

Tools and Prerequisites for Image Processing

EE4830 Digital Image Processing

<http://www.ee.columbia.edu/~xix/ee4830/>

Lecture 1, Jan 28th, 2008
Part 2 by Lexing Xie

Outline

- Review and intro in MATLAB
 - A light-weight review of linear algebra and probability
 - An introduction to image processing toolbox
- A few demo applications
- Image formats in a nutshell
- Pointers to image processing software and programming packages

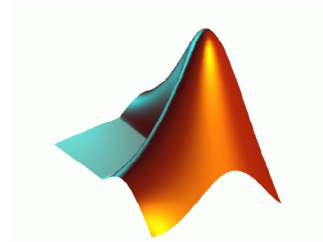
Matlab is ...



: a numerical computing environment and programming language. Created by The MathWorks, MATLAB allows easy matrix manipulation, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs in other languages.

■ Main Features:

- basic data structure is matrix
- optimized in speed and syntax for matrix computation



■ Accessing Matlab on campus

- Student Version
 - Matlab + Simulink \$99
 - Image Processing Toolbox \$59
 - Other relevant toolboxes \$29~59 (signal processing, statistics, optimization, ...)
- CUNIX and EE lab (12th floor) has Matlab installed with CU site-license

Why MATLAB?

- Shorter code, faster computation
- Focus on ideas, not implementation

$$f(x) = 2 \cdot \sin(x^3) / 3 + 4.56, x \in \{1, 3, 5, \dots, 9999\}$$

- **C:**

```
#include <math.h>
double x, f[500];
for( x=1.; x < 1000; x=x+2)
    f[(x-1)/2]=2*sin(pow(x, 3.))/3+4.56;
```

- **MATLAB:**

```
f=2*sin((1:2:1000).^3)/3+4.56;
```

But: scripting language, interpreted,

matrices

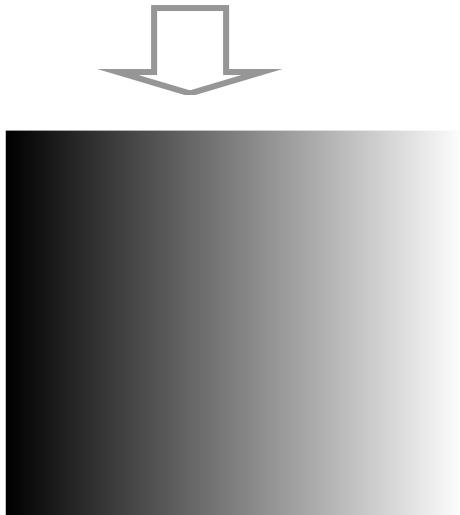
- ... are rectangular “tables” of entries where the entries are numbers or abstract quantities ...
- Some build-in matrix constructors
 - `a = rand(2), b = ones(2), c=eye(2),`
- Addition and scalar product
 - `d = c*2;`
- Dot product, dot-multiply and matrix multiplication
 - `c(:)'*a(:), d.*a, d*a`
- Matrix inverse, dot divide, etc.
 - `inv(a), a./d`

matrixes as images, and vice versa

- `x = 1 : 256;`
- `y = ones(256,1);`
- `a = x*y;`
`b = y*x;`

`size(a) = ?` `size(b) = ?`

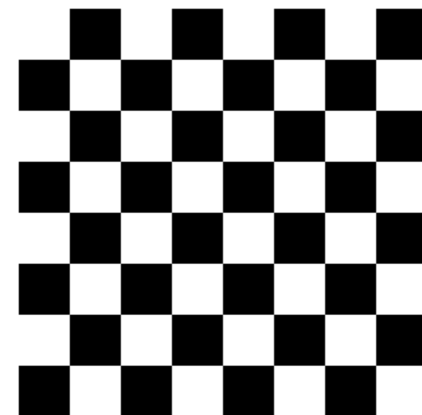
`imagesc(b); colormap(gray(256))`



`imagesc(checkerboard(32)>.5);`

or, from scratch:

```
b = ones(1,8); b(2:2:end)=0  
b = [b; b(end:-1:1)]  
b = repmat(b, [4 1])  
chessb = kron(b,ones(32));
```



256x256 chess board

eigen vectors and eigen values

- “eigenvectors” are exceptional vectors in the same direction as Ax

$$Ax = \lambda x$$

- λ are called *eigenvalues*

- Examples:

