Digital Image Processing (CS/ECE 545)
Lecture 1: Introduction to Image Processing and ImageJ

Prof Emmanuel Agu

Computer Science Dept.
Worcester Polytechnic Institute (WPI)
What is an Image?

- 2-dimensional matrix of Intensity (gray or color) values

\[ I(u, v) \in \mathbb{P} \quad \text{and} \quad u, v \in \mathbb{N}. \]
Example of Digital Images

a) Natural landscape
b) Synthetically generated scene
c) Poster graphic
d) Computer screenshot
e) Black and white illustration
f) Barcode
g) Fingerprint
h) X-ray
i) Microscope slide
j) Satellite Image
k) Radar image
l) Astronomical object
Imaging System

Example: a camera
Converts light to image

Credits: Gonzales and Woods
Digital Image?

- **Remember:** *digitization* causes a digital image to become an *approximation* of a real scene.
Digital Image

- Common image formats include:
  - 1 values per point/pixel (B&W or Grayscale)
  - 3 values per point/pixel (Red, Green, and Blue)
  - 4 values per point/pixel (Red, Green, Blue, + “Alpha” or Opacity)

- We will start with gray-scale images, extend to color later
What is image Processing?

- Algorithms that alter an input image to create new image
- Input is image, output is image

![Original Image](image)

**Image Processing Algorithm (e.g. Sobel Filter)**

![Processed Image](image)

- Improves an image for human interpretation in ways including:
  - Image display and printing
  - Image editing
  - Image enhancement
  - Image compression
Example Operation: Noise Removal

Think of noise as white specks on a picture (random or non-random)
Examples: Noise Removal

Example: Contrast Adjustment

Low Contrast  Original Contrast  High Contrast
Example: Edge Detection
Example: Region Detection, Segmentation
Example: Image Compression

Original, 2.1MB

JPEG Compression, 308KB (15%)
Example: Image Inpainting

Inpainting? Reconstruct corrupted/destroyed parts of an image

Credit: M. Bertalmio, G. Sapiro, V. Caselles, C. Ballester: Image Inpainting, SIGGRAPH 2000
Examples: Artistic (Movie Special )Effects
Applications of Image Processing

Biology

Astronomy

Credit: Dartmouth Electron Microscopy Facility

Credit: NASA, Jeff Hester, and Paul Scowen (Arizona State)
More info here
Applications of Image Processing

Medicine

Security, Biometrics

Credit: Dr. Janet Lainhart, UofU Psychiatry
Applications of Image Processing: Medicine

Original MRI Image of a Dog Heart

Edge Detection Image

Applications of Image Processing

Satellite Imagery

Credit: NASA

Personal Photos

Credit: Tom Fletcher
Applications of Image Processing: Geographic Information Systems (GIS)

- Terrain classification
- Meteorology (weather)

Applications of Image Processing: Law Enforcement

- Number plate recognition for speed cameras or automated toll systems
- Fingerprint recognition
Applications of Image Processing: HCI

- Face recognition
- Gesture recognition
Relationship with other Fields

**High-level**

**Computer Vision**
Object detection, recognition, shape analysis, tracking
Use of Artificial Intelligence and Machine Learning

**Image Analysis**
Segmentation, image registration, matching

**Low-level**

**Image Processing**
Image enhancement, noise removal, restoration, feature detection, compression
Key Stages in Digital Image Processing

- Image Acquisition
- Image Enhancement
- Image Restoration
- Morphological Processing
- Segmentation
  - Representation & Description
  - Object recognition

Problem Domain: Colour Image Processing

Representation & Description: Image Compression
Key Stages in Digital Image Processing:

- Image Acquisition
- Image Enhancement
- Image Restoration
- Morphological Processing
- Segmentation
- Representation & Description
- Output (digitized) image
- Problem Domain
- Example: Take a picture

- Colour Image Processing
- Image Compression
- Object recognition

Key Stages in Digital Image Processing: Image Enhancement

- Image Acquisition
- Image Restoration
- Morphological Processing
- Segmentation
- Representation & Description
- Object recognition

Example: Change contrast

Problem Domain

- Colour Image Processing
- Image Compression

Key Stages in Digital Image Processing: Image Restoration

- Problem Domain
- Image Acquisition
- Image Enhancement
- Image Restoration
- Morphological Processing
  - Example: Remove Noise
- Segmentation
  - Representation & Description
  - Object recognition
- Image Compression
- Colour Image Processing

Key Stages in Digital Image Processing: Morphological Processing

Image Acquisition → Image Enhancement → Image Restoration → Morphological Processing

