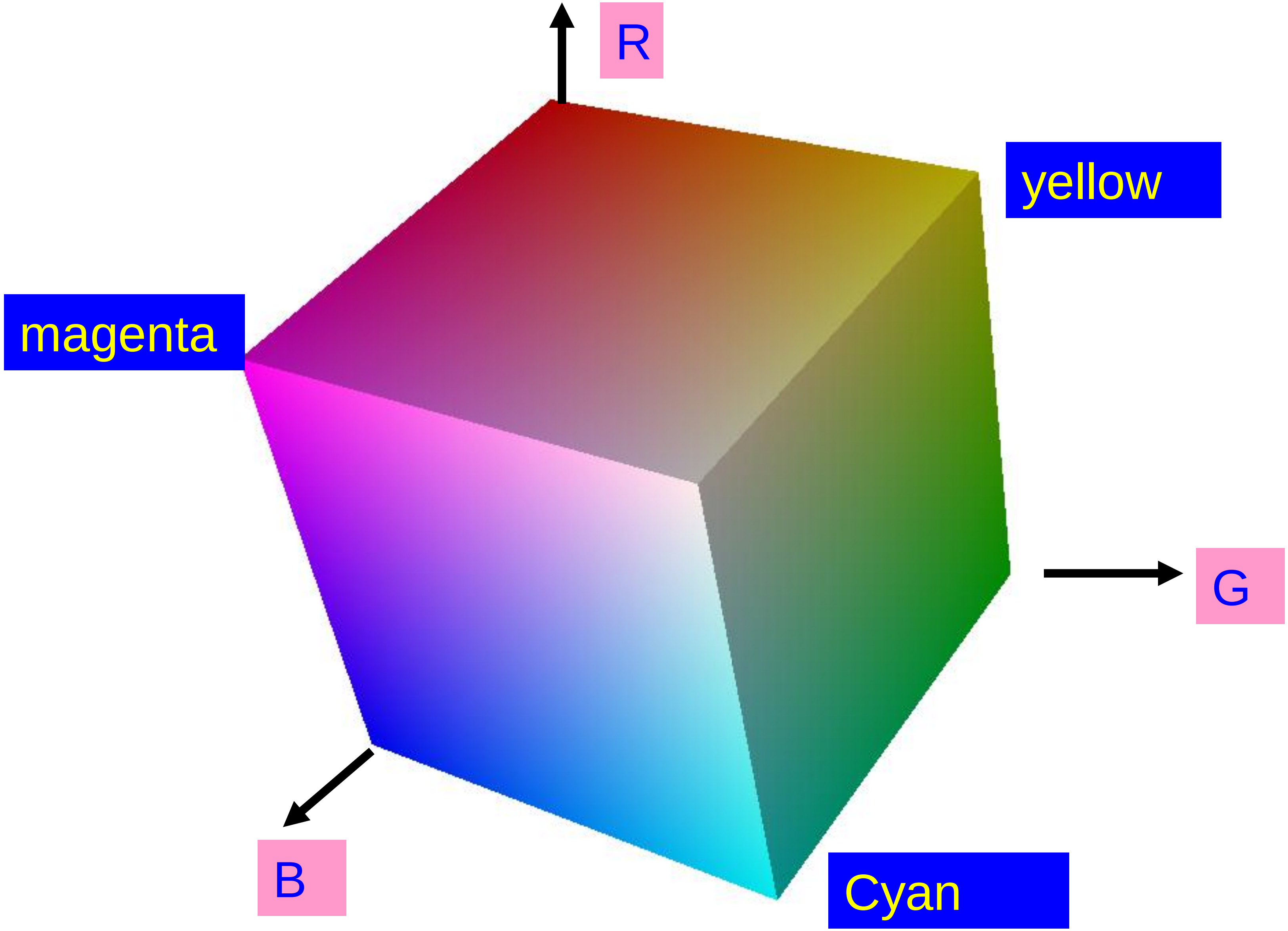
Or 16 million different colors

- The monitor shows true color
    - Human eyes discern 10 million color




Close-up of an LCD display panel

# RGB Cube



# RGB System

- Every color is represented as a mix of “pure” red, green and blue colors in different amount
- Equal amounts of the three colors give gray shades
- RGB cube
  - main diagonal line connecting the points (0,0,0) and (1,1,1) is the locus of all the grayscale value
- Color (0,0,0): dark
- Color (1,1,1): white
- Color (0.5,0.5,0.5): gray level between dark and white
- Color (1,0,0):  Red
- Color (0,1,0): Green
- Color (1,1,0): ?
- (1,0,1): ?
- (0,1,1): ?

