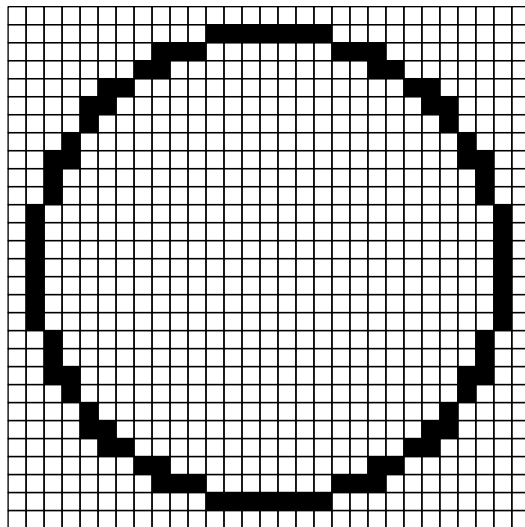
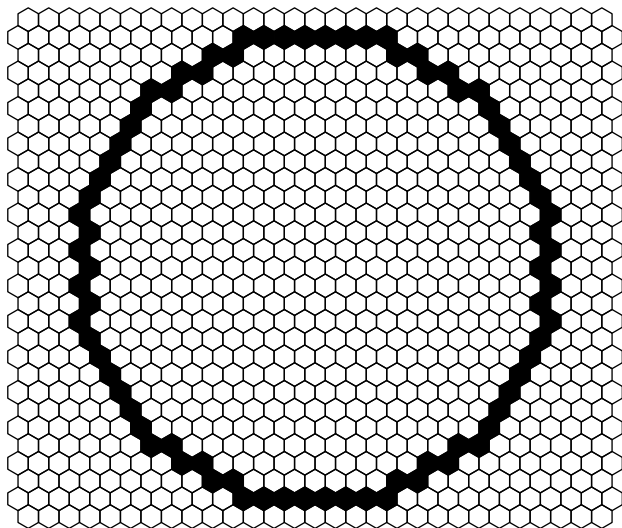


Bogusław Jackowski

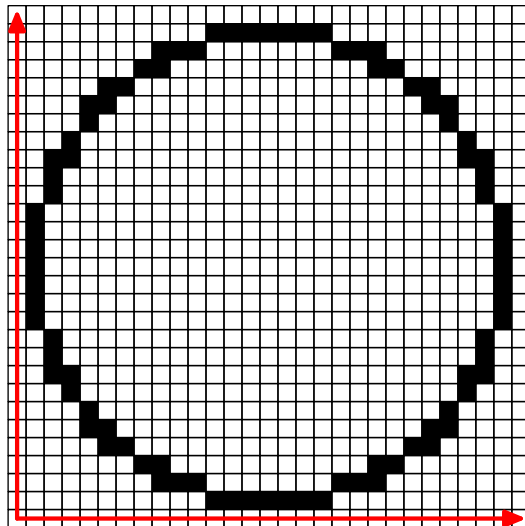
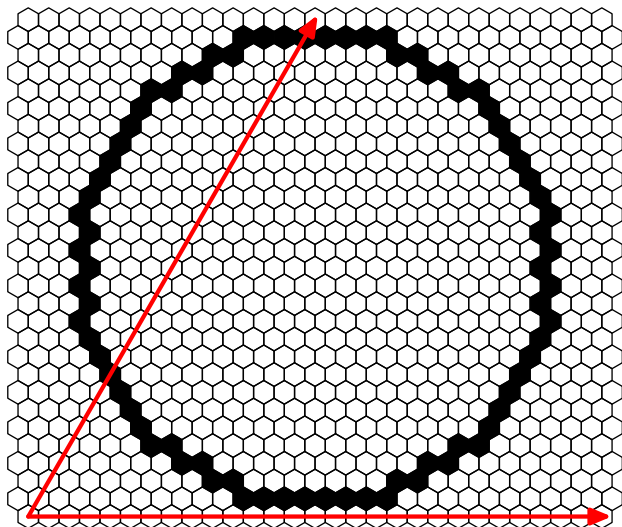
GRAFIKA DYSKRETNA

