Tömörítési eljárások

jpeg, gif, GSM, compress, pkzip, (arc, rar ...) redundáns adatok: tömörítés sikeres!

Veszteség nélküli és veszteséges eljárások

Futási hossz:
Pl. a 127 alatti karakter: darabszám a következő ismétlődő karakterre
128 feletti karakter: x-128 darab különböző karakter
Nem mindig hatásos: pl. egy ABABAB...

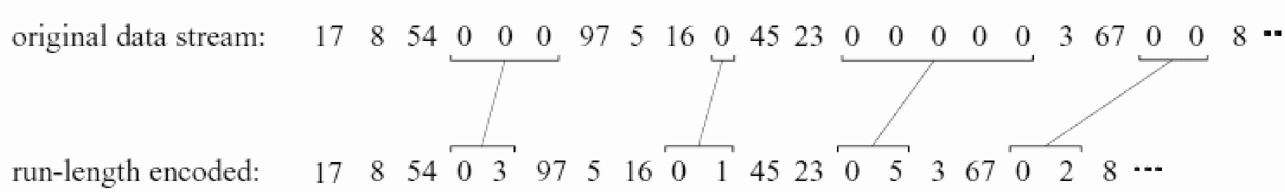


FIGURE 27-1
Example of run-length encoding. Each run of zeros is replaced by two characters in the compressed file: a zero to indicate that compression is occurring, followed by the number of zeros in the run.

Delta kódolás:
csak a különbség: kevesebb bit!

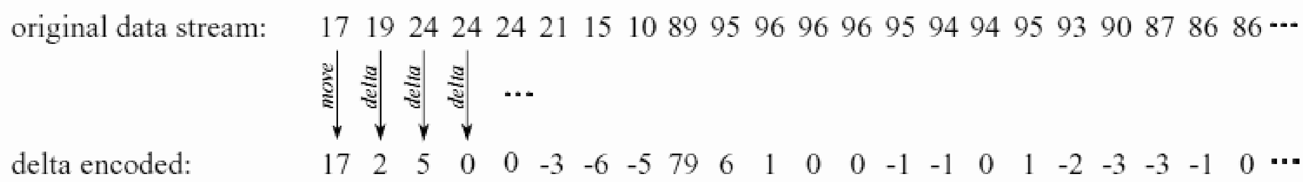
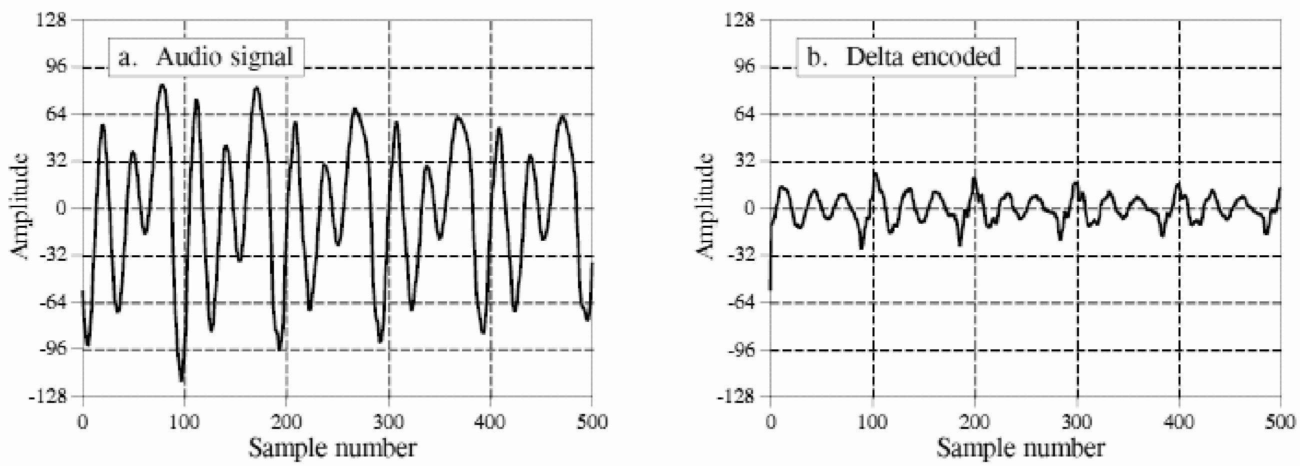


FIGURE 27-4
Example of delta encoding. The first value in the encoded file is the same as the first value in the original file. Thereafter, each sample in the encoded file is the difference between the current and last sample in the original file.