Image Compression

Segmentation

Representation & Description

Object recognition

Problem Domain

Colour Image Processing

Extract attributes useful for describing image

Key Stages in Digital Image Processing: Segmentation

1. Image Acquisition
2. Image Enhancement
3. Image Restoration
4. Morphological Processing
5. Segmentation
   - Representation & Description
   - Object recognition
6. Colour Image Processing
7. Image Compression

Divide image into constituent parts

Key Stages in Digital Image Processing: Object Recognition

- Problem Domain
- Image Acquisition
- Image Enhancement
- Image Restoration
- Morphological Processing
- Segmentation
- Representation & Description
- Colour Image Processing
- Image Compression

Image regions transformed suitable for computer processing

Key Stages in Digital Image Processing: Representation & Description

- Image Acquisition
- Image Enhancement
- Image Restoration
- Morphological Processing
- Segmentation
- Representation & Description
- Object recognition

Key Stages in Digital Image Processing:

- Image Compression
- Image Acquisition
- Image Restoration
- Morphological Processing
- Segmentation
- Representation & Description
- Object recognition
- Colour Image Processing
Key Stages in Digital Image Processing: Colour Image Processing

- Image Acquisition
- Image Enhancement
- Image Restoration
- Morphological Processing
- Segmentation
- Representation & Description
- Object recognition

Consider color images (color models, etc)

Colour Image Processing

Image Compression
Mathematics for Image Processing

- Calculus
- Linear algebra
- Probability and statistics
- Differential Equations (PDEs and ODEs)
- Differential Geometry
- Harmonic Analysis (Fourier, wavelet, etc)
About This Course

- Image Processing has many aspects
  - Computer Scientists/Engineers develop tools (e.g. photoshop)
    - Requires knowledge of maths, algorithms, programming
  - Artists use image processing tools to modify pictures
    - DOES NOT require knowledge of maths, algorithms, programming

Example: Portraiture photoshop plugin
Example: Knoll Light Factory photoshop plugin
Example: ToonIt photoshop plugin
About This Course

- Most hobbyists follow artist path. Not much math!
- **This Course: Image Processing for computer scientists and Engineers!!!**
- Teaches concepts, uses ImageJ as concrete example
- ImageJ: Image processing library
  - Includes lots of already working algorithms,
  - Can be extended by programming new image processing techniques
- **Course is NOT**
  - just about programming ImageJ
  - a comprehensive course in ImageJ. (Only parts of ImageJ covered)
  - about using packages like Photoshop, GIMP
About This Course

- Class is concerned with:
  - How to implement image processing algorithms
  - Underlying mathematics
  - Underlying algorithms

- This course is a lot of work. Requires:
  - Lots of programming in Java (maybe some MATLAB)
  - Lots of math, linear systems, fourier analysis
Administrivia: Syllabus Summary

- 2 Exams (50%), 5 Projects (50%)
- Projects:
  - Develop ImageJ Java code on any platform but must work in Zoolab machine
  - May discuss projects but turn in individual projects
- Class website: http://web.cs.wpi.edu/~emmanuel/courses/cs545/S14/
- Text:
- Cheating: Immediate ‘F’ in the course
- My advice:
  - Come to class
  - Read the text
  - Understand concepts before coding
Light And The Electromagnetic Spectrum

- Light: just a particular part of electromagnetic spectrum that can be sensed by the human eye.

- The electromagnetic spectrum is split up according to the wavelengths of different forms of energy.
Reflected Light

- The colours humans perceive are determined by nature of light reflected from an object.

- For example, if white light (contains all wavelengths) is shone onto a green object, it absorbs most wavelengths absorbed except the green wavelength (color).
Electromagnetic Spectrum and IP

- Images can be made from any form of EM radiation

From Wikipedia
Images from Different EM Radiation

- Radar imaging (radio waves)
- Magnetic Resonance Imaging (MRI) (Radio waves)
- Microwave imaging
- Infrared imaging
- Photographs
- Ultraviolet imaging telescopes
- X-rays and Computed tomography
- Positron emission tomography (gamma rays)
- Ultrasound (not EM waves)
Human Visual System: Structure Of The Human Eye

- The lens focuses light from objects onto the retina.
- Retina covered with light receptors called **cones** (6-7 million) and **rods** (75-150 million).
- Cones concentrated around fovea. Very sensitive to colour.
- Rods more spread out and sensitive to low illumination levels.
Image Formation In The Eye

- Muscles in the eye can change the shape of the lens allowing us to focus on near or far objects.
- An image is focused onto the retina exciting the rods and cones and sending signals to the brain.
Image Formation

- The Pinhole Camera (abstraction)
  - First described by ancient Chinese and Greeks (300-400AD)

\[ x = -\frac{f}{Z} X, \quad y = -\frac{f}{Z} Y \]
Thin Lens
Brightness Adaptation & Discrimination

- The human visual system can perceive approximately $10^{10}$ different light intensity levels.

