
Filtering (I)

Dr. Gerhard Roth

COMP 4900D
Winter 2012

Agenda

- Getting to know images.
- Image noise.
- Image filters.

Digital Images

An image - rectangular array of integers

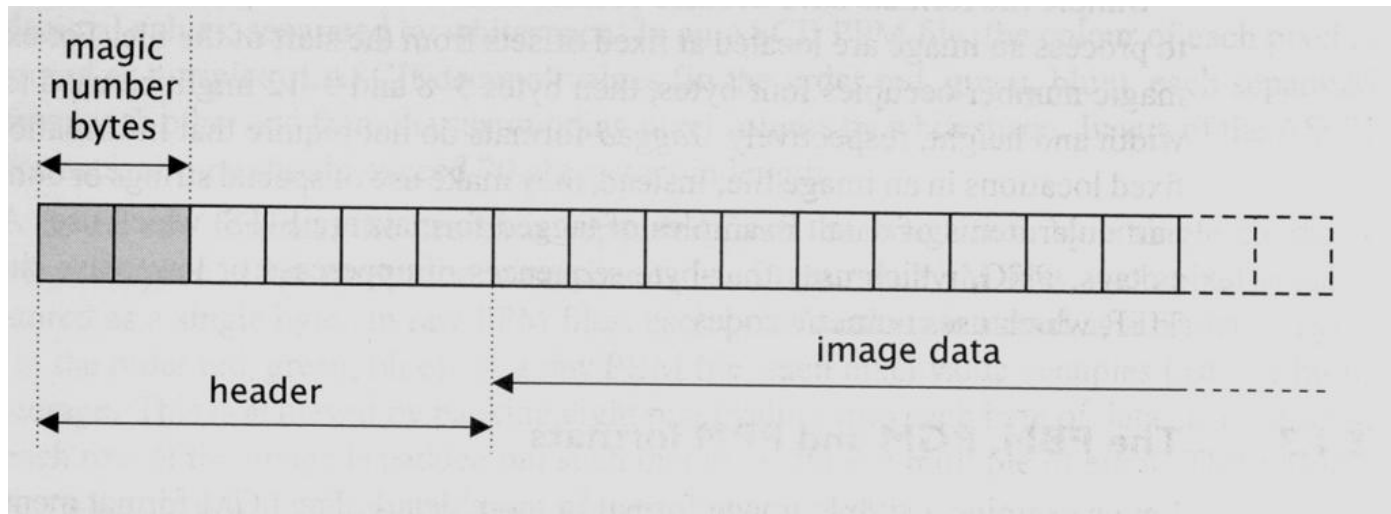
Each integer - the brightness or darkness of the image at that point

N: # of rows, M: # of columns, Q: # of gray levels

- $Q = 2^q$ (q is the # of bits/pixel)
- Storage requirements: $N \times M \times q$
- (e.g., for $N=M=1024$ and $q=8 \rightarrow$ storage = 1MB)

$$\begin{array}{cccc} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & f(1,1) & \dots & f(1,M-1) \\ \dots & \dots & \dots & \dots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{array}$$

Image File Formats



- Header – usually contains width, height, bit depth.
- Magic number – identify file format.
- Image data – stores pixel values line by line without breaks.

Common Image File Formats

GIF (Graphic Interchange Format, .gif)

PNG (Portable Network Graphics, .png)

JPEG (Joint Photographic Experts Group, .jpeg)

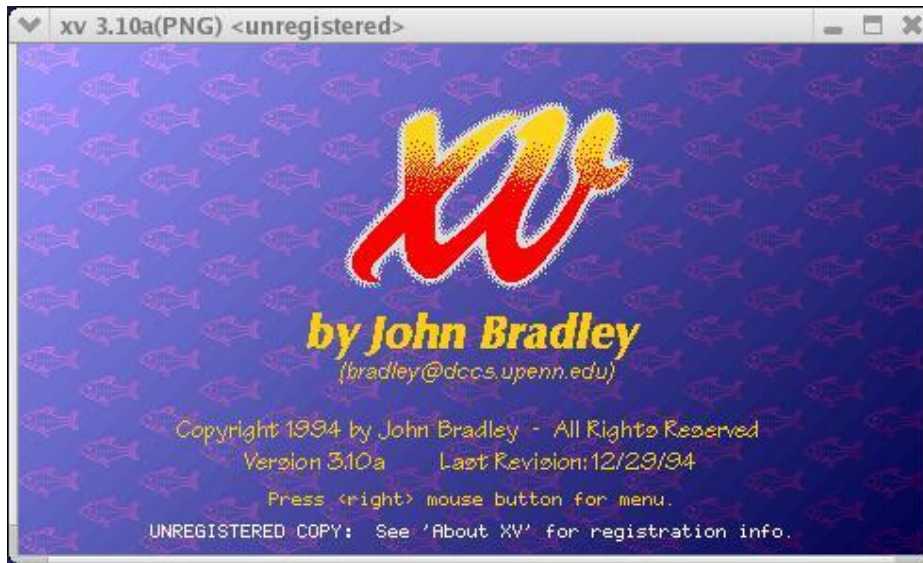
TIFF (Tagged Image File Format, tiff)

PGM (Portable Gray Map, .pgm)

FITS (Flexible Image Transport System, .fits)

...

