

# Einführung

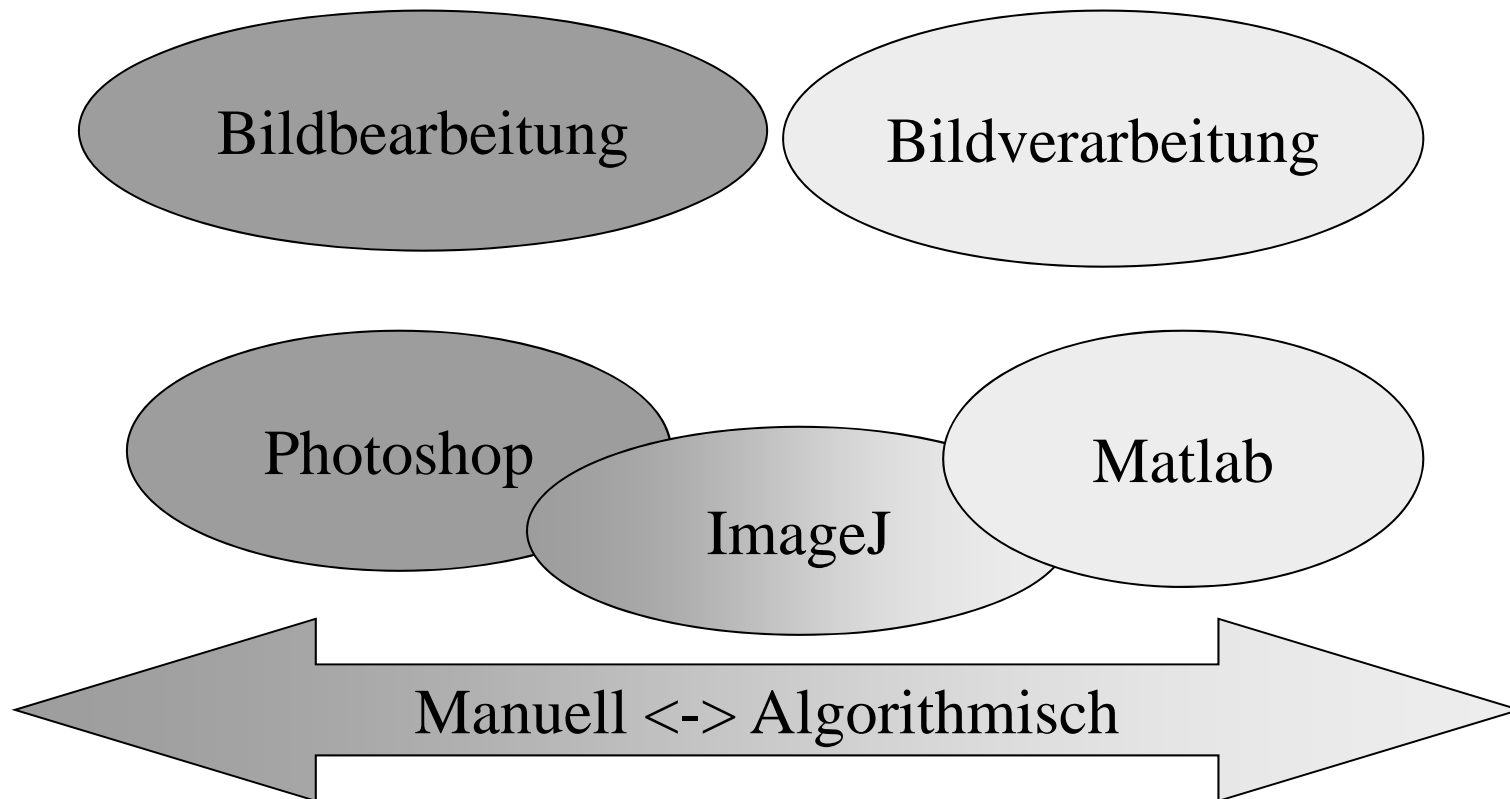
Medizin- und Bioinformatik  
Bakkalaureats-Studium  
5. Semester

Backfrieder-Hagenberg

# Bildverarbeitung

- Computer-Graphik
- Bildbearbeitung
  - manuell, semi-automatisch
  - gestalterisch
  - Setzen von Parametern
- Bildverarbeitung
  - algorithmische Verarbeitung,
  - Automatisierung angestrebt
  - Transformation, Verbesserung

# Kontext



# ImageJ

- Interaktives Bildverarbeitungspaket
- Verfügbar auf allen Plattformen
- Unterstützt:
  - Vielzahl von File-Formaten
  - 8, 16, 24 Bit Farb- und Grauwertbilder
  - Transformationen
  - Interaktive Segmentierung
- Frei verfügbar !!!