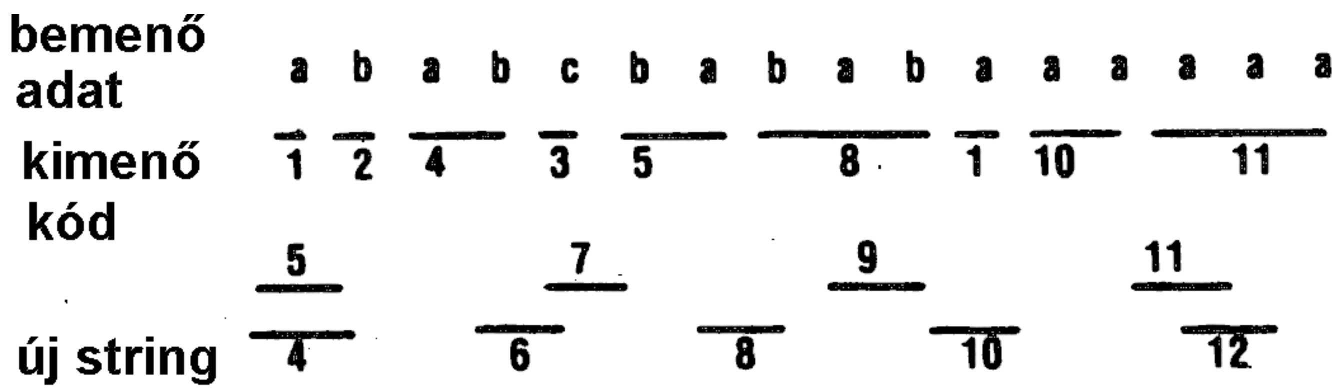


Lineáris Prediktív Kódolás (LPC)

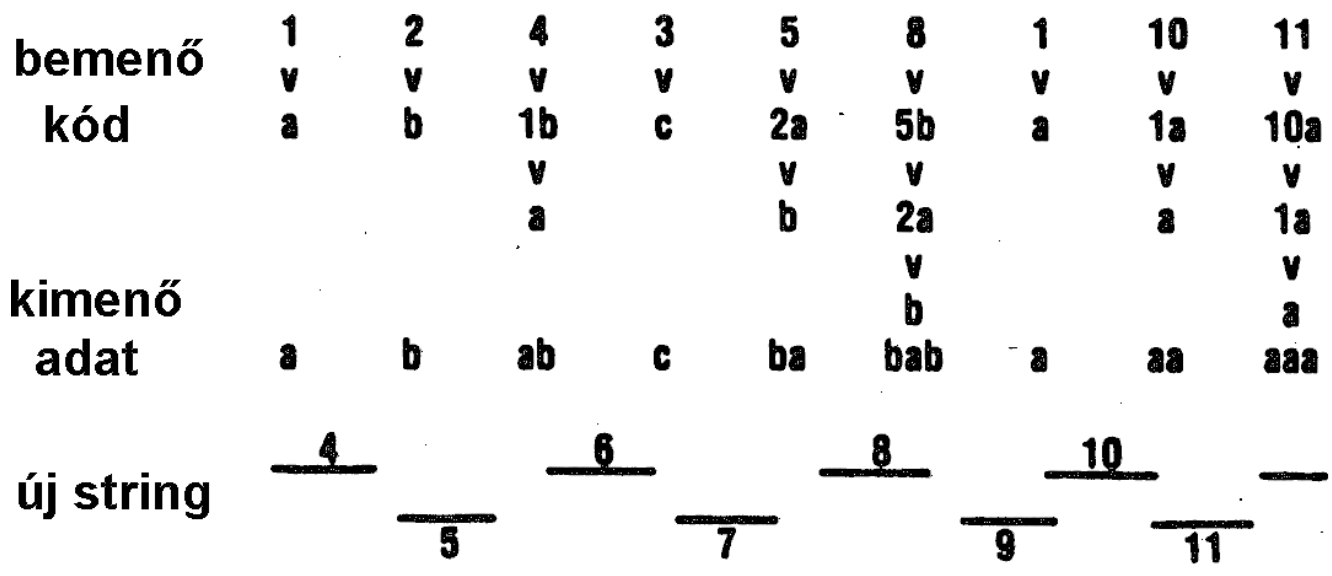
Lempel-Ziv-Welch eljárás alapötlet: szótár létrehozása

stringtábla		szótár	
a	1	a	1
b	2	b	2
c	3	c	3
ab	4	1b	4
ba	5	2a	5
abc	6-	4c	6
cb	7	3b	7
bab	8	5b	8
baba	9	8a	9
aa	10	1a	10
aa	11	10a	11
aaaa	12	11a	12

