

Kosinová transformace 36ACS

10. listopadu 2006

Martin “BruXy” Bruchanov
bruxy@regnet.cz

Úkolem *transformačního kódování* je převést hodnoty *vzájemně závislých* vzorků (podle toho jakou tvoří prostorovou mozaiku) na jiné vzorky v sousedství *na sebe nezávislé*, jejichž hodnoty by byly soustředěny do *menší* rozlohy matice, než u netransformovaných vzorků.

- ▶ Ukládání obrazu pomocí JPEG
- ▶ Motion-JPEG
- ▶ Animace ve formátu MPEG a variantách MPEG-2, MPEG-4
- ▶ Modifikovaná DCT v zvukových formátech MP3, AC-3, Ogg Vorbis a AAC
- ▶ Využití MPEG-2 pro DVB-T a MPEG-4 pro DVB-S

Diskrétní kosinová transformace (DCT) je lineární funkce $F : R^N \rightarrow R^N$, kde R jsou reálná čísla a N rozměry matice $N \times N$. Reálná čísla $x_0 \dots x_{N-1}$ jsou transformována na reálná čísla $X_0 \dots X_{N-1}$ pomocí následujících funkcí:

- ▶ DCT-I:

$$X_k = \frac{1}{2}(x_0 + (-1)^k x_{N-1}) + \sum_{n=1}^{N-2} x_n \cos \left[\frac{\pi}{N-1} nk \right]$$

- ▶ DCT-II: (nejčastěji používaná)

$$X_k = \sum_{n=0}^{N-1} x_n \cos \left[\frac{\pi}{N} \left(n + \frac{1}{2} \right) k \right]$$

- ▶ DCT-III: (inverzní DCT)

$$X_k = \frac{1}{2}x_0 + \sum_{n=1}^{N-1} x_n \cos \left[\frac{\pi}{N} n \left(k + \frac{1}{2} \right) \right]$$

Formát JPEG pro ukládání obrazu

- ▶ velice rozšířený formát *Joint Photographic Experts Group*;
- ▶ standard ISO a ITU, uvedený v roce 1992 (později vylepšený v 1997);
- ▶ používá ztrátovou kompresi za pomoci DCT;
- ▶ snížení kvality na 75 % je většinou ještě; nepozorovatelné a přitom kompresní poměr může být 20 : 1 až 25 : 1;
- ▶ je vhodný pro ukládání fotografií a „přírodních“ obrazů, pro počítačovou grafiku se nehodí;

Velikost souboru v závislosti na stupni JPEG komprese

Originál snímku v rozlišení 256×192 .



Bez komprese
147 456 B

Velikost souboru v závislosti na stupni JPEG komprese



Kvalita 75%
15 970 B

Velikost souboru v závislosti na stupni JPEG komprese



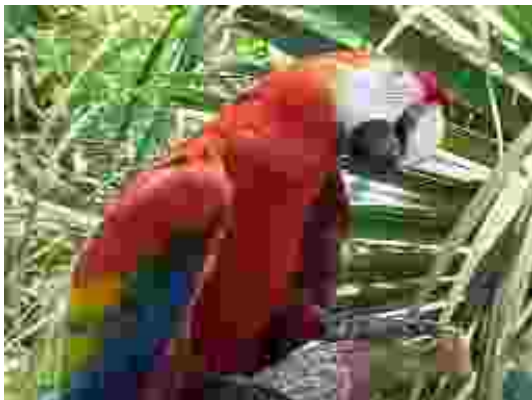
Kvalita 40%
9 564 B

Velikost souboru v závislosti na stupni JPEG komprese



Kvalita 10%
4 277 B

Velikost souboru v závislosti na stupni JPEG komprese



Kvalita 3%
2474 B

Kdy se JPEGu vyhnout velkým obloukem I.

JPEG je vhodný pro ukládání fotografií a „přírodních“ obrazů, pro počítačovou grafiku (grafy, schémata, ...) se nehodí.