BACHOTEK 1998

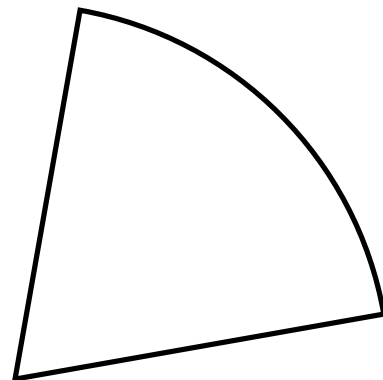
Motto: bit kształtuje świadomość
filozofia Hakunów



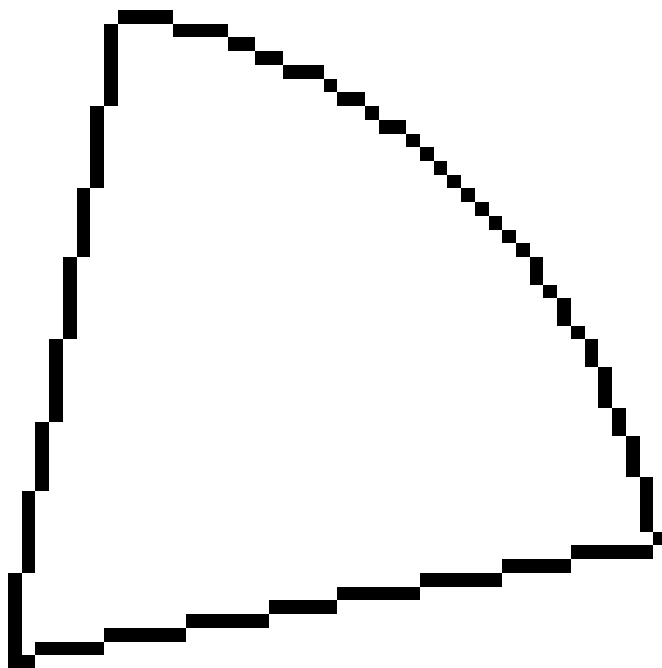
Mapa bitowa niekoniecznie musi być prostokątna...



... co niekoniecznie musi być wygodne



Grafika obwiedniowa pozwala na chwilę zapomnieć
o mapach bitowych...



... ale i tak wszystkie drogi prowadzą do map bitowych

	0	1	2	...	j	...	n
0							
1							
2							
...							
i					$V_{i,j}$		
...							
m							

Mapa bitowa to z formalnego punktu widzenia
tablica liczb $V_{i,j}$

	0	1	2	...	j	...	n
0	■	□	■	□	■	■	□
1	■	□	□	■	■	□	□
2	□	□	□	□	□	■	□
...	□	□	■	□	■	□	□
i	□	□	■	■	$V_{i,j}$	■	□
...	■	□	□	■	■	■	□
m	□	■	□	■	■	■	■

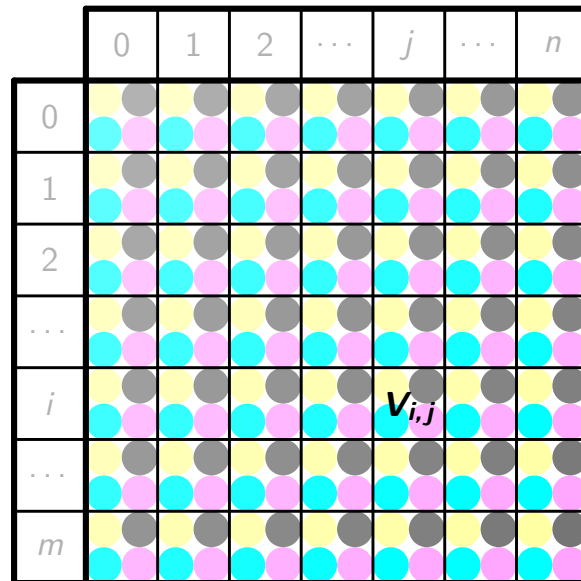
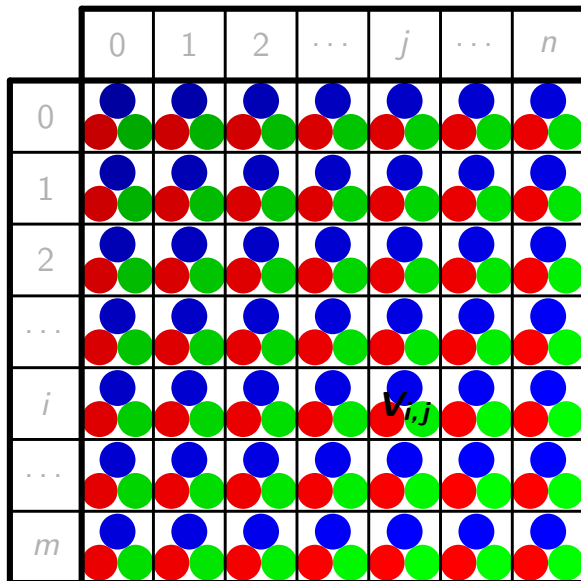
Dwubarwne mapy bitowe: $V_{i,j} = 0, 1$



