

## Aufgabe 3 Kompression

0							
1							
2							
3							
4							
5							
6							
7							

1. Geben Sie den Speicheraufwand zur Speicherung des Originalbildes an.
  - es reicht den Speicheraufwand vereinfacht als Anzahl der Werte zu zählen.  
Dass Werte unterschiedlicher Datentypen mit unterschiedlichen Anzahlen von Bits gespeichert werden (können), muss nicht berücksichtigt werden.

Wenn ein Bild nur eine begrenzte Anzahl von Farben enthält (z. B. nur grün, orange, gelb und blau), kann eine Farbpalette verwendet werden, um die Farben darzustellen. Die Farbpalette definiert eine begrenzte Anzahl von Farben, und anstatt den Farbwert speichert jedes Pixel die Nummer der entsprechenden Farbe. Für eine Farbpalette mit 4 Einträgen reichen 2 Bit pro Pixel aus, um die Nummer der Farbe zu speichern.<sup>[1]</sup> In diesem Bild kann auch vereinfacht angenommen werden, jede Zelle kann eine Farbe annehmen, also bei 64 Zellen (8x8) 1 Attribute was ebenfalls zum gleichen Ergebniss führt.

Somit beträgt der Speicheraufwand  $8 \times 8 \times 2 = 128$

2. Wählen Sie eine weitere Kompressionsmethode: Run Length, Value Point, Quadtree oder Lempel-Ziv (Version aus VL) und geben Sie die komprimierte Fassung (geeignet, d.h. tabellarisch oder als Baum) an. Wie hoch ist der Speicherbedarf der komprimierten Fassungen?

Run Length Codierung (RLC): Aufeinanderfolgende gleiche Attributwerte werden als ein Lauf zusammengefasst. Der Speicheraufwand ist bei 23 Läufen = **69**.

Als Zeilentabelle dargestellt:

Lauf	Zeile	Farbe	Zellen
1	0	gelb	4
2	0	orange	4
3	1	gelb	4
4	1	orange	4
5	2	gelb	1
6	2	orange	2
7	2	gelb	1
8	2	grün	1
9	2	orange	3
10	3	gelb	1
11	3	orange	2
12	3	gelb	1
13	3	grün	2
14	3	orange	2
15	4	blau	4
16	4	grün	3
17	4	orange	1
18	5	blau	6
19	5	grün	1
20	5	blau	1
21	6	blau	8
22	7	blau	1
23	7	orange	7

Speicherbedarf:

---

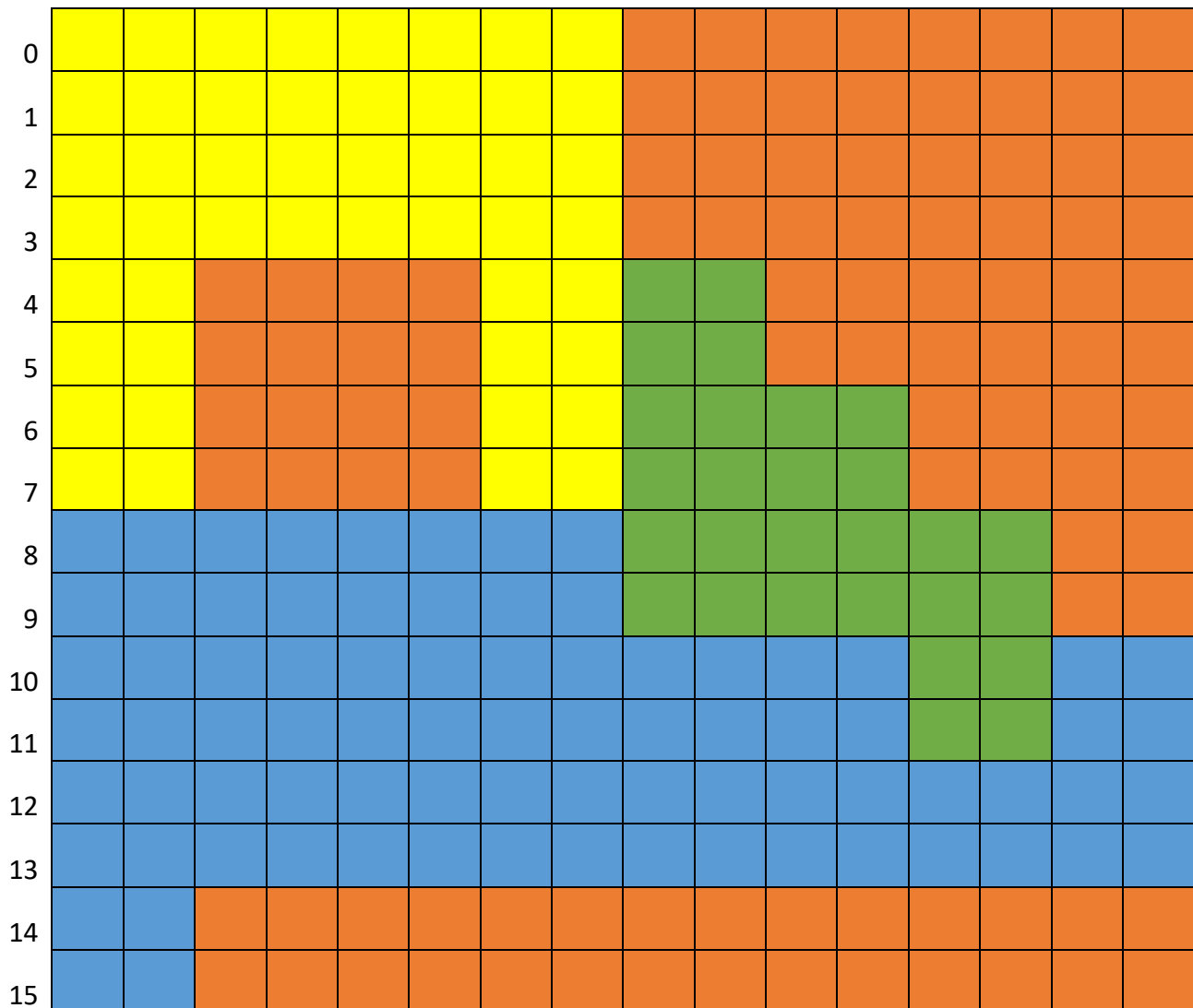


---

69

3. Wie verändert sich der Speicheraufwand in beiden Fällen, wenn das Bild als 16\*16 Bild gespeichert werden würde?

Bei einer Speicherung des Bildes in 16x16, hätte das Originalbild 256 Zellen, was bei vier Farben bedeutet, dass der Speicheraufwand für das unkomprimierte Bild 512 beträgt. Somit vervierfacht sich der Speicherbedarf.



Lauf	Zeile	Farbe	Zellen
1	0	gelb	8
2	0	orange	8
3	1	gelb	8
4	1	orange	8
5	2	gelb	8
6	2	orange	8
7	3	gelb	8
8	3	orange	8
9	4	gelb	2
10	4	orange	4
11	4	gelb	2
12	4	grün	2
13	4	orange	6
14	5	gelb	2
15	5	orange	4
16	5	gelb	2
17	5	grün	2
18	5	orange	6
19	6	gelb	2
20	6	orange	4
21	6	gelb	2
22	6	grün	4
23	6	orange	4

Lauf	Zeile	Farbe	Zellen
24	7	gelb	2
25	7	orange	4
26	7	gelb	2
27	7	grün	4
28	7	orange	4
29	8	blau	8
30	8	grün	6
31	8	orange	2
32	9	blau	8
33	9	grün	6
34	9	orange	2
35	10	blau	12
36	10	grün	2
37	10	blau	2
38	11	blau	12
39	11	grün	2
40	11	blau	2
41	12	blau	16
42	13	blau	16
43	14	blau	2
44	14	orange	14
45	15	blau	2
46	15	orange	14

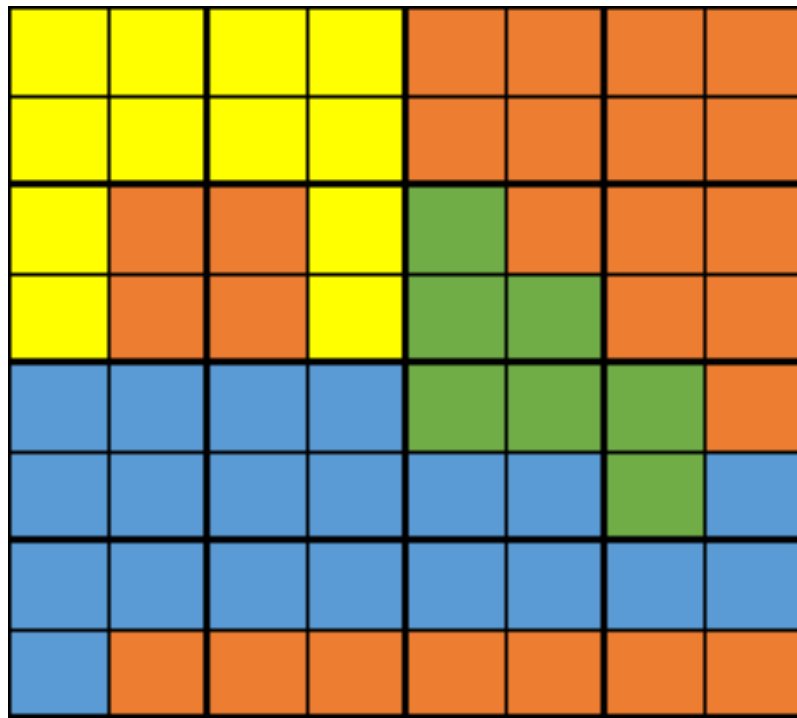
Speicherbedarf:

**138**

Für die komprimierte Version mit RLC dieses Bildes verdoppelt sich der Speicheraufwand und beträgt somit **138**.

4. Stellen Sie die nächste Ebene einer Pyramide zu dem angegebenen Rasterkartenausschnitt dar (4X4 Zellen), die jeweils vier Zellen aggregiert. Welche Aggregationsmethode ist hier sinnvoll (Mittelwert, Mehrheit)?

Da es sich um Farben handelt, ist die Aggregation über den Mittelwert somit nicht sinnvoll. Deshalb wird hier die Methode der Mehrheit genutzt. Das 8x8 Raster wird in 4x4 Blöcke unterteilt. Für jeden Block wird die Farbe ausgewählt, die am meisten darin vorkommt. Bei gleicher Anzahl kann z.B. randomisiert werden. Im unteren Beispiel wurde immer die Farbe für den Block gewählt, welche als erstes im 4x4 Block vorkommt. Somit entsteht folgendes Bild.



## **Quellen**

[1] Weiterbildung Informatik. Codierung von Bilddaten. Zugriffszeit 16.05.2023 12:01.  
<https://weiterbildung-informatik.wollw.de/chapter2/part1/sec2/>