

Erzeugung, Repräsentation  
und Speicherung von digitalen  
Bildern

## Inhalt

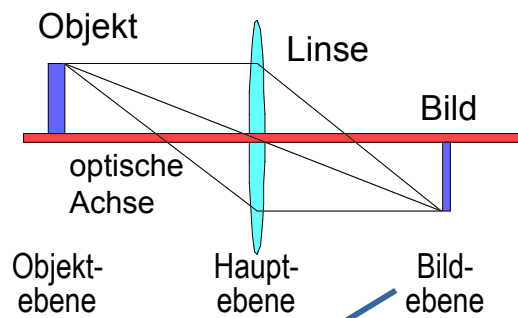
---

- ☐ Erzeugung digitaler Bilder
- ☐ Repräsentation von Bilddaten
- ☐ Speicherung von Bilddaten

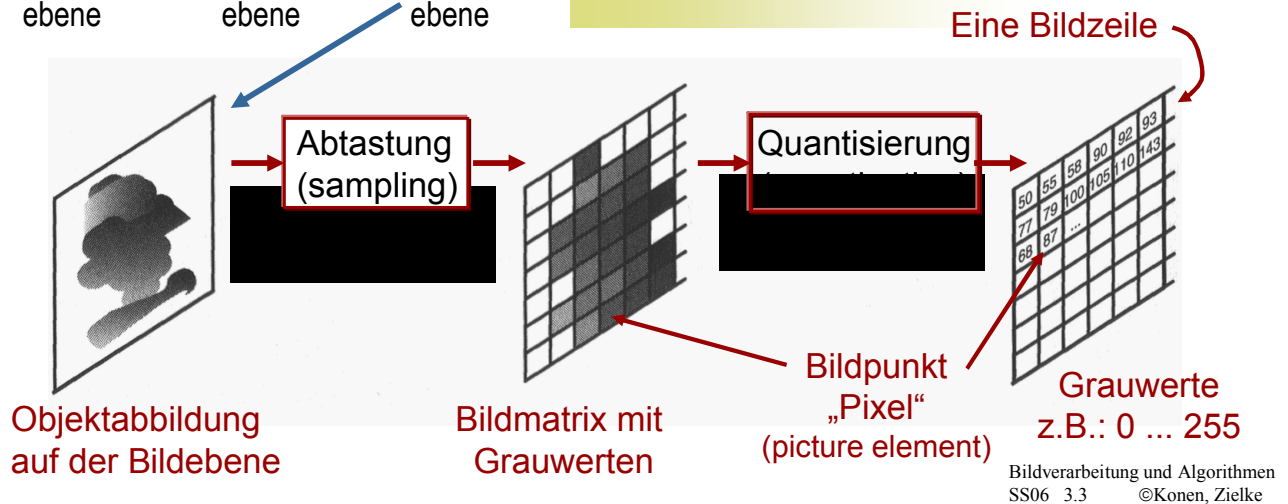


Was ist Quantisierung und wo findet  
Quantisierung bei der Bildaufnahme  
statt?

# Erzeugung digitaler Bilddaten



Bei der **Bilddigitalisierung** wird die Projektion eines Objekts auf der Bildebene (örtlich) **abgetastet**, und die einzelnen Intensitätswerte (Bildpunkte) werden **quantisiert**.



## Bildabtastung / Rasterung (image sampling)

- Die Projektionsfläche auf der Bildebene wird durch ein regelmäßiges Raster von Photosensoren abgetastet.
- Je nach Dichte der Abtastpunkte pro Flächeneinheit (bei Scannern gemessen in  $dpi = dots\ per\ inch$ ) entsteht ein hochaufgelöstes oder ein "grobkörniges" Bild.



