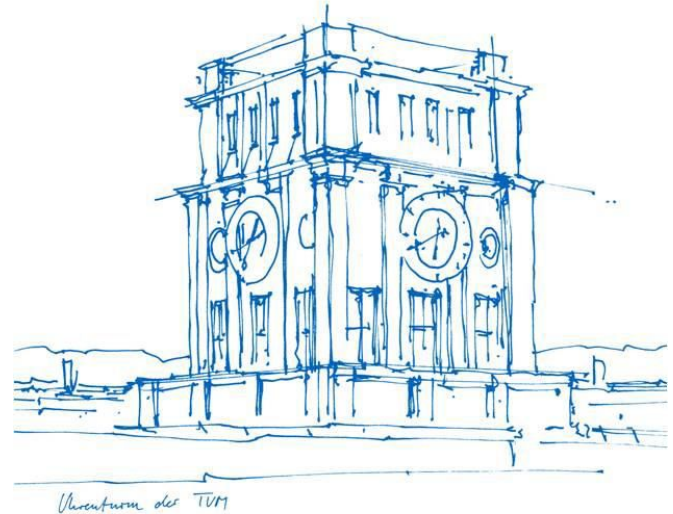


Bitmap Komprimierung

Philipp Czernitzki
Adam Karamelo

Technische Universität München
Fakultät für Informatik
Lehrstuhl für Rechnerarchitektur &
Parallele Systeme
München, 07. Februar 2022



Einführung

Bildkompression - Methode um Dateigröße zu minimieren

Laufängerkodierung / Run-length-encoding (RLE)

-> Algorithmus für verlustfreie Bildkompression

RLE - wird von meisten Bilder Formaten unterstützt wie BMP, TIFF and PCX

BMP - Dateiformat um Rastergrafiken zu erstellen

Bitmap Format (8bpp)

1	Bitmap File Header Informationen über BMP Datei	BYTES 14 FELDER 5
2	Bitmap Information Header Informationen über BMP Bild	BYTES 40 FELDER 11
3	Color Palette Farbpalette in 4-Tupel (r, g, b, reserviert)	BYTES ~
4	Pixeldaten enthält 1-Byte (8bpp) Pixelindizes	BYTES ~

Type(0x4d42), **Size**(Dateigröße)
Offbits(Anzahl der Metadaten)

Size(Header Typ)
Width, Height
Compression(BI_RGB, BI_RLE8)
Size Image

obligatorisch bei 8bpp Bitmap

Pixelindizes in 1-Byte

Bitmap Information Header Typen

- BitmapCoreHeader (12 Bytes)
- BitmapInfoHeader (40 Bytes)
- BitmapV4Header (108 Bytes)
- BitmapV5Header (124 Bytes)

Color Palette

(blau, grün, rot, reserved=0)

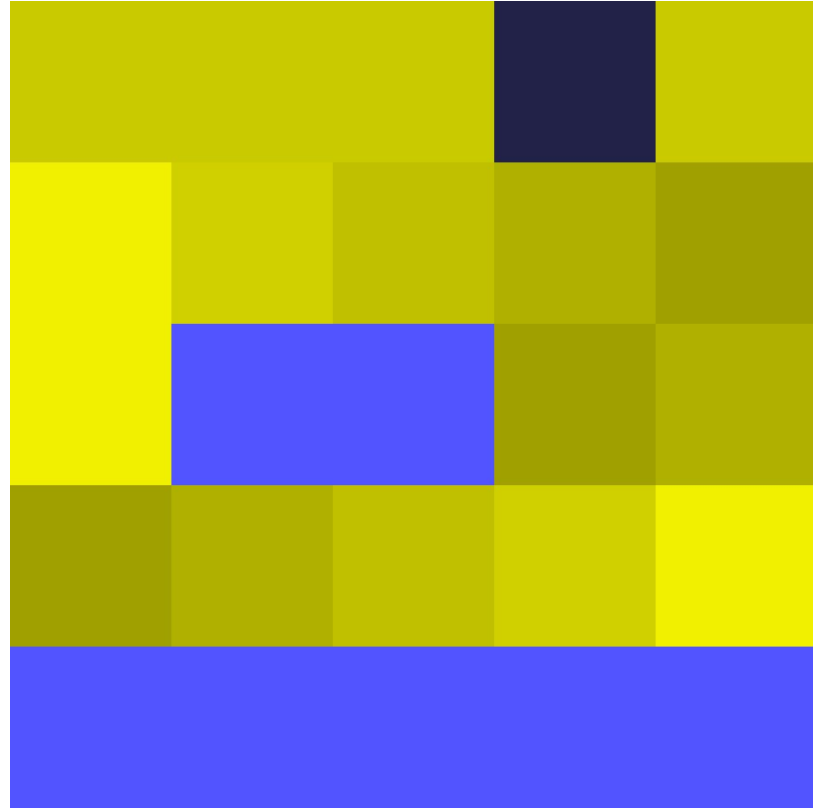
0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15

