Bildverarbeitung mit ImageJ

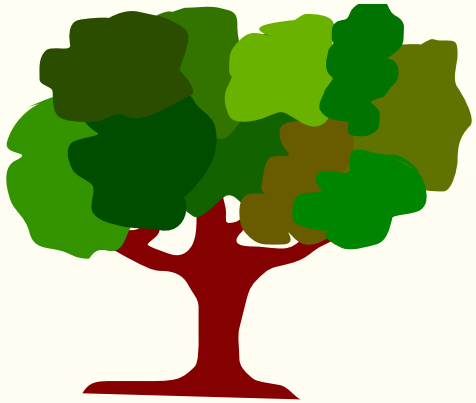
Benjamin Drayer

Lehrstuhl für Mustererkennung und Bildverarbeitung
Institut für Informatik

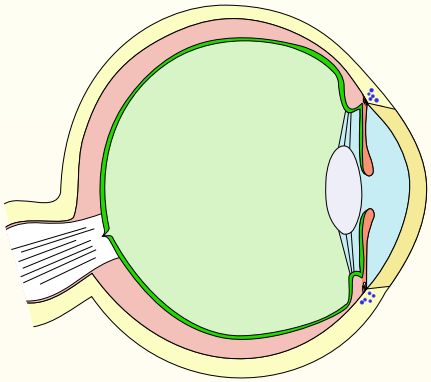
24. April 2013

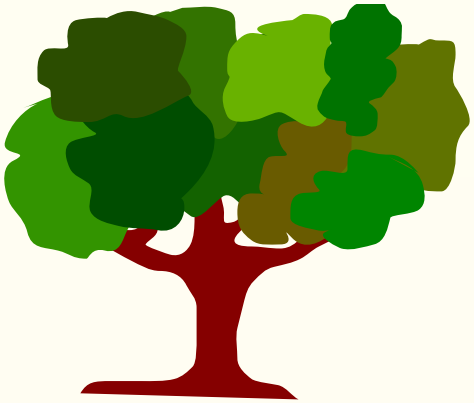
Inhalt

Wie entsteht ein Bild?

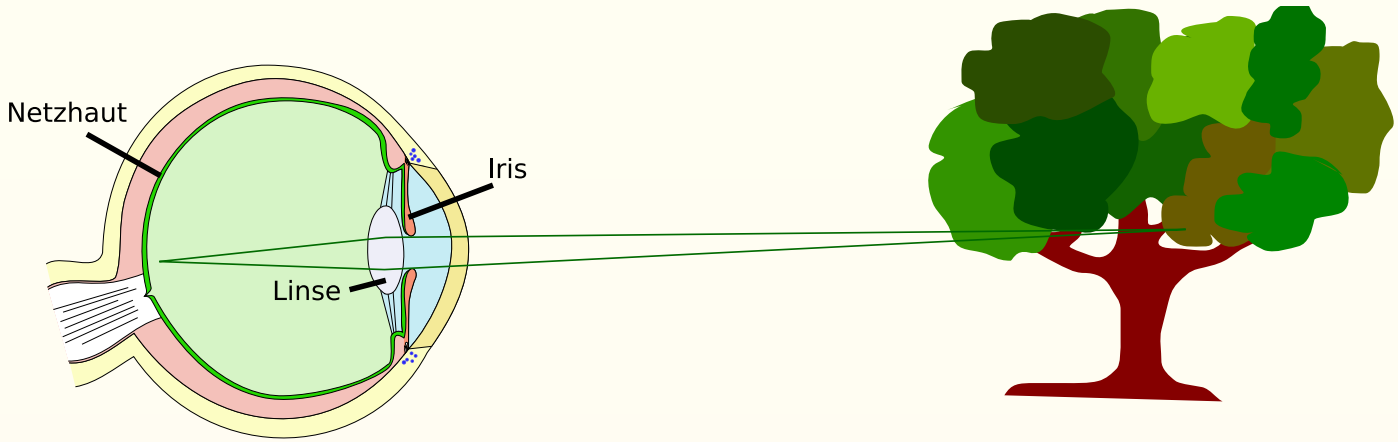


Wie entsteht ein Bild?



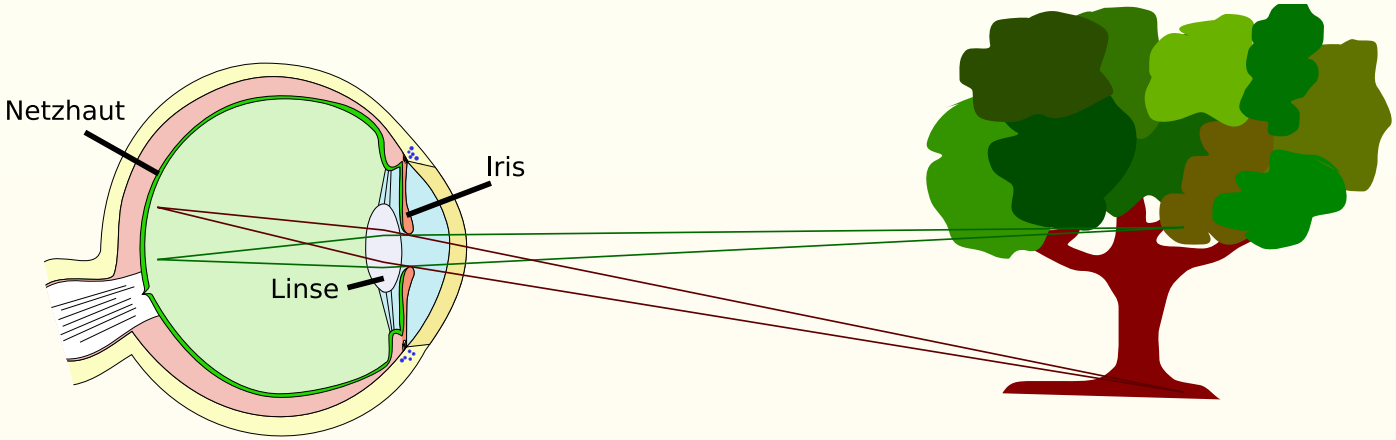


Wie entsteht ein Bild?