Abtastpunkte:  
 $160 \times 120$



Abtastpunkte:  
 $80 \times 60$

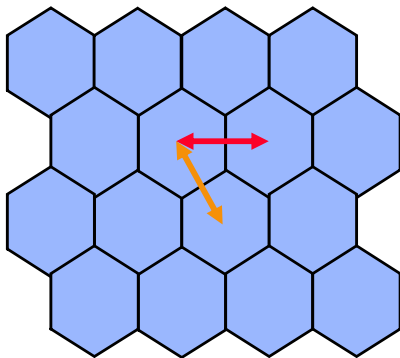
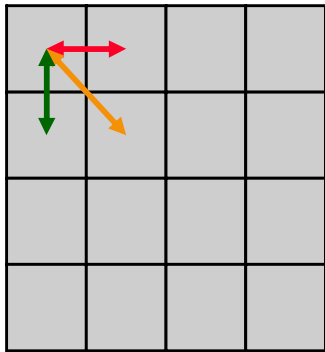


Abtastpunkte:  
 $40 \times 30$



Abtastpunkte:  
 $20 \times 15$

# Kartesisches und hexagonales Bildraster



- ☐ Technische bildverarbeitende Systeme benutzen (heute) fast ausschließlich ein kartesisches Basisgitter für das Bildraster:
  - ➔ rechteckige bzw. quadratische Pixel.
  - ➔ Pixelabstände (Zellenabstände) inhomogen.
  - ➔ einfache mathematische Behandlung (Matrix).
- ☐ Bei biologischen Augen sind die Photorezeptoren in einer "Wabenstruktur" angeordnet.
- ☐ Auch bei neueren technischen Entwicklungen (elektronische Chip- Augen) werden *hexagonale* Gitter benutzt:
  - ➔ bessere Chip-Flächenfüllung bei integrierter Signalverarbeitungselektronik.
  - ➔ Pixelabstände (Zellenabstände) sind homogen.
  - ➔ gleiche Trennflächen zwischen allen Nachbarn

Bildverarbeitung und Algorithmen  
SS06 3.5 ©Konen, Zielke

## Unterabtastung (undersampling)

- ☐ Ein Detail im Bild, das Ortsfrequenzen größer als die doppelte Abtastrate besitzt, bezeichnet man als unterabgetastet.

Korrekt abgetastete Bildzeile  
▲ die Detail-Frequenz ist wiedergegeben.

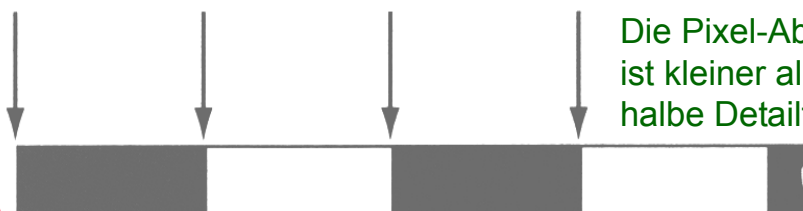


Die Pixel-Abtastrate ist doppelt so hoch wie die Detailfrequenz.

Original-Bildzeile mit Helligkeitsdetails



Unterabgetastete Bildzeile  
▲ die Detailfrequenz ist verfälscht (**Aliasing**).



Die Pixel-Abtastrate ist kleiner als die halbe Detailfrequenz.

Bildverarbeitung und Algorithmen  
SS06 3.6 ©Konen, Zielke

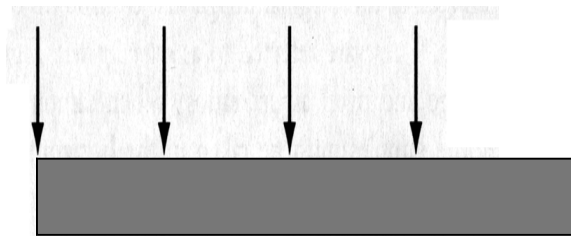
## Unterabtastung (2) (undersampling)

- ❑ Man mache sich klar, dass bei Unterabtastung je nach Abtastrate auch noch viel drastischere Effekte eintreten können: Wenn der Pfeil etwa alle zwei Kästchen kommt (also Detailfrequenz = Abtastfrequenz) und immer im "Schwarzen" des Originals liegt, dann wird der Sample&Hold-Wert immer schwarz sein!
- ❑ Dies ist der Effekt des **Aliasing**: Frequenzen, die für die Abtastrate zu hoch sind, sehen so aus wie tiefe. Die hohen Frequenzen geben sich sozusagen als jemand anderes aus, daher die Bezeichnung *Alias*.

