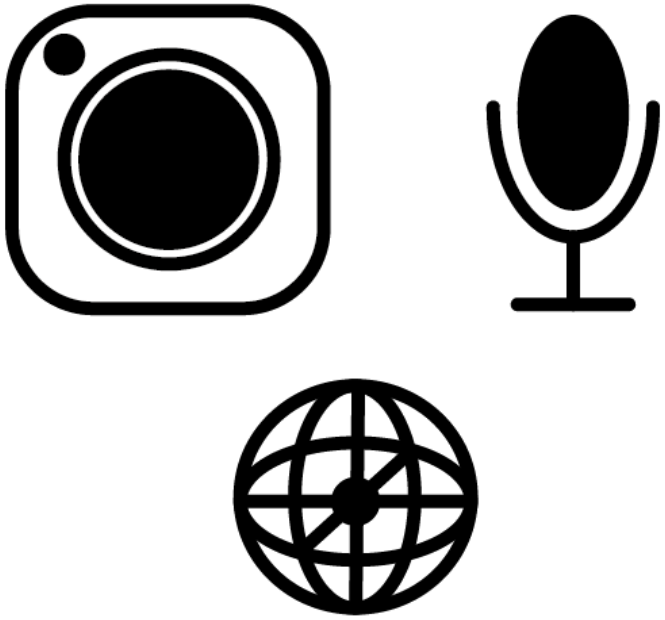


Sensordatenverarbeitung

SEGMENTIERUNG (4B)

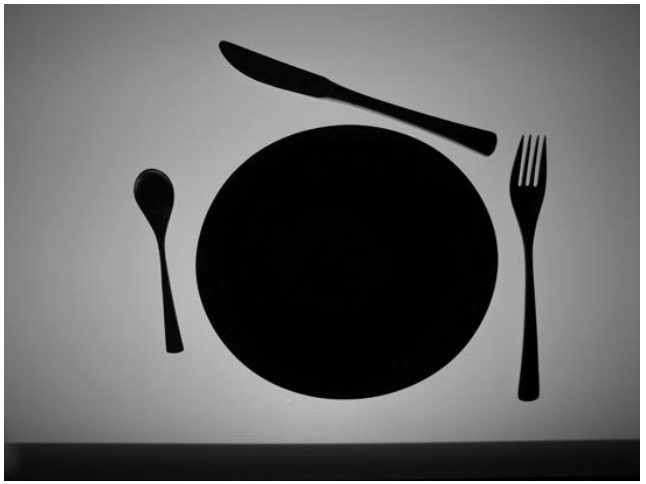
4.-8.11.24

Rahmen für segm.-getr. BV

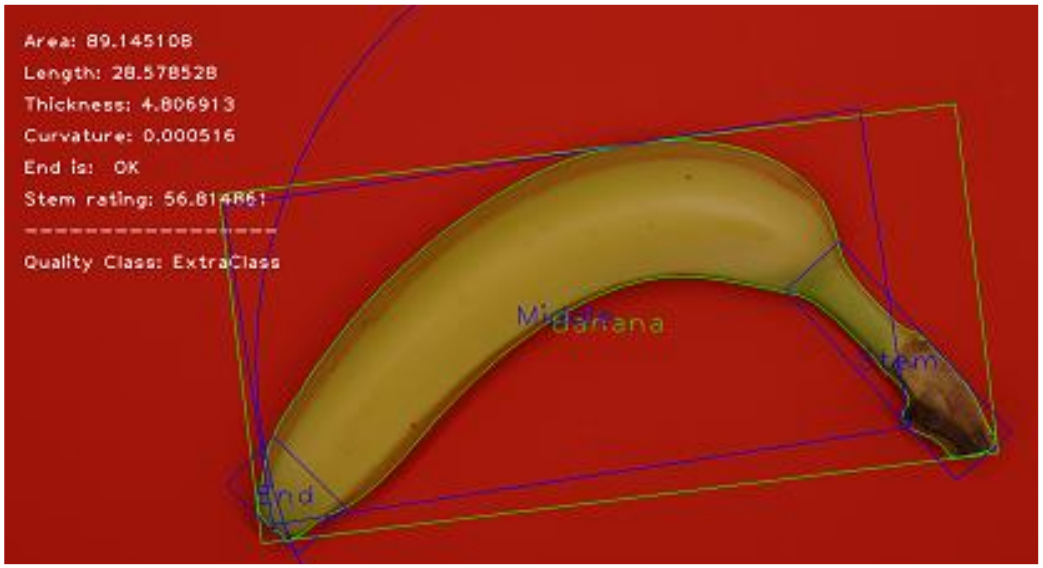
- Definiertes Licht, Hintergrund- und Objektfarben
  - per Pixel Klassifikation in Hintergrund / Objekt aufgrund von Farbe / Helligkeit
  - oder per Pixel Klassifikation in Kante / nicht-Kante aufgrund Nachbarpixeln
- Kamera von oben
  - 2D, direkte Korrespondenz Pixel←→mm
- Geometrisch verstandene Formen
  - Objektklassifikation über Maße oder andere geometrische Eigenschaften
  - Fehlerkennungen ähnlich ausfiltern
- Einfach genug um anwendungsspezifische Variationen zu entwerfen
  - in jedem Schritt viele Abwandelungsmöglichkeiten und Optionen
  - große Bedeutung für industrielle Bildverarbeitung
  - methodisch nicht state-of-the-art

Beispiele

- Erkennen von Teller, Gabel, Messer, Löffel
  - Alles liegt irgendwo im Bild aber berührt sich nicht
  - Position, Orientierung und Identität erkennen