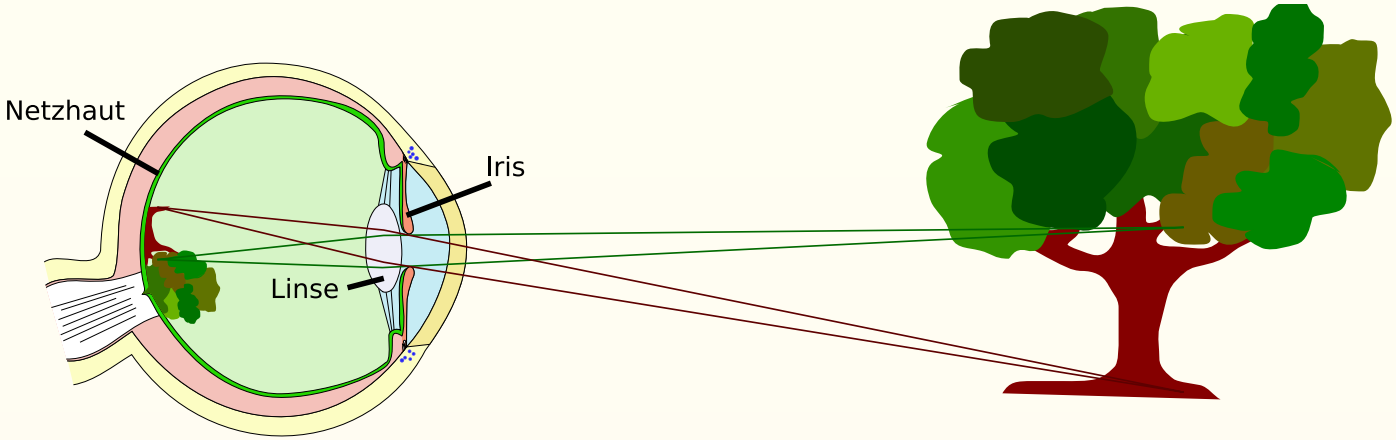
- Anteile des Spektrums des auf ein Objekts einfallenden Lichts werden reflektiert
- Die Linse projiziert das einfallende Licht auf die Netzhaut

Wie entsteht ein Bild?



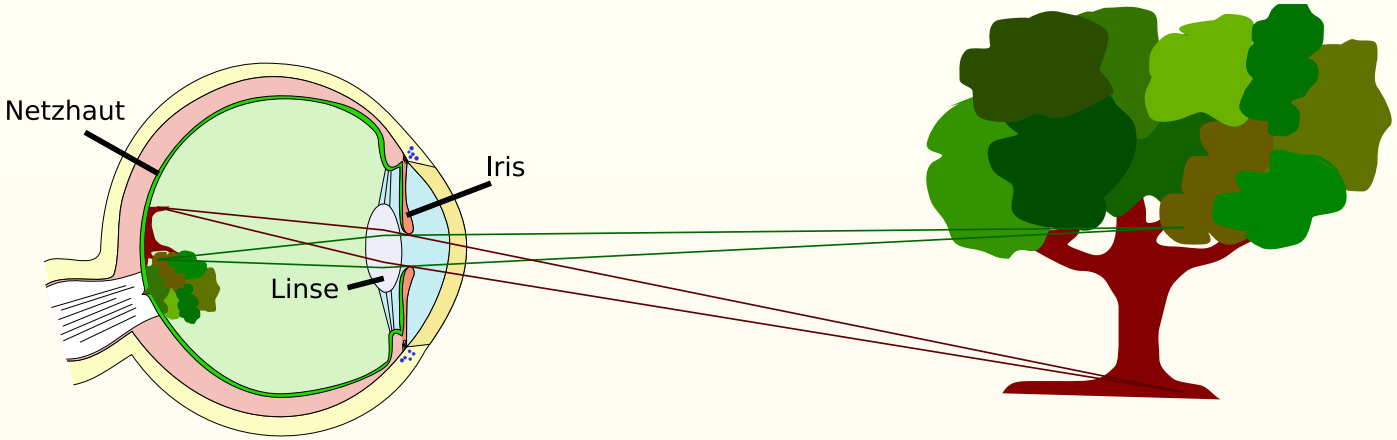
- Anteile des Spektrums des auf ein Objekts einfallenden Lichts werden reflektiert
- Die Linse projiziert das einfallende Licht auf die Netzhaut

Wie entsteht ein Bild?



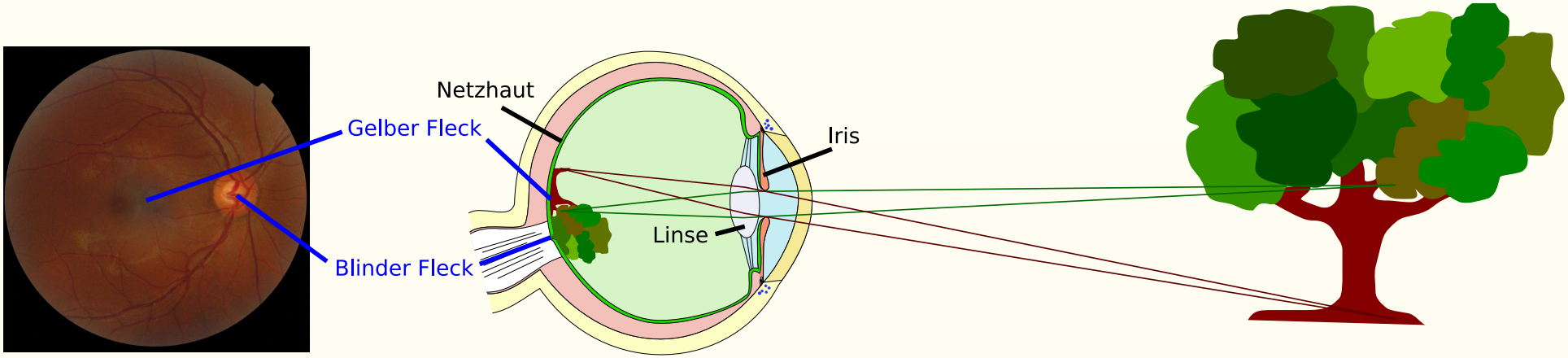
- Anteile des Spektrums des auf ein Objekts einfallenden Lichts werden reflektiert
- Die Linse projiziert das einfallende Licht auf die Netzhaut
- Das auf die Retina prozierte Bild steht auf dem Kopf

Wie entsteht ein Bild?



- Retina

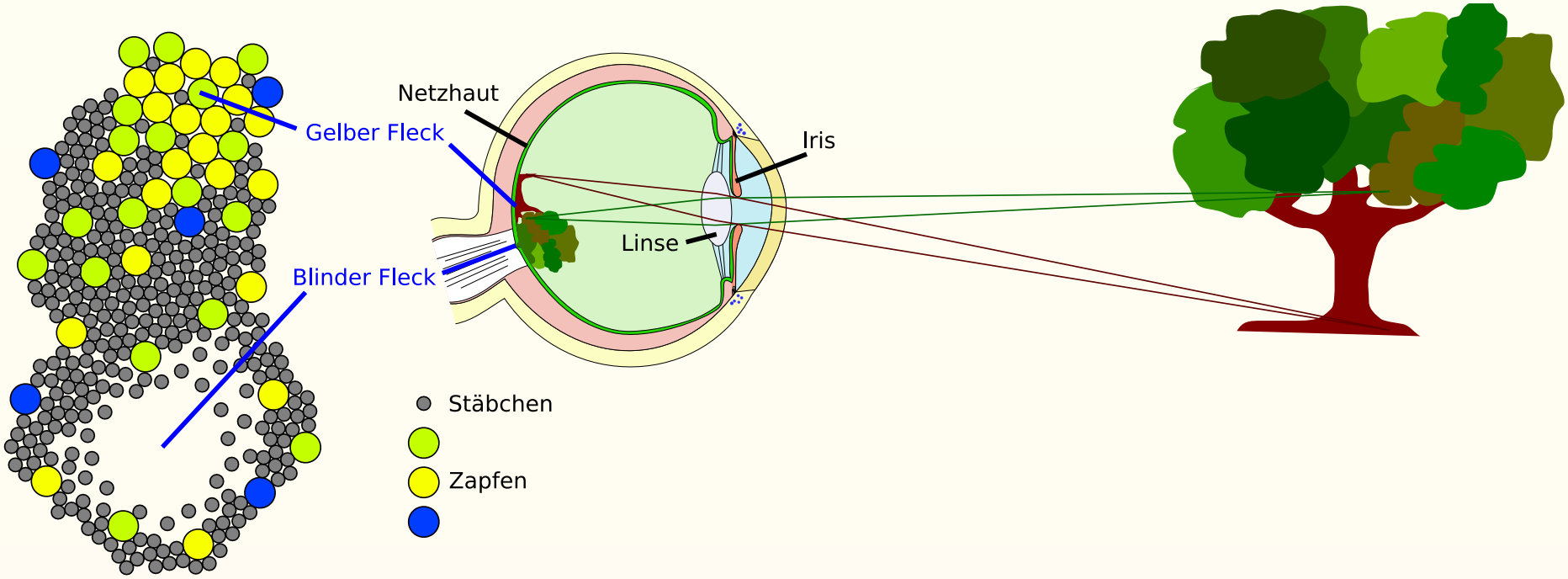
Wie entsteht ein Bild?



