



Medientechnik

Übung 4

Java Bildbearbeitung - Teil 1

Planung

Nr	Zeitraum	Thema
1	20.04. – 25.04.	Bildretusche mit Gimp
2	27.04. – 01.05.	GUI Programmierung
3	04.05. – 08.05.	Model-View Controller
4	18.05. – 22.05.	Bildfilter – Teil 1
5	26.05. – 29.05.	Video & Film Theorie Bildfilter – Teil 2
6	01.06. – 03.06.	Audio-Aufnahme und –Bearbeitung Video & Film Theorie
7	08.06. – 11.06	Audio-Aufnahme und –Bearbeitung

Java2D – Bildbearbeitung

- Teil 1:
 - Bilder laden
 - Konvolution
 - Filter von JH Labs
- Teil 2 (nächste Woche):
 - Eigene Filter

Bilddateien per Code laden

- Einlesen von Bilddateien umfasst komplexe Algorithmen
 - Decodieren des Bildformats
 - Einlesen lokal aus Datei oder über eine URL
 - Berücksichtigung von langsamen Festplatten- und Netzzugriffen
- Observer-Modell: Anzeigefunktion wird wieder aufgerufen, wenn Daten nachgeladen sind

Bilder laden mit Java

- Java: Diverse Möglichkeiten zum Laden eines Bildes (Image)
 - Standard-AWT-Methode (`MediaTracker`)
 - Swing-Methode (`ImageIcon`)
- Laden eines `BufferedImage`:
 - `ImageIO`
 - Spezielle Codecs (`Sun-JPEG-Codec` meist in Standardinstallation enthalten)

BufferedImage laden

Alternative 1

```
public BufferedImage loadImage(String filename)
    throws IOException{
    return ImageIO.read(new File(filename));
}
```

Alternative 2

```
import com.sun.image.codec.jpeg.JPEGCodec;
import com.sun.image.codec.jpeg.JPEGImageDecoder;
import sun.awt.image.ImageFormatException;

[...]

public BufferedImage loadImage(String filename)
    throws IOException, ImageFormatException {
    InputStream in = new FileInputStream(filename);
    JPEGImageDecoder decoder = JPEGCodec.createJPEGDecoder(in);
    BufferedImage img = decoder.decodeAsBufferedImage();
    in.close();
    return img;
}
```

Fehlermeldung bei Sun imports

- aufgrund einer Eclipse-Einstellung werden teilweise Fehler (“Access restriction...”) bei Verwendung der Sun Pakete angezeigt
- Lösung:
 - Window → Preferences → Java → Compiler → Errors/Warnings → Deprecated and restricted API
 - auf „Warning“ oder „Ignore“ stellen

Java - Bildfilter

Klasse	Hilfsklassen	Effekte	"in place"? (src = dst)
ConvolveOp	Kernel	Weich- und Scharfzeichnen, Kantenerkennung	nein
Affine TransformOp	java.awt.geom. AffineTransform	Geometrische Transformationen	nein
LookupOp	LookupTable, ByteLookupTable, ShortLookupTable	Inversion, Farbtrennung, Aufhellung, Thresholding	ja
RescaleOp		Aufhellen, Abdunkeln	ja
Color ConvertOp	java.awt.Color. ColorSpace	Farbraum-konversion	ja