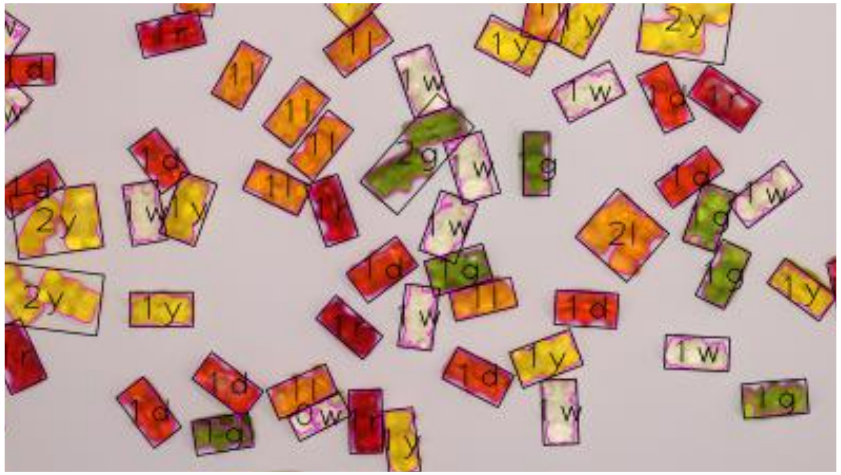


- Qualitätskontrolle von Bananen
  - Banane liegt irgendwo im Bild
  - prüft Maße, Krümmung, Farbe im Zentralteil, vorhandenen Stiel
  - Daniel Krause, Bernd Poppinga, Lukas Post, Alexander Stöwing



Beispiele

- Gummibären erkennen
  - Gummibären verschiedener Geschmacksrichtung frei verteilt
  - Position, Orientierung und Geschmacksrichtung erkennen
  - Stefan Hillmich, Erik Keshishian, Ralf Morawe



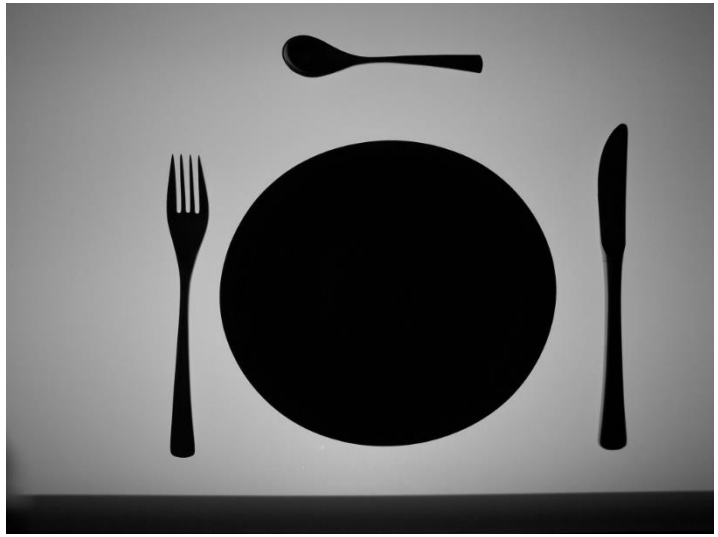
- Spielkartenerkennen
  - frei und überlappend auf Untergrund liegend
  - Symbole voll sichtbar
  - Karten erkennen
  - Nils Drebing, Eike Externest, Sebastian Kühl

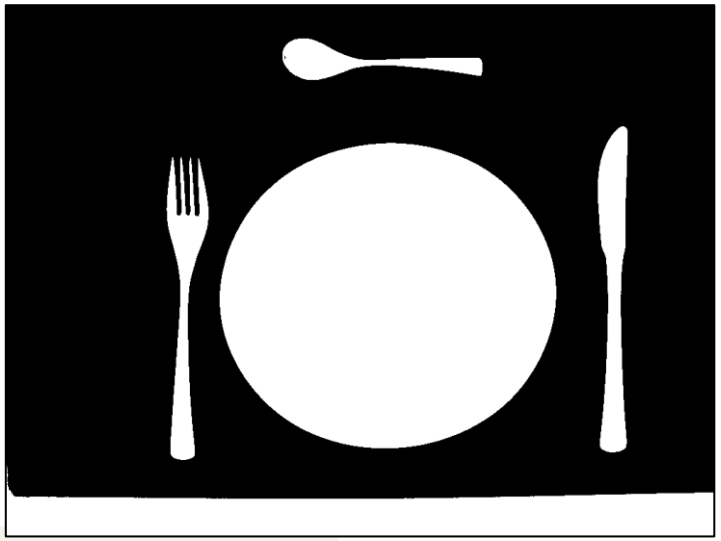
Segmentierungsgetriebene BV

- Segmentierung / Binarisierung
  - Klassifikation einzelner Pixel aufgrund von Helligkeit / Farbe
  - Klassifikation einzelner Pixel als Kante
- Gruppierung
  - zusammengehörende Regionen bilden
  - Herausforderung: berührende Objekte trotzdem trennen
  - Herausforderung: fälschlich getrennte Objekte vereinen
- Kennzahlen bilden
  - geometrische (Länge, Breite, Fläche, Krümmung, ...)
  - farbliche (mittlere Farbe, # Pixel einer Farbe)
  - Dicke quer an verschiedenen Stellen längs
- Filtern & Klassifizieren
  - An bestimmten Stellen auf den Objekten nach weiteren Merkmalen suchen