Image Manipulation Tools



Linux: xv, gimp

Windows: Photoshop,
Paint Shop Pro



OpenCV + Ch – read and display images

```
#ifdef _CH_
#pragma package <opencv>
#endif

#include "cv.h"
#include "highgui.h"

// Load the source image. HighGUI use.
IplImage *image = 0, *gray = 0;

int main( int argc, char** argv )
{
    char* filename = argc == 2 ? argv[1] : (char*)"fruits.jpg";

    if( (image = cvLoadImage( filename, 1)) == 0 )
        return -1;

    // Convert to grayscale
    gray = cvCreateImage(cvSize(image->width,image->height), IPL_DEPTH_8U, 1);
    cvCvtColor(image, gray, CV_BGR2GRAY);

    // Create windows.
    cvNamedWindow("Source", 1);
    cvNamedWindow("Result", 1);

    // Show the images.
    cvShowImage("Source", image);
    cvShowImage("Result", gray);

    // Wait for a key stroke
    cvWaitKey(0);
    cvReleaseImage(&image);
    cvReleaseImage(&gray);

    cvDestroyWindow("Source");
    cvDestroyWindow("Result");

    return 0;
}
```

Noise

Noise in Computer Vision:

- Any entity (in images, data, or intermediate results) that is not interesting for the main computation

Can think of noise in several contexts:

- Spurious pixel fluctuations, introduced by image acquisition
- Numerical inaccuracies, introduced by the computer's limited precision, round-offs or truncations, etc
- Entities that do not belong to any meaningful region (edges that do not correspond to actual object contours)

Image Noise

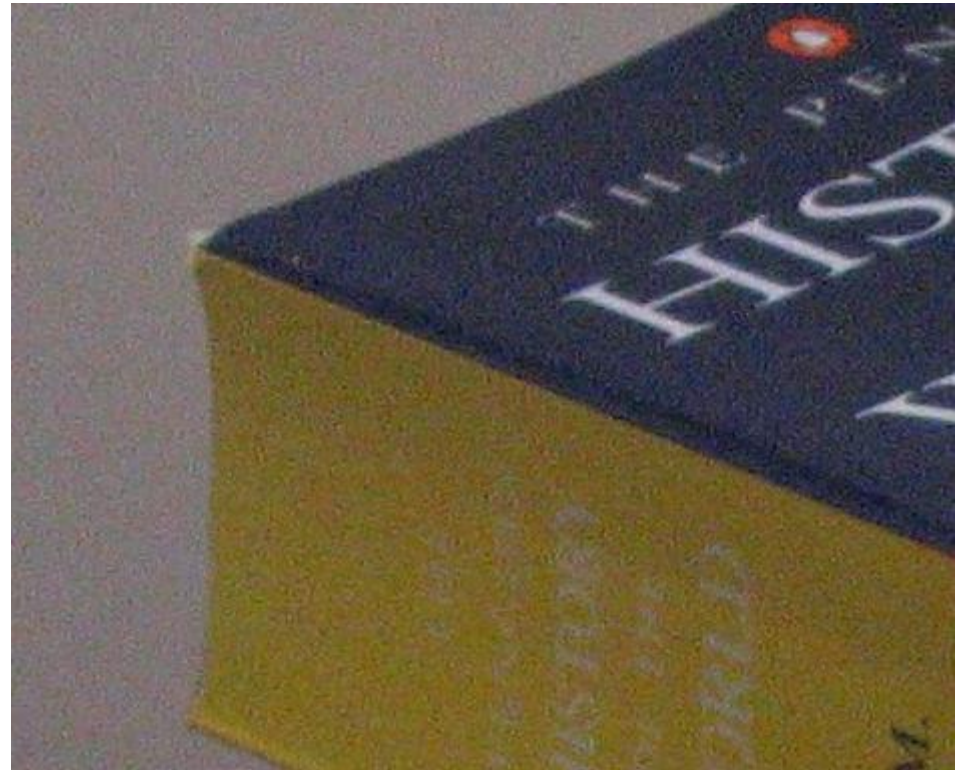


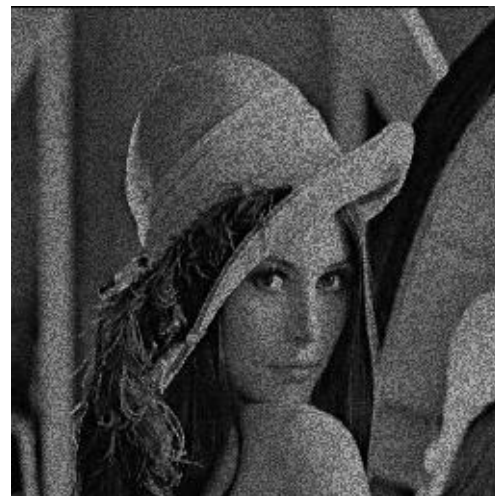
Image Noise

Additive and random noise:

$$\hat{I}(x, y) = I(x, y) + n(x, y)$$

$I(x, y)$: the true pixel values

$n(x, y)$: the (random) noise at pixel (x, y)



Random Variables

A random variable is a variable whose values are determined by the outcome of a random experiment.

A random variable can be **discrete** or **continuous**.

Examples

1. A coin is tossed ten times. The random variable X is the number of tails that are noted. X can only take the values 0, 1, ..., 10, so X is a discrete random variable.
2. A light bulb is burned until it burns out. The random variable Y is its lifetime in hours. Y can take any positive real value, so Y is a continuous random variable.

Mean, Variance of Random Variable

Mean (Expected Value)

$$\mu = E(X) = \sum_i x_i p(x_i) \quad (\text{discrete})$$

$$\mu = E(X) = \int_{-\infty}^{\infty} xp(x)dx \quad (\text{continuous})$$

Variance

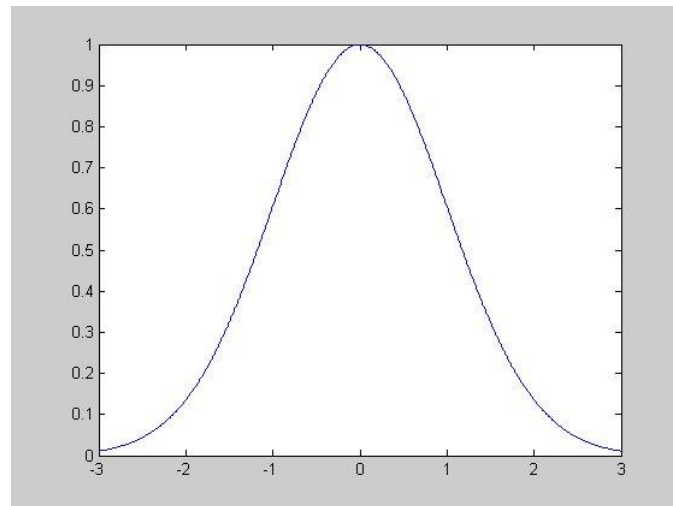
$$\sigma^2 = E((X - \mu)^2)$$

$\sigma = \sqrt{\sigma^2}$ is called standard deviation

Gaussian Distribution

Single variable

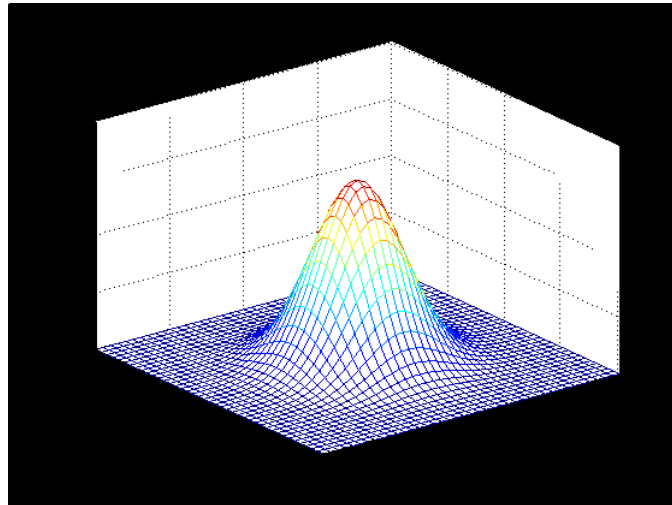
$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2 / 2\sigma^2}$$



Gaussian Distribution

Bivariate with zero-means and variance σ^2

$$G_{\sigma}(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x^2 + y^2)}{2\sigma^2}\right)$$



Gaussian Noise

Is used to model additive random noise

- The probability of $n(x,y)$ is $e^{\frac{-n^2}{2\sigma^2}}$
- Each has zero mean
- The noise at each pixel is independent



Impulsive Noise

- Alters random pixels
- Makes their values very different from the true ones

Salt-and-Pepper Noise:

- Is used to model impulsive noise



$$I_{sp}(h, k) = \begin{cases} I(h, k) & x < l \\ i_{\min} + y(i_{\max} - i_{\min}) & x \geq l \end{cases}$$

x, y are uniformly distributed
random variables

l, i_{\min}, i_{\max} are constants