Original-Bildzeile  
mit Helligkeitsdetails



Bei der Abtastung läuft ein Marker von li nach re. Bei jedem Pfeil wird der aktuelle Grauwert abgelesen und für das folgende Pixel fortgeschrieben (Sample & Hold)



Bildverarbeitung und Algorithmen  
SS06 3.7 ©Konen, Zielke

## Abtastung und Quantisierung (Zusammenfassung)

- ❑ Unter **Bildabtastung bzw. Rasterung** versteht man die Aufteilung des Bildes in diskrete Bildpunkte (Pixel) mit festgelegten Abständen (Diskretisierung der örtlichen Dimensionen x und y).
- ❑ Unter **Quantisierung** versteht man die Bewertung der Helligkeit (Intensität) eines Pixels mittels einer festgelegten Grauwert- bzw. Farben-Menge, z.B. natürliche Zahlen von 0 bis 255.
- ❑ **Digitalisierung** besteht aus Abtastung und Quantisierung.
- ❑ Ein digitales Bild ist immer nur eine Annäherung (Approximation) der Originalabbildung.

Bildverarbeitung und Algorithmen  
SS06 3.8 ©Konen, Zielke

# Graustufen Quantisierung (grey-level quantization)

256 Graustufen



- Bei einem digitalen Bild ist die Anzahl der möglichen Grau- bzw. Farbstufen begrenzt.
- Die minimal benötigte Anzahl von Stufen hängt vom Bildinhalt und von dem Zweck der Bildauswertung ab!



64 Graustufen  
6 Bit/Pixel



16 Graustufen  
4 Bit/Pixel



8 Graustufen  
3 Bit/Pixel



4 Graustufen  
2 Bit/Pixel



2 Graustufen  
1 Bit/Pixel

"Binärbild"

Bildverarbeitung und Algorithmen  
SS06 3.9 ©Konen, Zielke

## Technische Grenzen bei der Bildquantisierung

- Die physikalischen Grenzen des darstellbaren Intensitätsbereichs des Bildes werden in erster Linie durch den Bildsensor bestimmt  
→ Dynamikbereich.
- Weitere limitierende Faktoren:
  - Auflösung bzw. Geschwindigkeit des analog-zu-digital Wandlers (ADC)
  - Begrenzung der Übertragungsrate und/oder des Speicherplatzes für das Bild.

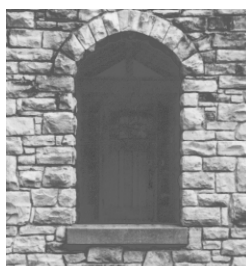
Kameraaufnahme von  
Autogen-Schweißen mit  
konventionellem CCD-Sensor:



... mit hochdynamischem  
CMOS-Sensor:

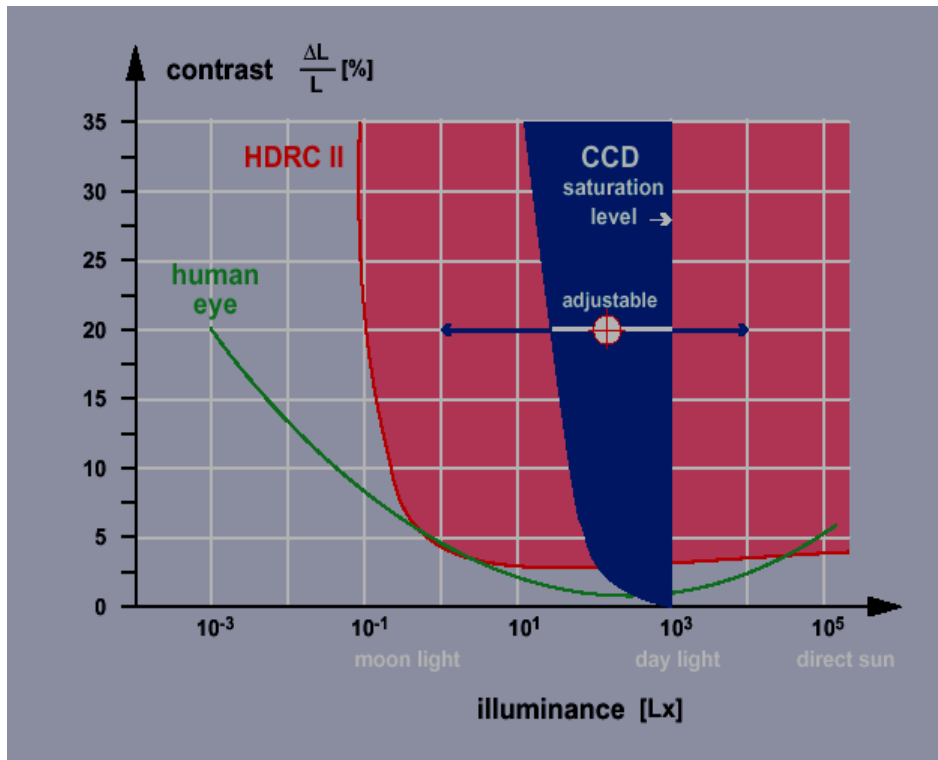


Versuch,  
unzureichende  
Bilddynamik  
zu  
kompensieren  
→



Bildverarbeitung und Algorithmen  
SS06 3.10 ©Konen, Zielke

## Kontrastauflösung des menschlichen Auges, eines Std.-CCD-Chips und eines High-Dynamic-Range-CMOS -Chips (HDRC)



□ Maximale Helligkeitsdynamik in einem Bild

- CCD: 2...3 Dekaden
- HDRC: 4...6 Dekaden
- Auge: 8...10 Dekaden

Bildverarbeitung und Algorithmen  
SS06 3.11 ©Konen, Zielke

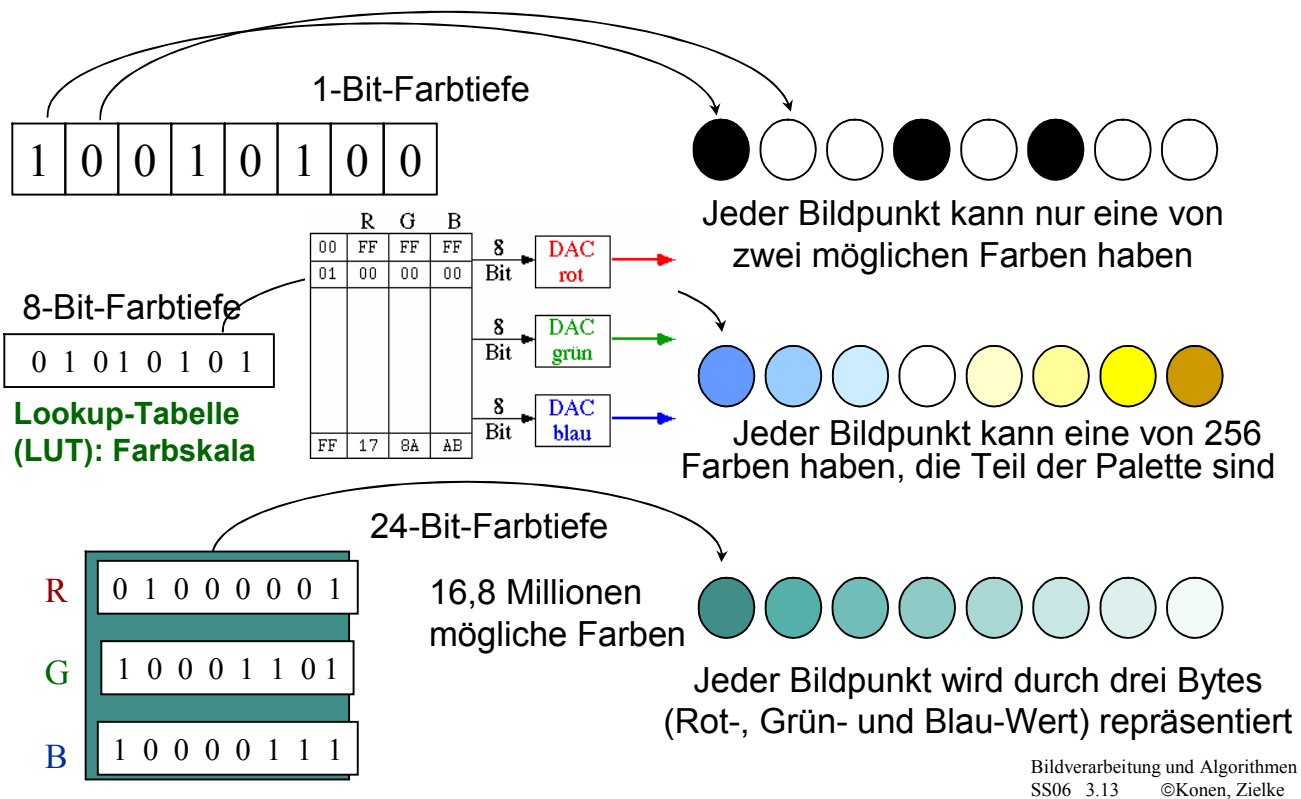
## Inhalt

- Erzeugung digitaler Bilder
- **Repräsentation von Bilddaten**
- Speicherung von Bilddaten