Binarisierung / Segmentierung

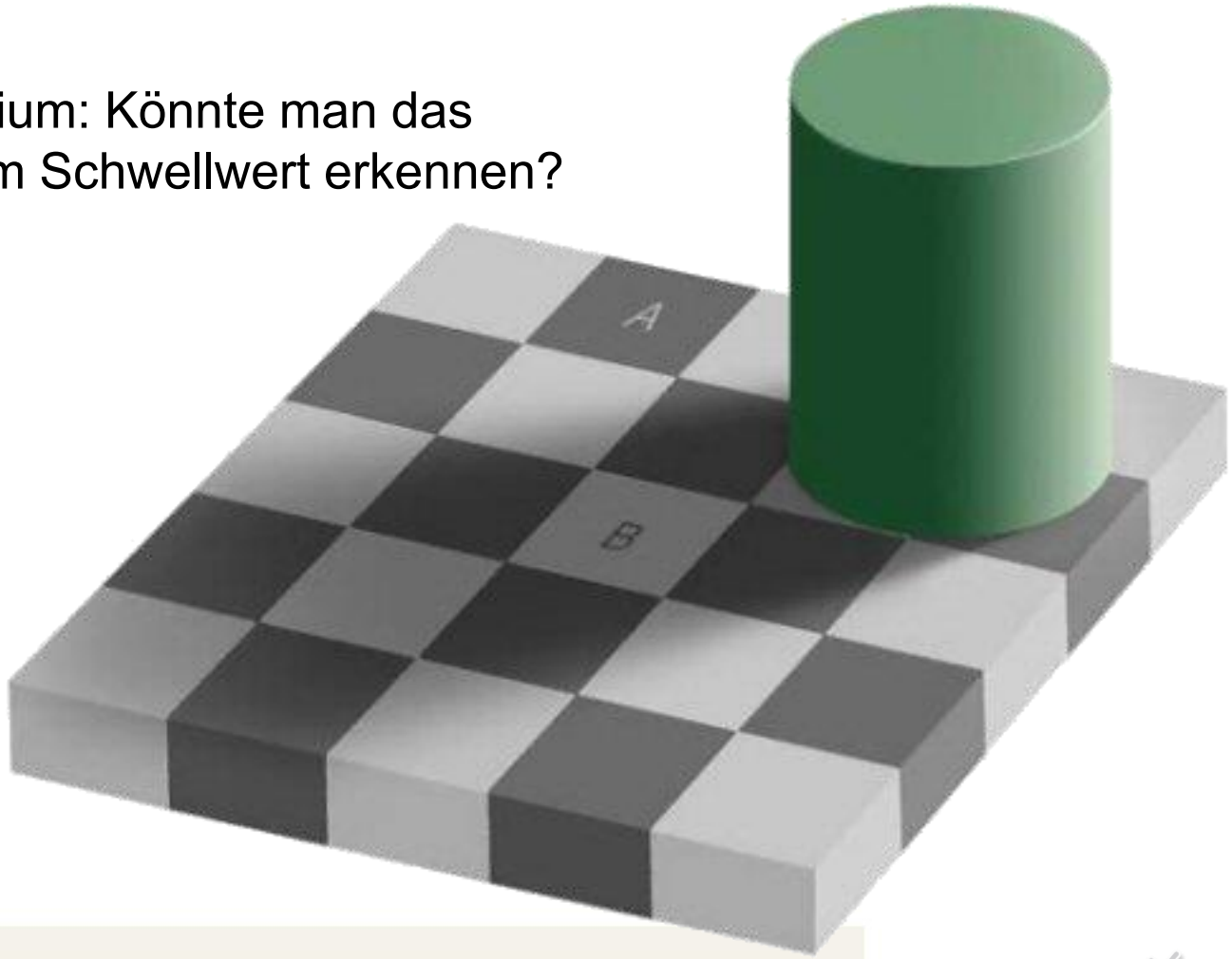
- Klassifikation einzelner Pixel
  - RGB → {Hintergrund, Objekttyp1, Objekttyp2, ...}
  - äquivalent pro Objekt ein Binärbild Hintergrund vs. Objekt
  - Intern: {0, 255} oder {0,1,2,3,...}
  - Darstellung für Menschen zum Analysieren: Schwarz/Weiß oder mit Farben oder Farbe über Originalbild
- Grauwertbild: Schwellwert
  - OpenCV: `threshold`
- Woher kommt der Schwellwert?
  - manuell einstellen (aber dann einen für alle Bilder)
  - mit Gimp o.ä ausprobieren
  - oder aus Grauwertverteilung berechnen (Otsu-Algorithmus, `THRESH_OTSU`)





Helligkeitssegmentierung

- Frage an das Auditorium: Könnte man das Schachbrett mit einem Schwellwert erkennen?