- $A = [.8 \ .3; .2 \ .7]$
- $[v, d] = \text{eig}(A);$
- $A*v(:, 1)$
- $A*v(:, 2)$

- `eigshow`

- properties of λ :

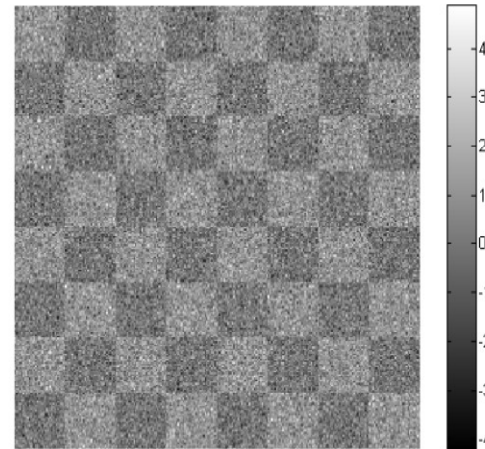
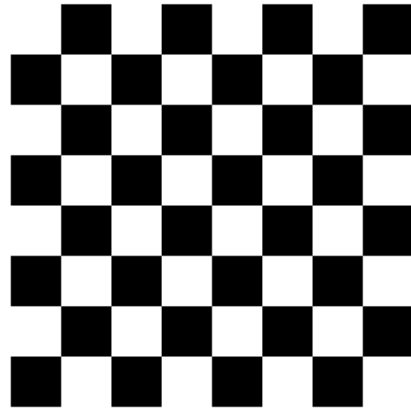
- $\sum_{i=1}^n a_{ii} = \sum_{i=1}^n \lambda_i = \text{trace}(A)$

- $\lambda_1 \cdot \lambda_2 \cdot \dots \cdot \lambda_n = \det(A)$

- eigen-vectors and values are useful for:

- Getting exponents of a matrix A^{100000}
- Image compression
- Object recognition
- The search algorithm behind Google
- ...

matlab quiz



- Chessboard + noise
 - `x = chessb + randn(256);`
- How to get the minimum and maximum value of `x` (in one line, with one function call) ?

`[min(x(:)) max(x(:))]`

the handy, esp. if `x` is more than three dimensions

`prctile(x(:), [0 100])`

the obscure, but exactly one function call.

probability

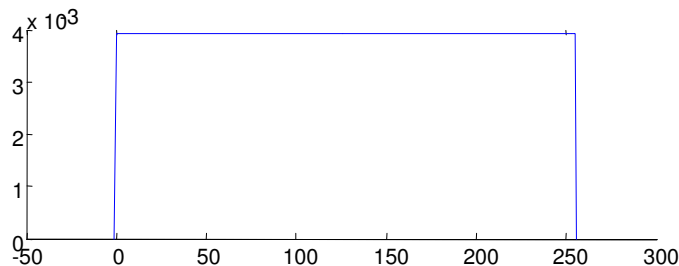
- *probability* refers to the *chance* that a particular event (or set of events) will occur.



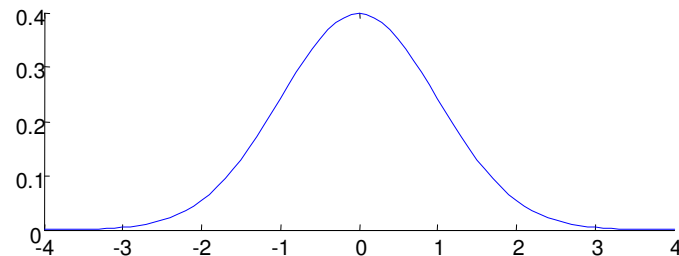
$\Pr(\text{head})=1/2,$
 $\Pr(\text{tail})=1/2$

- *probability density function* $p(x)$ is a non-negative intergrable function $\mathbb{R} \rightarrow \mathbb{R}$ such that for any interval $[a, b]$:
 $\Pr(x \in [a,b]) = \int_a^b p(x)dx$

```
p = pdf('uniform', -1:256, 0, 255);  
plot(-1:256, p)
```

