

Bildverarbeitung und Algorithmen

Prof. Dr. Wolfgang Konen

**Einführung in
ImageJ**



Tools in der Bildverarbeitung

□ Früher:

- fast so viele BV-Tools wie BV-Formate
- Lösungen nur schwer auf andere Systeme übertragbar

□ Der Grund: Performance

- BV = viele Pixel = hohe Rechenleistung

□ Heute: Extensibility/Plattformunabhängigkeit im Vordergrund

Was ist ImageJ? (1)

□ **Open-Source-Projekt, initiiert durch Wayne Rusband, NIH**

□ **Philosophie:**

- kompakter, aber dennoch mächtiger Kern von Basisoperationen
 - ◆ Bilder einlesen / schreiben
 - ◆ Bilder **b**earbeiten
 - ◆ Bildverbesserung
 - ◆ Bilder analysieren
- einfach erweiterbar durch Java-Plugins, die on-the-fly erstellt und dem System hinzugefügt werden können
- inzwischen große User-Gemeinde, darum große Zahl an Plugins
- <https://imagej.nih.gov/ij/>
- <https://imagej.nih.gov/ij/plugins>
-

ImageJ – Allgemeine Eigenschaften

□ **Runs Everywhere:**

- ImageJ läuft auf
 - ◆ Linux,
 - ◆ Mac OS X,
 - ◆ Windows.

□ **Open Source:**

- ImageJ and its [Java source code](#) are freely available and in the [public domain](#). No license is required.

□ **User Community:**

- ImageJ has a large and knowledgeable worldwide user community. More than 1200 users and developers subscribe to the [ImageJ mailing list](#).

Wie kann man ImageJ einsetzen?

auf 4 Ebenen

- 1. ImageJ als Toolbox nutzen (manuelle Bildbearbeitung und Analyse)**
- 2. Macros aufzeichnen oder schreiben**
 - komplexere Abläufe automatisieren
- 3. ImageJ-Plugins schreiben**
 - erfordert Java-Kenntnisse
 - Rapid Prototyping >> schnelle Integration
- 4. ImageJ in eigene Applikationen einbinden**

Wir nutzen vor allem 1. und 3., manchmal auch 2.