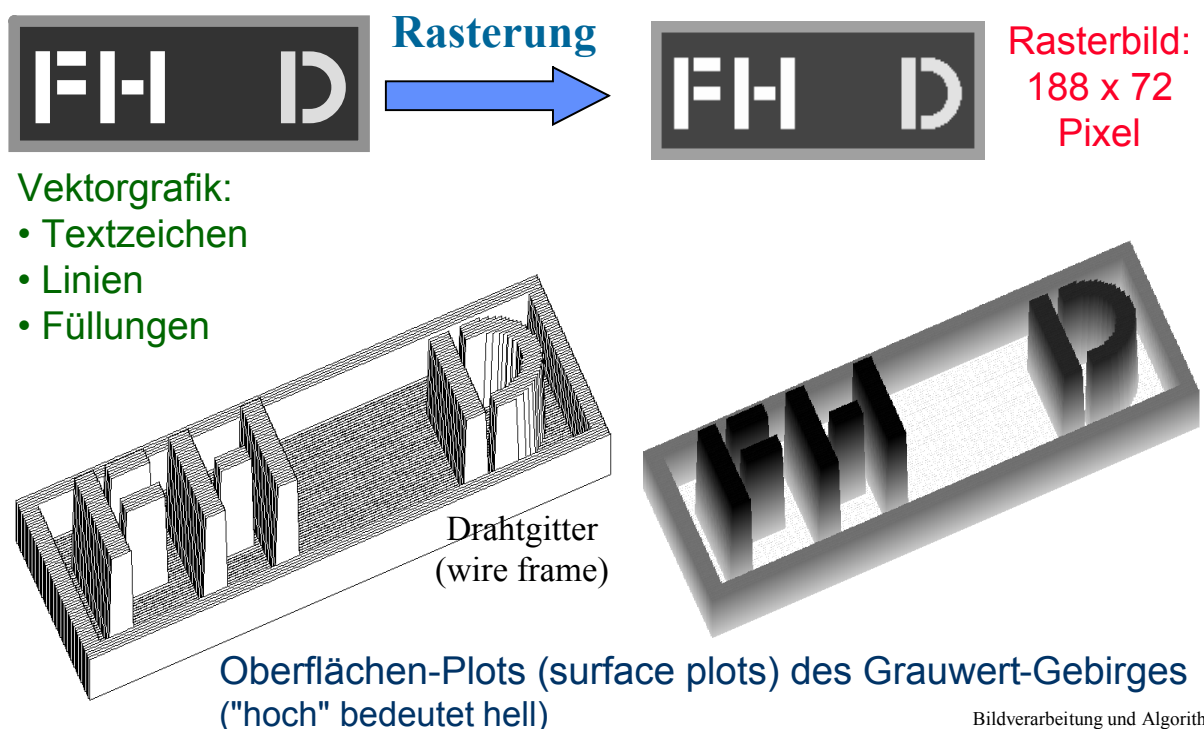
Bildverarbeitung und Algorithmen  
SS06 3.12 ©Konen, Zielke