- miden bejövő adat: új szó a maximális egyezésű szótárelemmel
- tömörített kódot a maximális egyezésű szótárelem
- szótár mérete tipikusan 2^{12}-2^{16}
- ha szótár megtelik $\longrightarrow$ letörlik (csak az induló szótár marad meg ASCII!)



- visszaállítás/dekódolás: szótár újra építése
- adatfolyamban!!
- minden kód rekurzívan helyettesítődik a prefix kódjával+ a követő karakterrel
- HW megvalósítás



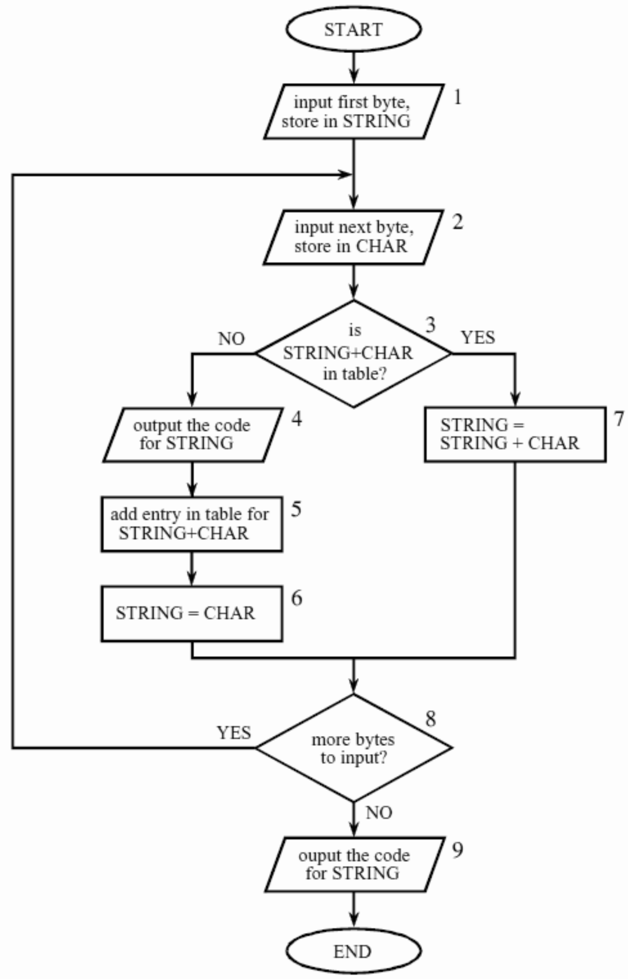


FIGURE 27-7
LZW compression flowchart. The variable, *CHAR*, is a single byte. The variable, *STRING*, is a variable length sequence of bytes. Data are read from the input file (box 1 & 2) as single bytes, and written to the compressed file (box 4) as 12 bit codes. Table 27-3 shows an example of this algorithm.