256 colors

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69
88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105
124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177
196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213
34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51
70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87
106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123
142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195
214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231

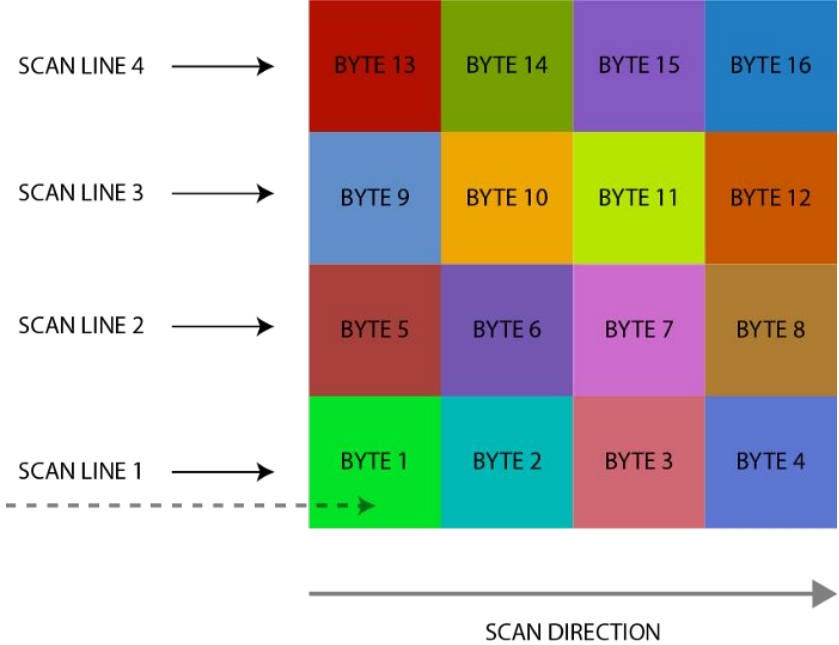
232	233	234	235	236	237	238	239	240	241	242	243
244	245	246	247	248	249	250	251	252	253	254	255

Eine 5x5 Bitmap

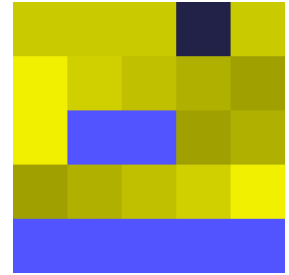


● BYTE 1	BYTE 2	BYTE 3	BYTE 4
BYTE 5	BYTE 6	BYTE 7	BYTE 8
BYTE 9	BYTE 10	BYTE 11	BYTE 12
BYTE 12	BYTE 14	BYTE 15	BYTE 16

PIXEL DATA

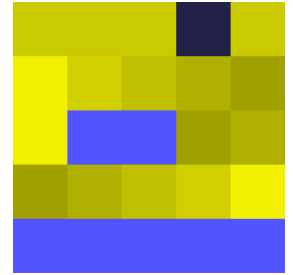


Bitmap ohne Lauflängenkodierung



00000000	42	4d	7e	00	00	00	00	00	00	00	56	00	00	00	28	00	BM~.....V...(.
00000010	00	00	05	00	00	00	05	00	00	00	01	00	08	00	00	00
00000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	08	00
00000030	00	00	00	00	00	00	ff	54	52	00	00	a0	a0	00	00	b0TR.....
00000040	b0	00	00	c0	c0	00	00	d0	d0	00	00	f0	f0	00	00	ca
00000050	ca	00	49	22	22	00	00	00	00	00	00	00	00	00	01	02	..I"".....
00000060	03	04	05	00	00	00	05	00	00	01	02	00	00	00	05	04
00000070	03	02	01	00	00	00	06	06	06	07	06	00	00	00		
0000007e																	

