Image Processing I

15-463: Rendering and Image Processing

Alexei Efros

…with most slides shamelessly stolen from Steve Seitz and Gonzalez & Woods

Today

Image Formation

Range Transformations

- Point Processing

Programming Assignment #1 OUT

Reading for this week:

- Gonzalez & Woods, Ch. 3
- Forsyth & Ponce, Ch. 7

Image Formation

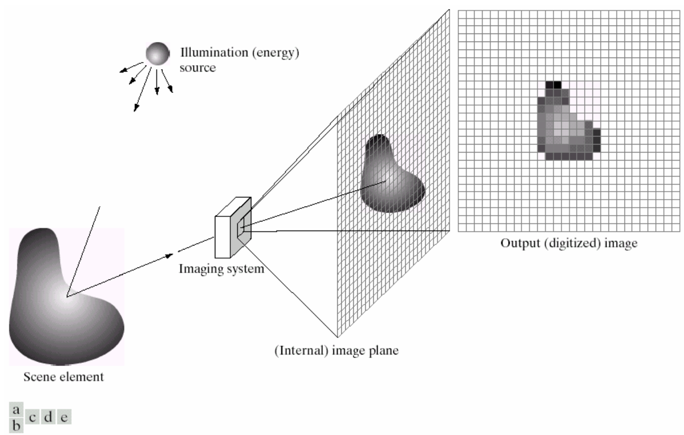


FIGURE 2.15 An example of the digital image acquisition process. (a) Energy ("illumination") source. (b) An element of a scene. (c) Imaging system. (d) Projection of the scene onto the image plane. (e) Digitized image.

*f(x,y) = reflectance(x,y) * illumination(x,y)*

Reflectance in [0,1], illumination in [0,inf]

Sampling and Quantization

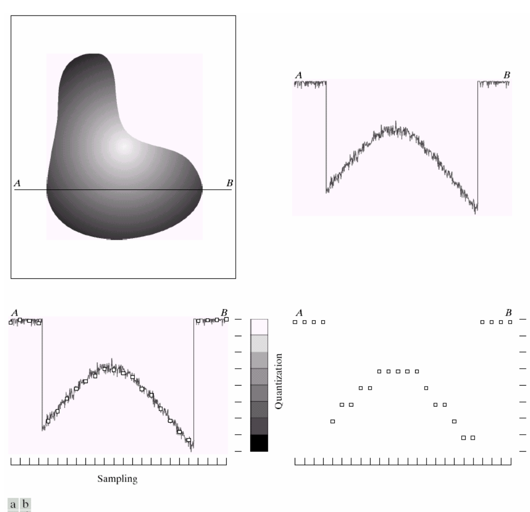
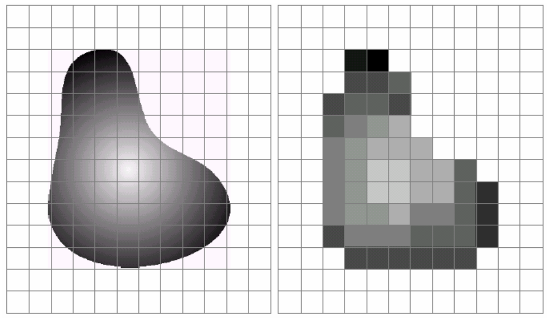


FIGURE 2.16 Generating a digital image. (a) Continuous image. (b) A scan line from *A* to *B* in the continuous image, used to illustrate the concepts of sampling and quantization. (c) Sampling and quantization. (d) Digital scan line.

Sampling and Quantization



a b

FIGURE 2.17 (a) Continuos image projected onto a sensor array. (b) Result of image sampling and quantization.

What is an image?