- However, at any one time we can only discriminate between a much smaller number – *brightness adaptation*.

- Similarly, *perceived intensity* of a region is related to the light intensities of the regions surrounding it.
Brightness Adaptation & Discrimination: Mach Band Effect

Perceived intensity overshoots or undershoots at areas of intensity change
Brightness Adaptation & Discrimination

An example of *simultaneous contrast*

All inner squares have same intensity but appear darker as outer square (surrounding area) gets lighter

Image Acquisition

- Images typically generated by *illuminating a scene* and absorbing energy reflected by scene objects.

Image Sensing

- Incoming energy (e.g. light) lands on a sensor material responsive to that type of energy, generating a voltage.
- Collections of sensors are arranged to capture images.

Spatial Sampling

- Cannot record image values for all \((x,y)\)
- Sample/record image values at discrete \((x,y)\)
- Sensors arranged in grid to sample image
Image (Spatial) Sampling

● A digital sensor can only measure a limited number of samples at a discrete set of energy levels

● **Sampling** can be thought of as:
  Continuous signal $\times$ comb function
Image Quantization

- **Quantization**: process of converting continuous analog signal into its digital representation
- Discretize image $l(u,v)$ values
- Limit values image can take
Image Sampling And Quantization

Sampling and quantization generates approximation of a real world scene

Image as Discrete Function

After spatial sampling and quantization, an image is a discrete function. The image domain $\Omega$ is now discrete:

$$\Omega \subset \mathbb{N}^2,$$

and so is the image range:

$$I : \Omega \rightarrow \{1, \ldots, K\},$$

where $K \in \mathbb{N}$. 
Image as a Function

A simple image

Image function as a height field
Representing Images

- Image data structure is 2D array of pixel values
- Pixel values are gray levels in range 0-255 or RGB colors
- Array values can be any data type (bit, byte, int, float, double, etc.)
Spatial Resolution

• *The spatial resolution* of an image is determined by how fine/coarse sampling was carried out

• **Spatial resolution**: smallest discernable image detail

  • Vision specialists talk about image resolution

  • Graphic designers talk about *dots per inch* (DPI)
Spatial Resolution

Spatial Resolution: Stretched Images

Intensity Level Resolution

- **Intensity level resolution**: number of intensity levels used to represent the image
  - The more intensity levels used, the finer the level of detail discernable in an image
  - Intensity level resolution usually given in terms of number of bits used to store each intensity level

<table>
<thead>
<tr>
<th>Number of Bits</th>
<th>Number of Intensity Levels</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>0, 1</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>00, 01, 10, 11</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>0000, 0101, 1111</td>
</tr>
<tr>
<td>8</td>
<td>256</td>
<td>00110011, 01010101</td>
</tr>
<tr>
<td>16</td>
<td>65,536</td>
<td>1010101010101010</td>
</tr>
</tbody>
</table>
Intensity Level Resolution

256 grey levels (8 bits per pixel)  128 grey levels (7 bpp)  64 grey levels (6 bpp)  32 grey levels (5 bpp)

16 grey levels (4 bpp)  8 grey levels (3 bpp)  4 grey levels (2 bpp)  2 grey levels (1 bpp)

Saturation & Noise

**Saturation**: highest intensity value above which color is washed out

**Noise**: grainy texture pattern
Resolution: How Much Is Enough?

- The big question with resolution is always *how much is enough*?
  - Depends on what is in the image (*details*) and what you would like to do with it (*applications*)
  - Key questions:
    - Does image look aesthetically pleasing?
    - Can you see what you need to see in image?
Resolution: How Much Is Enough?

- **Example:** Picture on right okay for counting number of cars, but not for reading the number plate.
Intensity Level Resolution

Low Detail

Medium Detail

High Detail
Image File Formats

- Hundreds of image file formats. Examples
  - Tagged Image File Format (TIFF)
  - Graphics Interchange Format (GIF)
  - Portable Network Graphics (PNG)
  - JPEG, BMP, Portable Bitmap Format (PBM), etc

- Image pixel values can be
  - **Grayscale**: 0 – 255 range
  - **Binary**: 0 or 1
  - **Color**: RGB colors in 0-255 range (or other color model)
  - **Application specific** (e.g. floating point values in astronomy)
How many Bits Per Image Element?