Příklad: 16 barev, rozlišení 256×192



Bez komprese, 24 654 B

Kdy se JPEGu vyhnout velkým obloukem II.



Kvalita 75 %
7 300 B

Kdy se JPEGu vyhnout velkým obloukem II. – lupa

Už při kvalitě 75 % je na ostrých barevných přechodech patrné zkreslení.



Kdy se JPEGu vyhnout velkým obloukem III.



Kvalita 50 %
5 298 B

Kdy se JPEGu vyhnout velkým obloukem IV.



Kvalita 25 %
3 766 B

Kdy se JPEGu vyhnout velkým obloukem V.



Kvalita 1 %
1 378 B

Lépe použít formáty s bezztrátovou kompresí

Formát	Kvalita	Soubor
Windows Bitmap	100 %	24 654 B
JPEG	100 %	17 740 B
JPEG	75 %	7 300 B
JPEG	50 %	5 298 B
TIFF, komprese PackBits	100 %	4 352 B
Windows Bitmap RLE	100 %	3 984 B
TIFF, komprese LZW	100 %	3 850 B
JPEG	25 %	3 766 B
GIF	100 %	1 569 B
JPEG	1 %	1 378 B
Portable Network Graphics	100 %	1 111 B

Příznivé velikosti souboru možno dosáhnout vhodným kompromisem mezi rozlišením a počtem barev.

Postup kódování JPEG souboru

- ▶ lidské oko vnímá jas a barvy s různou citlivostí;
- ▶ neukládá obrazová data v barevném kódování *RGB*, ale používá *YCrCb* a poměr jednotlivých složek $4:2:0$;
- ▶ rozdělí na čtvercové bloky o rozměrech 8×8 pixelů;
- ▶ těchto 64 bodů je pak převedeno z prostorové oblasti (x, y) na frekvenční oblast (i, j) za pomoci DCT;
- ▶ výslednou kvalitu a vnímání obrazu nejvíce ovlivní kvantizace jednotlivých členů pomocí kvantizační tabulky;
- ▶ výsledná matice je linearizována na posloupnost, která obsahuje velké množství nul;
- ▶ posloupnost se rozdělí do kategorií a nadbytečnost nul se omezí RLE kódováním;
- ▶ nakonec se vše zakóduje pomocí Huffmanovy transformace.

Převod RGB→YCrCb a dekompozice na bloky



$$\begin{aligned} Y &= 0,30R + 0,59G + 0,11B \\ Cr &= -0,17R - 0,33G - 0,5B \\ Cb &= +0,5R - 0,42G - 0,08B \end{aligned}$$

Koeficienty pro náš makroblok

Osmibitové hodnoty 0–255 vyjadřující jas pixelu jsou před provedením DCT převedeny na rozsah –128–127.

107	103	98	84	45	8	4	1
109	93	114	97	90	66	11	–3
118	90	110	96	93	89	72	33
106	97	99	100	96	97	92	79
98	108	106	97	93	106	94	94
103	103	98	105	98	106	87	109
99	107	109	98	103	93	98	97
84	80	89	103	100	105	92	81

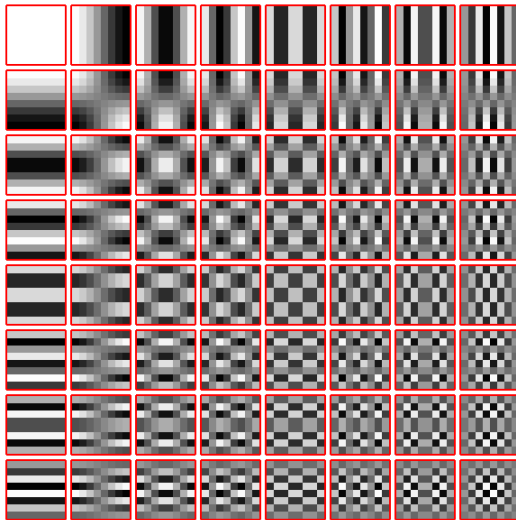
Definice dvourozměrné DCT v JPEGu

$$\text{DCT}(i, j) = \frac{1}{4} C(i) \cdot C(j) \sum_{x=0}^7 \sum_{y=0}^7 p(x, y) \cdot \cos \left[\frac{(2x+1)i\pi}{16} \right] \cdot \cos \left[\frac{(2y+1)j\pi}{16} \right],$$

$$\text{kde } C(a) = \begin{cases} \frac{1}{\sqrt{2}} & \text{pro } a = 0; \\ 1 & \text{v ostatních případech.} \end{cases}$$