Co oznacza zero, a co jeden, nie jest oczywiste

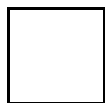
	0	1	2	...	j	...	n
0							255
1							
2							
...							
i					$V_{i,j}$		
...							
m	0						

Cieniowane mapy bitowe: $V_{i,j} = 0, 1, \dots, 255$



Kolorowe mapy bitowe:

RGB: $V_{i,j} = (r, g, b)$; CMYK: $V_{i,j} = (c, m, y, k)$



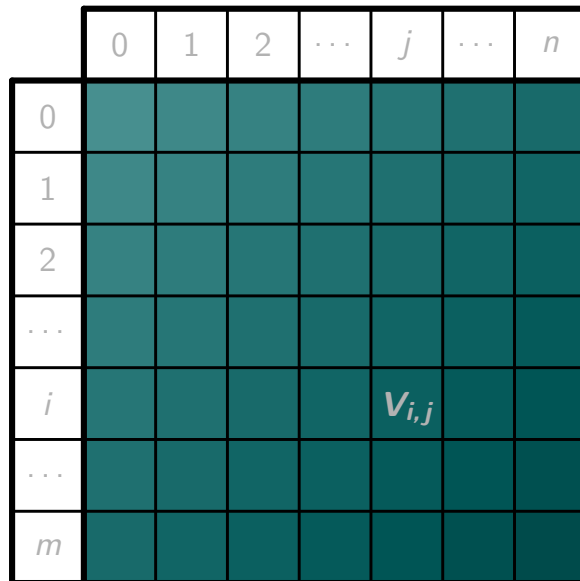
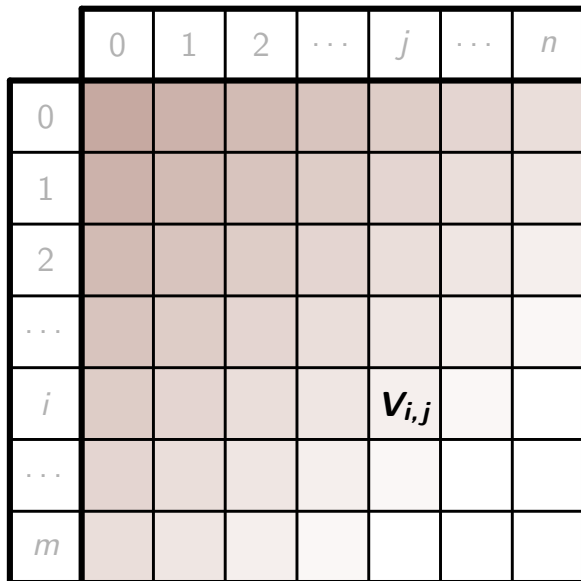
$(1, 1, 1)$

$(0, 0, 0, 0)$



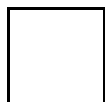
$(0, 0, 0)$

$(c, m, y, 1)$



Kolorowe mapy bitowe:

RGB: $V_{i,j} = (r, g, b)$; CMYK: $V_{i,j} = (c, m, y, k)$



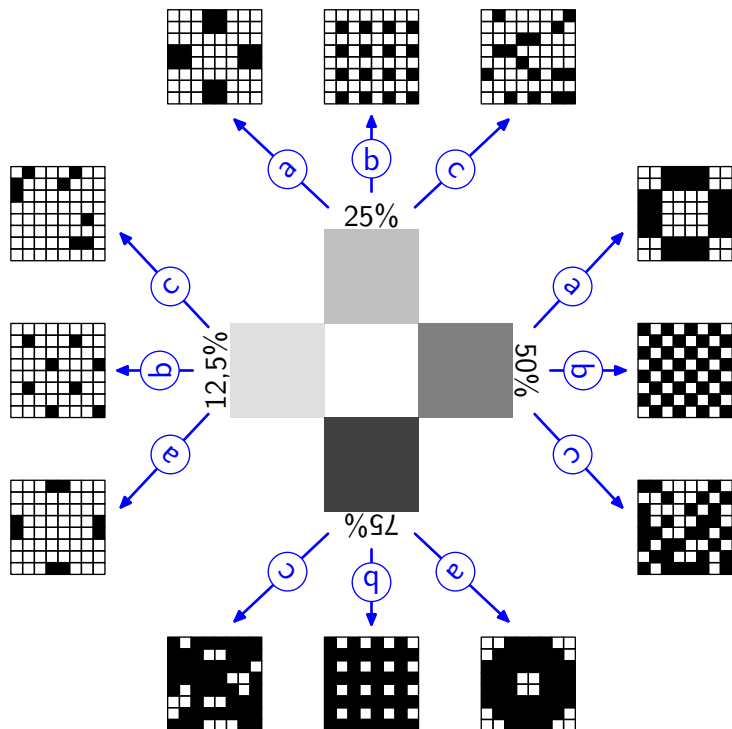
$(1, 1, 1)$

$(0, 0, 0, 0)$

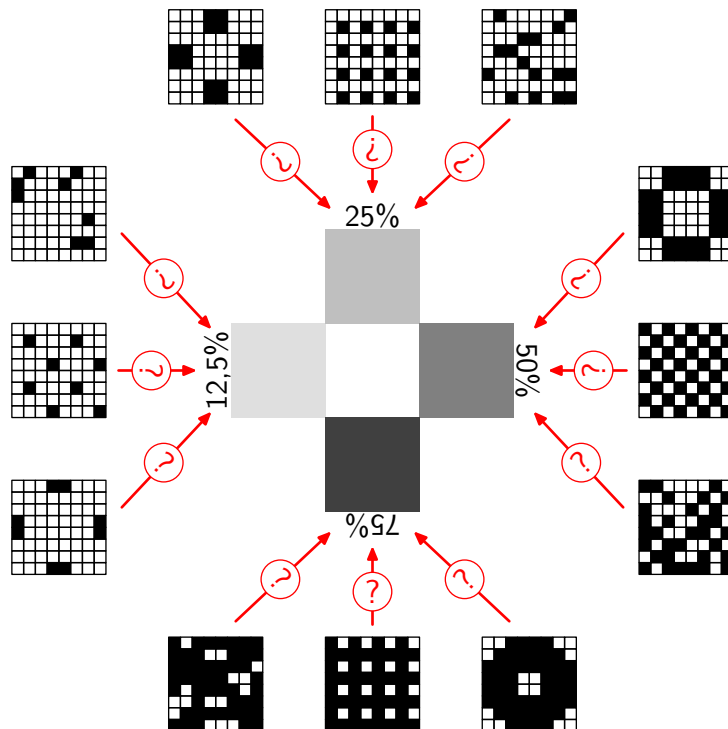


$(0,0,0)$

$(c, m, y, 1)$



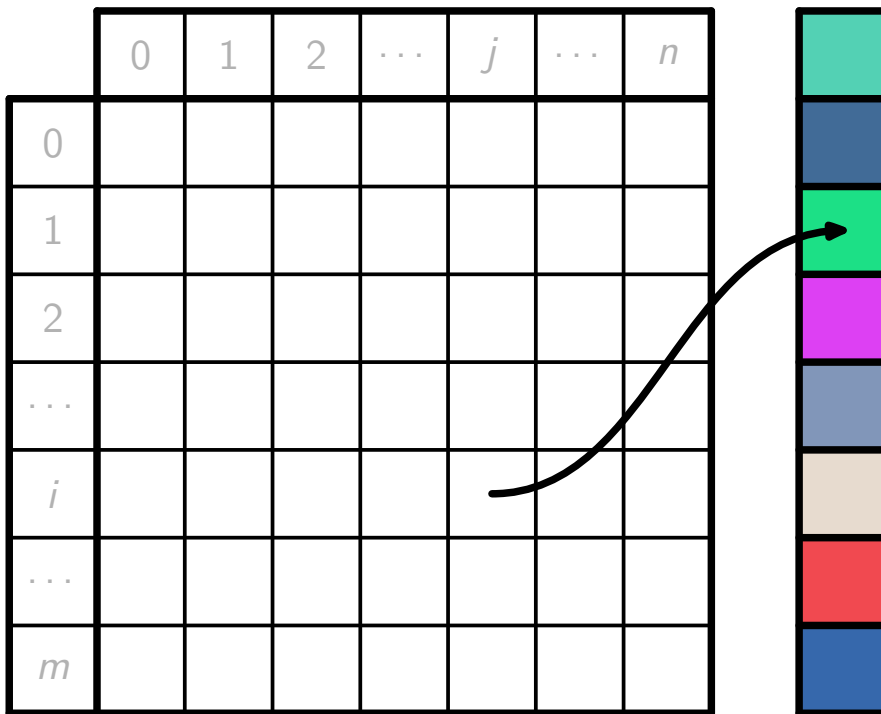
Raster pozwala wyrazić szeroką gamę odcieni na urządzeniach czarno-białych



Zamiana zrastrowanych map bitowych na cieniowane
nie jest zadaniem trywialnym



Mapy bitowe z maską:
dodatkowa informacja może posłużyć do wybrania
elementów zdjęcia



Paletowe mapy bitowe:
 $V_{i,j}$ wskazuje jeden z kolorów palety

Motto: na opuchliznę tylko kompres
medycyna niekonwencjonalna

Dane wejściowe:

0 0 0 0 0 0 255 255 255 0 63 127 127 127 127

Wynik:

6 0 3 255 2 0 63 4 127

liczba powtórzeń:

$$x \rightarrow x - 1$$

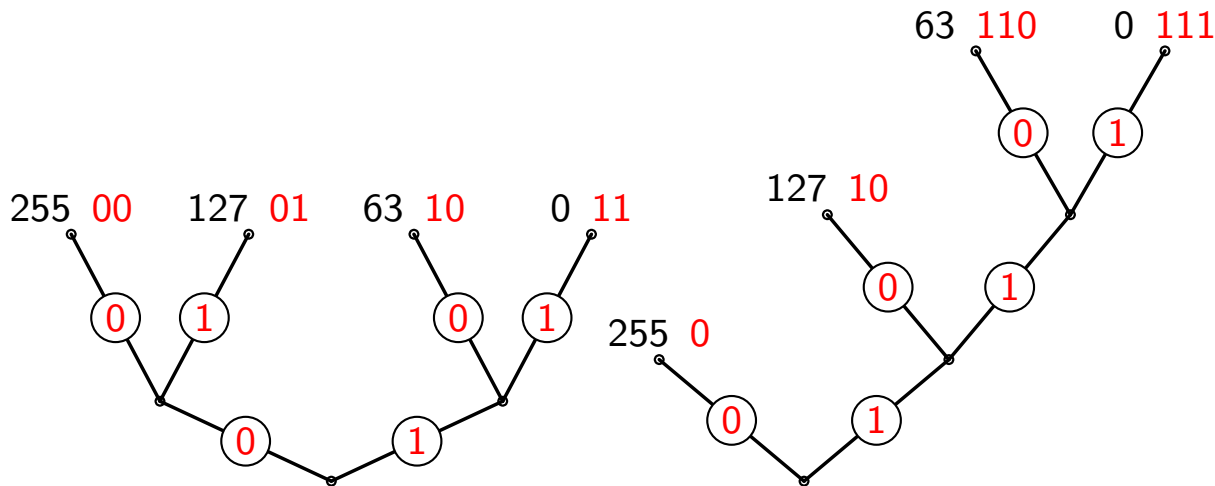
długość fragmentu nie kodowanego:

$$x \rightarrow 127 + x$$

Wynik „jednokolorowy”:

5 0 2 255 129 0 63 3 127

W kompresji RLE (*run length encoding*) kluczową rolę odgrywają znaczniki, zawierające informację o ewentualnym powtarzaniu się danych



00101110

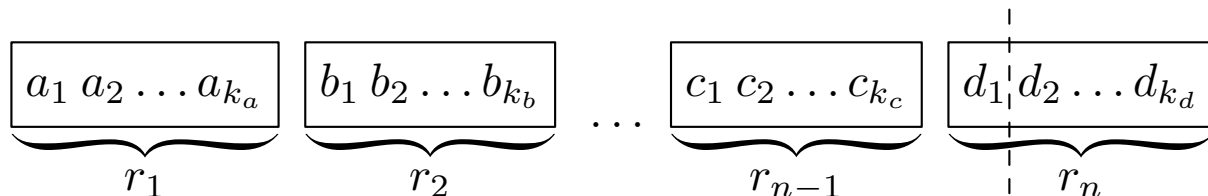
255 63 0 63

255 255 127 0 255

Kompresja Huffmana zamienia dane na ciągi bitów różnej długości; kluczem do odkodowania jest optymalnie skonstruowane drzewo binarne

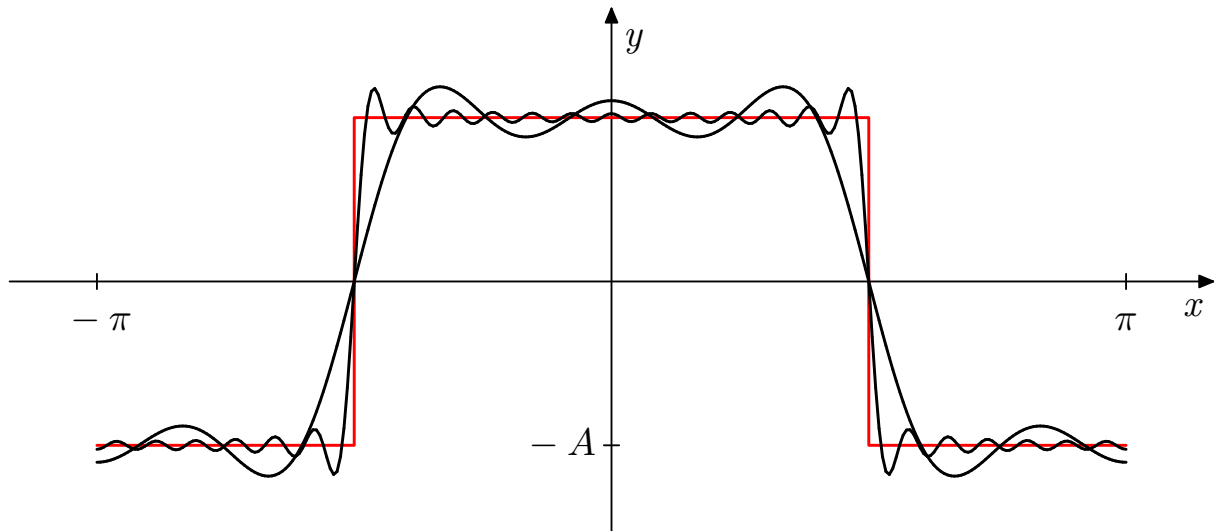
krok	wejście	hasło	słownik	wynik
0	abacccab	a	a b c	<i>nic</i>
1	abacccab	b	a b c ab	0
2	abacccab	a	a b c ab ba	0 1
3	abacccab	c	a b c ab ba ac	0 1 0
4	abacccab	c	a b c ab ba ac cc	0 1 0 2
5	abacccab	cc	a b c ab ba ac cc	0 1 0 2
6	abacccab	a	a b c ab ba ac cc cca	0 1 0 2 6
7	abacccab	ab	a b c ab ba ac cc cca	0 1 0 2 6
8	abacccab		a b c ab ba ac cc cca	0 1 0 2 6 3

Kompresja LZW zamienia ciąg danych na ciąg wskaźników („numerów haseł”) w tworzonym *a vista* słowniku