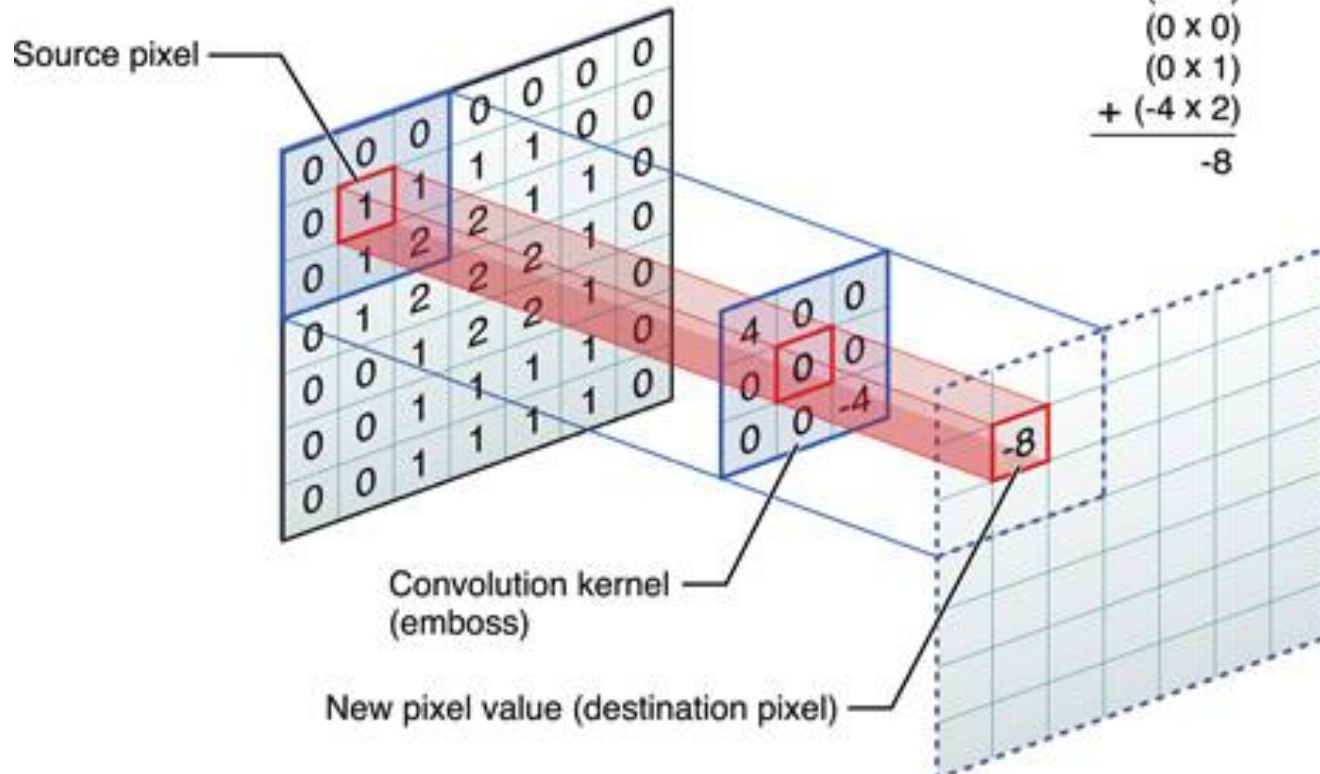
Konvolution

- Idee:
 - Farbwert eines Pixels und umliegender Pixel wird kombiniert
 - Die Kombination wird mit einer sog. “Kernel Operation” gemacht
≈ Matrix Berechnung
- Konvolution = “Faltung”
Kernel = “Faltungsmatrix”

Konvolution: Allgemeines Beispiel

Center element of the kernel is placed over the source pixel. The source pixel is then replaced with a weighted sum of itself and nearby pixels.

$$\begin{array}{r}
 (4 \times 0) \\
 (0 \times 0) \\
 (0 \times 0) \\
 (0 \times 0) \\
 (0 \times 1) \\
 (0 \times 1) \\
 (0 \times 0) \\
 (0 \times 1) \\
 (0 \times 1) \\
 \hline
 + (-4 \times 2) \\
 \hline
 -8
 \end{array}$$



Kernel

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

Identitäts-Kernel

$$\begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}$$

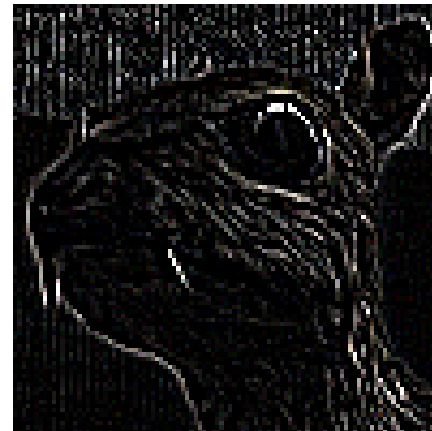
$$\begin{pmatrix} 4 & 12 & 0 \\ 0 & \mathbf{1} & 0 \\ 0 & 0 & 45 \end{pmatrix} \times \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} = \mathbf{1}$$

Achtung: Identitäts-Kernel ist keine Identitätsmatrix!