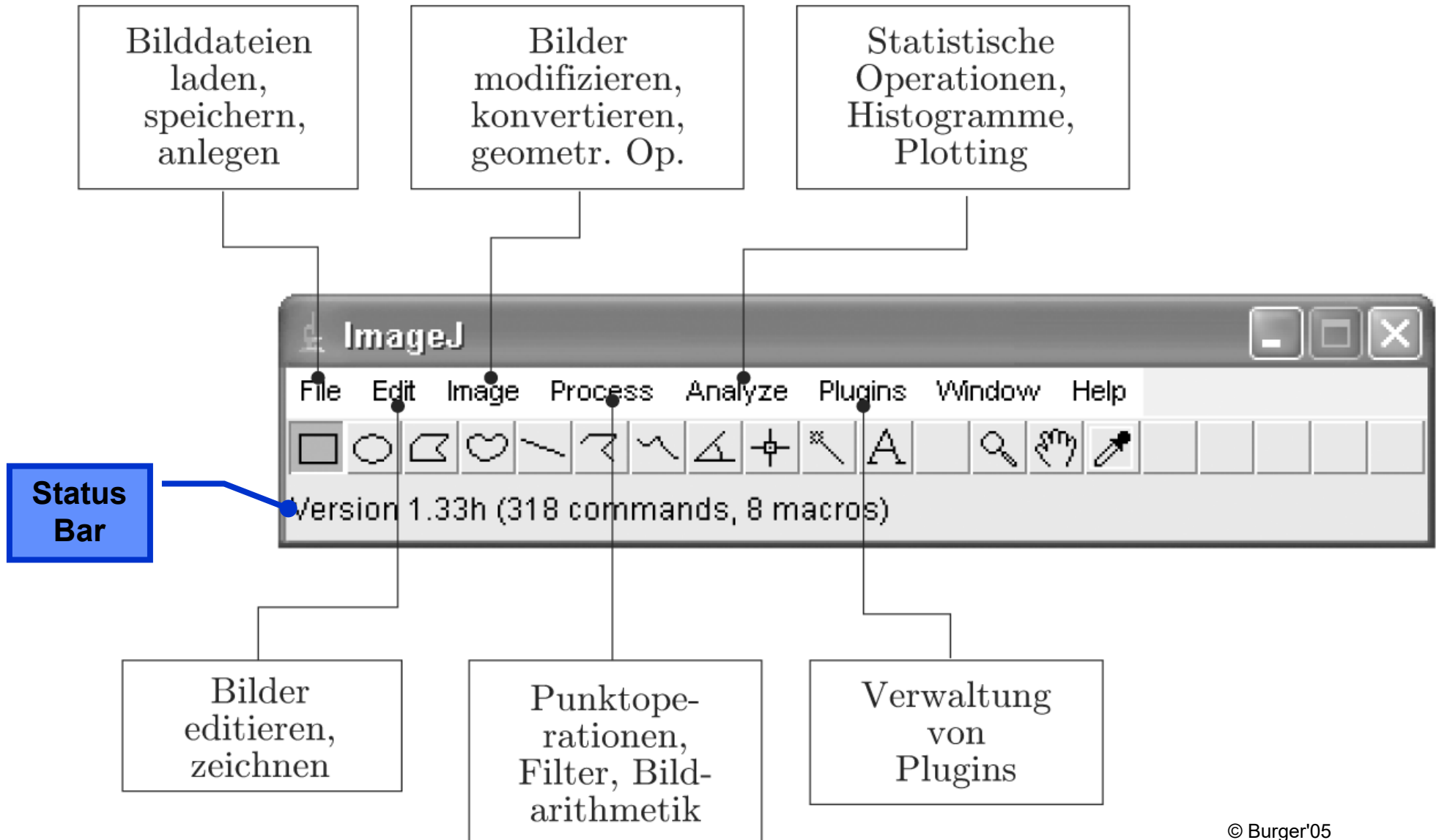
Basiskonzepte ImageJ

The screenshot displays the ImageJ software interface with several open windows:

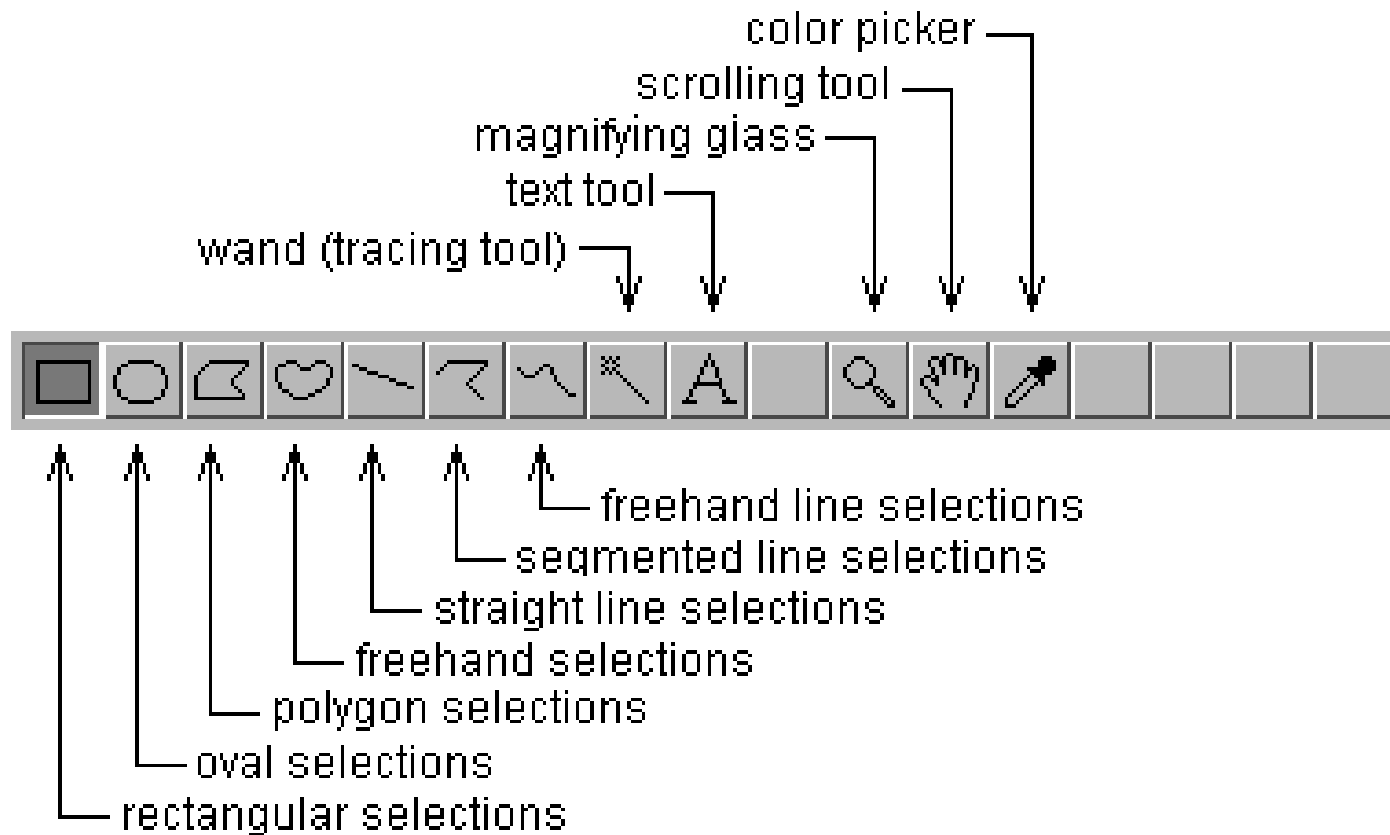
- ImageJ**: The main application window with a menu bar (File, Edit, Image, Process, Analyze, Plugins, Window, Help) and a toolbar. The status bar shows "Location = (144,85), value=0,66,200".
- mri.gif (2:1)**: A grayscale MRI scan of a brain slice with a yellow outline highlighting a specific region.
- FluorescentCells.jpg**: A multi-color fluorescence image showing cells with green and red filaments and blue nuclei.
- 1D Gel**: A grayscale image of a gel electrophoresis pattern with a yellow box highlighting a vertical lane.
- blobs.gif**: A grayscale image with red blobs representing segmented objects.
- Drawing...**: A drawing window showing the same red blobs with black outlines and numbered from 1 to 6.
- Results**: A table displaying the analysis results for the segmented blobs.