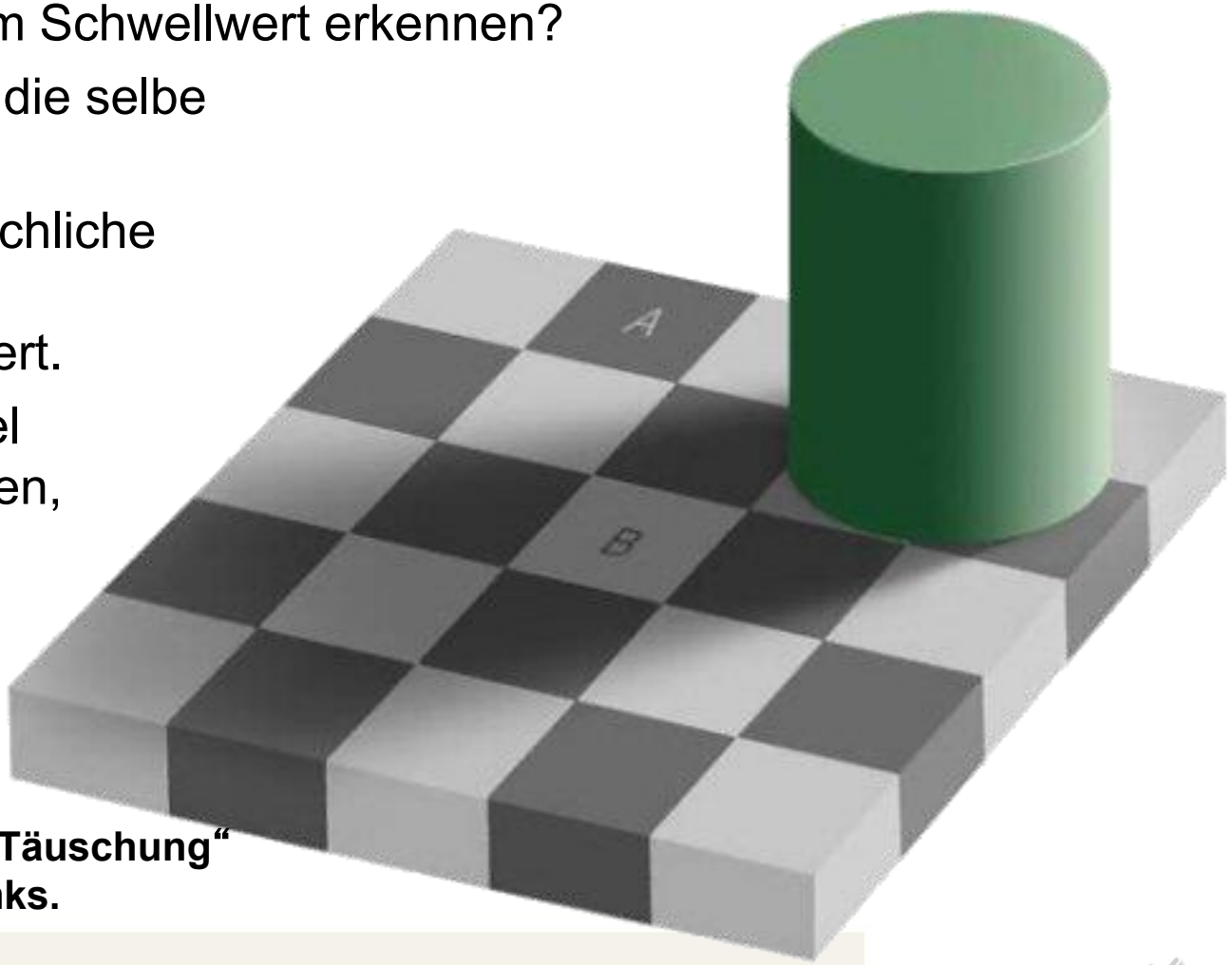
Helligkeitssegmentierung

- Frage an das Auditorium: Könnte man das Schachbrett mit einem Schwellwert erkennen?
- Nein! A und B haben die selbe Helligkeit!

Quelle: wikipedia „Optische Täuschung“
Mit vielen Beispielen und Links.

Helligkeitssegmentierung

- Frage an das Auditorium: Könnte man das Schachbrett mit einem Schwellwert erkennen?
- Nein! A und B haben die selbe Helligkeit!
- Folgerung: Der menschliche Blick trügt oft, weil er unbewusst interpretiert.
- Szenarien können viel einfacher aussehen, als sie sind.



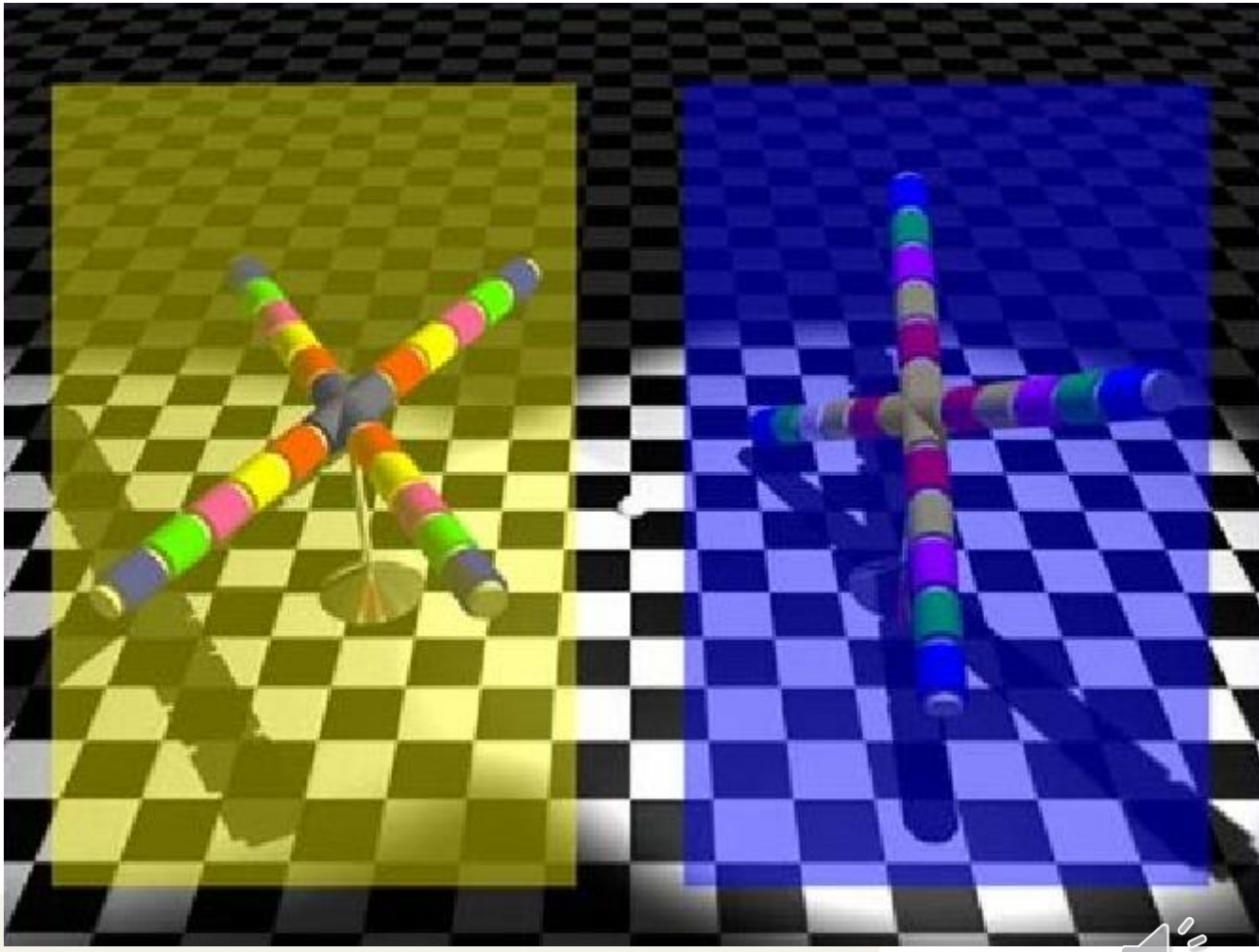
Quelle: wikipedia „Optische Täuschung“
Mit vielen Beispielen und Links.

