

# BILDVERARBEITUNG MIT RACKET – TEIL II

BILDVERARBEITUNGSPRAKTIKUM, SOMMERSEMESTER 2015  
UNIVERSITÄT HAMBURG, FACHBEREICH INFORMATIK  
LEONIE DRESCHLER-FISCHER, DAVID MOSTELLER UND BENJAMIN SEPPKE

## 1. AUFGABE: VON GRAUWERT- ZU FARBBILDERN

In Teil I haben wir noch mit Grauwert-Bildern gearbeitet, nun wollen wir uns den Umgang mit Farbbildern anschauen. Dazu vergleichen wir erst einmal wie die Bilder in Racket repräsentiert sind, nachdem wir DrRacket gestartet haben:

### BILDER LADEN

Damit man ein Bild laden kann, muss man zunächst die VIGRA-Anbindung einbinden.

```
(require vigracket)
```

Wir laden heute zwei Bilder, das altbekannte Blox-Bild und ein Farbbild:

```
(define blox (load-image  
              (build-path vigracket-path "images/blox.gif")))  
  
(define lenna (load-image  
              (build-path vigracket-path "images/lenna_face.png")))
```

Zunächst können jetzt beide Bilder betrachtet werden, mit der bekannten Funktion:

```
(show-image lenna „Das neue Bild“)
```

Wir können auch direkt `lenna` oder `blox` eingeben, um zu schauen, wie die Bilder intern repräsentiert sind.

## 2. AUFGABE: KANALREPRÄSENTATION VON FARBBILDERN

Eine typische Vorgehensweise ist die Aufteilung von Farbbildern in ihre einzelnen Kanäle. Auch die Vigracket funktioniert in vielen Punkten so. Man kann aber auch selbst die enthaltenen Kanäle betrachten:

### ZERLEGUNG IN FARBKANÄLE

Probiert nun das Lenna-Bild in seine Rot, Grün und Blau-Komponenten zu zerlegen. Hierfür gibt es vorgefertigte Funktionen:

```
image->red, image->green und image->blue
```

Schaut euch die Ergebnisse an, und notiert Auffälligkeiten!

## KANAL-KOMBINATION ZU EINEM FARBBILD

Wie kann man aus diesen Einzelkanalbildern wieder ein Farbbild zusammensetzen? Stellt das Originalbild aus seinen Kanalbildern wieder her, und stellt zusätzlich ein Bild mit vertauschten Kanälen (z.B. rot ↔ grün Kanal vertauscht) her.

Tipp: Denkt an Listen!

## OPERATIONEN AUF FARBBILDERN

Farbbilder können meist wie normale Grauwert-Bilder behandelt werden, insbesondere lassen sich alle Operationen von gestern darauf anwenden. Einige probieren wir nun noch einmal auf dem Lenna-Bild:

```
(gsmooth img scale) und (ggradient img scale)
```

Probiert z.B. die Werte: 1.0 und 2.0 für den Parameter `scale`. Was fällt, insbesondere im Vergleich mit Grauwertbildern, bei den Ergebnissen auf?

# PAUSE

## 3. AUFGABE: SELEKTIVE GLÄTTUNG

Ähnlich zur gestrigen Aufgabe 3 wollen wir auch für Farbbilder einen Schwellenwert-Filter definieren. Leider können wir dazu im Moment `image-map` nicht so einfach verwenden, da wir drei unterschiedliche Kanäle in unserem Lenna-Bild haben.

- Wendet die Funktion zur Schwellenwert-Filterung auf den Rotkanal an. Erstellt verschiedene Ergebnisbilder mit Schwellenwerten von: 180.0, 200.0 und 220.0
- Erstellt auf dieser Grauwert-Maske eine Farbwert-Maske, die für jeden Kanal gleich ist. Tipp: Schaut euch Aufgabe 2 (Kanal-Kombination) von heute noch einmal an!
- Schreibt unter Zuhilfenahme von `image-map` eine Funktion, die ein Farbbild, eine Maske und eine Glättungsskala als Parameter annimmt. Die Funktion soll als Ergebnisbild ein Bild zurückgeben, dass nur an den maskierten Gebieten geglättet worden ist.

*Anmerkung:*

Nach einem ähnlichen Prinzip funktionieren die einfachen Haut-Weichzeichnungsfilter, die in den meisten Digitalkameras als digitale Filter enthalten sind!

## ZUSATZAUFGABE 1: SEGMENTIERUNG VON BILDERN

Siehe Referenz-Handbuch zur Vigracket, Beispiel 3, Abschnitt 6.3.

## ZUSATZAUFGABE 2: BILDHISTOGRAMME

- a) Schreibt – vorerst nur für Grauwert-Bilder – eine Funktion, welche die Anzahl der Farbwerte in ein Histogramm überführt (z.B. einen `vector` mit 256 Einträgen).
- b) Schreibt eine kleine Plotting-Funktion, der die Histogramme in einem Bild ausgibt, welches ähnlich einem Block-Diagramm die einzelnen Anzahlen auf die Höhe des Bildes skaliert einträgt.