- Retina

  Blinder Fleck Austritt des Sehnervs
  Gelber Fleck Zentrales Sehfeld

Wie entsteht ein Bild?



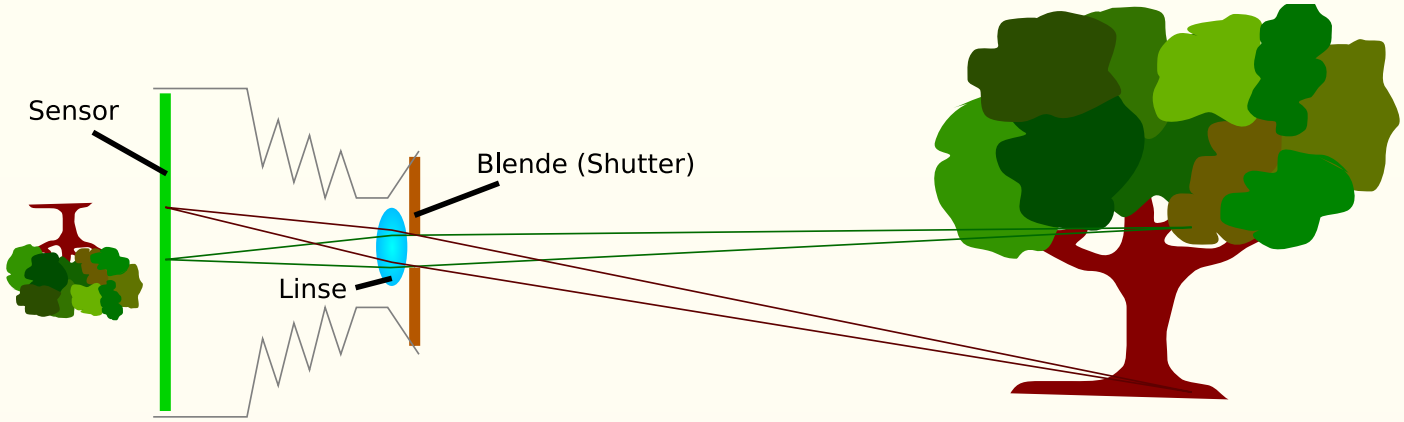
- Retina

  Blinder Fleck Austritt des Sehnervs $\Rightarrow$ Keine Rezeptoren!
  Gelber Fleck Zentrales Sehfeld $\Rightarrow$ Nur Zapfen!

- Rezeptoren

  Stäbchen Hohe Lichtempfindlichkeit, nur Helligkeit
  Zäpfchen Niedrige Lichtempfindlichkeit, Spektral Selektiv

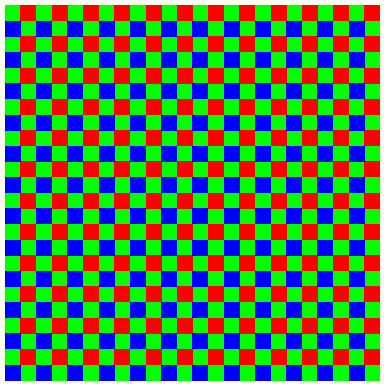
Wie entsteht ein Bild?

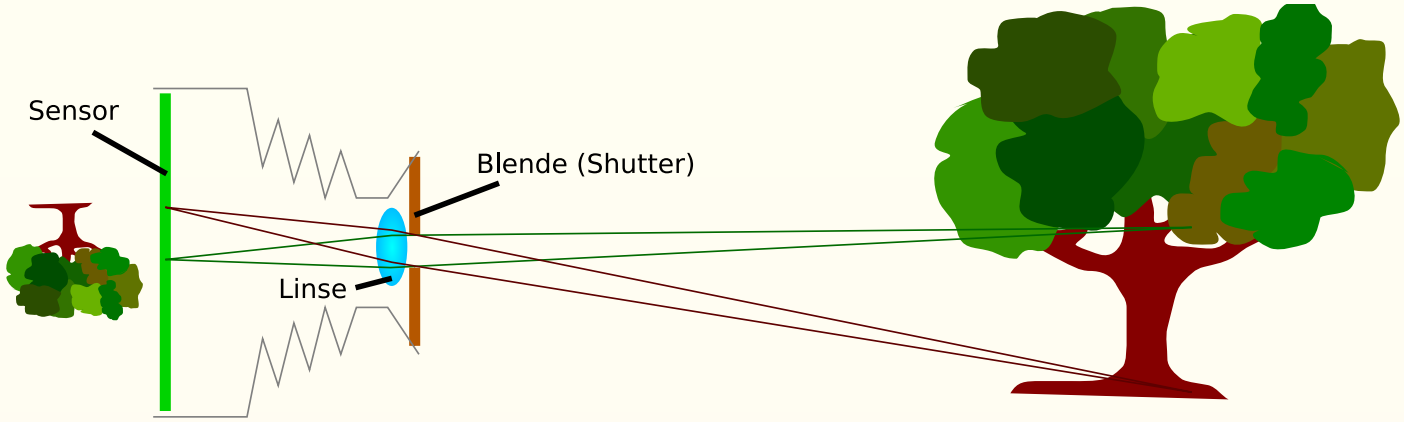


- Das Auge und eine Kamera Teilen einige Eigenschaften

Blende	Die Blende entspricht der Iris des Auges
Linse	Sie fokussiert in beiden Fällen das einfallende Licht
Sensor	Der Sensor einer Kamera erfüllt die Aufgaben der Retina des Auges

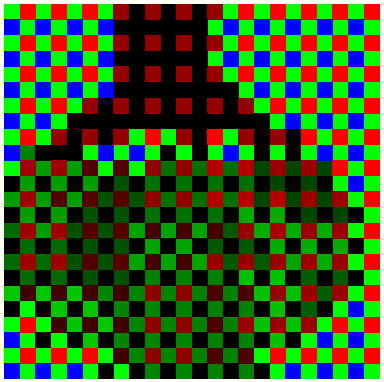
Wie entsteht ein Bild?

