

Peter Ellenberg
Matr.Nr.: 08613998
Gruppe 10

René Valdéz-Voges
Matr.Nr.: 04732700
Gruppe 08

Praktikum
Bilddatencodierung

Prof. Dr. Nischwitz

Fachbereich Informatik
7. Semester Informatik in der Wirtschaft

Fachhochschule München

Inhaltsangabe:

2. <i>Praktikumsaufgabe</i>	3
Aufgabe 2.1	3
Aufgabe 2.2	5
Aufgabe 2.3	6
Aufgabe 2.4	8

2. Praktikumsaufgabe

Aufgabe 2.1

Unterabtastung und Quantisierung

Es soll wie in dem Vorlesungsbeispiel ein Bild Quantisiert werden.

Hierzu wurde mit den Modul Geometrie/Skala aus dem Bild "Wespe" drei verschiedene Größen verwendet: 256x256, 128x128 und 64x64.

Für jede größere Rasterung wurden die Anzahl der Graustufen pro Farbe R-G-B quantisiert. Das Modul thresh aus dem Themenkreis Segmentierung stellt die Funktion "Äquidensiten" zur Verfügung. Mittels dieses Moduls können Graustufenintervalle definiert und entsprechende Mittelwerte angegeben werden. Nachdem für jedes Bild die drei Bildkanäle R-G-B einzeln extrahiert wurden, konnte das Modul thresh/Äquidensiten auf jedes einzelne Graubild angewendet werden. Mit dem Modul basis/merge wurden die drei einzelnen Graubilder wieder zu einem Bild, das jetzt aber weniger Farben besitzt, zusammengesetzt. Da die Farben aus den Originalbilder nicht gleichverteilt sind, sondern viele homogene Bereiche aufweisen, werden in den reduzierten Bildern weniger Farben dargestellt.

In einer Konfigurationsdatei werden diese Intervalle angegeben:

Für 16 Graustufen pro Farbe (Mögliche Kombinationen für Farbwerte: 4096 (16*16*16 Graustufen))

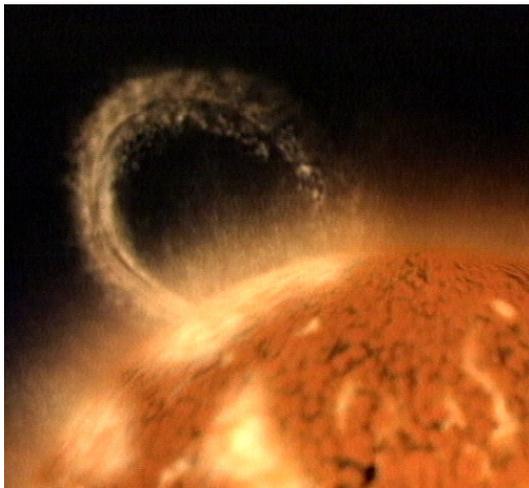
Untere Grenze	Obere Grenze	Mittelwert
0	15	8
16	31	24
32	47	40
48	63	56
64	79	72
80	95	88
96	111	104
112	127	120
128	143	136
144	159	152
160	175	168
176	191	184
192	207	200
208	223	216
224	239	232
240	255	248

Für 4 Graustufen pro Farbe (Mögliche Kombinationen für Farbwerte: 64 (4*4*4 Graustufen))

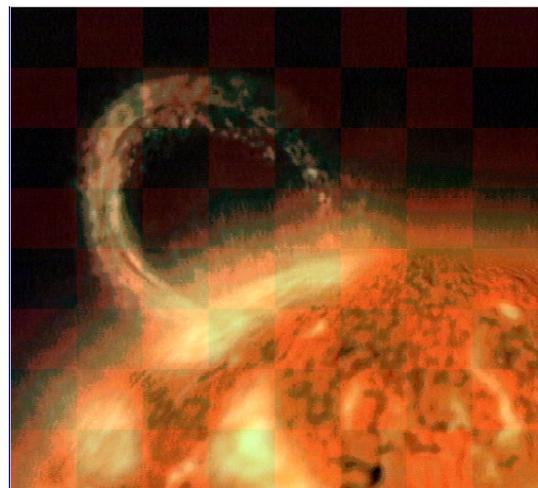
Untere Grenze	Obere Grenze	Mittelwert
0	63	32
64	127	96
128	191	160
192	255	224