File	Area	Mean	Major	Minor	Angle
1	425	195.95	28.02	19.31	71.22
2	426	201.84	31.33	17.31	17.59
3	676	198.99	35.72	24.10	166.25
4	361	197.21	23.70	19.39	172.83
5	610	189.72	46.20	16.81	64.39
6	641	192.62	39.75	20.53	122.64

Menus



Tools

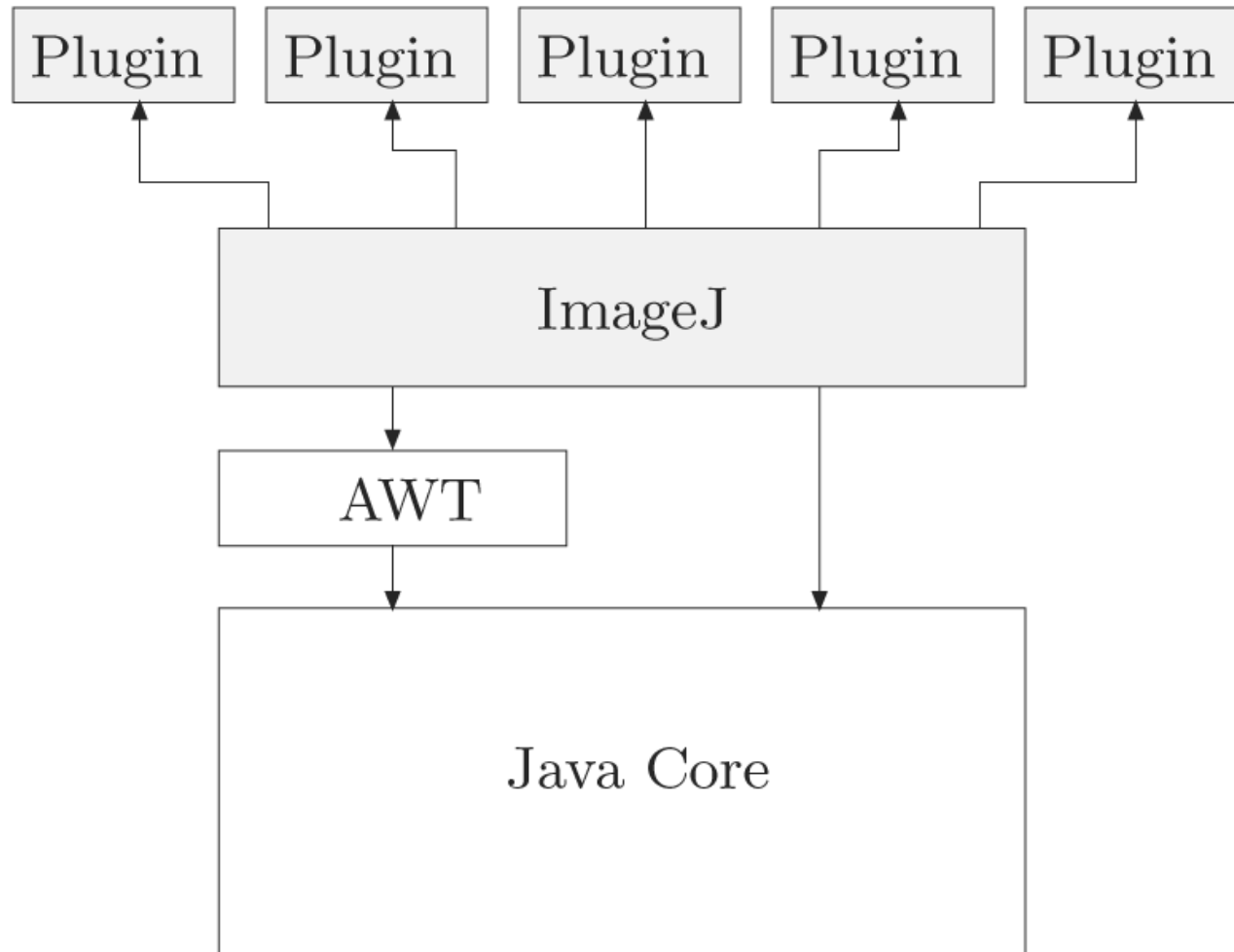


Software Demonstration

Walk Through
ImageJ +
Documentation

□ ... dann Übungen Ü1-Ü4

Software-Konzept



Einführung Plugin-Programmierung

Die Einführung erfolgt anhand der Handouts [Burger05] u. [Bailer01] sowie praktischer Übungen, und zwar nach folgendem Fahrplan:

- Walk-Through S. 31-35 aus [Burger05]: **Erstes Plugin-Beispiel**
- ... selber ausprobieren am Rechner
- Walk-Through S. 4-8 aus [Bailer01]: **ImageJ Class Structure**
- Walk-Through S. 12-16 aus [Bailer01]: **Image Representation, ROI**
- ... dann Übungen Ü5-Ü6
- [Bailer01, S. 9-10]: **Inverter_-Beispiel erweitert: ROI, schneller**
- ... dann Übungen Ü7-Ü9

Zusammenspiel der Klassen ImagePlus, ImageProcessor, ImageStack

ImagePlus

```
protected Image img;  
protected ImageProcessor ip;  
private ImageStack stack;  
private int currentSlice;
```

ImageStack

```
private int nSlices;  
private Object[] stack;  
private String[] label;  
private ImageProcessor ip;
```

ImageProcessor

```
protected Image img;
```

ByteProcessor extends ImageProcessor

```
protected byte[] pixels;  
public Image createImage();
```

ColorPr extends

```
protected int[] pixels;  
public Image createImage();
```

holds the pixels[]-data for the nSlices slices

only placeholder, getProcessor() writes on it

Nützliche weitere Hinweise

- Mit Plugins – New – PluginFilter... bekommt man direkt ein Template mit den richtigen `imports` und Methoden
- Ein `Rectangle` ist eine Klasse mit den public Members
 - ◆ `x,y`: obere linke Ecke
 - ◆ `width,height`
- Mit `Rectangle ip.getRoi()` bekommt man das Bounding Rectangle einer ROI im `ImageProcessor ip` zurück
- **ACHTUNG:** `ROI implus.getRoi()` aus der Klasse `ImagePlus` ist eine andere Methode (!!)
- Der Plot erscheint in einem `PlotWindow` [Bailer01, 6.6.3] erst, nachdem dessen Methode `draw()` aufgerufen wurde (!)