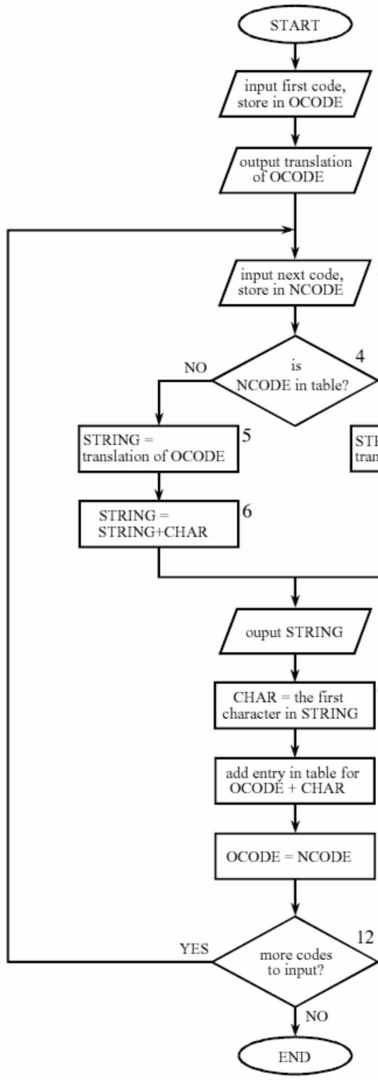
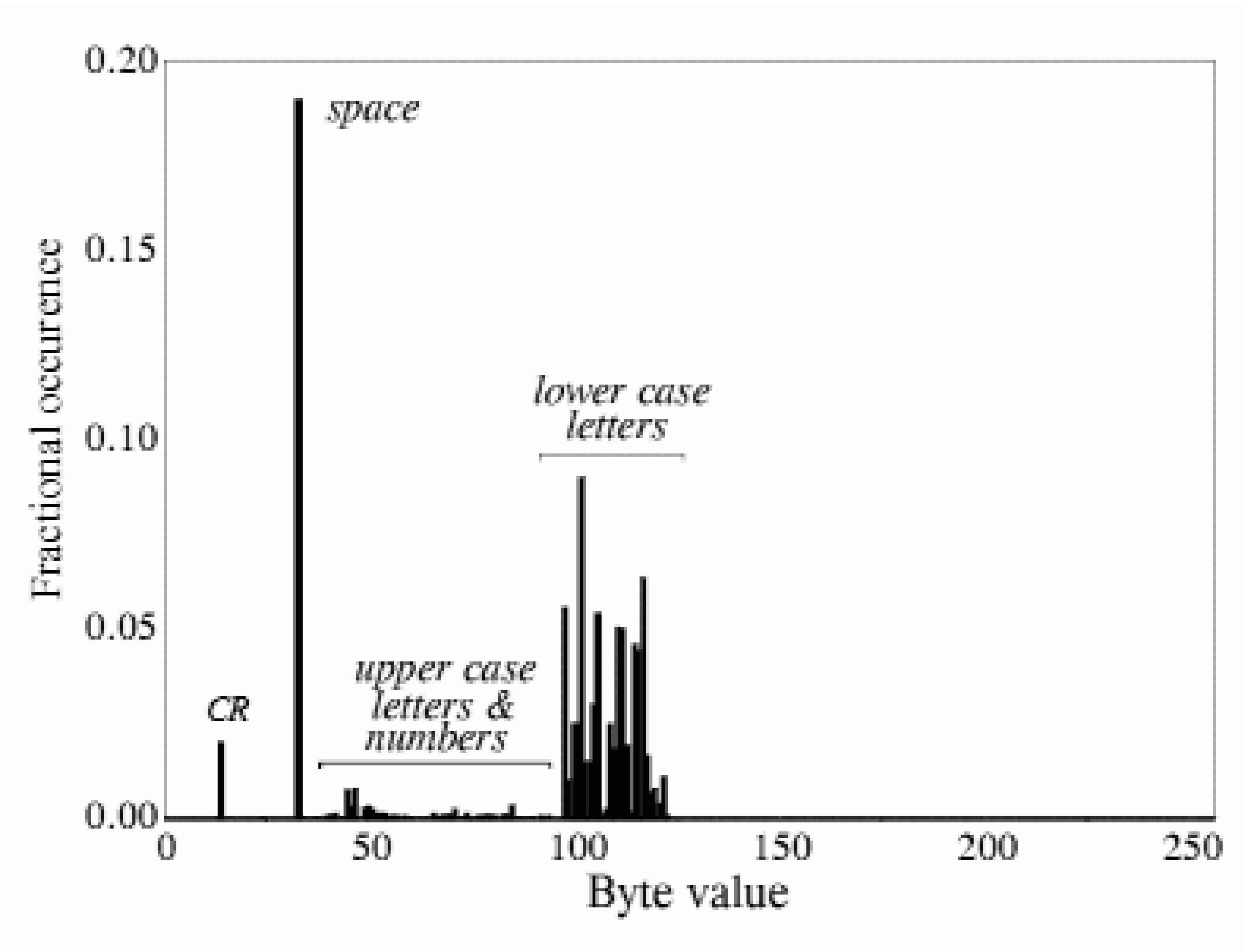


FIGURE 27-8
LZW uncompression flowchart. The variables, *OCODE* and *NCODE* (oldcode and newcode), hold the 12 bit codes from the compressed file, *CHAR* holds a single byte, *STRING* holds a string of bytes.