Beispiel zur Unterabtastung und zur Quantisierung:

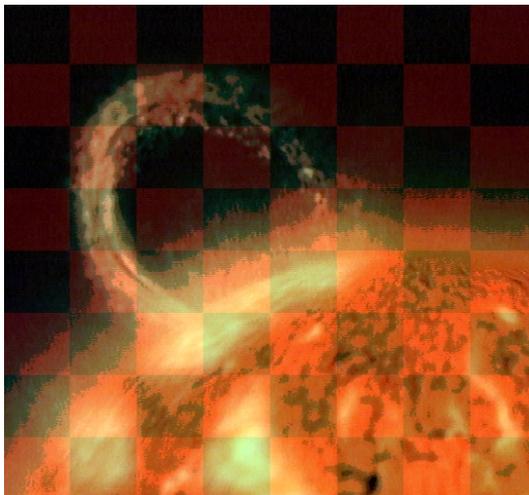
		
256x256, 16777216 Farben 1572864 bit, 192 KB	256x256, 4096 Farben 786432 bit, 96 kByte	256x256, 64 Farben 393216 bit, 48 kByte
		
128x128, 16777216 Farben 393216 bit, 48 kByte	128x128, 4096 Farben 196608 bit, 24 kByte	128x128, 64 Farben 98304 bit, 12 kByte
		
64x64, 16777216 Farben 98304 bit, 12 kByte	64x64, 4096 Farben 49152 bit, 6 kByte	64x64, 64 Farben 24576 bit, 3 kByte

Aufgabe 2.2

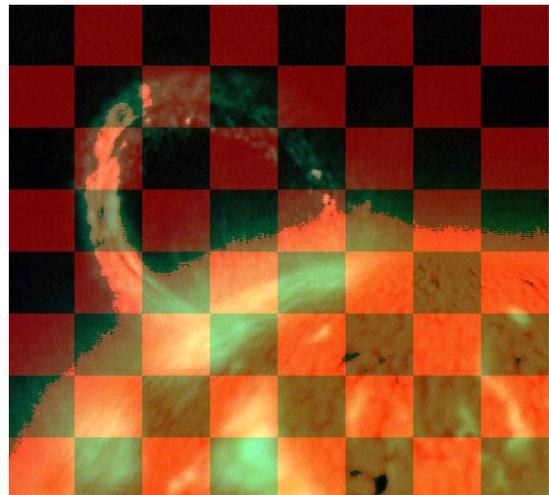
Originalbild nova.bld



Bitebene 5, 1.Kanal durch schach.bld ersetzt



Bitebene 4,5, 1.Kanal durch schach.bld ersetzt



Bitebene 4,5,6, 1.Kanal durch schach.bld ersetzt

Wie zu sehen ist, ist das Schachbrett auf Bitebene 5 sehr schwach zu erkennen. Je niedriger die Bitebene ist, um so schlechter ist das eingefügte Bild zu erkennen. In den folgenden Bildern werden erst 2, dann drei Bitebenen ersetzt. Dadurch tritt das Schachbrettmuster immer stärker in den Vordergrund. Wir haben in unserem Beispiel nur im Kanal 1, also „Rot“, ersetzt. Dadurch sind die weißen Felder des Schachbretts intensiv rot geworden. (255 Grauwert im Schachbrett wird zu 255 Grauwert im Kanal „Rot“).

Dieses Skript ersetzt im Originalbild „nova.bld“ die vierte, fünfte und sechste Bitebene des ersten Kanals durch die Bitebenen des Bildes „schachbrett.bld“.