Farbsegmentierung

- Idee: Einen bestimmten Bereich von Farben als Objekt klassifizieren
- Beispiel rechts: Rote Gummibärchen an ihrer Farbe segmentiert
- Kann mehr Klassen von Objekten unterscheiden als Helligkeitssegmentierung
- Problem: Beleuchtung beeinflusst die Farbe eines Objektes im Bild

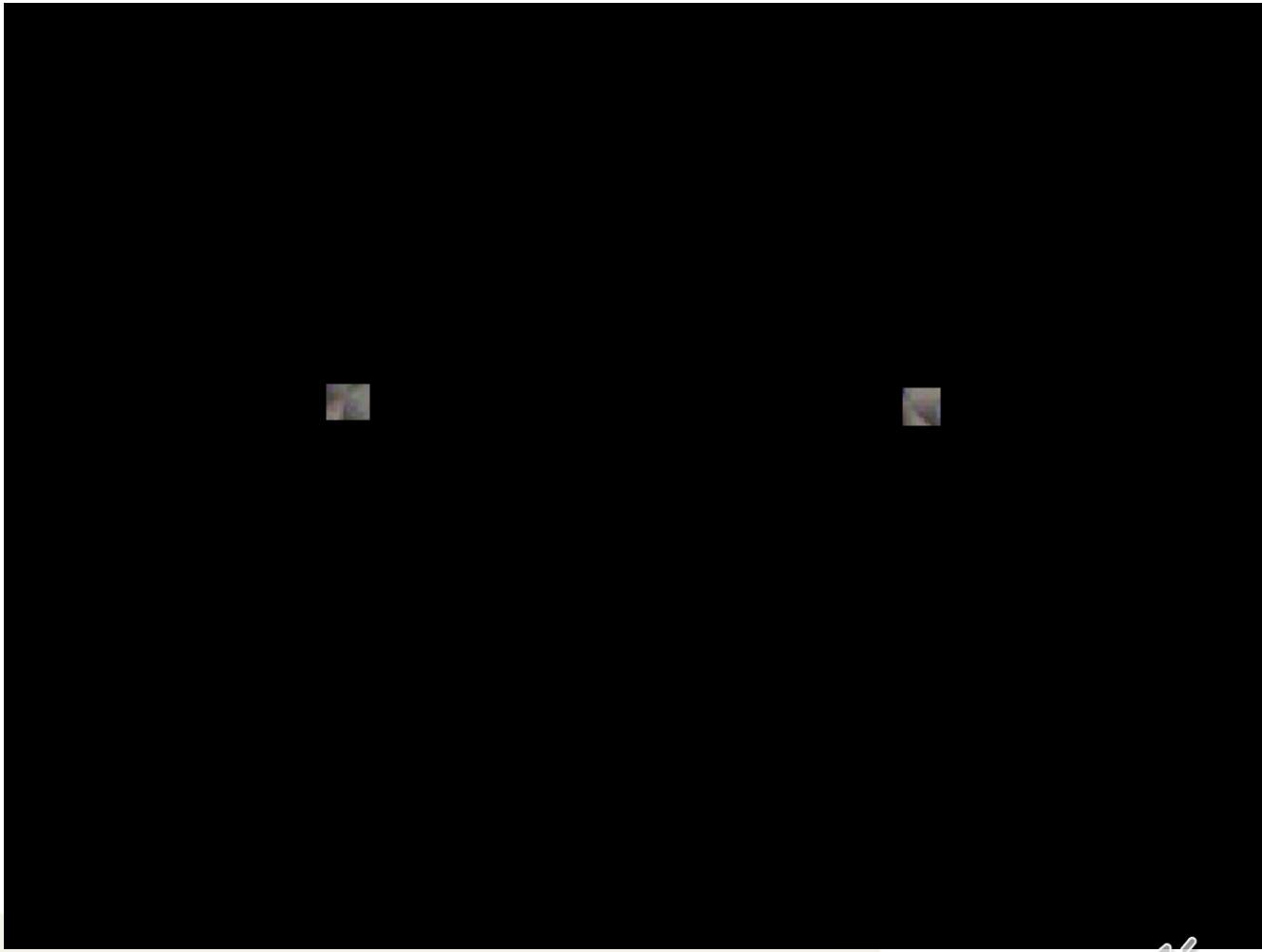
Farbsegmentierung

- Frage an das Auditorium: Welche Farbe haben die Kreuzungen?



Farbsegmentierung

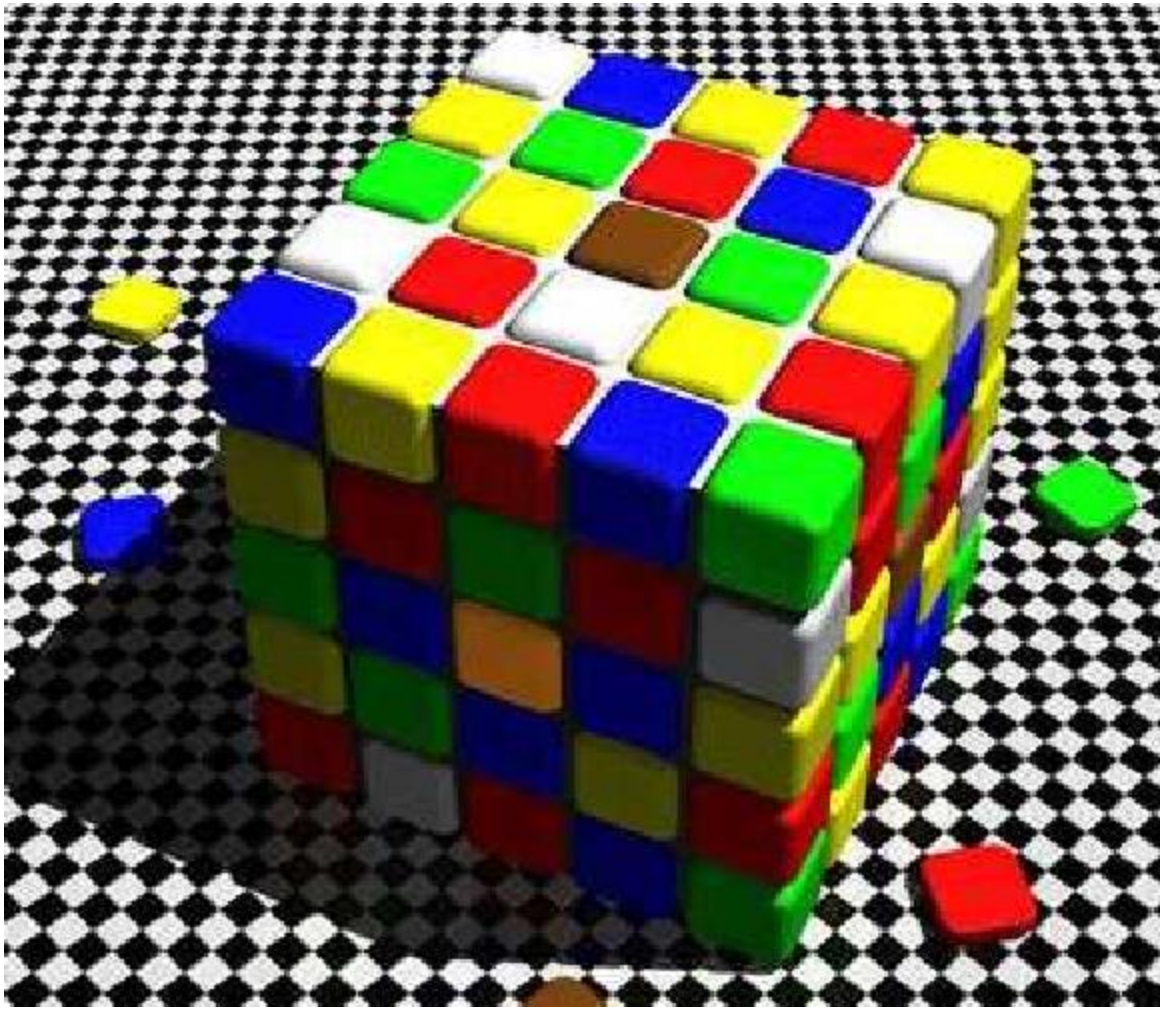
- Frage an das Auditorium: Welche Farbe haben die Kreuzungen?
- Das selbe Grau!