Huffman-kódolás:
Bejövő jelek eloszlása előzetesen pontosan ismert



sorbarendezés előfordulásuk valószínűségében
gyakoriak - kevés bit
ritkák - sok bit
rekurzívan a két legkisebb valószínűségű jel helyett új jelet vezet be, a két jel

valószínűségének együttes valószínűségével

		lépések			
Jel	**Valószínűség**	1	2	3	4
a_2	0.4	0.4	0.4	0.4	0.6
a_6	0.3	0.3	0.3	0.3	0.4
a_1	0.1	0.1	0.2	0.3	
a_4	0.1	0.1	0.1		
a_3	0.06	0.1			
a_5	0.04				

Kódolás: pl. nagyobb valószínűségű jel 0, a kisebb 1
Visszalépünk egyet a rendezésben, s.í.t.
Prefix kódolás

		lépések			
Jel	**Valószínűség**	1	2	3	4
a_2	0.4	0.4	0.4	0.4	0.6
a_6	0.3	0.3	0.3	0.3	0.4
a_1	0.1	0.1	0.2	0.3	
a_4	0.1	0.1	0.1		
a_3	0.06	0.1			
a_5	0.04				

Dekódolás: táblázattal
Egyértelmű kódhatárok!

			lépések							
Jel	**Val.**	**Kód**	1		2		3		4	
a_2	0.4	1	0.4	1	0.4	1	0.4	1	0.6	0
a_6	0.3	00	0.3	00	0.3	00	0.3	00	0.4	1
a_1	0.1	011	0.1	011	0.2	010	0.3	01		
a_4	0.1	0100	0.1	0100	0.1	011				
a_3	0.06	01010	0.1	0101						
a_5	0.04	01011								

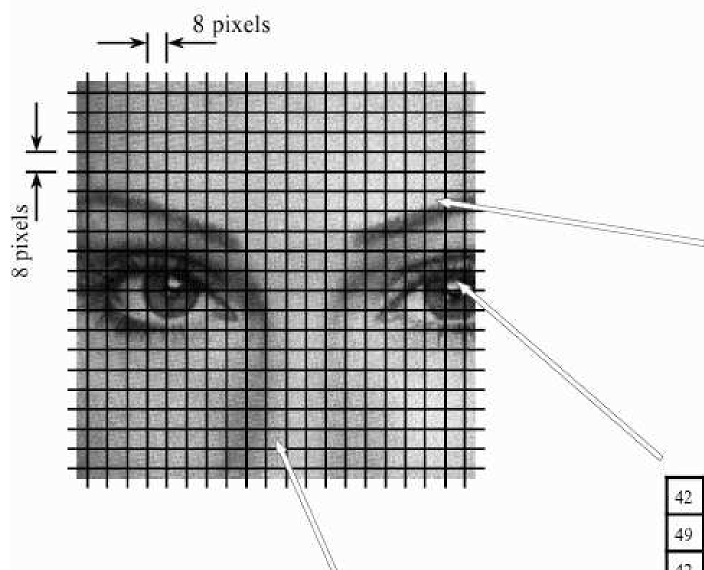
Kérdés: Mekkora a példában szereplő karaktersorozat és annak Huffman kódolásának Shannon-entrópiája?

A statikus kódtábla felépítéséhez ismerni kell a jelet!

Aritmetikai kódolás

JPEG kódolás

Veszteséges kódolás (l. pl. wavelet) - hosszú kutatómunka, több ajánlás
JPEG Baseline coding
Kép felosztása 8×8-as területekre (lokálisan adaptív)



231	224	224	217	217	203	189	196
210	217	203	189	203	224	217	224
196	217	210	224	203	203	196	189
210	203	196	203	182	203	182	189
203	224	203	217	196	175	154	140
182	189	168	161	154	126	119	112
175	154	126	105	140	105	119	84
154	98	105	98	105	63	112	84