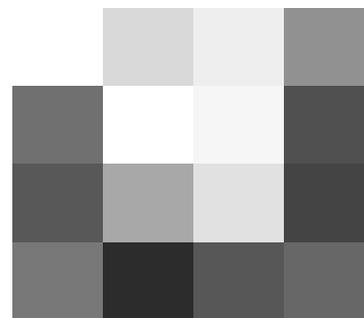
```
# Schachbrettmuster in Bitebene 5 schreiben
$IWEXE/bitrep "e1b'nova' kanal=1 e2b'schach' schwelle=127 bitebene=5 ab'nova5schach"
# Schachbrettmuster in Bitebene 4 schreiben
$IWEXE/bitrep "e1b'nova5schach' kanal=1 e2b'schach' schwelle=127 bitebene=4
ab'nova45schach"
# Schachbrettmuster in Bitebene 6 schreiben
$IWEXE/bitrep "e1b'nova45schach' kanal=1 e2b'schach' schwelle=127 bitebene=6
ab'nova456schach"
```

Beobachtung beim Konvertieren in ein verlustbehaftetes Format (z.B. JPEG):
Die zuvor ersetzten Bitebenen enthalten nicht mehr die "versteckten" Bilder, die Bitebenen sind nach der Konvertierung nicht mehr in ihrer Originalform vorhanden.

Aufgabe 2.3



altmuc.bld



4x4 Bildauschnitt

255	217	238	145
112	255	246	80
88	168	225	68
120	44	87	103

Grauwerte

Zuerst muss das arithmetische Mittel berechnet werden:

n (Anzahl Werte) = 16

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

$$\bar{x} = 153$$

Daraus können wir nun eine Bitmaske erstellen. Alle Grauwerte, die größer als das arithmetische Mittel sind, werden auf 1 gesetzt. Die anderen auf 0.

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

Bitmaske

Dann wird die Streuung der Grauwerte abweichend vom arithmetischen Mittel berechnet:

q (Werte über Mittel) = 7

$$\sigma^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$$

$$\sigma = \frac{1}{15} \sqrt{10404 + 4096 + 7225 + 64 + 1681 + 10404 + 8649 + 5329 + 4225 + 225 + 5184 + 7225 + 1089 + 11881 + 4356 + 2500}$$

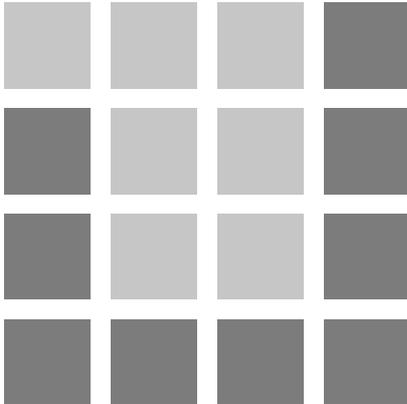
Streuung $\sigma = 72$

Nun werden die Parameter a und b ermittelt.

$$a = \bar{x} - \sigma \left(\frac{q}{n-q} \right)^{\frac{1}{2}} = 153 - 72 * \left(\frac{7}{9} \right)^{\frac{1}{2}} = 125$$

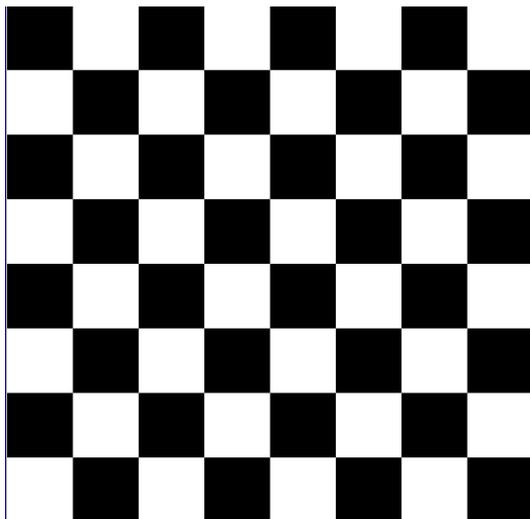
$$b = \bar{x} + \sigma \left(\frac{n-q}{q} \right)^{\frac{1}{2}} = 153 + 72 * \left(\frac{9}{7} \right)^{\frac{1}{2}} = 199$$

Diese Werte werden nun auf die Bitmaske angewendet, und überall, wo eine 1 kodiert war, wird der Grauwert von „b“, wo eine 0 kodiert war „a“ gesetzt.



Decoder

Aufgabe 2.4



Im Bild schach.bld existieren nur zwei Grauwerte, die beide mit derselben Häufigkeit auftreten.

$$H = -\sum p(g) \cdot \log_2 p(g)$$

$$p(\text{weiss}) = p(\text{schwarz}) = 0.5$$

$$H = -2(0.5 \cdot \log_2 0.5) = -2 \cdot (0.5 \cdot -1) = 1$$