Quelle: http://www.echalk.co.uk/amusements/OpticalIllusions/colourPerception/colourPerception.html

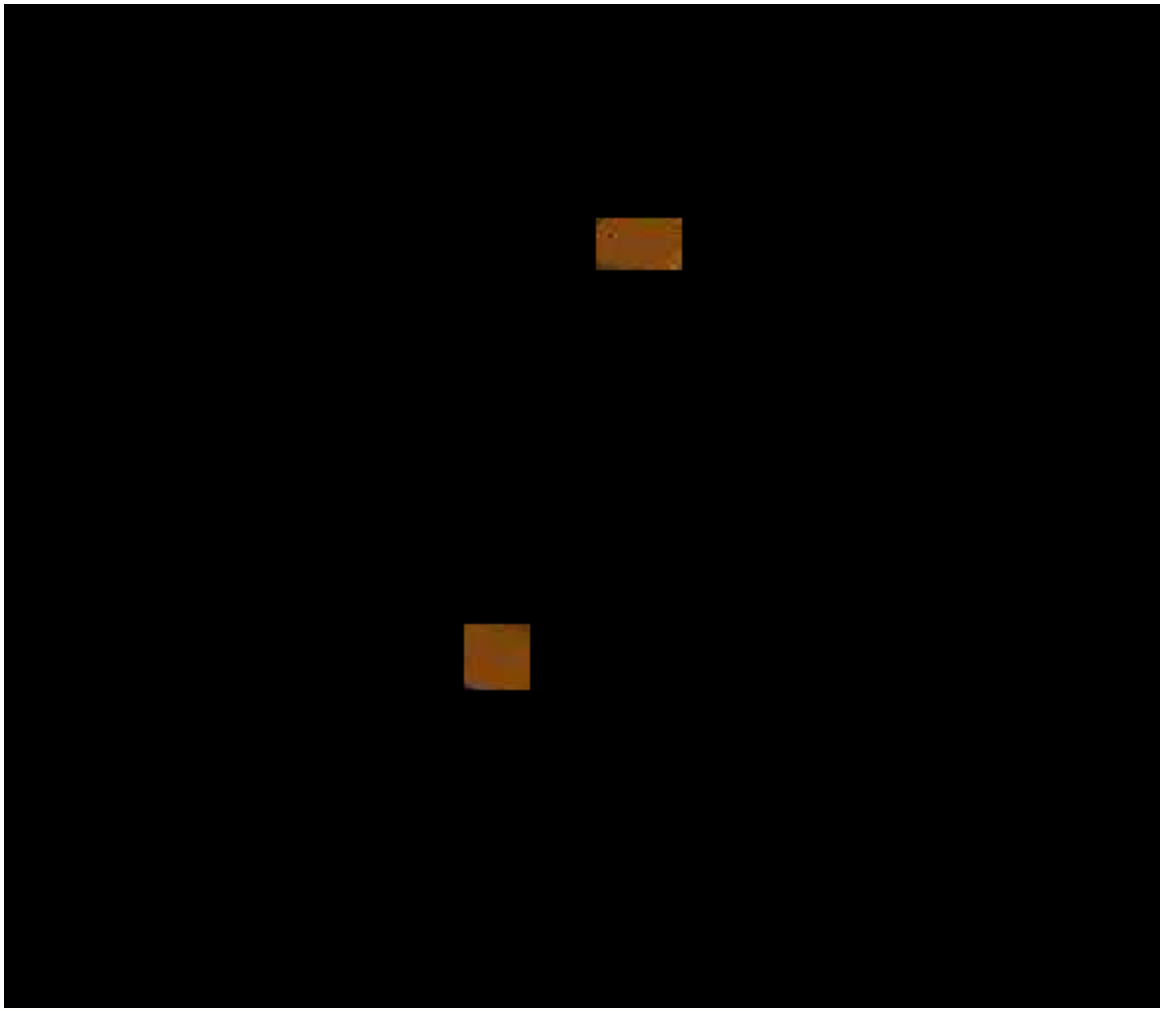
Farbsegmentierung

- Welche Farbe haben die mittleren Quadrate der oberen und vorderen Fläche?