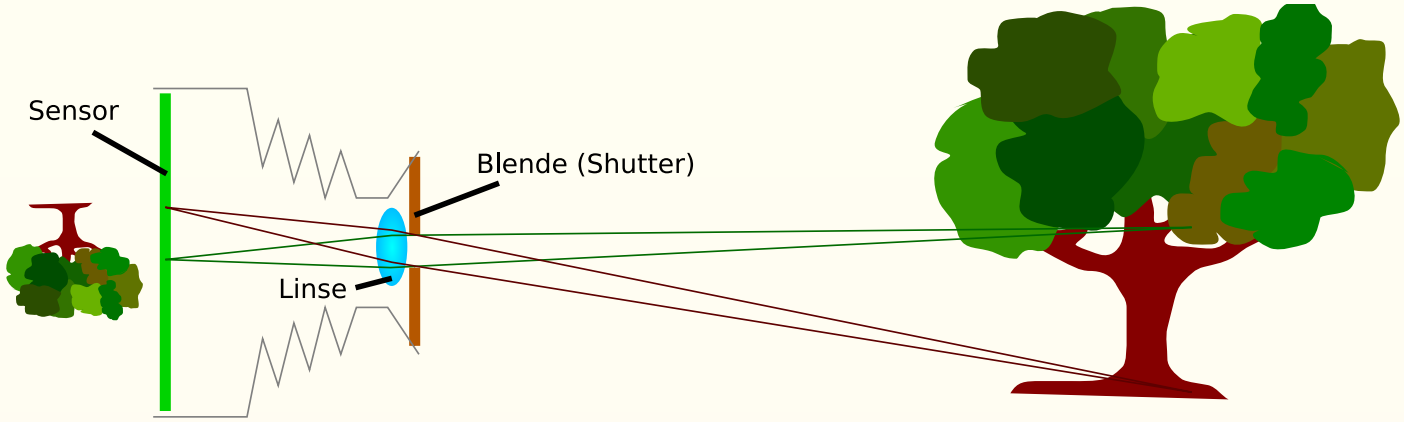




- Der Sensor besteht aus (quadratischen) lichtempfindlichen „Rezeptoren” für Rot, Grün und Blau
- Die Signale einer bestimmten Sensorfläche werden zu einem Farbwert kombiniert

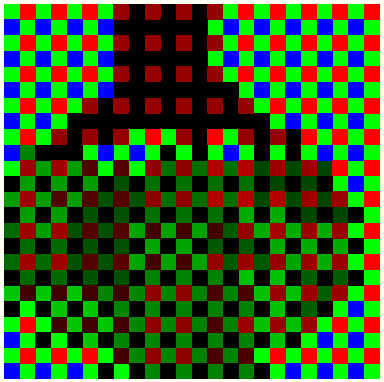
Wie entsteht ein Bild?

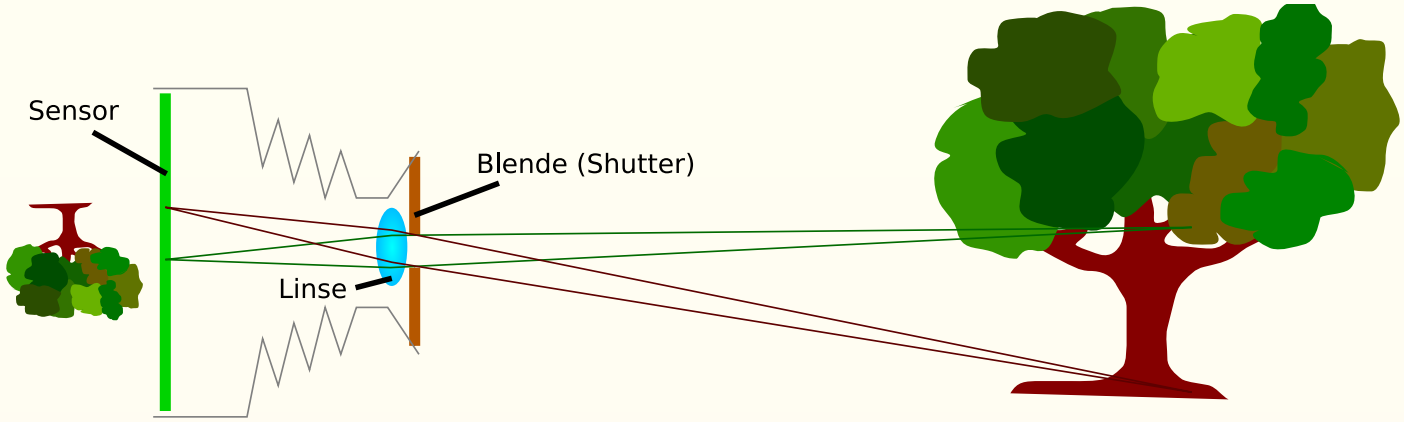




- Der Sensor besteht aus (quadratischen) lichtempfindlichen „Rezeptoren" für Rot, Grün und Blau
- Die Signale einer bestimmten Sensorfläche werden zu einem Farbwert kombiniert
- Die Lichtintensität wird in elektrische Ströme umgewandelt

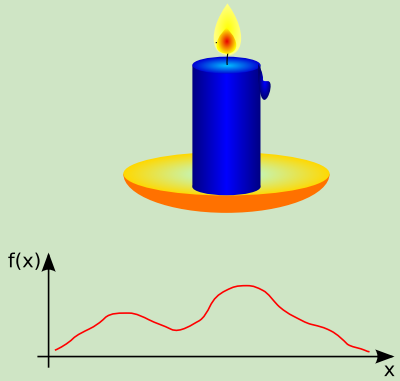
Wie entsteht ein Bild?