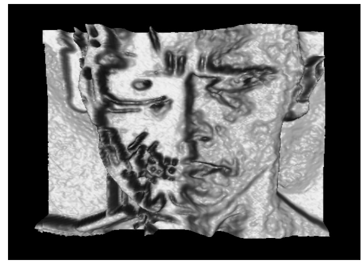
We can think of an **image** as a function, *f*, from R^2 to R:

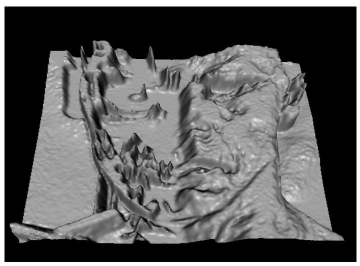
- $f(x, y)$ gives the **intensity** at position (x, y)
- Realistically, we expect the image only to be defined over a rectangle, with a finite range:
  - f: $[a,b]$x$[c,d] \rightarrow [0,1]$

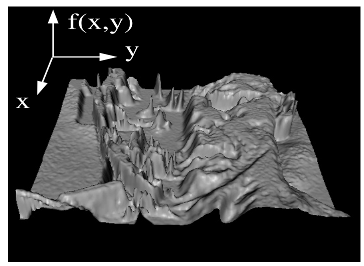
A color image is just three functions pasted together. We can write this as a "vector-valued" function:

$$f(x, y) = \begin{bmatrix} r(x, y) \\ g(x, y) \\ b(x, y) \end{bmatrix}$$

Images as functions







What is a digital image?

We usually operate on **digital** (**discrete**) images:

- **Sample** the 2D space on a regular grid
- **Quantize** each sample (round to nearest integer)

If our samples are Δ apart, we can write this as:

$f[i, j] = \text{Quantize}\{ f(i\,\Delta, j\,\Delta) \}$

The image can now be represented as a matrix of integer values

$j \rightarrow$, $i \downarrow$

62	79	23	119	120	105	4	0
10	10	9	62	12	78	34	0
10	58	197	46	46	0	0	48
176	135	5	188	191	68	0	49
2	1	1	29	26	37	0	77
0	89	144	147	187	102	62	208
255	252	0	166	123	62	0	31
166	63	127	17	1	0	99	30

Image processing

An **image processing** operation typically defines a new image g in terms of an existing image f.

We can transform either the range of f.

$$g(x, y) = t(f(x, y))$$

Or the domain of *f*:

$$g(x, y) = f(t_x(x, y), t_y(x, y))$$

What kinds of operations can each perform?

Point Processing

The simplest kind of range transformations are these independent of position x,y:

$$g = t(f)$$

This is called point processing.

What can they do?

What's the form of *t*?

Important: every pixel for himself – spatial information completely lost!

Basic Point Processing

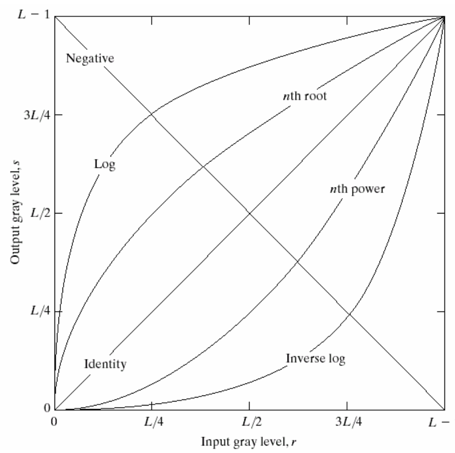
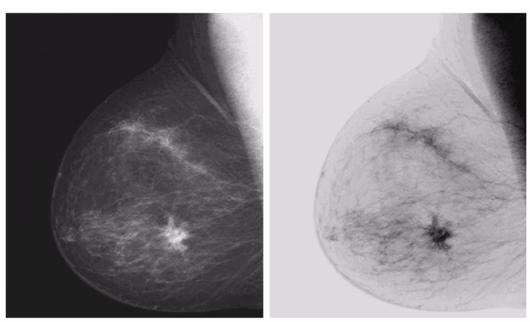


FIGURE 3.3 Some basic gray-level transformation functions used for image enhancement.

Negative



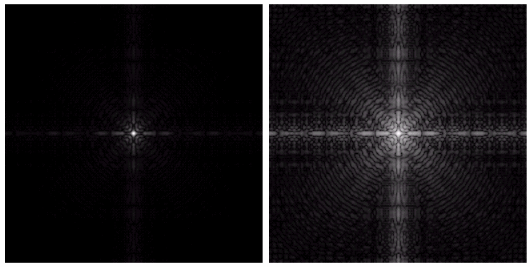
a b

FIGURE 3.4 (a) Original digital mammogram. (b) Negative image obtained using the negative transformation in Eq. (3.2-1). (Courtesy of G.E. Medical Systems.)