Farbsegmentierung

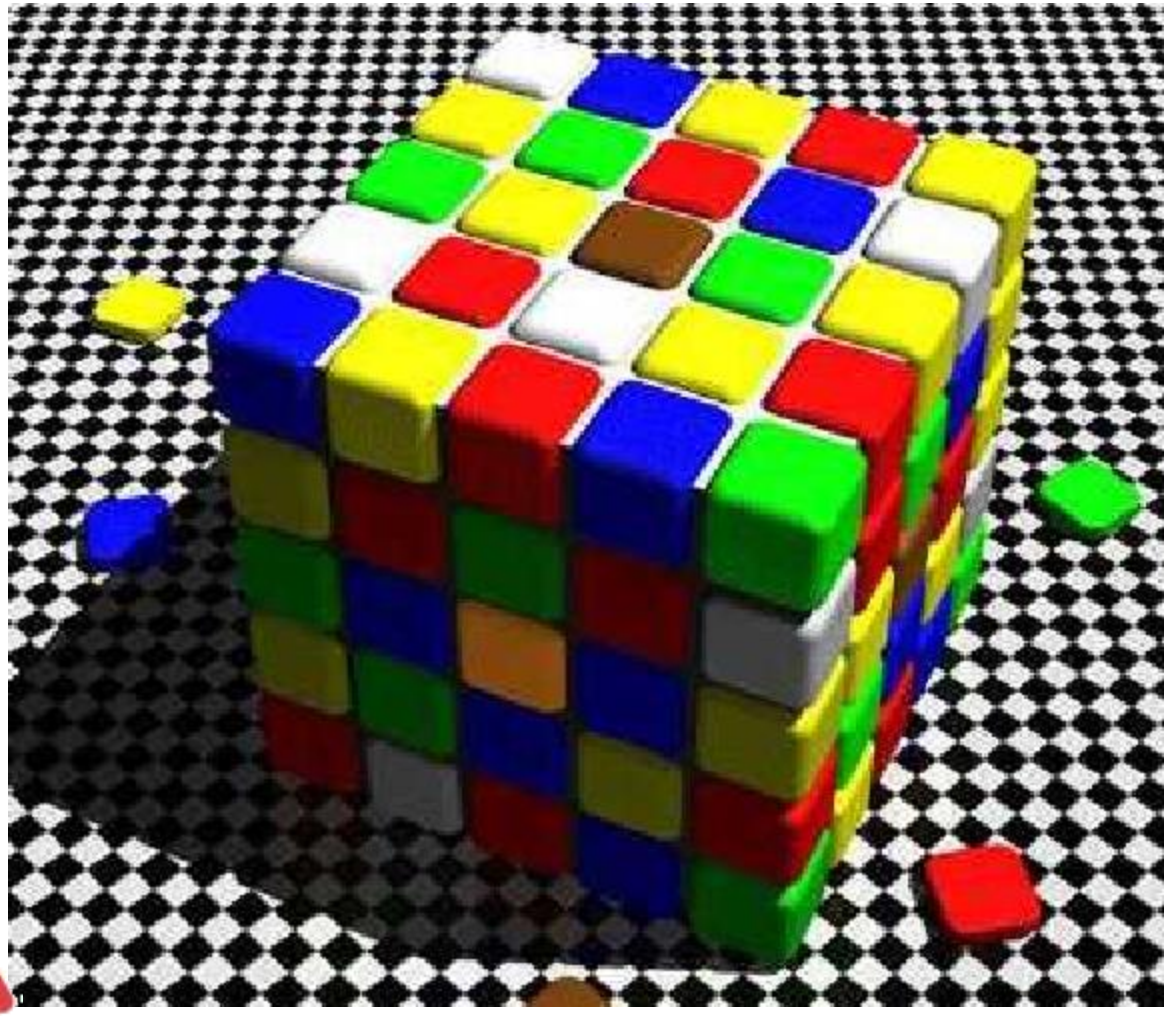
- Welche Farbe haben die mittleren Quadrate der oberen und vorderen Fläche?
- Dieselbe!



Quelle:
http://www.echalk.co.uk/amusements/OpticalIllusions/colourPerception/colourPerception.html

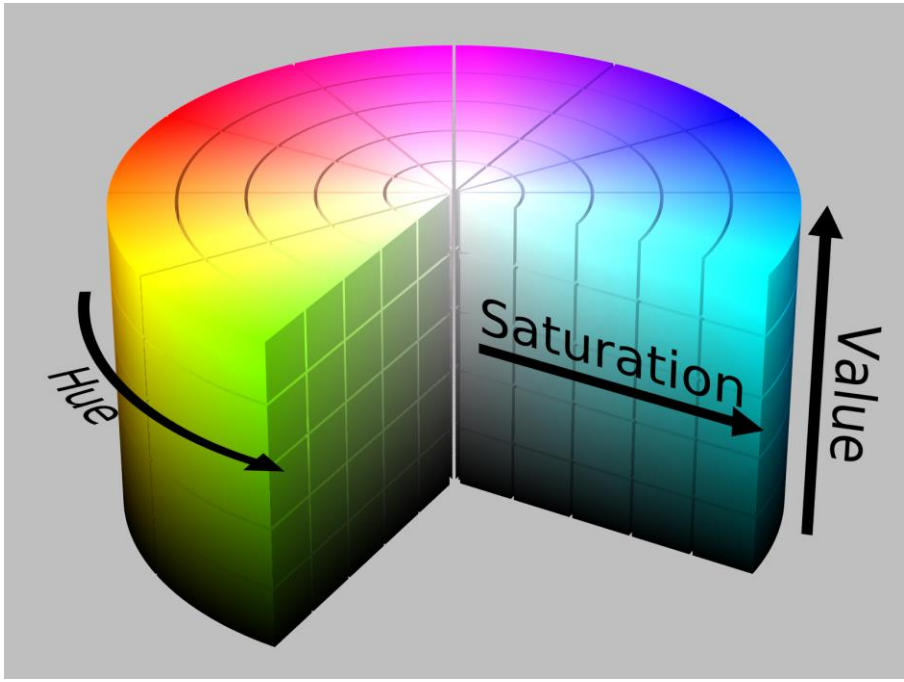
Farbsegmentierung

- Welche Farbe haben die mittleren Quadrate der oberen und vorderen Fläche?
- Dieselbe!
- Folgerung: Der menschliche Blick trügt oft, weil er unbewusst interpretiert.
- Szenarien können viel einfacher aussehen, als sie sind.
- Für Farbe kontrolliertes Licht