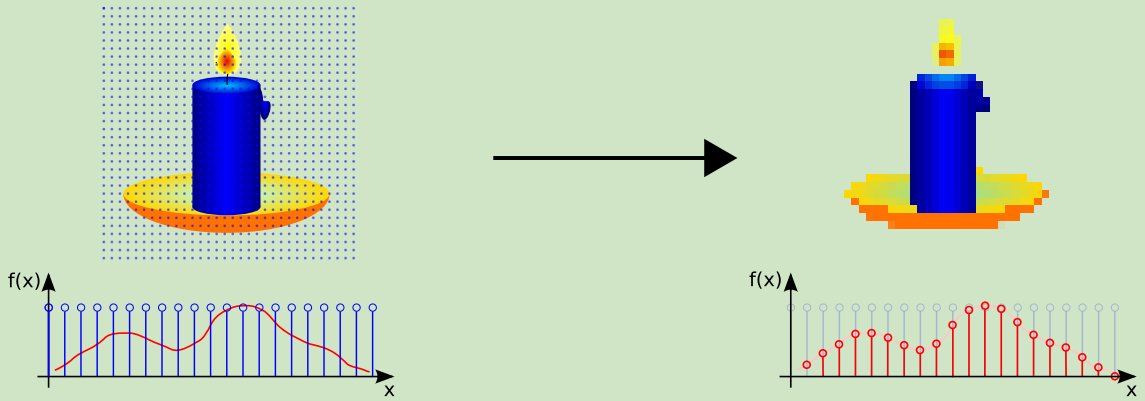
- Der Sensor besteht aus (quadratischen) lichtempfindlichen „Rezeptoren” für Rot, Grün und Blau
- Die Signale einer bestimmten Sensorfläche werden zu einem Farbwert kombiniert
- Die Lichtintensität wird in elektrische Ströme umgewandelt
- Diese Ströme werden quantisiert (oft 0 – 255)

Abtastung



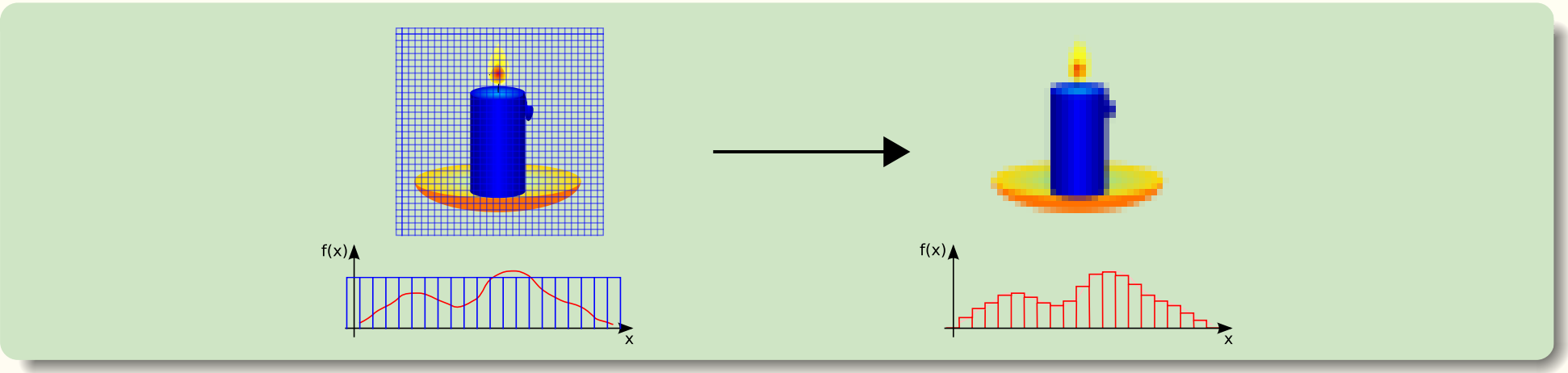
- Kontinuierliche 2-D Intensitätsverteilung auf dem Sensor

Abtastung



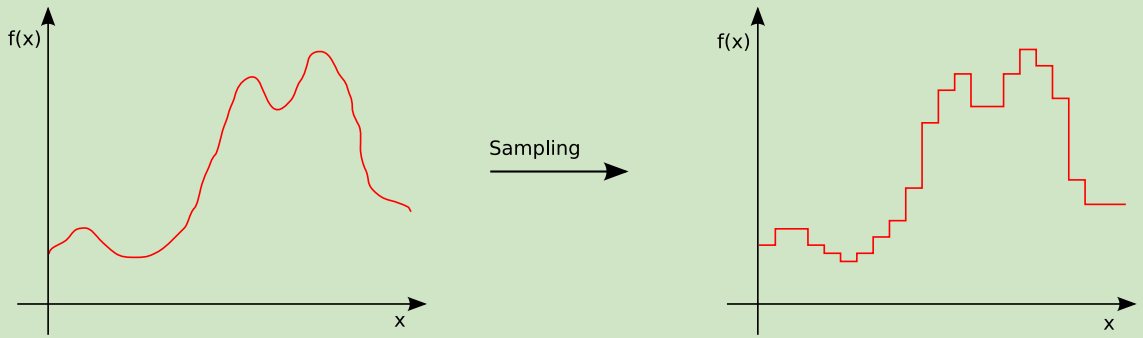
- Kontinuierliche 2-D Intensitätsverteilung auf dem Sensor
- Modell: Abtastung über ein Raster aus δ-Funktionen

Abtastung



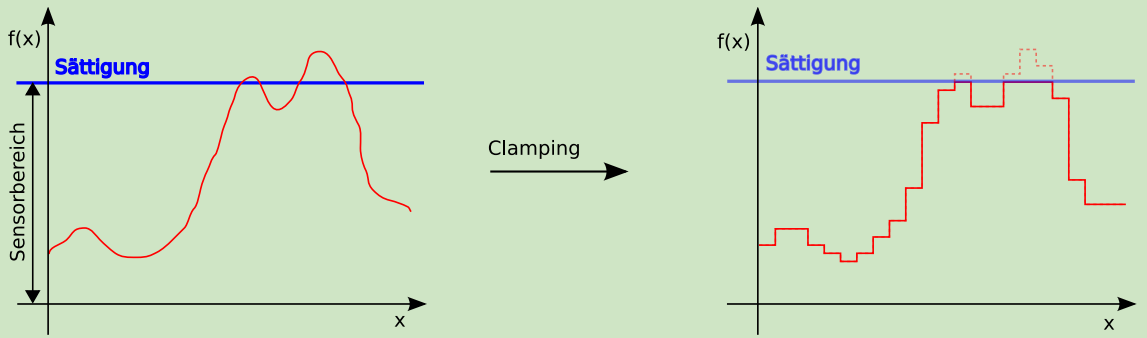
- Kontinuierliche 2-D Intensitätsverteilung auf dem Sensor
- Modell: Abtastung über ein Raster aus δ-Funktionen
- Sensor: Integration der Lichtmenge in jedem Sensorelement (Pixel)
  - über die Fläche des Sensorelements (wenige 100 μm^2)
  - über die Belichtungszeit (wenige Millisekunden)

Quantisierung



Sampling: Vorherige Folie

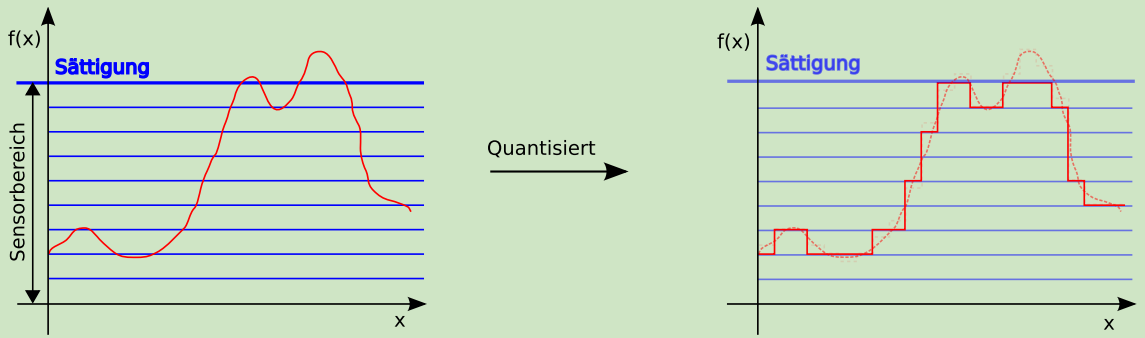
Quantisierung



Sampling: Vorherige Folie

Clamping: Sensorsättigung führt zu einer oberen Grenze für die darstellbare Helligkeit