# Darstellung von Rasterbildern im Speicher

## Bitmuster repräsentieren die Pixel-Farben



## Verschiedene Darstellungsformen der selben Bildinformation (1)

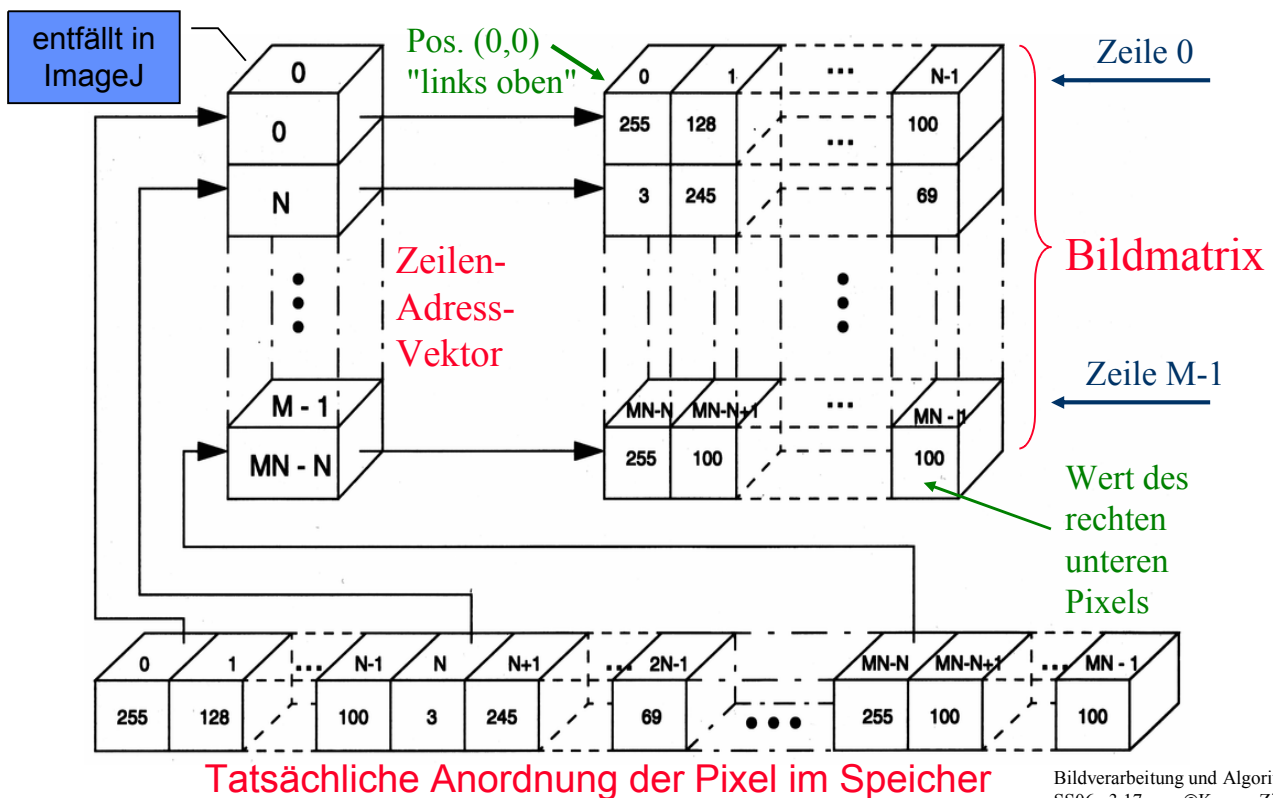








## Bilddaten-Organisation im Hauptspeicher z.B. bei dem Programmpaket ImageJ



### Klasse ImageProcessor in ImageJ (Definitionen in der Programmiersprache Java)

```
package ij.process;

public abstract class ImageProcessor {
    ...
    int width;
    int height
}

public class ByteProcessor extends ImageProcessor {
    protected byte[] pixels;
    ...
}
```

