

CME 192: Introduction to MATLAB

Lecture 7

Stanford University

February 5, 2019

Outline

Review

Images

Movies

Review

Lecture 6

- ▶ Ordinary Differential Equations
- ▶ Solving Equations
- ▶ Function Approximation

Outline

Review

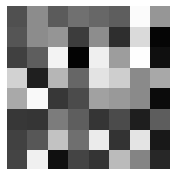
Images

Movies

Grayscale Images

- ▶ a matrix of pixels representing brightness
- ▶ dark to bright represented by range
 - integers: 0 - 255 (8-bit)
 - doubles: 0.0 - 1.0
- ▶ "black & white" means just black or white

80	140	91	112	103	94	251	150
85	140	149	60	141	46	218	5
64	94	243	3	239	157	254	15
210	31	182	102	227	206	138	168
169	249	55	75	161	153	138	11
55	53	124	94	53	75	30	93
65	87	194	111	235	51	228	24
68	240	16	68	52	187	132	39



Color Images

- ▶ various encodings
- ▶ with transparency (e.g. PNG) or without (e.g. JPEG)
- ▶ usually 3 colors + 1 (optional) transparency
- ▶ e.g. RGB: Red, Green, Blue
- ▶ colors represented by adding primary colors

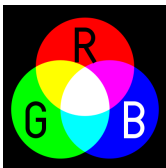


Figure:
Additive Colors

Red

$$\begin{bmatrix} 116 & 194 & 154 \\ 190 & 250 & 185 \\ 14 & 88 & 225 \end{bmatrix}$$

Green

$$\begin{bmatrix} 132 & 165 & 84 \\ 115 & 7 & 156 \\ 104 & 27 & 122 \end{bmatrix}$$

Blue

$$\begin{bmatrix} 86 & 117 & 197 \\ 93 & 207 & 187 \\ 21 & 156 & 249 \end{bmatrix}$$



Reading Images from Files

- ▶ image format decoding isn't easy
 - MATLAB provides tools
 - in other programming languages, search for a helper library
- ▶ binary image encoding takes less space than matrix representation

10,000 × 10,000 pixels \implies 300 MB of memory

```
1 % rgb image
2 I = imread('cherry.jpg');
3 % red layer
4 R = I(:, :, 1);
5 % green layer
6 G = I(:, :, 2);
7 % blue layer
8 B = I(:, :, 3);
```

Manipulating Images

- ▶ Treat like matrices
 - interpolate
 - filter
 - logically index
- ▶ Image specific functions

<code>rgb2gray</code>	three layered RGB image to single 2D layer of grayscale
<code>imresize</code>	resizes image by a scalar scaling factor
<code>im2double</code>	coverts from integer (0 - 255) to double (0.0 - 1.0) representation
<code>im2uint8</code>	coverts from double (0.0 - 1.0) to integer (0.0 - 255) representation
<code>im2bw</code>	converts a grayscale image to just black or white with threshold

Writing Images

- ▶ choose image encoding via file extension
- ▶ if 3 layers, then RGB
- ▶ if 1 layer, then Grayscale

```
1 % rgb image
2 I = imread('cherry.jpg');
3
4 % convert I to grayscale
5 I = rgb2gray(I);
6
7 % write rgb image to file
8 % save as PNG
9 imwrite(I, 'g_cherry.png');
```

Outline

Review

Images

Movies

Treating Movies

- ▶ for analysis
 - treat frames as individual images
 - uncompress to matrix image format
- ▶ making movies
 - make every frame separately
 - build every frame from scratch (as opposed to adding to a previous frame)

Reading Movies

- ▶ missing from Octave (which has poor support for videos)

```
1 v = VideoReader('movie.avi');
2
3 v.Duration
4 v.FrameRate
5 v.Height
6
7 while hasFrame(v)
8     frame = readFrame(v);
9     imshow(frame);
10
11     pause(1 / v.FrameRate);
12 end
```

Writing Movies Procedure

- ▶ create a video writer
 - choose compression
 - configure options
- ▶ open the video writer
- ▶ create a figure
- ▶ repeat
 - clear the figure
 - plot to it
 - enforce same scaling and "zoom"
 - convert a figure to a frame (image-like)
 - write the frame
- ▶ close the video writer (flushes all data from RAM to disk)

Writing Movies Code

- ▶ missing from Octave (which has poor support for videos)

```
1 v = VideoWriter('my_video.mp4', 'MPEG-4');
2 % v = VideoWriter('my_video.avi', 'Uncompressed AVI');
3 v.FrameRate = 60 % 30 by default
4 % open video after setting options
5 open(v);
6 % simple harmonic oscillator
7 x = [5.0, 0];
8 f = @(x) [x(2); -x(1)];
9 tspan = [0, 1 / 60]; % single frame
10 % 10 second video
11 figure(1, 'Position', [0, 0, 500, 300]);
12 for i = 1:600
13     [T, X] = ode45(f, tspan, x);
14     x(:) = X(end, :);
15     clf(); % clear current figure
16     scatter(1, x(1));
17     % always keep the same range
18     xlim([-1, 1]);
19     ylim([-1, 1]);
20     % things look weird when not to scale
21     axis('equal');
22     % get frame from current figure
23     frame = getframe(gcf());
24     writeVideo(v, frame);
25 end
26 close(v);
```

Writing Movies Checklist

- ▶ extension must match chosen compression
- ▶ open the video writer before writing to it
- ▶ make sure frame rate matches time evolution
- ▶ clear figure before each frame
- ▶ enforce same scaling (`xlim`, `ylim`, `axis('equal')`)
- ▶ write a frame, not the plot, to the video writer
- ▶ close the video writer