Log

a b

FIGURE 3.5 (a) Fourier spectrum. (b) Result of applying the log transformation given in Eq. (3.2-2) with $c = 1$.



Power-law transformations

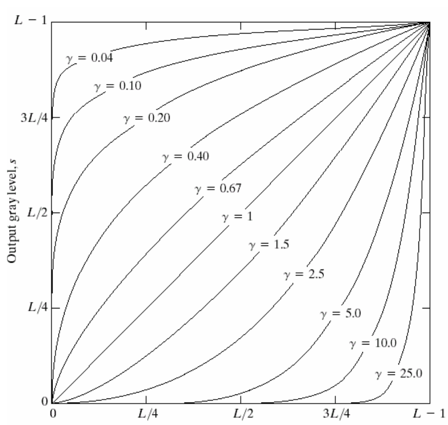
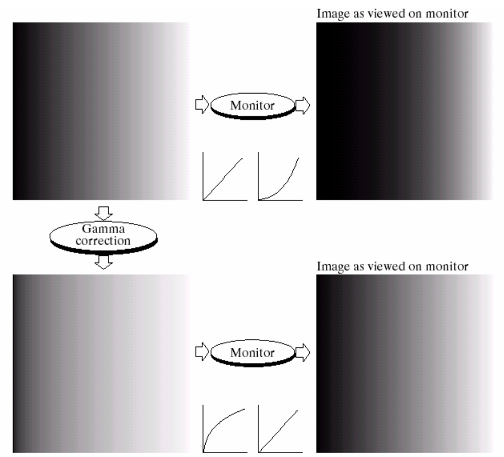


FIGURE 3.6 Plots of the equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases).

Gamma Correction



a b
c d

FIGURE 3.7 (a) Linear-wedge gray-scale image. (b) Response of monitor to linear wedge. (c) Gamma-corrected wedge. (d) Output of monitor.

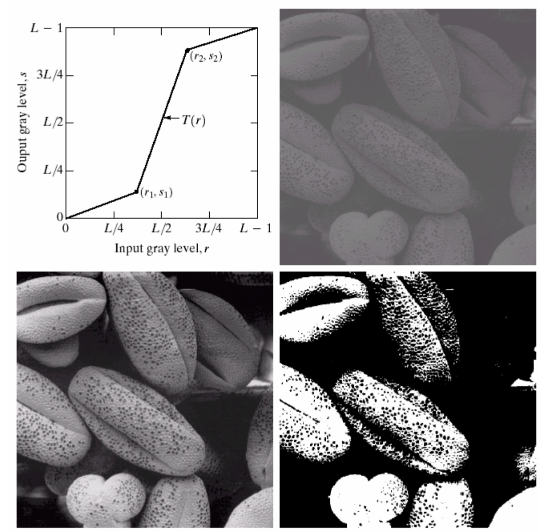
Gamma Measuring Applet:
http://www.cs.berkeley.edu/~efros/java/gamma/gamma.html

Image Enhancement

a b
c d

FIGURE 3.9 (a) Aerial image. (b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 3.0, 4.0,$ and 5.0, respectively. (Original image for this example courtesy of NASA.)

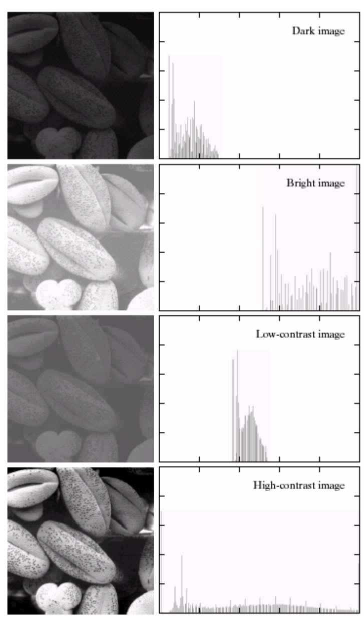
Contrast Streching



a b
c d

FIGURE 3.10 Contrast stretching. (a) Form of transformation function. (b) A low-contrast image. (c) Result of contrast stretching. (d) Result of thresholding. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