Podczas dekompresji danych, utworzonych przez algorytm LZW, z ciągu wskaźników odtwarzany jest słownik, a na podstawie słownika – dane źródłowe; jeśli po odkodowaniu wskaźników r_1, r_2, \dots, r_{n-1} wskaźnik r_n wskazuje na puste jeszcze miejsce w słowniku, to zachodzi równość

$$c_1 c_2 \cdots c_{k_c} d_1 = d_1 d_2 \cdots d_{k_d}$$



$$y(x) = \frac{A}{2} + \frac{2A}{\pi} \left(\cos x - \frac{1}{3} \cos 3x + \frac{1}{5} \cos 5x - \frac{1}{7} \cos 7x + \dots \right)$$

Podstawowym algorytmem wykorzystanym w kompresji JPEG jest dyskretna transformacja cosinusowa, mająca wiele wspólnego z aproksymacją funkcji za pomocą wielomianów trygonometrycznych

$$Q_{u,v} = \frac{1}{4} C_u C_v \sum_{x=0}^7 \sum_{y=0}^7 (q_{x,y} - 128) \cos \frac{\pi u (2x+1)}{16} \cos \frac{\pi v (2y+1)}{16}$$

$$q_{u,v} = 128 + \frac{1}{4} \sum_{u=0}^7 \sum_{v=0}^7 C_u C_v Q_{u,v} \cos \frac{\pi u (2x+1)}{16} \cos \frac{\pi v (2y+1)}{16}$$

gdzie $C_u = 1/\sqrt{2}$ dla $u = 0$, a w przeciwnym razie $C_u = 1$;
podobnie jest określona wartość C_v .

Dyskretna transformacja cosinusowa obliczana jest dla „kafelków” 8×8 pikseli, na które rozbijana jest mapa bitowa; wynik można interpretować jako tablicę amplitud składowych harmonicznych

	0	1	2	3	4	5	6	7
0	255	250	243	234	222	205	179	96
1	219	216	212	207	198	186	165	96
2	182	182	181	179	174	166	151	96
3	146	149	151	151	150	147	138	96
4	109	115	120	124	127	127	124	96
5	73	81	89	96	103	108	110	96
6	36	47	58	69	79	88	96	96
7	0	14	28	41	55	69	83	96

Tablica poziomów szarości dla przykładowego „kafelka”

	0	1	2	3	4	5	6	7
0	10,37	43,25	-74,03	39,88	-35,12	21,85	-15,74	7,06
1	422,67	191,34	-47,74	42,79	-22,40	19,31	-10,22	5,34
2	0,07	0,02	-0,09	-0,05	0,07	0,08	-0,04	-0,09
3	44,22	19,90	-5,31	4,55	-2,06	1,74	-1,24	0,86
4	0,63	0,16	0,30	-0,25	-0,88	-0,08	-0,26	0,47
5	12,72	5,89	-1,32	1,19	-0,77	0,54	-0,31	0,20
6	-0,16	-0,05	0,21	0,13	-0,16	-0,19	0,09	0,23
7	2,76	1,29	-0,90	-0,11	0,38	0,04	-0,47	0,07

Tablica amplitud – wynik obliczenia DCT

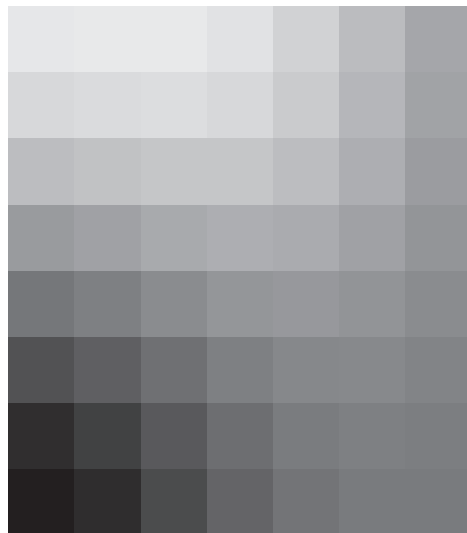
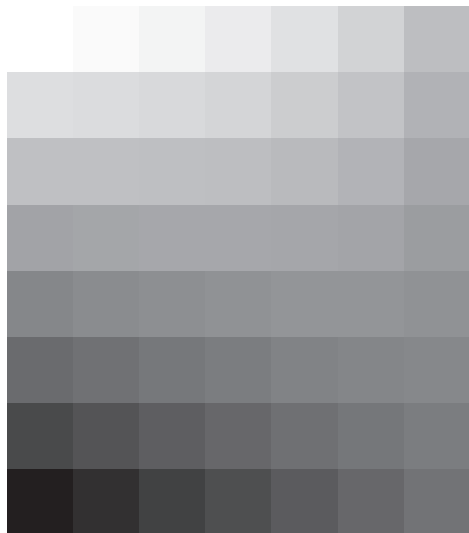
	0	1	2	3	4	5	6	7
0	0	0	-100	0	0	0	0	0
1	400	200	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0