# Programmbeispiel

## "Binarisierung"

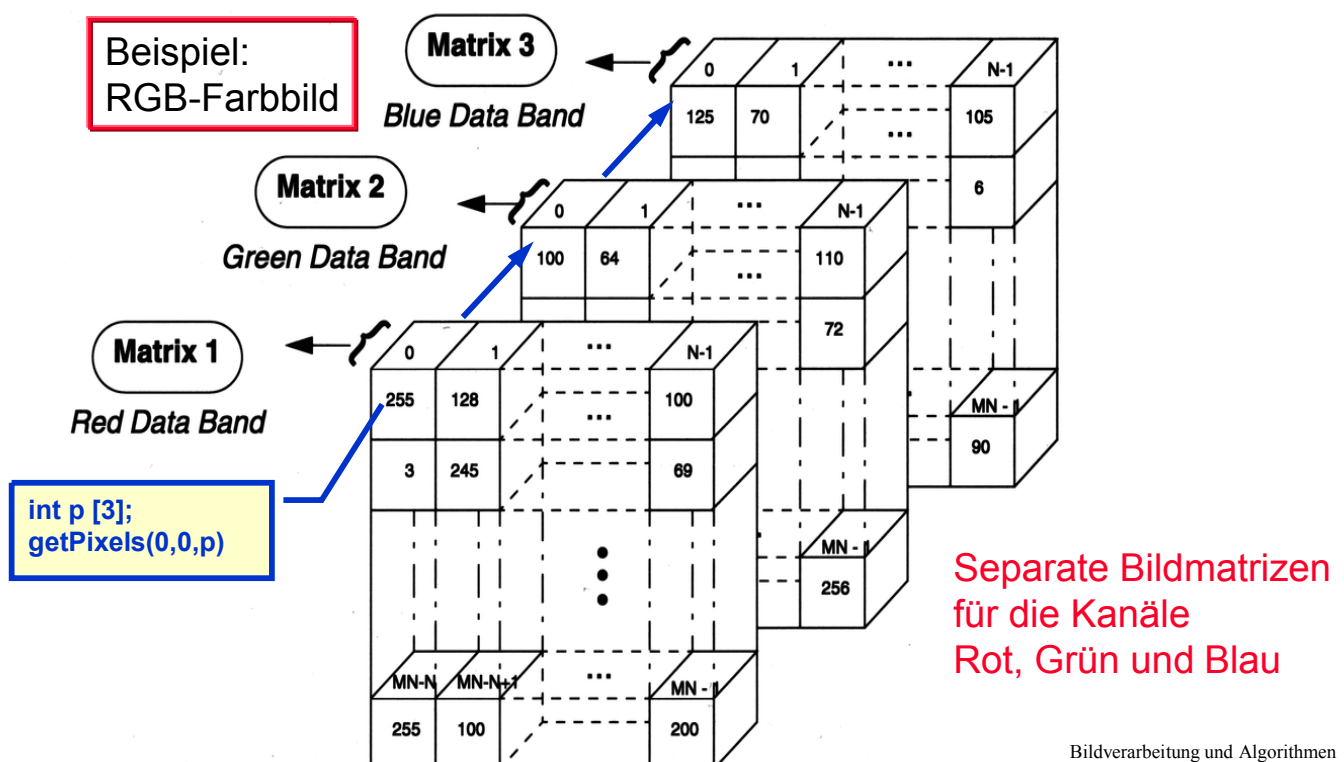
```
import ij.*;
import ij.process.*;
int white = 255;
int black = 0;
int threshval = 54;

void run(ImageProcessor ip) {
    for (r=0; r < no_of_rows; r++) {
        for (c=0; c < no_of_cols; c++) {
            int p = ip.getPixel(r,c);
            ip.putPixel(r,c,black);
            if (p > (byte) threshval)
                ip.putPixel(r,c, white);
        }
    }
}
```

Kern des Algorithmus:  
"gehe durch die gesamte Bildmatrix und setze alle Pixel, die heller als der Schwellwert sind auf weiß, die anderen auf schwarz"

Bildverarbeitung und Algorithmen  
SS06 3.19 ©Konen, Zielke

## ImageJ Datenstruktur für mehrkanalige Bilder: ColorProcessor



Bildverarbeitung und Algorithmen  
SS06 3.20 ©Konen, Zielke

- ☐ Erzeugung digitaler Bilder
- ☐ Repräsentation von Bilddaten
- ☐ **Speicherung von Bilddaten**



Welche Dateiformate für Bilder kennen Sie? Was wissen Sie über diese Formate?

Bildverarbeitung und Algorithmen  
SS06 3.21 ©Konen, Zielke

## Dateiformate zur Archivierung von Bildern auf Externspeichern Raster/Bitmap-Formate (1)

---

- ☐ GIF - Graphics Interchange Format (Compuserve)
  - meistverbreitetes Grafikformat im Internet
  - technisch gleichwertig und lizenzmäßig besser: **PNG** (Portable Networks Graphics)
  - LZW-Kompression (spezielle verlustfreie Lauflängen-Codierung (benannt nach den Entwicklern Lempel, Ziv und Welch):
    - ◆ bis ca. 5:1, bei nur einer Farbe bis ca. 30:1
    - ◆ 256 Farben (neue Version: 16 Millionen Farben)
    - ◆ bewegte Bilder möglich (animated GIF)
  - Lokale Farbskala:  
Definition für einzelnes Bild möglich (Vorrang gegenüber globaler Farbskala)
  - Daten:  
sequentieller Modus oder "interlaced mode"  
(gestattet Gesamtbilddarstellung, bevor das File vollständig übertragen ist)

