CSE 7

Introduction to Programming in MATLAB

Chapters 3.3-3.5, 5.1.5, and 13.2
Images and Plotting
How Computers Store Images

• Images stored as a mosaic of tiny squares*
• Each square is exactly one color
  – Square is called a **pixel**
• Like the ancient art form of tile mosaics, or the melting beads kits kids play with today

* Vector graphics are a somewhat different method of storing images, but we’ll set those aside for now—and even vector images are displayed as a mosaic of pixels.
Square Pixels Are Inherently Rough/ Jagged

• Hard to make smooth edges and curves with only square tiles!
  – If more and smaller tiles used, the smoother (less *pixelated*) the image
To a computer, images are a matrix of pixels.

But remember: to a computer, a color is made of three numbers: a value each for Red, Green, Blue.
In MATLAB, RGB images are matrices of size [Height X Width X 3].

Note: only the circled numbers are correct for the colors shown, the rest are made up.
Image Matrices

- 3-layered image matrices -- Read ‘rainbow.jpg’ into im
- Subset im to im2 – e.g. (155:184, 145:164, :)
- 1 layer of an image – Get the red layer in im2
- Boolean Matrix – Check if the red layer is strong (>200)

Can load all of them as an image in MATLAB!

How can a boolean matrix change some parts of an image?
“Green screen” special effects

• Actors (or objects) filmed in front of a screen with a very specific shade of **green**

• Later, a computer program checks the color of each **pixel** of the movie
  - **Pixels** matching that specific shade of **green** are replaced with **pixels** from another image/movie
  - **Pixels** not matching that specific shade of **green** are left alone
“Green screen” special effects

- Actors wear green to be invisible in the final movie
  - Two men on the left are moving a puppet
  - Man on the right, if he doesn’t put his green shirt back on, will appear to be a floating torso
Matrix summary

- Individual items: **elements**
  - Denoted by a variable with two subscripts e.g. \( a_{i,j} \)
- Addition and subtraction needs matrices of the same size
- Multiplication requires number of columns in first matrix equals the number of rows in second matrix
- One horizontal line: **row vector**
- One vertical line: **column vector**
- Applications: graphing, imaging, science, electronics

```
x = [ 4 : .2 : 20 ];
y = [ 4 : .2 : 20 ]*4;
plot ( x );
hold on;
plot ( y, 'r' );
ylabel ( 'y' );
xlabel ( 'x' );
legend ( '[4:.2:20]', '[4:.2:20]*4', 'Location', 'Northwest' );
title ( 'My first plot' );
```
**subplot function**

- Creates matrix (or vector) in a Figure Window so multiple plots are viewed at once
- For matrix $m \times n$, `subplot(m,n,i)` refers to element $i$ ($i$ is integer in range 1 to $m*n$)
- Elements in Figure Window numbered *row-wise*

<table>
<thead>
<tr>
<th>Plot 1</th>
<th>Plot 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plot 3</td>
<td>Plot 4</td>
</tr>
</tbody>
</table>

```plaintext
>> subplot ( 2, 2, 1 )
>> subplot ( 2, 2, 2 )
>> subplot ( 2, 2, 3 )
>> subplot ( 2, 2, 4 )
```
Other plot Functions

- **clf** clears the figure window
- **figure** creates a new figure window (can # figure(2))
- **hold** toggle that keeps current graph in figure window
- **legend** displays strings in a legend
- **grid** displays grid lines
- **bar** bar chart

**Note:** make sure to use enough points to get a “smooth” graph
myVector = rand( 1000, 1 );
hist( myVector);