Kantenerkennung

- Edge detection kernel (Beispiel):

$$\begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$



Quelle: Michael Plotke, [https://en.wikipedia.org/wiki/Kernel_\(image_processing\)](https://en.wikipedia.org/wiki/Kernel_(image_processing))

Kantenerkennung

Rechenbeispiel:

$$\begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix} \times \begin{pmatrix} 1 & 10 & 1 \\ 1 & \mathbf{10} & 1 \\ 1 & 10 & 1 \end{pmatrix} \Rightarrow$$

$$\begin{aligned} & (-1) * 1 + (-1) * 10 + (-1) * 1 + \\ & (-1) * 1 + 8 * \mathbf{10} + (-1) * 1 + \\ & (-1) * 1 + (-1) * 10 + (-1) * 1 \\ & = \\ & \mathbf{54} \end{aligned}$$

Weitere Kernel

- Schärferen

$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$



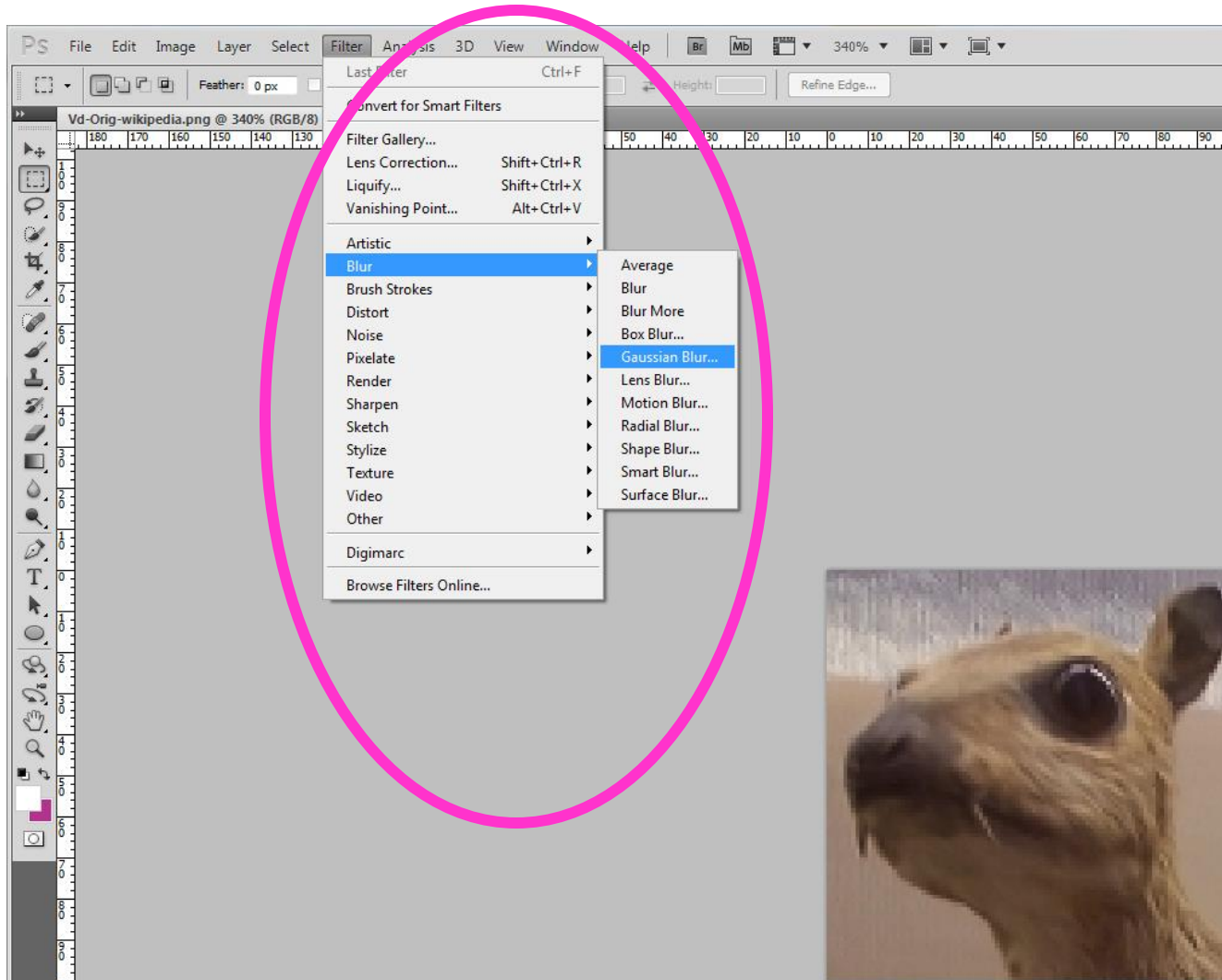
- Gaussscher Weichzeichner

$$\frac{1}{16} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$$



Quelle: Michael Plotke, [https://en.wikipedia.org/wiki/Kernel_\(image_processing\)](https://en.wikipedia.org/wiki/Kernel_(image_processing))