# Ressourcen

- Main Page
  - <http://rsb.info.nih.gov/ij/>
- Documentation
  - <http://rsb.info.nih.gov/ij/docs/index.html>
- Plugins
  - <http://mtd.fh-hagenberg.at/depot/imaging/imagej/>  
Von Werner Bailer, FH-Hagenberg

# Plugin

- ImageJ durch Plugins erweiterbar
- Java-Klasse
- Subfolder ‚Plugin‘ in Installations-Folder
- Integrierte Entwicklungsumgebung (rudimentär)

# Grundstruktur

```
import ij.*;
import ij.plugin.filter.PlugInFilter;
import ij.process.*;
import java.awt.*;

public class demo_ implements PlugInFilter {
    public int setup(String arg, ImagePlus imp) {
        if (arg.equals("about")) { showAbout(); return DONE;}
        return DOES_8G+DOES_STACKS+SUPPORTS_MASKING;
    }
    public void run(ImageProcessor ip) {
        Here comes my code
    }
    void showAbout() {IJ.showMessage("About Demo_...",
        "These are my strings ");
    }
}
```

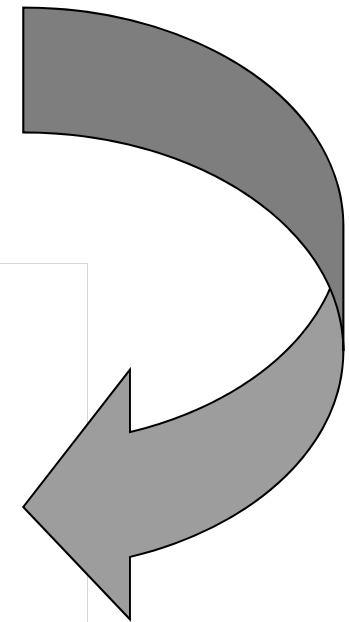
Drei  
Methoden



# Setup

- Aufruf von Info-Strings
- Rückgabe der Status-Maske

```
public int setup(String arg, ImagePlus imp) {  
    if (arg.equals("about")) {showAbout(); return DONE;}  
    return DOES_8G+DOES_STACKS+SUPPORTS_MASKING;  
}
```





# Showabout

- Definition eines Info Strings

```
void showAbout() {  
    IJ.showMessage("About Demo_...",  
        "These are my strings ");  
}
```

# Run-Methode

- Kernstück des Plugins
- Operationen auf das aktuelle Bild
- Übergabe des image-Prozessors
  - Pixel
  - Dimensionen
  - Regionen

# Run Methode

```
public void run(ImageProcessor ip) {  
    int breite=ip.getWidth();  
    int hoehe=ip.getHeight();  
    byte [] pixel_ref=ip.getPixels();  
    for (int i=0; i<breite*hoehe; i++)  
        { ...  
    }  
}
```

# Java für ImageJ

- run-Methode analog zu C-Routine
- Keine typische Java-Programmierung
- kleines Sub-set an Instruktionen
- Leitfaden „java-jump.pdf“

# Typen

- char 16 Bit unicode-character
- byte 8 Bit +/- Ganzzahl
- short 16 Bit
- int 32 Bit
- long 64 Bit
- float 32 Bit (IEEE)
- double 64 Bit (IEEE)

# Analogien zu C

- Variablen wie in C
- Zuweisungen  $=, +=, *= \dots$
- Vergleichsoperatoren  $==, >= \dots$
- Arithmetische Operatoren  $+, -, \%$
- Logische Operatoren  $\&\&, ||, !$

# Mathematische Funktionen

- im Package `java.lang.Math`
- `Math.abs()`
- `Math.sqrt()`
- `Math.sin()`

```
public static double mysin(double arg, double freq) {  
    int amp;  
    amp=4;  
    return amp*Math.sin(freq*arg);  
}
```

# Felder

- Definieren und Allokieren von Feldern

- ein-dimensionales Feld

```
byte[] a=new byte[100];
```

- zwei-dimensionales Feld

```
byte [][] b=new byte [50][100]
```

- Zugriff

```
a[12]=16; b[49][92]=4;
```