42	28	35	28	42	49	35	42
49	49	35	28	35	35	35	42
42	21	21	28	42	35	42	28
21	35	35	42	42	28	28	14
56	70	77	84	91	28	28	21
70	126	133	147	161	91	35	14
126	203	189	182	175	175	35	21
49	189	245	210	182	84	21	35

154	154	175	182	189	168	217	175
154	147	168	154	168	168	196	175
175	154	203	175	189	182	196	182
175	168	168	168	140	175	168	203
133	168	154	196	175	189	203	154
168	161	161	168	154	154	189	189
147	161	175	182	189	175	217	175
175	175	203	175	189	175	175	182

FIGURE 27-9
JPEG image division. JPEG transform compression starts by breaking the image into 8×8 groups, each containing 64 pixels. Three of these 8×8 groups are enlarged in this figure, showing the values of the individual pixels, a single byte value between 0 and 255.

Kódolás blokkonként: Karhunen-Loeve (főkomponens) transzformáció lenne a legjobb, de FFT is jó, és egyszerűbb:
Ne legyen komplex kimenet: diszkrét cosinus transzformációt (DCT) alkalmazunk

DCT: mintasorrend $1, 2, 3, 4, 5, 6, 7, 8, 7, 6, 5, 4, 3, 2, 1$+FFT $\longrightarrow$ szimmetria miatt valós lesz! Azaz 8×8 valós értékből 8×8 valós értéket készít!

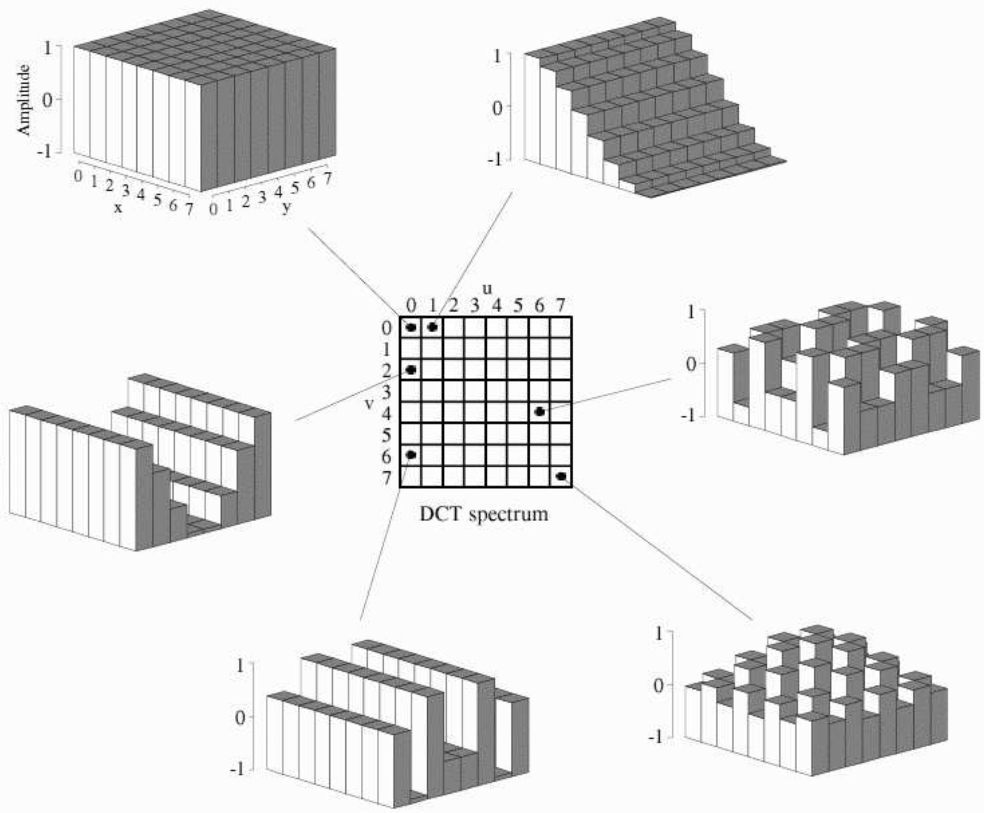


FIGURE 27-10
The DCT basis functions. The DCT spectrum consists of an 8×8 array, with each element in the array being an amplitude of one of the 64 basis functions. Six of these basis functions are shown here, referenced to where the corresponding amplitude resides.