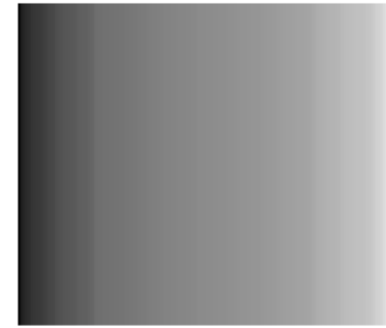
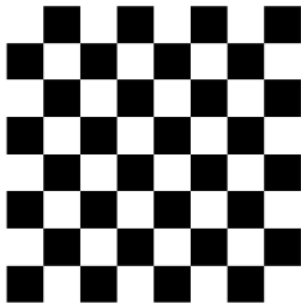


```
p = pdf('normal', -4:.1:4, 0, 1);  
plot(-4:.1:4, p)
```

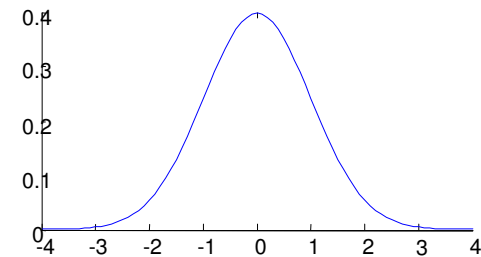
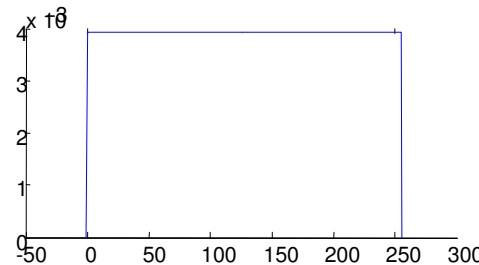


probability

- Suppose you're blind-folded and points to a point in a cardboard with the following prints, after a friend rotates and shifts it randomly (i.e. randomly draw a pixel from the following images)



$$p(\blacksquare) = 1/2 \quad p(\square) = 1/2$$



$$p(\blacksquare) = p(\square) = \dots = p(\square) = 1/256$$

mean and std

- Mean

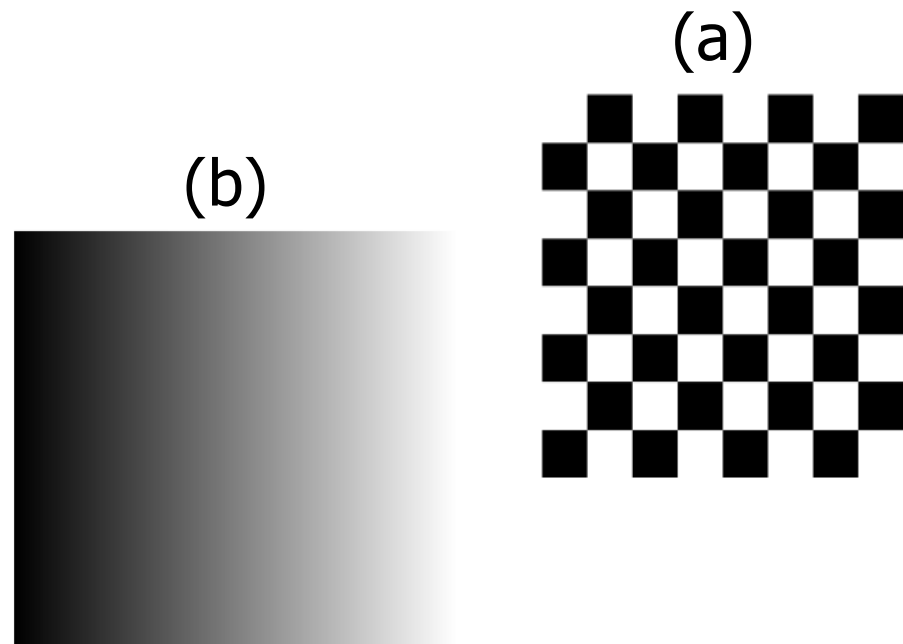
- $m_x = E[x] = \int x p(x) dx$

- Standard-deviation

- $\sigma_x^2 = E[(x-m_x)^2] = \int (x-m_x)^2 p(x) dx$

(a) and (b) are aforementioned gray-scale images with values between [0,1]. Which one of the following holds, if any?

$m_a < m_b$ ~~X~~ $m_a = m_b$
 $\sigma_a < \sigma_b$ ~~X~~ $\sigma_a > \sigma_b$



MATLAB (contd.)