Tablica amplitud po kwantyzacji, to znaczy po zaokrągleniu do wielokrotności liczby 100; kwantyzacja jest momentem, w którym zachodzi utrata jakości danych

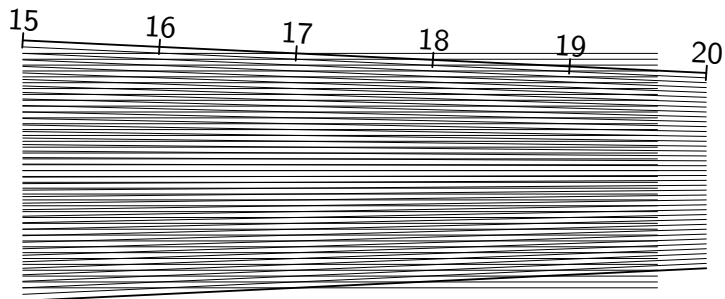
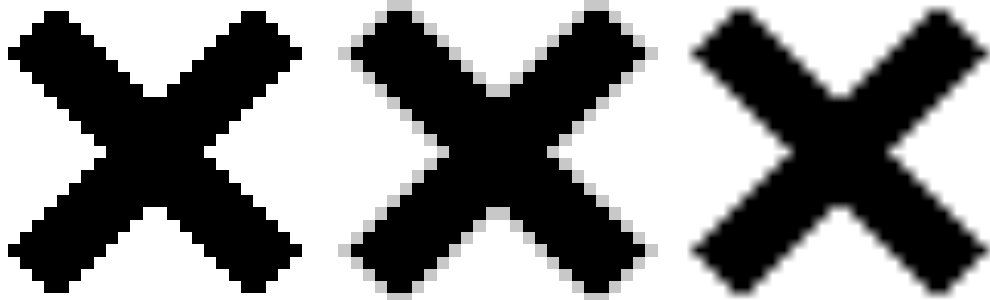
	0	1	2	3	4	5	6	7
0	255	250	243	234	222	205	179	96
1	219	216	212	207	198	186	165	96
2	182	182	181	179	174	166	151	96
3	146	149	151	151	150	147	138	96
4	109	115	120	124	127	127	124	96
5	73	81	89	96	103	108	110	96
6	36	47	58	69	79	88	96	96
7	0	14	28	41	55	69	83	96

0	1	2	3	4	5	6	7
229	231	231	223	204	177	150	133
211	215	217	211	195	170	145	130
178	184	189	189	178	159	137	124
135	143	154	160	156	143	127	116
88	99	116	129	132	126	116	107
45	59	80	100	110	111	105	100
12	28	53	77	94	99	97	94
0	11	38	65	85	93	93	90

Porównanie danych wyjściowych (tablica po lewej)
z danymi uzyskanymi przez zastosowanie
odwrotnej transformacji DCT do skwantyzowanych
amplitud (tablica po prawej)



Wizualne porównanie danych wyjściowych („kafelek” po lewej) z danymi uzyskanymi przez zastosowanie odwrotnej transformacji DCT do skwantyzowanych amplitud („kafelek” po prawej)



Ziarnista struktura map bitowych bywa przykra; najczęściej można natrafić na problemy z tym związane podczas skalowania map bitowych i przy obróbce grafiki zawierającej elementy rastra