Quantisierung

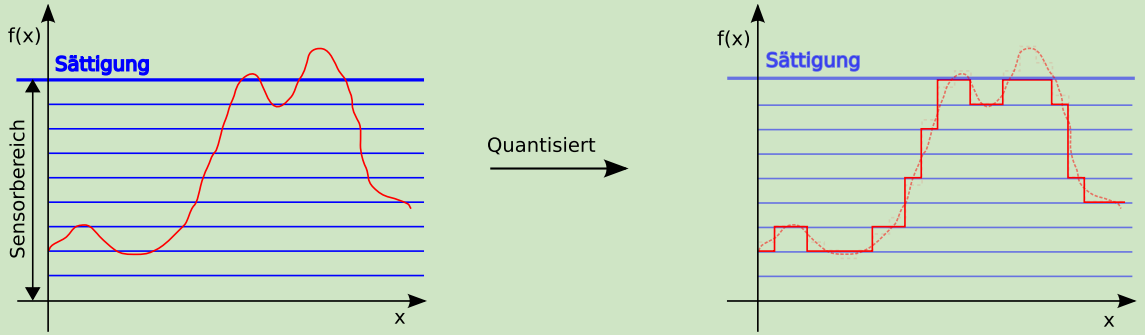


Sampling: Vorherige Folie

Clamping: Sensorsättigung führt zu einer oberen Grenze für die darstellbare Helligkeit

Quantisierung: Abbildung des durch den Sensor wahrnehmbaren Helligkeitsbereichs in einen diskreten Wertebereich

Quantisierung



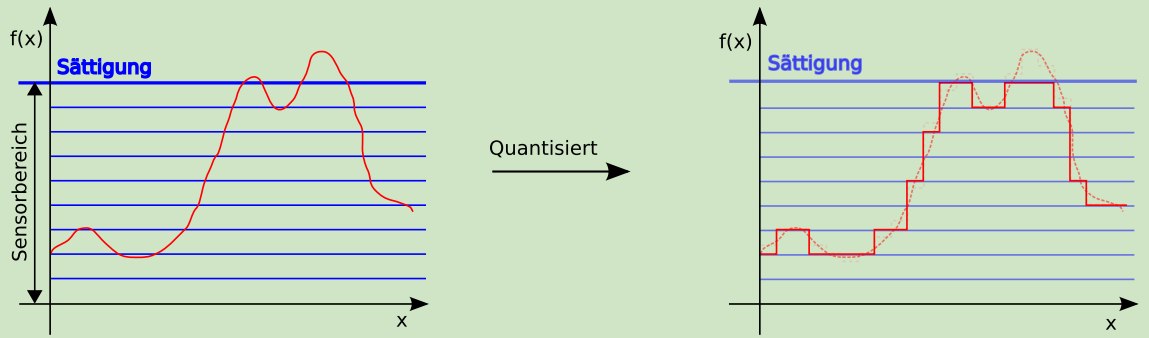
Sampling: Vorherige Folie

Clamping: Sensorsättigung führt zu einer oberen Grenze für die darstellbare Helligkeit

Quantisierung: Abbildung des durch den Sensor wahrnehmbaren Helligkeitsbereichs in einen diskreten Wertebereich

Windowing: Nur ein Ausschnitt der Welt wird betrachtet

Quantisierung



Sampling: Vorherige Folie

Clamping: Sensorsättigung führt zu einer oberen Grenze für die darstellbare Helligkeit

Quantisierung: Abbildung des durch den Sensor wahrnehmbaren Helligkeitsbereichs in einen diskreten Wertebereich

Windowing: Nur ein Ausschnitt der Welt wird betrachtet

Störungen: Das Signal ist durch unerwünschte Nebeneffekte gestört

- **Systematisch:** Unschärfe, Linsenverzerrungen, Chromatische Fehler
- **Statistisch:** Rauschen

Das digitale Bild

Definition (Digitales Bild)

Helligkeits/Farb-Werte auf einem regulären Gitter I_{ij}

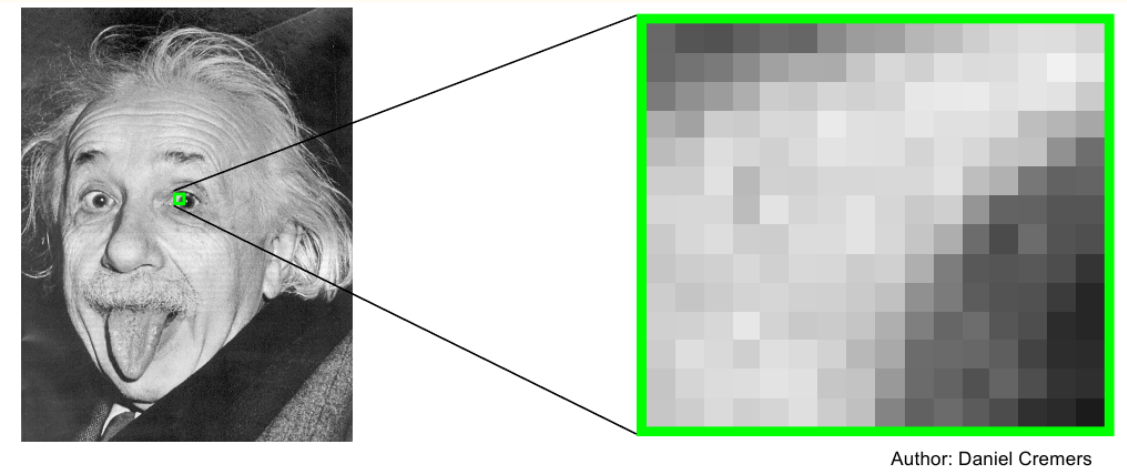
$$I : (\Omega \subset \mathbb{R}) \to \mathbb{R}$$
$$(x, y) \mapsto I(x, y)$$

Das digitale Bild

Definition (Digitales Bild)

