Image-Processing Laboratories

Image Processing I
2016
Goal

➤ Practice image processing, illustrate fundamental aspects
➤ Write some basic image-processing algorithms
➤ Apply image-processing routines to images

<table>
<thead>
<tr>
<th>Date</th>
<th>Session</th>
<th>Time</th>
<th>Topics</th>
<th>Graded</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oct 20</td>
<td>IP-LAB-0</td>
<td>10:15 - 11:00</td>
<td>Introduction</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>IP-LAB-1</td>
<td>11:15 - 13:00</td>
<td>Pixelwise + Fourier</td>
<td>Yes</td>
</tr>
<tr>
<td>Dec 1</td>
<td>IP-LAB-2</td>
<td>10:15 - 13:00</td>
<td>Digital Filtering</td>
<td>Yes</td>
</tr>
<tr>
<td>Dec 8</td>
<td>IP-LAB-3</td>
<td>10:15 - 13:00</td>
<td>Tomography</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Computer rooms

➡ Microsoft Windows: Computer of the STI
➡ 1 machine / student (Microsoft Windows)
➡ Open-source software: Eclipse, ImageJ + (Word)

<table>
<thead>
<tr>
<th>Room</th>
<th>T.A.</th>
<th>Last name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO4</td>
<td>Daniel Schmitter</td>
<td>AAA - to be defined</td>
</tr>
<tr>
<td>CO5</td>
<td>Ferréol Soulez</td>
<td>to be defined</td>
</tr>
<tr>
<td>CO6</td>
<td>Denis Fortun</td>
<td>to be defined</td>
</tr>
<tr>
<td>CM103</td>
<td>Daniel Sage</td>
<td>to be defined</td>
</tr>
<tr>
<td>CM110</td>
<td>Mike McCann</td>
<td>to be defined - ZZZ</td>
</tr>
</tbody>
</table>

Student-friendly environment for the session

➡ Software package available at 10:15
➡ Only on authenticated computers of the classrooms

Do not change!
Be on time!
Procedure

➡ Step 1: *one week before the session*

▶ Get a good understanding of the theory and read the instructions on the handouts

➡ Step 2: *10h15-13h00 in the computer room*

▶ Write the Java code and test algorithms on images
▶ Submit your *code.java* and your *report.doc*

➡ Step 3: *one week after the session*

▶ Give back your graded report and your code
IP-LAB

Rules of the laboratories

- Individual work, we encourage interactions as long as they remain oral
- Electronic or printed sharing of material is forbidden
- Neither electronic nor printed preparation is allowed
- Only on the appropriate computer (no personal laptop)
- Time limited: 13h00 (15 min. to submit)
- Negotiation are mandatory ti make up for missed laboratories

<table>
<thead>
<tr>
<th>5</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Submission</td>
<td>On-time</td>
</tr>
<tr>
<td>50</td>
<td>Code.java + report.doc</td>
</tr>
</tbody>
</table>

Grading on 60 points

A human-readable code that meets the specifications and that run in the provided environment.
Correct answers to questions and right resulting images.

http://bigwww.epfl.ch/teaching/iplab/
Image Processing in Java

ImageJ

Image Processing I
2016

Daniel Sage, BM 4.135, daniel.sage@epfl.ch
Java: a general-purpose language

Powerful language
- Object-oriented
- Multithread
- Automatic garbage collector → no pointers
- Rich standard libraries: AWT, Swing, JAI, Java3D, ...

Complete language
- Extensive compilation-time checking
- Java does GUI, Java knows images
- Java is ready for the Net

Learning
- Strong influence of C/C++
- Simple object model
- Few low-level facilities

Drawbacks
- Memory management
- Versions (Java SE 8)
- Slow?

Free online book: “Thinking in Java” by Bruce Eckel
Java: Cross-Platform

Compilation phase

**Source**
Code.java

```
public class AdvancedMath {
    public double cube(double d) {
        return d*d*d;
    }
}
```

**Java compiler**

```
0: iconst_2
1: istore_1
2: iload_1
3: sipush 1000
4: if_icmpge 44
5: iconst_2
6: istore_2
7: iload_1
```

**Bytecode**

code.class

soft.jar

**Packaging**

**Archive**
soft.jar

**Developer**

End User

MacOS  Linux  Win.
## A cup of Java: one-slide guide

<table>
<thead>
<tr>
<th>Description</th>
<th>Syntax</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comments</td>
<td>// or /* ...*/</td>
<td>/* This is my comment */</td>
</tr>
<tr>
<td>End of statement / assignment</td>
<td>; =</td>
<td>a = 4 + k;</td>
</tr>
<tr>
<td>Arithmetic and logical</td>
<td>+ - * / ++</td>
<td>n = n + 10;</td>
</tr>
<tr>
<td>operations</td>
<td>- - &amp;&amp;</td>
<td></td>
</tr>
<tr>
<td>Access to data and methods</td>
<td>.</td>
<td>w = image.getWidth();</td>
</tr>
<tr>
<td>Naming convention</td>
<td></td>
<td>int width = 256;</td>
</tr>
<tr>
<td>△ case sensitive</td>
<td></td>
<td>image.rotateClockwise();</td>
</tr>
<tr>
<td></td>
<td></td>
<td>public class Image</td>
</tr>
<tr>
<td></td>
<td></td>
<td>double PI = 3.1415;</td>
</tr>
<tr>
<td>Primitive data type</td>
<td></td>
<td>double boltzmann = 1.38E-23;</td>
</tr>
<tr>
<td>△ integer division</td>
<td></td>
<td>int a = -1;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>int c = a / 2; // gives 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>double c = a / 2.0; // gives -0.5</td>
</tr>
<tr>
<td>Math functions</td>
<td></td>
<td>double ad = (double)a;</td>
</tr>
<tr>
<td>Access to element of an</td>
<td>[...]</td>
<td></td>
</tr>
<tr>
<td>array</td>
<td></td>
<td>int n = array.length;</td>
</tr>
<tr>
<td>△ arrays start at the index 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Loop</td>
<td>for(int i=-4; i&lt;=4; i++) { a = (10 + i)*i;</td>
<td></td>
</tr>
<tr>
<td>Test</td>
<td></td>
<td>if (a==b &amp;&amp; c&lt;3) {</td>
</tr>
</tbody>
</table>