Image Histograms

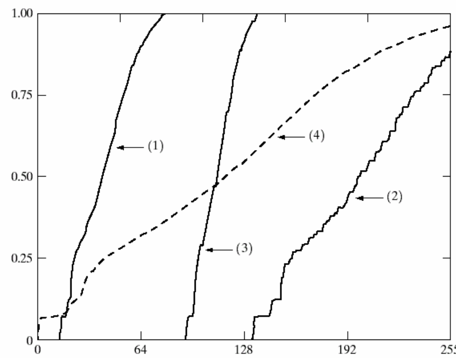


a b

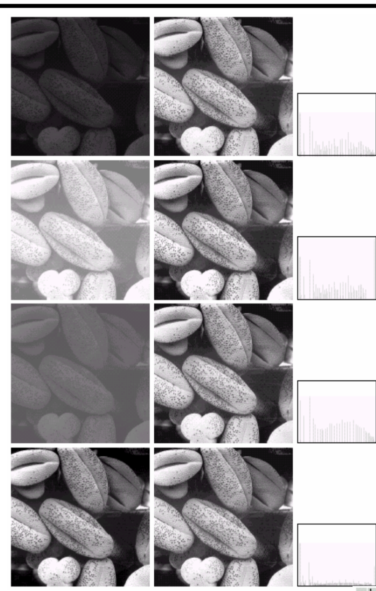
FIGURE 3.15 Four basic image types: dark, light, low contrast, high contrast, and their corresponding histograms. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

Cumulative Histograms

FIGURE 3.18 Transformation functions (1) through (4) were obtained from the histograms of the images in Fig.3.17(a), using Eq. (3.3-8).



Histogram Equalization



a b c

FIGURE 3.17 (a) Images from Fig. 3.15. (b) Results of histogram equalization. (c) Corresponding histograms.

Histogram Matching

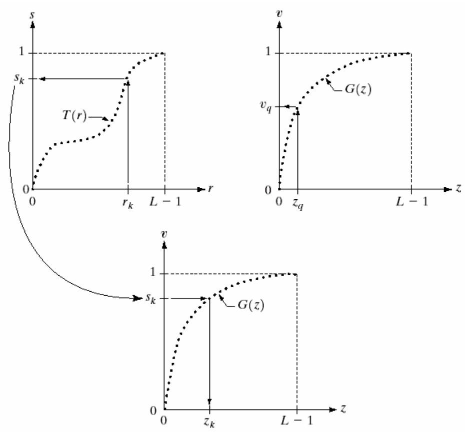


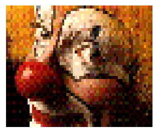
FIGURE 3.19 (a) Graphical interpretation of mapping from r_k to s_k via $T(r)$. (b) Mapping of z_q to its corresponding value v_q via $G(z)$. (c) Inverse mapping from s_k to its corresponding value of z_k.

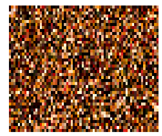
Match-histogram code

```
Match-histogram (im1,im2)
  im1-cdf = Make-cdf(im1)
  im2-cdf = Make-cdf(im2)
  inv-im2-cdf = Make-inverse-lookup-table(im2-cdf)
  Loop for each pixel do
       im1[pixel] =
         Lookup(inv-im2-cdf,
                Lookup(im1-cdf,im1[pixel]))
```

Neighborhood Processing (filtering)

Q: What happens if I reshuffle all pixels within the image?





A: It's histogram won't change. No point processing will be affected…

Need spatial information to capture this.

Programming Assignment #1

Easy stuff to get you started with Matlab

- James will hold tutorial this week

Distance Functions

- SSD
- Normalized Correlation

Bells and Whistles

- Point Processing (color?)
- Neighborhood Processing
- Using your data (3 copies!)
- Using your data (other images)