Helligkeits/Farb-Werte auf einem regulären Gitter I_{ij}

$$I : (\Omega \subset \mathbb{R}) \to \mathbb{R}$$
$$(x, y) \mapsto I(x, y)$$

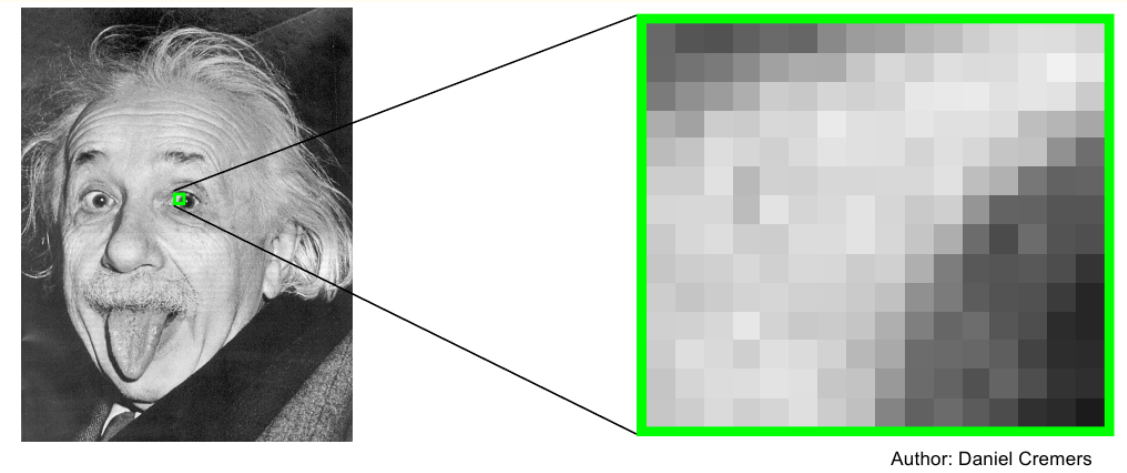


Das digitale Bild

Definition (Digitales Bild)

Helligkeits/Farb-Werte auf einem regulären Gitter I_{ij}

$$I : (\Omega \subset \mathbb{R}) \to \mathbb{R}$$
$$(x, y) \mapsto I(x, y)$$

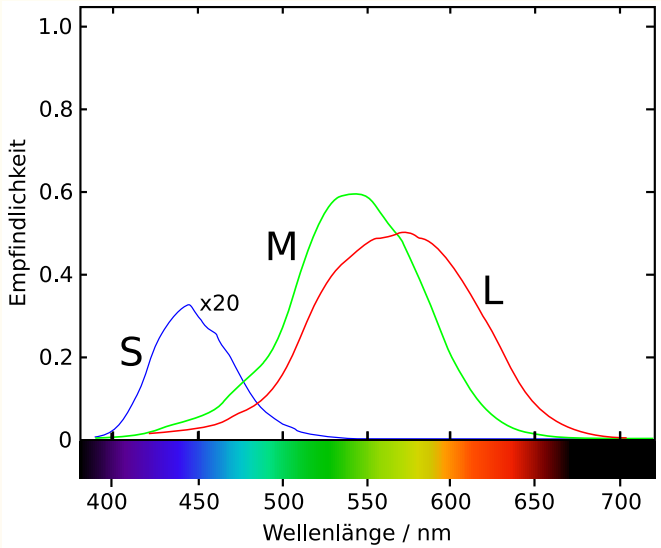


Was ist Farbe?

Photorezeptoren im menschlichen Auge:

- Stäbchen (Rods)
  Hohe Lichtempfindlichkeit in breitem Spektrum
- Zapfen (Cones)
  Geringere spektral selektive Lichtempfindlichkeit

  S-Zapfen Supra-Frequency Cones
  M-Zapfen Middle-frequency Cones
  L-Zapfen Low-Frequency Cones

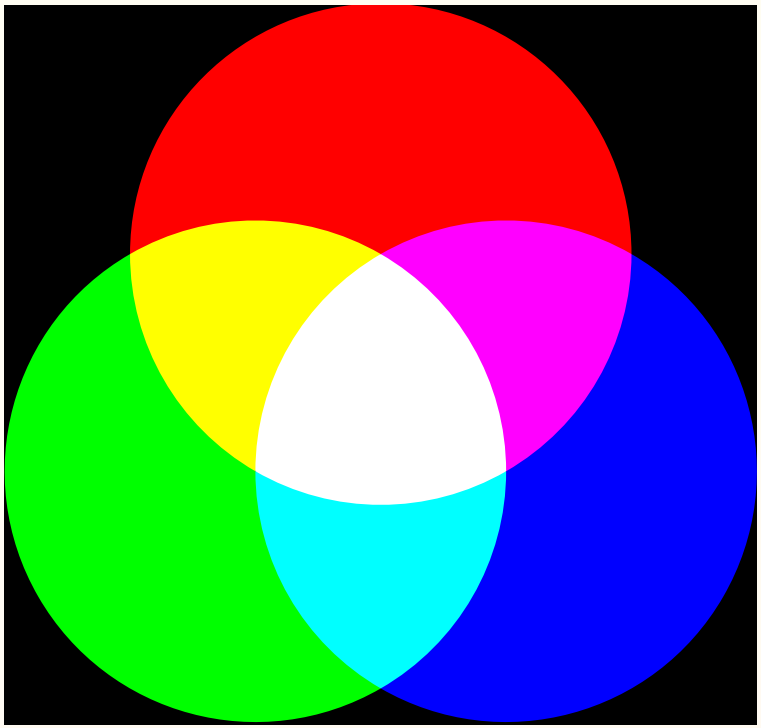


- Farbempfinden über Linearkombination des Zapfen-Signale
- Geringere Empfindlichkeit für Blautöne

Farbräume (RGB)

Farbbeschreibung (3 Kanäle):

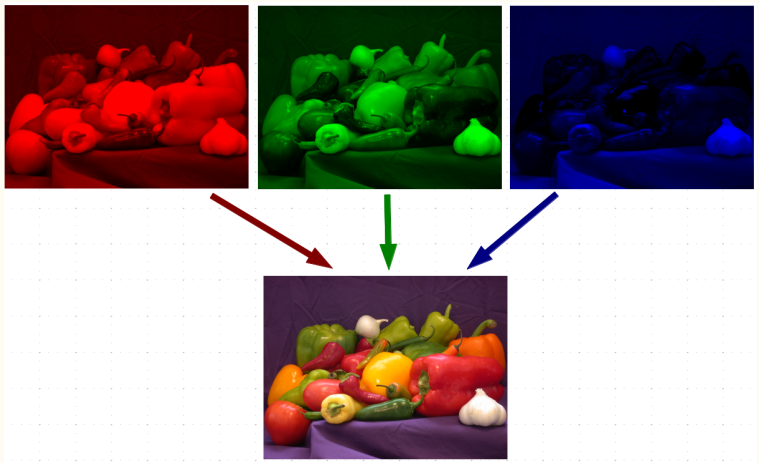
- **R**ot
- **G**rün
- **B**lau



Farbräume (RGB)

Farbbeschreibung (3 Kanäle):