Bitmap mit Lauflängenkodierung



00000000	42	4d	80	00	00	00	00	00	00	00	00	56	00	00	00	28	00	BM.....V...(.
00000010	00	00	05	00	00	00	05	00	00	00	01	00	08	00	01	00	
00000020	00	00	2a	00	00	00	00	00	00	00	00	00	00	00	08	00	...*.....	
00000030	00	00	00	00	00	00	ff	54	52	00	00	a0	a0	00	00	b0TR.....	
00000040	b0	00	00	c0	c0	00	00	d0	d0	00	00	f0	f0	00	00	ca	
00000050	ca	00	49	22	22	00	05	00	00	00	00	05	01	02	03	04	..I"".....	
00000060	05	00	00	00	00	05	05	00	00	01	02	00	00	00	00	05	
00000070	05	04	03	02	01	00	00	00	03	06	01	07	01	06	00	01	
00000080																		

Einfarbige Bitmap mit Lauflängenkodierung

00000000	42	4d	62	00	00	00	00	00	00	00	3a	00	00	00	28	00	BMb.....:...(.
00000010	00	00	96	00	00	00	0a	00	00	00	01	00	08	00	01	00
00000020	00	00	28	00	00	00	00	00	00	00	00	00	00	00	01	00	..(.....
00000030	00	00	00	00	00	00	b7	00	ff	00	96	00	00	00	96	00
00000040	00	00	96	00	00	00	96	00	00	00	96	00	00	00	96	00
*																	
00000060	00	01															..



SIMD - Optimierung

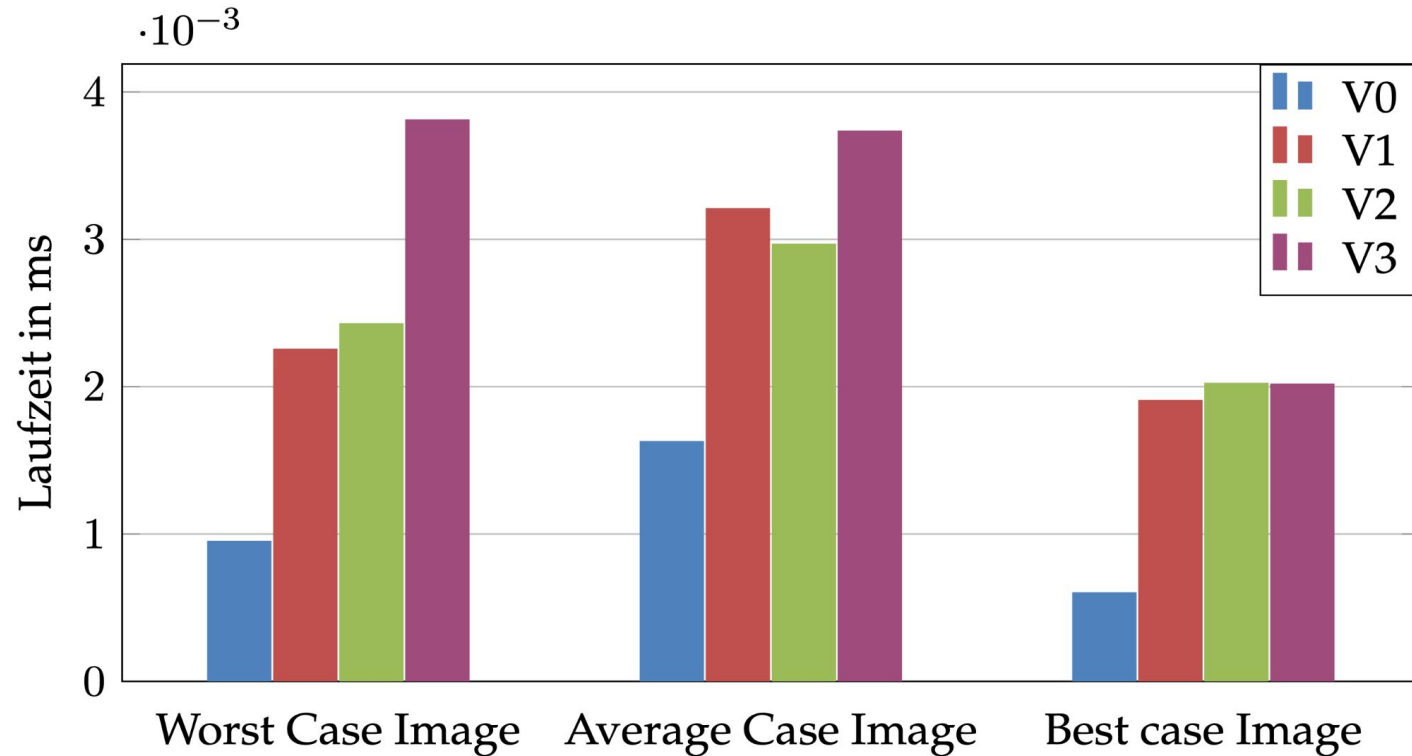
`_mm_cmpeq_epi8(pixels, pixelsShifted)` - 16 Byte Blöcke vergleichen