**Grayscale (Intensity Images):**

| Chan. | Bits/Pix. | Range   | Use                                                   |
|-------|-----------|---------|                                                      |
| 1     | 1         | 0...1   | Binary image: document, illustration, fax            |
| 1     | 8         | 0...255 | Universal: photo, scan, print                        |
| 1     | 12        | 0...4095| High quality: photo, scan, print                     |
| 1     | 14        | 0...16383| Professional: photo, scan, print                     |
| 1     | 16        | 0...65535| Highest quality: medicine, astronomy                 |

**Color Images:**

| Chan. | Bits/Pix. | Range       | Use                                                   |
|-------|-----------|-------------|                                                      |
| 3     | 24        | [0...255]^3 | RGB, universal: photo, scan, print                   |
| 3     | 36        | [0...4095]^3| RGB, high quality: photo, scan, print                |
| 3     | 42        | [0...16383]^3| RGB, professional: photo, scan, print                |
| 4     | 32        | [0...255]^4 | CMYK, digital prepress                              |

**Special Images:**

<table>
<thead>
<tr>
<th>Chan.</th>
<th>Bits/Pix.</th>
<th>Range</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>16</td>
<td>-32768...32767</td>
<td>Whole numbers pos./neg., increased range</td>
</tr>
<tr>
<td>1</td>
<td>32</td>
<td>±3.4 \cdot 10^{38}</td>
<td>Floating point: medicine, astronomy</td>
</tr>
<tr>
<td>1</td>
<td>64</td>
<td>±1.8 \cdot 10^{308}</td>
<td>Floating point: internal processing</td>
</tr>
</tbody>
</table>
**Introduction to ImageJ**

- **ImageJ**: Open source Java Image processing software
- Developed by Wayne Rasband at Nat. Inst for Health (NIH)
  - Many image processing algorithms *already implemented*
  - New image processing algorithms can also be implemented easily
  - Nice click-and-drag interface

Wayne Rasband (right)
ImageJ: Key Features

- **Interactive tools** for image processing of images
  - Supports many image file formats (JPEG, PNG, GIF, TIFF, BMP, DICOM, FITS)

- **Plug-in mechanism** for implementing new functionality, extending ImageJ

- **Macro language + interpreter**: Easy to implement large blocks from small pieces without knowing Java
ImageJ Software Architecture

- ImageJ uses Java’s windowing system (AWT) for display.
- Programmer writes plugins to extend ImageJ.
  - Already implemented plugins available through ImageJ’s plugins menu.
ImageJ Plugins

- **Plugins**: Java classes that implement an interface defined by ImageJ

- Two types of plugins
  - **Plugin**: Requires no image to be open first
  - **PlugInFilter**: Passed currently open image, operates on it

- We will mostly focus on **PlugInFilters**

- Two methods defined
  - `int setup(String arg, ImagePlus im)`: Does initialization, verifies plugin capabilities matches input image
  - `int run(ImageProcessor ip)`: Does actual work. Passed image (ip), modifies it, creates new images
First ImageJ Example: Invert Image

- **Task:** Invert 8-bit grayscale \((M \times N)\) image
- Basically, replace each image pixel with its complement

\[
I(u, v) \leftarrow 255 - I(u, v)
\]

- We shall call plugIn **My_Inverter**
  - Name of Java Class: **My_Inverter**
  - Name of source file: **My_Inverter.java**
  - “_” underscore makes ImageJ recognize source file as plugin
  - After compilation, automatically inserted into ImageJ menu
First ImageJ Example: Invert Image

```java
import ij.ImagePlus;
import ij.plugin.filter.PlugInFilter;
import ij.process.ImageProcessor;

public class My_Inverter implements PlugInFilter {

    public int setup (String arg, ImagePlus im) {
        return DOES_8G; // this plugin accepts 8-bit grayscale images
    }

    public void run (ImageProcessor ip) {
        int w = ip.getWidth();
        int h = ip.getHeight();

        // iterate over all image coordinates
        for (int u = 0; u < w; u++) {
            for (int v = 0; v < h; v++) {
                int p = ip.getPixel(u, v);
                ip.putPixel(u, v, 255-p); // invert
            }
        }
    }

    // end of class My_Inverter
```
Compiling ImageJ Plugins

1. Place plugIn source code (My_Inverter.java) in sub-directory of ImageJ install location <ij>/plugins/
2. Open grayscale image from samples (since plugin requires image to be open)
3. Compile in run plugin by going to menu
   Plugins->Compile and Run...

- **Note:** On startup, ImageJ loads all plugins in the <ij>plugins/sub-directory
- ImageJ can also be used with eclipse IDE (large programs)
References

- University of Utah, CS 4640: Image Processing Basics, Spring 2012
- Digital Image Processing slides by Brian Mac Namee