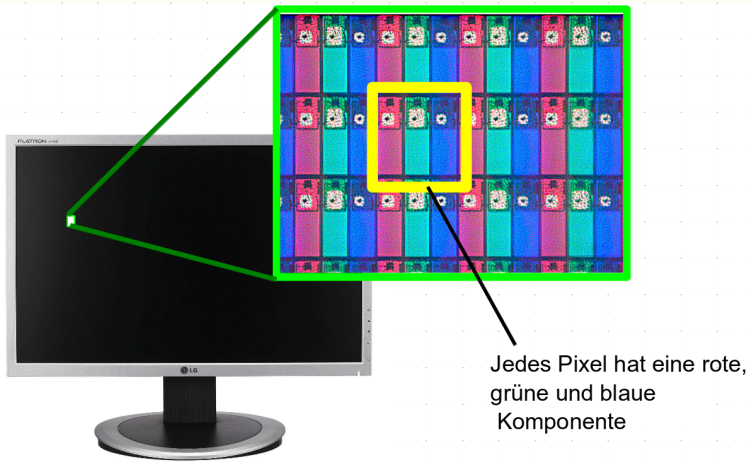
- Rot
- Grün
- Blau



Farbräume (RGB)

Farbbeschreibung (3 Kanäle):

- **R**ot
- **G**rün
- **B**lau



Eigenschaften:

- Optimiert für Monitordarstellung
- Standardspeicherformat
- Dreifarbentheorie
  (Additive Farbmischung)
- ✘ Farbbeschreibung nicht intuitiv!

ImageJ

ImageJ Installation

Downloadseite:
http://rsb.info.nih.gov/ij/download.html

Verfügbare Plattformen:

- Linux (32Bit und 64Bit)
- Windows (32Bit und 64Bit)
- Mac OS X (64Bit)

Installation:

Linux Tarball in ein Verzeichnis entpacken. Starten mit `./run`

andere Installer ausführen, bzw. den Hinweisen folgen

Architektur eines ImageJ Plugins

ImageJ Usage:	http://rsbweb.nih.gov/ij/docs/index.html
ImageJ Programming:	http://www.imagingbook.com/index.php?id=102
Java API:	http://java.sun.com/j2se/1.5.0/docs/api/

Beispiel (My_Inverter.java)

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

public class My_Inverter implements PlugInFilter {

  public int setup(String arg, ImagePlus imp) {
    if (IJ.versionLessThan("1.37j"))
      return DONE;
    else
      return DOES_8G+SUPPORTS_MASKING;
  }

  public void run(ImageProcessor ip) {
    Rectangle r = ip.getRoi();
    for (int y = r.y; y < (r.y + r.height); y++)
      for (int x = r.x; x < (r.x + r.width); x++)
        ip.set(x, y, 255 - ip.get(x, y));
  }

}
```

Plug-In Typen

PlugIn Allgemeine Basisklasse für alle Plugins

- `void run(String arg)`

Plug-In Typen

PlugIn Allgemeine Basisklasse für alle Plugins

- `void run(String arg)`

PlugInFilter Plugin zur Manipulation von Bildern

- `int setup(String arg, ImagePlus imp)`
  Initialisierung des PlugIns mit möglichen Rückgabewerten:
  `DONE`, `DOES_16`, `DOES_32`, `DOES_8G`, `DOES_8C`, `DOES_RGB`, `DOES_STACKS`, `NO_CHANGE`, `NO_IMAGE_REQUIRED`, `NO_UNDO`, `ROI_REQUIRED`, `STACK_REQUIRED`
- `void run(ImageProcessor ip)`

Plug-In Typen

PlugIn Allgemeine Basisklasse für alle Plugins

- `void run(String arg)`

PlugInFilter Plugin zur Manipulation von Bildern

- `int setup(String arg, ImagePlus imp)`
  Initialisierung des PlugIns mit möglichen Rückgabewerten:
  `DONE`, `DOES_16`, `DOES_32`, `DOES_8G`, `DOES_8C`, `DOES_RGB`, `DOES_STACKS`, `NO_CHANGE`, `NO_IMAGE_REQUIRED`, `NO_UNDO`, `ROI_REQUIRED`, `STACK_REQUIRED`
- `void run(ImageProcessor ip)`

PlugInFrame Plugin zur Darstellung von AWT Fenstern

Wichtige Klassen

ImagePlus Abgeleitet von `java.awt.Image`
Basisklasse für die Bildvisualisierung

Wichtige Klassen

ImagePlus Abgeleitet von `java.awt.Image`
Basisklasse für die Bildvisualisierung

ImageProcessor Basisklasse zur Bildmanipulation, z.B.

- `int getPixel(int x, int y)`
- `void putPixel(int x, int y, int value)`

Wichtige Klassen

ImagePlus Abgeleitet von `java.awt.Image`
Basisklasse für die Bildvisualisierung

ImageProcessor Basisklasse zur Bildmanipulation, z.B.

- `int getPixel(int x, int y)`
- `void putPixel(int x, int y, int value)`
- `void smooth()`
- `void findEdges()`

Wichtige Klassen

ImagePlus Abgeleitet von `java.awt.Image`
Basisklasse für die Bildvisualisierung

ImageProcessor Basisklasse zur Bildmanipulation, z.B.

- `int getPixel(int x, int y)`
- `void putPixel(int x, int y, int value)`
- `void smooth()`
- `void findEdges()`
- `void scale(double xScale, double yScale)`

Wichtige Klassen

ImagePlus Abgeleitet von `java.awt.Image`
Basisklasse für die Bildvisualisierung

ImageProcessor Basisklasse zur Bildmanipulation, z.B.

- `int getPixel(int x, int y)`
- `void putPixel(int x, int y, int value)`
- `void smooth()`
- `void findEdges()`
- `void scale(double xScale, double yScale)`
- `void drawLine(int x1, int y1, int x2, int y2)`
- `void fill()`

Wichtige Klassen

ImagePlus Abgeleitet von `java.awt.Image`
Basisklasse für die Bildvisualisierung

ImageProcessor Basisklasse zur Bildmanipulation, z.B.

- `int getPixel(int x, int y)`
- `void putPixel(int x, int y, int value)`
- `void smooth()`
- `void findEdges()`
- `void scale(double xScale, double yScale)`
- `void drawLine(int x1, int y1, int x2, int y2)`
- `void fill()`

NewImage Factory zur Erstellung von Bildern

Wichtige Klassen

ImagePlus Abgeleitet von `java.awt.Image`
Basisklasse für die Bildvisualisierung

ImageProcessor Basisklasse zur Bildmanipulation, z.B.

- `int getPixel(int x, int y)`
- `void putPixel(int x, int y, int value)`
- `void smooth()`
- `void findEdges()`
- `void scale(double xScale, double yScale)`
- `void drawLine(int x1, int y1, int x2, int y2)`
- `void fill()`

NewImage Factory zur Erstellung von Bildern

GenericDialog Klasse zur einfachen Erstellung von Dialogen

Plugin-Erstellung in der Praxis

Bis hier Fragen?

Referenzen

ImageJ-Homepage: http://rsb.info.nih.gov/ij/

Tutorial: http://www.imagingbook.com/

Empfohlene Literatur:

[1] Burger und Burge. **Digitale Bildverarbeitung - Eine Einführung mit Java und ImageJ** 2. überarb. Aufl., Deutsch, Springer, Berlin, 2006.

[2] Gonzalez and Woods. **Digital Image Processing** 2nd Edition, Prentice Hall, 2002