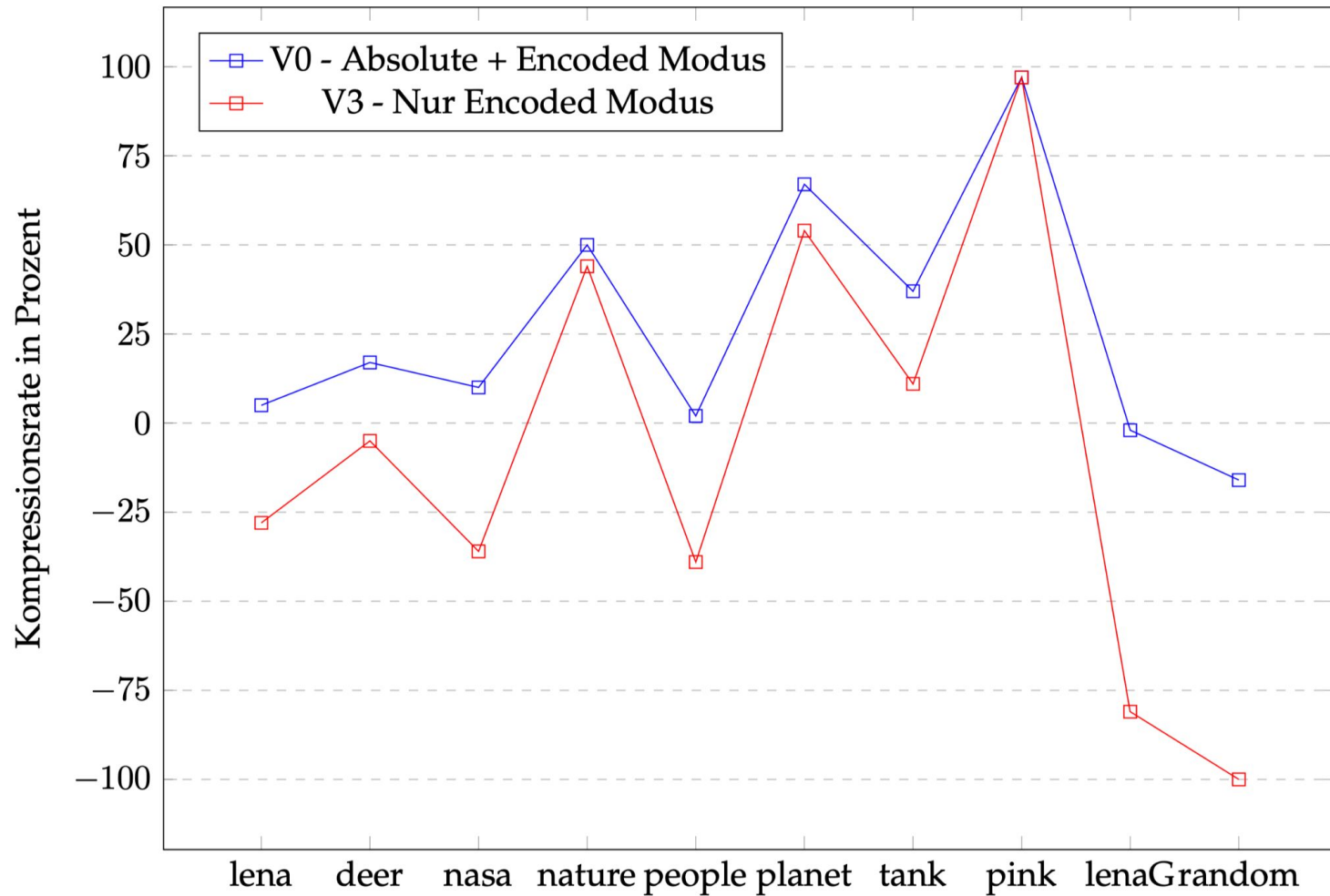
0x00 0x01 0x02 0x02 0x03 0x04 0x04 0x04 0x05