### Java Crash course
- 5 October 2016, 12:00-13:00, CM 110 (20 computers)
- 12 October 2016, 17:00-18:00, CM 110 (20 computers)
Easy IP programming

ImageAccess.class

- High-level and robust methods
- Access to the pixel
  - putPixel() getPixel()
- Access to part of an image
  - getRow() getColumn() getNeighborhood()
- Show for debugging
  - image.show("title")
- Independent of the type
  - double
- Boundary conditions
  - mirror

Example:

double d = image.getPixel(10, 10);
d = d * 2.0;
image.putPixel(10, 10, d);

http://bigwww.epfl.ch/teaching/iplab/docs/
/**
 * Applies a binary threshold.
 */

public ImageAccess threshold(ImageAccess in, double t) {
    int nx = in.getWidth();
    int ny = in.getHeight();
    ImageAccess out = new ImageAccess(nx, ny);
    double pixel = 0.0;
    for (int x = 0; x < nx; x++) {
        for (int y = 0; y < ny; y++) {
            pixel = in.getPixel(x, y);
            if (pixel > t) {
                pixel = 255.0; // white
            } else {
                pixel = 0.0; // black
            }
            out.putPixel(x, y, pixel);
        }
    }
    return out;
}
Notion of readability

Correct coding but unreadable!

```java
/*** Applies a binary threshold. */
public ImageAccess threshold(ImageAccess in, double t) {
    int nx = in.getWidth();
    int ny = in.getHeight();
    ImageAccess out = new ImageAccess(nx, ny);
    double pixel = 0.0;
    for (int x = 0; x < nx; x++) {
        for (int y = 0; y < ny; y++) {
            pixel = in.getPixel(x, y);
            if (pixel > t) { pixel = 255.0; } else { pixel = 0.0; }
            out.putPixel(x, y, pixel);
        }
    }
    return out;
}
```

Guidelines for readability

- Indentation
- Natural spacing as text
- Meaningful naming
- Simplicity (translate formula)
- Comments
- Implicit conventions (e.g. x for horizontal axis)

Guidelines for readability are:

- Use a TAB to indent
General-purpose image-processing and image-analysis software
Free, open-source, ImageJ is in the public domain
Largely used in life science, material science, microscopy, ...
Java  →  multi-platform
Allow automation by macros
Open architecture  →  extensible by plug-ins

The ImageJ ecosystem: an open platform for biomedical image analysis
J. Schindelin, C. Rueden, M. Hiner, and K. Eliceiri
Molecular Reproduction and Development, 2015

NIH Image to ImageJ: 25 years of image analysis
http://www.nature.com/nmeth/journal/v9/n7

Analyse d’image scientifique, le monde ImageJ
D. Sage, Flash Informatique EPFL, 2012
http://flashinformatique.epfl.ch/spip.php?article2526

http://rsb.info.nih.gov/ij/
ImageJ’s Community
➡ Written by one person
➡ Thousands of developers
➡ Hundreds of thousands of users

Related Projects
➡ Bio-formats
➡ OME Open microscopy
➡ TrakEM: electron microscopy

Distribution of ImageJ
➡ ImageJ (NIH)
➡ Fiji
➡ MicroManager, ...

ImageJ as library
➡ ImageJ 2.0
➡ Icy (Pasteur institute)
➡ KNIME, ...

http://imagej.net/
ImageJ - Installation

Quick and easy

▷ Ready in less than 1 minute
▷ Self-contained folder
▷ Customization
  - Memory setting
  - Third-party plugins

JVM: Java Virtual Machine

▷ Linux: pre-installed Java or use the bundled distribution with Java
▷ Windows: pre-installed Java or use the bundled distribution with Java
▷ Mac OS X: pre-installed Java

http://rsb.info.nih.gov/ij/
Pointwise operations
  ➡ Threshold, contrast, histogram
  ➡ Arithmetic operations

Spatial processing
  ➡ Linear filtering
  ➡ Morphology operations

Image transformation
  ➡ Fourier

Image analysis
  ➡ Particle analyzer
  ➡ Region of interest

File formats
  ✦ Standard formats: GIF, PNG, JPG, TIFF (8-bit, 16-bit, 32-bit, z-stack)
  ✦ Unformatted data: raw data
  ✦ Biomedical formats: DICOM, ...
  ✦ BioFormats: more than 120 formats including proprietary formats