Konvolution in der Praxis



Konvolution: Probleme

- Wie gehen wir mit dem Bildrand um?
 - Ignorieren / Unverändert lassen
 - Erweitern (Pixel wiederholen)
 - Pixel von der gegenüberliegenden Seite verwenden.
 - Bild zuschneiden
- Werte außerhalb des Farbraums?
 - Normalisieren

Konvolution in Java ausführen

- Hilfsklasse für ConvolveOp (Konvultionsfilter in Java)

1. Werte Array anlegen:

```
float[] values = {  
    1/9f, 1/9f, 1/9f,  
    1/9f, 1/9f, 1/9f,  
    1/9f, 1/9f, 1/9f  
};
```

2. Kernel konstruieren mit Breite, Höhe und Werte Array:

```
Kernel kernel = new Kernel(3, 3, values);  
ConvolveOp cOp= new ConvolveOp(kernel);
```

3. Filter anwenden:

```
ConvolveOp convolutionFilter= new ConvolveOp(kernel);  
convolutionFilter.filter( sourceImage,  
                        destinationImage);
```

Material / Tutorials

- <http://www.javaworld.com/article/2076764/java-se/image-processing-with-java-2d.html>
- http://www.tutorialspoint.com/java_dip/index.htm
- <http://www.jhllabs.com/ip/blurring.html>

Weitere Filter

Beispiel JH Labs

- Color Adjustment Filters
- Distortion and Warping Filters
- Effects Filters
- Texturing Filters
- Blurring and Sharpening Filters
- Edge Detection
- Transitions
- Alpha Channel Filters



[TwirlFilter](#) - Distort an image by twisting



[WarpFilter](#) - A general grid image warp



[WaterFilter](#) - Simulate water ripples

Effects Filters



[BlockFilter](#) - Mosaic or pixellate an image



[BorderFilter](#) - Add a border



[ChromeFilter](#) - Simulate chrome



[ColorHalftoneFilter](#) - Color halftoning effect.



[CrystallizeFilter](#) - Make an image look like stained glass

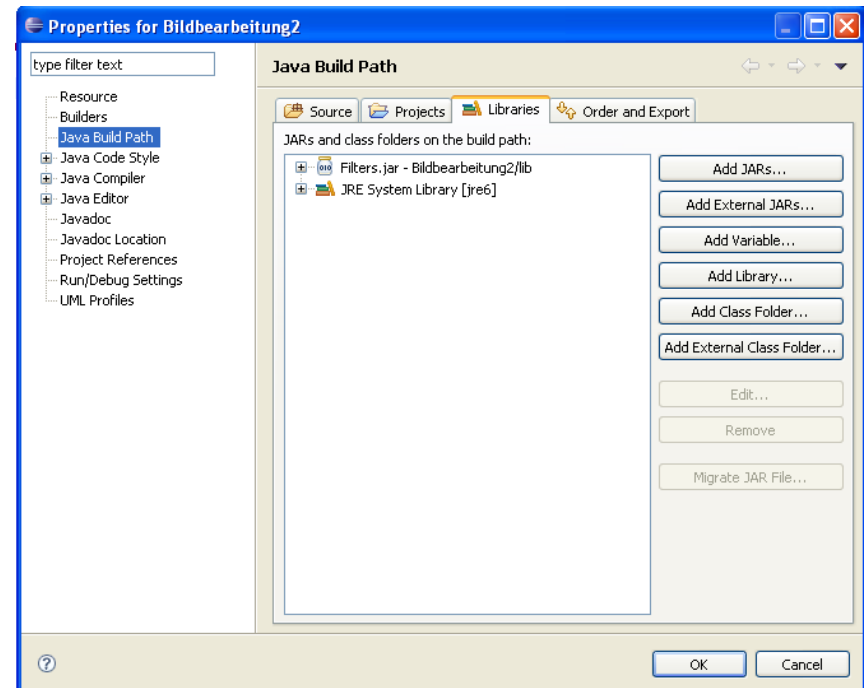


[EmbossFilter](#) - Simple embossing

Quelle: <http://www.jhlab.com/ip/filters/index.html>

JHLabs Filter einbinden

- Filter.jar herunterladen
<http://www.jhlabs.com/ip/filters/download.html>
 - In Projekt einbinden
 - neuer Ordner “lib”, jar reinkopieren
- Project Properties → Java Build Path → Add External Jar
- Filter aussuchen und verwenden