A DCT értékeket kvantáljuk (max. 11 bit!):

Original Group

a. Eyebrow

231	224	224	217	217	203	189	196
210	217	203	189	203	224	217	224
196	217	210	224	203	203	196	189
210	203	196	203	182	203	182	189
203	224	203	217	196	175	154	140
182	189	168	161	154	126	119	112
175	154	126	105	140	105	119	84
154	98	105	98	105	63	112	84

b. Eye

42	28	35	28	42	49	35	42
49	49	35	28	35	35	35	42
42	21	21	28	42	35	42	28
21	35	35	42	42	28	28	14
56	70	77	84	91	28	28	21
70	126	133	147	161	91	35	14
126	203	189	182	175	175	35	21
49	189	245	210	182	84	21	35

c. Nose

154	154	175	182	189	168	217	175
154	147	168	154	168	168	196	175
175	154	203	175	189	182	196	182
175	168	168	168	140	175	168	203
133	168	154	196	175	189	203	154
168	161	161	168	154	154	189	189
147	161	175	182	189	175	217	175
175	175	203	175	189	175	175	182

DCT Spectrum

d. Eyebrow spectrum

174	19	0	3	1	0	-3	1
52	-13	-3	-4	-4	-4	5	-8
-18	-4	8	3	3	2	0	9
5	12	-4	0	0	-5	-1	0
1	2	-2	-1	4	4	2	0
-1	2	1	3	0	0	1	1
-2	5	-5	-5	3	2	-1	-1
3	5	-7	0	0	0	-4	0

e. Eye spectrum

70	24	-28	-4	-2	-10	-1	0
-53	-35	43	13	7	13	1	3
23	9	-10	-8	-7	-6	5	-3
6	2	-2	8	2	-1	0	-1
-10	-2	-1	-12	2	1	-1	4
3	0	0	11	-4	-1	5	6
-3	-5	-5	-4	3	2	-3	5
3	0	4	5	1	2	1	0

f. Nose spectrum

174	-11	-2	-3	-3	6	-3	4
-2	-3	1	2	0	3	1	2
3	0	-4	0	0	0	-1	9
-4	-6	-2	1	-1	4	-10	-3
1	2	-2	0	0	-2	0	-5
3	-1	3	-2	2	1	1	0
3	5	2	-2	3	0	4	3
4	-3	-13	3	-4	3	-5	3

Quantization Error

g. Using 10 bits

0	0	0	0	-1	0	0	0
-1	0	0	0	0	0	0	-1
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	1	0	0	0	-1	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

h. Using 8 bits

0	-3	-1	-1	1	0	0	-1
1	0	-1	-1	0	0	0	-1
-1	-2	1	0	-2	0	-2	-2
-1	-2	-1	2	0	2	0	1
0	-2	1	0	0	1	0	0
0	-4	-1	0	1	0	0	0
0	-2	0	1	-1	-1	1	-1
-1	-3	1	1	1	-3	-2	-1

i. Using 5 bits

-13	-7	1	4	0	0	10	-2
-22	6	-13	5	-5	2	-2	-13
-9	-15	0	-17	-8	8	12	25
-9	16	1	9	1	-5	-5	13
-20	-3	-13	-16	-19	-1	-4	-22
-11	6	-8	16	-9	-3	-7	6
-14	10	-9	4	-15	3	3	-4
-13	19	12	9	18	5	-5	10

kvantálási tábla:

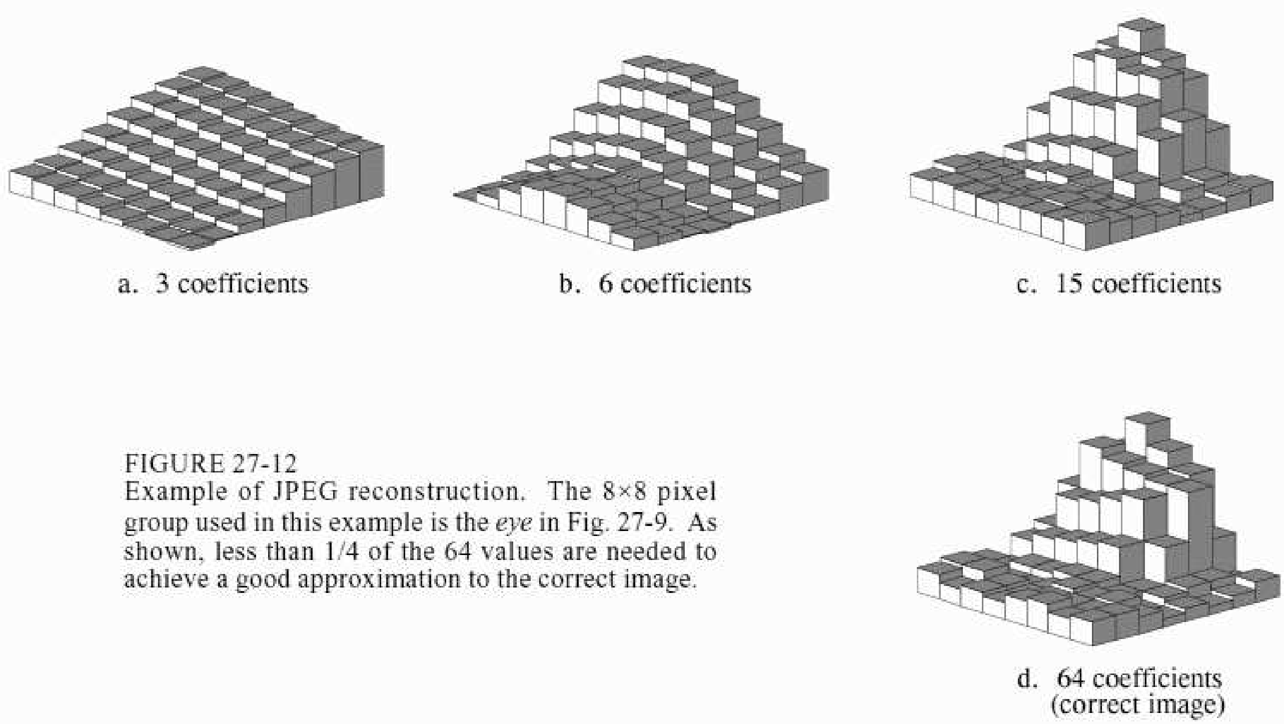


FIGURE 27-12
Example of JPEG reconstruction. The 8×8 pixel group used in this example is the *eye* in Fig. 27-9. As shown, less than 1/4 of the 64 values are needed to achieve a good approximation to the correct image.

a. Low compression

1	1	1	1	1	2	2	4
1	1	1	1	1	2	2	4
1	1	1	1	2	2	2	4
1	1	1	1	2	2	4	8
1	1	2	2	2	2	4	8
2	2	2	2	2	4	8	8
2	2	2	4	4	8	8	16
4	4	4	4	8	8	16	16

b. High compression

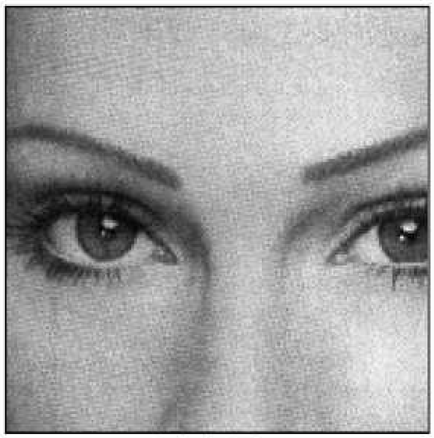
1	2	4	8	16	32	64	128
2	4	4	8	16	32	64	128
4	4	8	16	32	64	128	128
8	8	16	32	64	128	128	256
16	16	32	64	128	128	256	256
32	32	64	128	128	256	256	256
64	64	128	128	256	256	256	256
128	128	128	256	256	256	256	256

Lineáris sorozattá alakítás:

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

egymás mellé kerülnek, a homogén, egyszínű felületek:
AC komponensek futási hossz kóddal tömörítve

DC komponensek: delta kóddal tömörítve (lassan váltooznak)



a. Original image

b. With 10:1 compression

FIGURE 27-15
Example of JPEG distortion. Figure (a) shows the original image, while (b) and (c) shows restored images using compression ratios of 10:1 and 45:1, respectively. The high compression ratio used in (c) results in each 8×8 pixel group being represented by less than 12 bits.

c. With 45:1 compression

MPEG: mozgó JPEG