Farbsegmentierung

- Beobachtung (VL 4a): Beleuchtung beeinflusst
  - am stärksten Helligkeit
  - mittelstark Sättigung
  - am schwächsten Farbton
- Farbmodell HSV beschreibt Farbe durch Hue, Saturation und Value
  - Hue: Winkel im Farbkreis parametrisiert „Buntfarbe“
  - Saturation: Mischen der Farbe mit Weiß
  - Value: Mischen der Farbe mit Schwarz
- HSV Formel
- Quelle: https://en.wikipedia.org/wiki/HSL_and_HSV



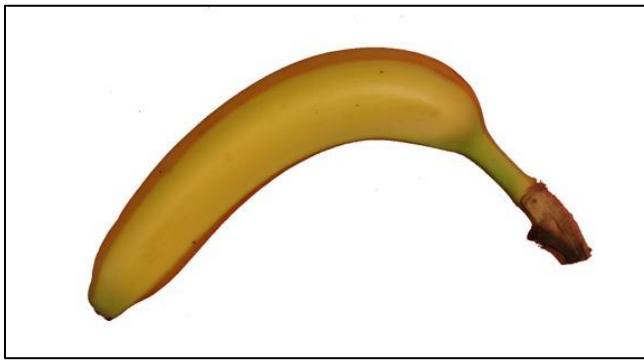
Quelle: SV_color_solid_cylinder.png: SharkD , https://en.wikipedia.org/wiki/HSL_and_HSV#/media/File:HSV_color_solid_cylinder_saturation_gray.png

Farbsegmentierung

- Farbbilder
  - nach HSV konvertieren (Hue, Saturation, Value)
  - Pixel innerhalb eines vorkonfigurierten HSV Bereichs als Objekt akzeptieren
  - [Hmin, Smin, Vmin], [Hmax, Smax, Vmax]
  - OpenCV: `inRange`
- Warum HSV-Bereich?
  - Beleuchtung beeinflusst stark V, etwas S
  - Daher kleiner H-Bereich
  - Nicht als RGB-Bereich ausdrückbar
- Zyklischer Hue
  - Hue bildet Farbkreis (0=180=Rot)
  - Rot ist daher oft zyklischer Hue-Bereich
  - z.B. [170..10] = [170..180] ∪ [0..10]
  - benötigt Hilfsfunktion (vgl. `cyclicInRangeImage, segment-inrange.py`)

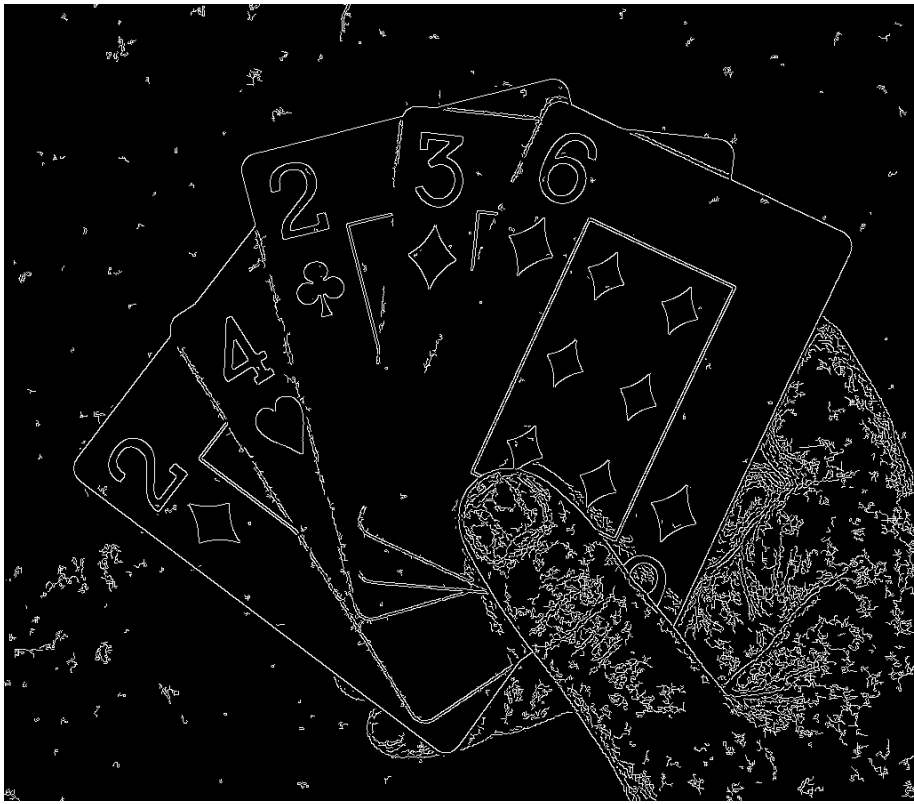
Farbsegmentierung

- Woher kommt der HSV-Bereich
  - manuell einstellen
  - mit einem Tool interaktiv einstellen (`segment-inrange.py`)
  - durch Feinjustieren der Werte
  - oder Klicken auf Objektpixel
- Beispiel: Rote Gummibärchen
  - [172, 114, 102] ... [10, 255, 255]
- Variante
  - Hintergrundfarbe erkennen und invertieren
- Beispiel: Hintergrund der Bananen
  - [0, 232, 140] ... [8, 255, 229]



Kantenerkennung

- Kantenerkennung mit Canny-Algorithmus
  - Sobel-Filter der auf Kanten anspricht (siehe später)
  - Nur lokale Maxima
  - 2 Schwellwerte für Starten und Fortführen einer Kante
  - OpenCV: `Canny`
- Vorteil
  - kein Bezug auf absolute Farben
- Nachteil
  - Linien oft nicht geschlossen
  - viel "Müll" an weichen Übergängen
- Beispiel: Karten
  - Symbole erkannt



Zusammenfassung

- Farbe eines Objektes im Bild hängt von Beleuchtung ab
  - Festes Licht verwenden
- Segmentierungsbasierte Bildverarbeitung
  - Kamera von oben, festes Licht, fester Hintergrund
- Segmentierung / Binarisierung
  - Grauwertschwellwert
  - HSV-Bereich
  - VG oder HG erkennen
  - Alternative: Kanten