# IMAGE method

**How to visualize a 2-D data as an image?**
**-Use "imagesc" method in Matlab**

Question: Visualize the following 2-D array

```
myimage = [1   0   1   0   1;
           0   1   0   1   0;
           1   0   1   0   1;
           0   1   0   1   0;
           1   0   1   0   1]
```

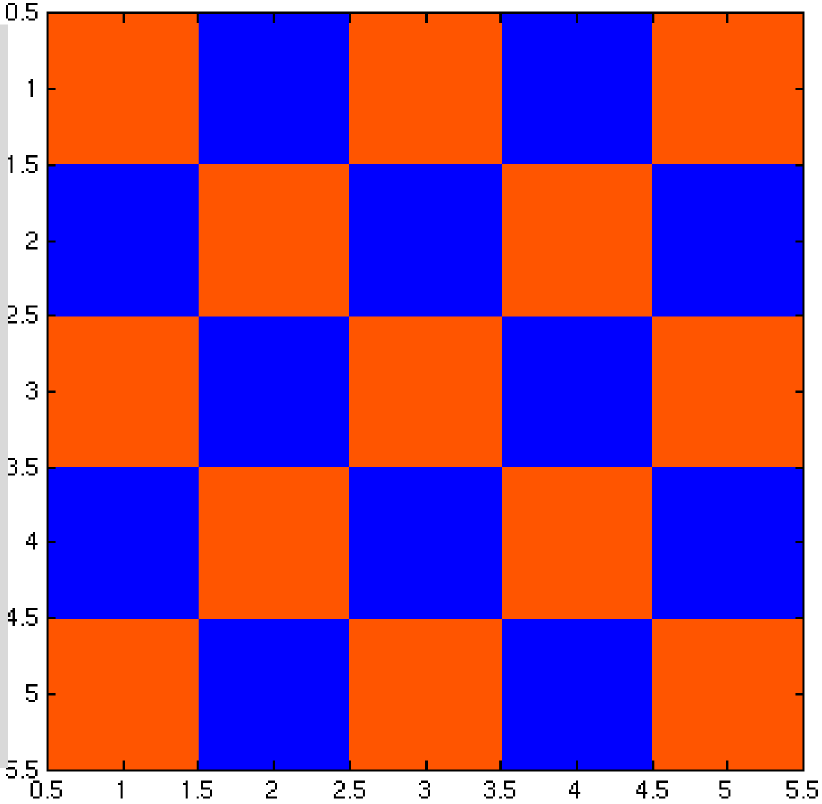
# IMAGE method – “imagesc”

Answer:

```
myimage = [1  0  1  0  1;
           0  1  0  1  0;
           1  0  1  0  1;
           0  1  0  1  0;
           1  0  1  0  1]

%visualize the 2-D data array
imagesc(myimage)

%make the image in a square shape
Axis square
```



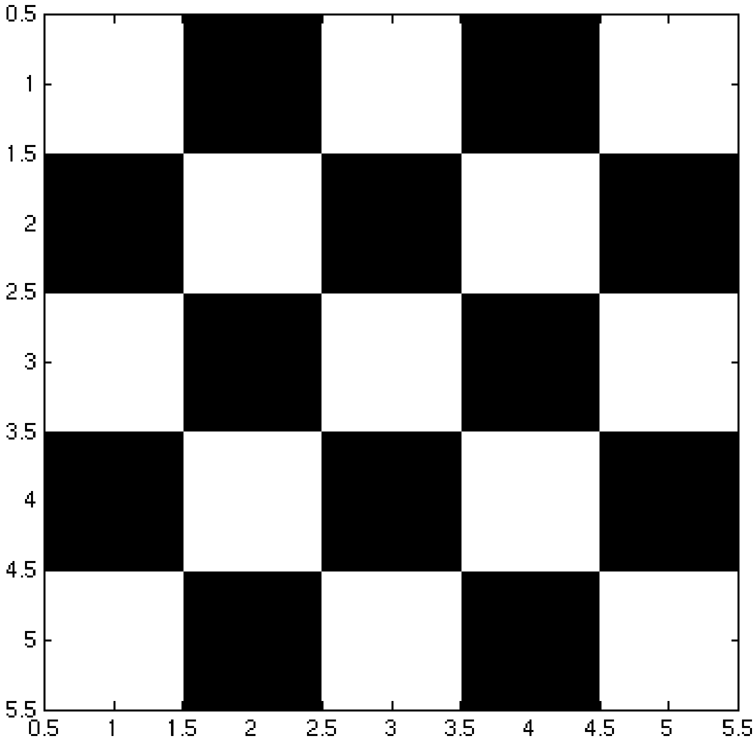
# Colormap method –”colormap”

**How to change the color to black and white?**

Answer:

%define a colormap with only two colors
mycolormap=[0,0,0; 1,1,1]

colormap(mycolormap)



# Colormap method –"colormap"

Question:

For the colormap defined below:

```
>mycolormap = [   0,      0,      0;
                0.5,  0.5,  0.5;
                  1       1       1;
                  1       0       0;
                  1       1       0;
                  0       0      1]
```

How many colors in total are defined in mycolormap? What are these colors?

# Colormap method –"colormap"

Answer:

Six colors in total. They are (1) black, (2) gray, (3) white, (4) red, (5) magenta, and (6) blue, respectively.

# The End of Chapter 3

# The End of Section 3:

# Visualization

# (Dec. 05, 2013 ends Here)