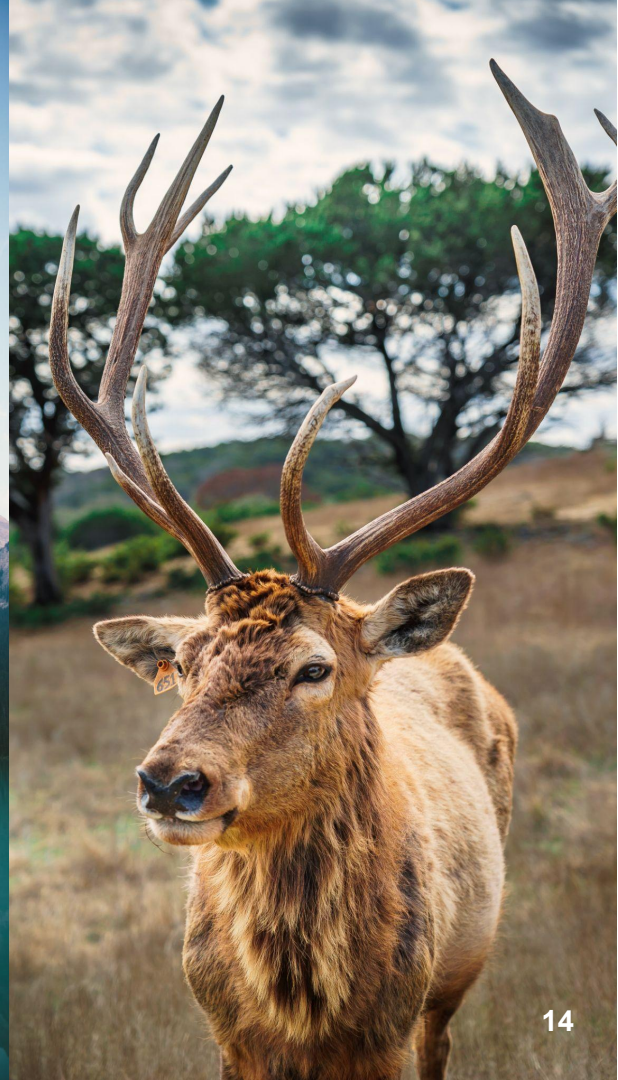
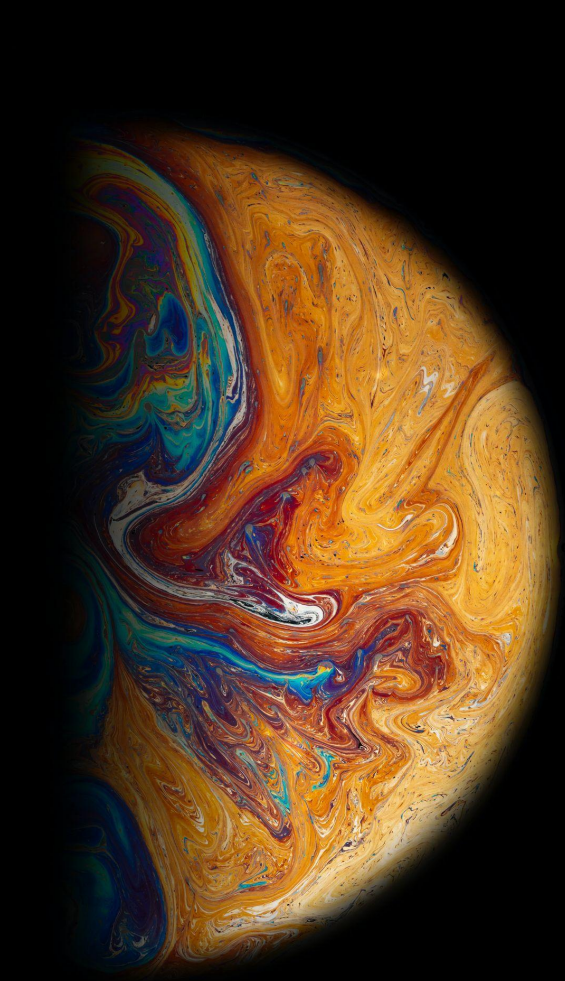
0x01 0x02 0x02 0x03 0x04 0x04 0x04 0x05

0x00 0x00 0xff 0x00 0x00 0xff 0xff

Performanzanalyse







Komprimierung

Kompressionsrate

= unkomprimierte Größe / komprimierte Größe

= $257/244 = 1.053$

Die Kompressionsrate ist 1.053 : 1

Speicherplatz Zugewinn

= $1 - (\text{komprimierte Größe} / \text{unkomprimierte Größe})$

= $0,05 = 5\%$





Danke für die Aufmerksamkeit!

Wie wünschen euch EXZELLENZ!