- M-files:
 - functions
 - scripts
- Language constructs
 - Comment: %
 - if .. else... for... while... end
- Help:
 - help function_name, helpwin, helpdesk
 - lookfor, demo

Image Processing Toolbox

- File I/O and display
 - `imread()`, `imwrite()`
 - `imshow()`, `image()`, `imagesc()`, `movie()`
- ? how different are these two images?



`cu_home_low.bmp` (382 KB)

`cu_home_low_j40.jpg` (29KB)

```
im1 = imread('cu_home_low_treebranch.bmp');  
im2 = imread('cu_home_low_treebranch_j40.jpg');  
  
sqrt( sum( (im1(:)-im2(:)).^2 ) / prod(size(im1)) )  
  
imshow(im1- im2)
```

Image Processing Toolbox (contd)

- Linear operations
 - `fft2()`, `dct2()`, `conv2()`, `filter2()`
- Non-linear operations
 - `median()`, `dilate()`, `erode()`, `histeq()`
- Statistics and analysis
 - `imhist()`, `mean2()`, `corr2()`, `std2()`
- Colormap and type conversions
 - `colormap()`, `brighten()`, `rgbplot()`
 - `rgb2ycbcr()`, `hsv2rgb()`, `im2uint8()`...

Outline

- Review and intro in MATLAB
 - A light-weight review of linear algebra and probability
 - An introduction to image processing toolbox

- introduction and pointers to other image processing software and programming packages

Demo of image processing software

- Enhancement
“equalize” (lecture 4)
- Compression (lecture 12)
- Color manipulation (lecture 3)
with GIMP www.gimp.org

before



after



- “unshake” <http://www.hamangia.freemove.co.uk/> (lecture 7)

before



after



Image Processing Software

- Bitmap editing: Adobe Photoshop, Macromedia Fireworks
- Vector graphics editing: Adobe Illustrator, Corel Draw
- Consumer photo tools: Picassa, ACDSee, Windows Paint, XV, Photoshop Elements
- ...
- GIMP

Send me <xlx@ee.columbia.edu> your suggestions of image editing/processing tools!

Video processing software

- Player
 - Windows media player, Real, Quicktime, iTunes, intervideo WinDVD, ...
- Format conversion
 - ffmpeg
- Editing
 - Adobe premier, muvee,

Resource sites .. <http://doom9.net/>

Image Processing Toolboxes

- In C/C++
 - IPL ... <http://www.cs.nott.ac.uk/~jzg/nottsvision/old/index.html>
 - OpenCV <http://sourceforge.net/projects/opencvlibrary>
<http://tech.groups.yahoo.com/group/OpenCV/>
 - ImageMagick <http://www.imagemagick.org/>
 - Insight Toolkit ITK (medical image) <http://www.itk.org/>
 - List of tools at mathtools.net
http://www.mathtools.net/C_C_/Image_Processing/

- In Java
 - Java Media APIs: JAI, JMF, Java image I/O ...
<http://java.sun.com/javase/technologies/desktop/media/>
 - http://www.mathtools.net/Java/Image_Processing/index.html

- Other
 - Python Imaging Library (PIL) <http://www.pythonware.com/products/pil/>
numpy, scipy

Image Data Types

- Basic unit in disk: byte (8 bits)
- Images are stored as unsigned integers (0-255)

- Depends on the color space and the precision / bit depth
 - 1bit, 4bit, 8bit, 24bit, 32bit (+alpha channel), indexed colors (gif, 2-8 bits)



- In MATLAB:
 - uint8→double→uint8

File Formats

- Why different file formats?
 - Convenient to use
 - Compact representation
- How many formats do we have?
 - e.g. 30+ in a consumer image software (ACDSee)
 - There are much more out there:
raster, vector, metafile, ... and growing
- Basic structure: Header + Data