Bildverarbeitung und Algorithmen  
SS06 3.22 ©Konen, Zielke

## Dateiformate zur Archivierung von Bildern auf Externspeichern Raster/Bitmap-Formate (2)

---

### □ TIFF-Format

(Tagged Image Format, Firma Aldus Cooperation)

- eines der wichtigsten Formate
- hohe Flexibilität - komplizierter Aufbau
- optionale Komprimierung (nach den Methoden Run-Length-Codierung, LZW, JPEG, Human)
- Speicherung mehrerer Bilder in einer Datei möglich

### □ BMP-Format

(Bitmap von MS-Windows)

- optionale Kompression
- RGB-Farbmodell
- Farbtabelle:  
je Eintrag 4 Byte (R,G,B,leer)
- Bilddaten: Zeilenweise Speicherung (letzte Zeile zuerst, Zeilenlänge muß Vielfaches von vier sein)

Bildverarbeitung und Algorithmen  
SS06 3.23 ©Konen, Zielke

## Dateiformate zur Archivierung von Bildern auf Externspeichern Raster/Bitmap-Formate (3)

---

### □ JPEG-Format

(Joint Photographic Expert Group Format)

- stark verbreitetes Format im Internet
- Transformation in den YUV-Farbraum
- verlustbehaftetes Kompressionsverfahren (Artefakte bei Strichzeichnungen, Kompression bis ca. 20:1)
- 24 Bit-Farbtiefe (16 Millionen Farben)
- sehr gut geeignet für Graustufen- und Farbbilder aller Farbtiefen
- ungeeignet für S/W-Bilder (höhere Datenmenge als bei GIF-Bildern)

Bildverarbeitung und Algorithmen  
SS06 3.24 ©Konen, Zielke

---



Exit-Point

Bildverarbeitung und Algorithmen  
SS06 3.25 ©Konen, Zielke

## Datenstrukturen für digitale Bilder (1)

---

In Abhängigkeit von Bildeigenschaften (z.B. Zeichnungen oder Grauwertbild), Bildverarbeitungsalgorithmen, Speicherkapazität usw. werden angepasste Datenstrukturen verwendet:

### ☐ Pixel-Matrix

- Speicherung der kompletten Bildmatrix
- einfacher direkter Zugriff zu jedem Pixel
- keine spezielle Unterstützung von Verarbeitungsalgorithmen
- keine Bildkompression

### ☐ Lauflängencodes

- Für jede Zeile Speicherung von Paaren (Anzahl gleichwertiger Pixel, Bildfunktionswert)
- Bildkompression
- Codierung von topologischen Strukturen möglich (z.B. Zusammenhangskomponenten mithilfe einer übergeordneten Graphenstruktur)

Bildverarbeitung und Algorithmen  
SS06 3.26 ©Konen, Zielke

## Datenstrukturen für digitale Bilder (2)

---

### □ Richtungscode

- Speziell zur Darstellung von Linienzügen (Konturen) in Binärbildern
- Bildkompression
- Unterstützung von Algorithmen  
(z.B. Berechnung von Fläche und Länge, Transformationen)

### □ Baumstrukturen

- Quad-Tree:
  - ♦ rekursive Zerlegung des Bildes in vier Teile, solange Bildteil inhomogen ist (Blätter = homogene Quadranten)
- Pyramiden:
  - ♦ Speicherung eines Bildes mit der Kantenlänge  $N = 2^k$  in  $k + 1$  Ebenen wachsender Auflösung
- Kompression und Unterstützung von Auswerte-Algorithmen

## Datenstrukturen für digitale Bilder (3)

---

### □ Vektorbilder

- Speicherung der parametrischen Beschreibung von Bildelementen  
(z.B. Polygone) und deren charakterisierenden Attribute  
(z.B. Füllfarbe von Flächen)
- Kompakte Darstellung
- Effektive und genaue Ausführung von Bildtransformationen  
(insbesondere geometrischen) und anderen Verarbeitungsschritten

### □ Zellenkomplexe

- Speicherung der Topologie und Geometrie durch Bildelemente unterschiedlicher Dimension  
(Verzweigungspunkte, Linien, homogene Flächen)
- Kompression von segmentierten Bildern
- Unterstützung der Bildanalyse