Filter verwenden



Übersicht: <http://www.jhllabs.com/ip/filters/index.html>

```
import com.jhllabs.image.SolarizeFilter;
import javax.imageio.ImageIO;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;

public class UseJHLabsFilterMain {
    public static void main(String[] args) {
        String filename = "test.jpg";
        try {
            BufferedImage destImage =
                ImageIO.read(new File(filename));
            BufferedImage tempImage = new BufferedImage(
                destImage.getColorModel(),
                destImage.getRaster(),
                destImage.getColorModel().isAlphaPremultiplied(),
                null
            );
            SolarizeFilter solarizeFilter = new SolarizeFilter();
            solarizeFilter.filter(tempImage, destImage);
        } catch (IOException e) { e.printStackTrace(); }
    }
}
```

Übungsblatt 3

- Bildfilter Funktionalität implementieren
- Bildverarbeitung (Theorie)


 LMU München, Übungen zur Vorlesung Medieninformatik im Sommersemester 2015 

Übungsblatt 3: Bildfilter


Aufgabe 1: **4 Punkte**



a) Erklären Sie kurz den Aufbau eines Histogramms (z. B. Achsenbeschriftungen). Welche Informationen kann man aus einem Histogramm ablesen? (1 Punkt)

b) Deuten Sie das folgende Histogramm. Wie sieht das Bild in etwa aus? Welche Probleme gibt es? (1 Punkt)



c) In der Vorlesung haben Sie die „Rule of Thirds“ kennengelernt. Das folgende Bild wurde entsprechend in 3 horizontale Abschnitte aufgeteilt. Zeichnen Sie zwei Histogramme (grobe Zeichnung genügt) für das obere und mittlere Drittel und erklären Sie Ihre Überlegungen dazu. (2 Punkte)



 LMU München, Übungen zur Vorlesung Medieninformatik im Sommersemester 2015 

Aufgabe 2: Konvolution

3 Punkte

a) Erklären Sie die Funktionsweise einer Konvolution (Faltung) mit Hilfe einer Kernelmatrix auf Bildern. Welche Probleme treten an den Rändern auf? (1,5 Punkte)

b) Was passiert bei der Anwendung der folgenden Filter? Das f steht für „float“. (je 0,5 Punkte)

a)
$$\begin{bmatrix} 1/9f & 1/9f & 1/9f \\ 1/9f & 1/9f & 1/9f \\ 1/9f & 1/9f & 1/9f \end{bmatrix}$$

b)
$$\begin{bmatrix} 1f & 1f & 1f \\ 1f & -8f & 1f \\ 1f & 1f & 1f \end{bmatrix}$$

c)
$$\begin{bmatrix} -1f & -1f & -1f \\ -1f & 9f & -1f \\ -1f & -1f & -1f \end{bmatrix}$$


Aufgabe 3: Bildfilter in JAVA

19 Punkte

a) Der vorhandene Bildbetrachter aus den letzten Übungsblättern soll nun mit einigen Fotobearbeitungseffekten ausgestattet werden. Dabei sollen die Checkboxes im Menü „Filter“ der De-/Aktivierung der Effekte dienen, wobei weiterhin MVC und das Observer-Pattern eingesetzt werden sollen. Das übrige Programmverhalten wurde in Übungsblatt 2 definiert.

Aufbauend auf den bereits angelegten GUI-Elementen sollen folgende Filter selbst implementiert und nutzbar gemacht werden (falls Parameter erforderlich sind, finden Sie die passenden heraus!). Fügen Sie ggf. weitere Schaltflächen zur GUI hinzu und passen Sie die Funktionalität der Buttons an! (je 2 Punkte pro Filter)

- Black/White/Threshold-Filter



Wrap-up Quiz

1. Was ist ein BufferedImage und warum wird es benötigt?
2. Woher kommt der Begriff "Kernel"?
3. Was bedeutet Konvolution auf Deutsch?
4. Was ist der „Identitäts-Kernel“?
5. Wie sieht ein Kernel aus, mit dem man ein Bild weichzeichnen kann?
6. Kann man per Konvolution Bilder in Schwarz-Weiß umwandeln?
7. Wie heißt die Klasse in Java, mit der die Konvolution durchgeführt wird?



Vielen Dank!

WELCHE FRAGEN GIBT ES? 😊