Format Comparison

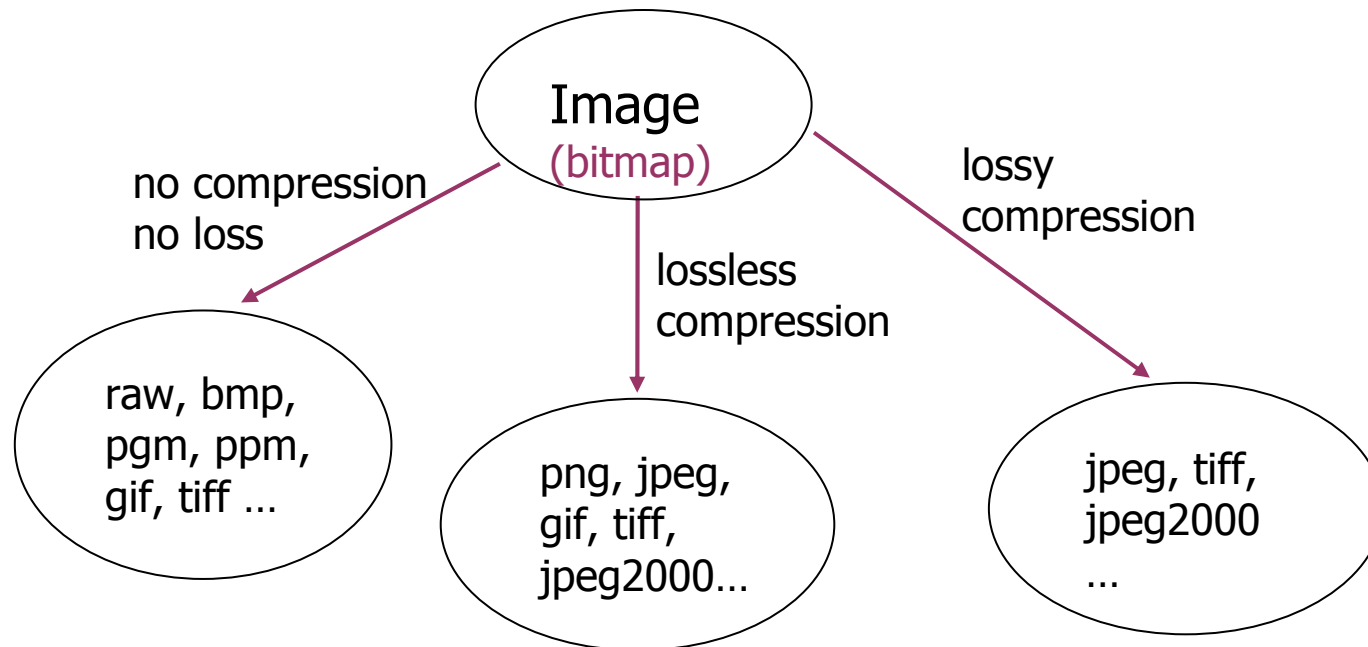
Two 256x256 color images

Format	RAW	BMP	GIF	PNG	JPG
Lossy?	N	N	N	N	Y
Compressed?	N	N	Y	Y	Y
	192K	193K	52.2K	106K	16K
	192K	193K	5K (4bit)	23K	20K
Fine prints	Raw data	Header ~1K	Look-up table + data		Quality factor 80


Why do the two images have different sizes as GIF/PNG/JPG files ?

Image Format Classification

- Types that MATLAB supports:
 - BMP, JPEG, PNG, GIF, TIFF, XWD, HDF, PCX, ...
 - Other open-source libraries from “google”



Resources and pointers

- Google, Wikipedia, Mathworld ... 

- Getting Help in Matlab
 - Matlab help, Image Processing Demos
 - DIP matlab tutorial online
 - Usenet groups

Summary

- Review of matrixes and probability
- MATLAB for image processing

- Data type and file formats
- Resources and pointers

< the end;  &  >

Working With Matrices in MATLAB

- Everything is treated as a matrix
- Elementary matrix manipulation
 - `zeros()`, `ones()`, `size()`, `eig()`, `inv()`
- Operators and special characters
 - `a(:,1:2:256)=b'.*c`
- String
 - `imstr=['this is lena'];`
`imglena=imread([imstr(9:end),'.png']);`
 - `ischar()`, `num2str()`, ...

- Review of linear algebra

- Point operation and matrix operations
- Eigen vectors, .. eigen values
- Images as matrices, and matrices as images ...
- Question: max/min, subsampling,

- Review of probability

- Coin-tossing, pdf, cdf, gaussian pdf
- Expectations, std, variance
- Question: pdf shape, expectation/expected value,

- Matlab

- Getting started
 - Image I/O and display
 - Matrix manipulation
 - Image processing demos
- The daily practice of image manipulation
 - Image processing tools in C, Java, ... and everything else
 - Data types and file formats
 - Resources, pointers and getting help