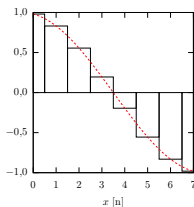
Na první pozici $i = 0, j = 0$ se pak ukládá tzv. *DC člen*, ten představuje *stejnoseměrnou složku* harmonického rozkladu a obsahuje průměr všech 64 hodnot z bloku. Na ostatních pozicích jsou pak *AC členy* neboli střídavá složka a jejich hodnotu ovlivňuje odchylka jednotlivých bodů od DC členu. Zjednodušeně řečeno, DCT se snaží blok 8×8 „napasovat“ jako lineární kombinaci obrazců daných předchozím vzorcem.

Obrazový blok je lineární kombinací obrazců DCT

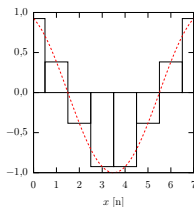


Výpočet jednorozměrné DCT I.

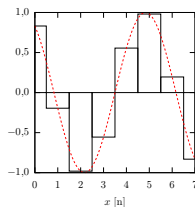
$i = 1$



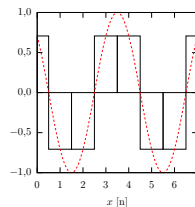
$i = 2$



$i = 3$

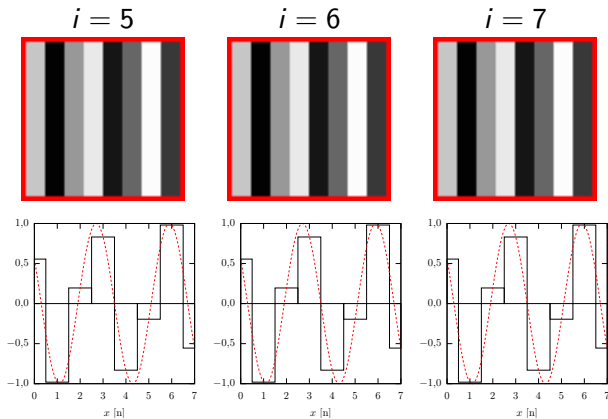


$i = 4$



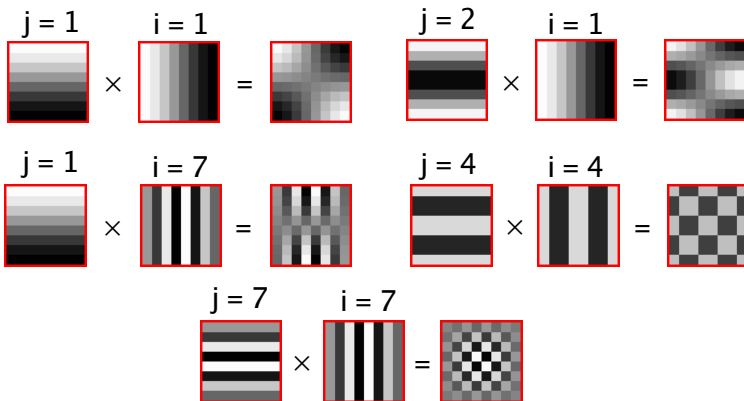
$$\cos \left[\frac{(2x + 1)i\pi}{16} \right]$$

Výpočet jednorozměrné DCT II.



$$\cos\left[\frac{(2x+1)i\pi}{16}\right]$$

Výpočet dvourozměrné DCT

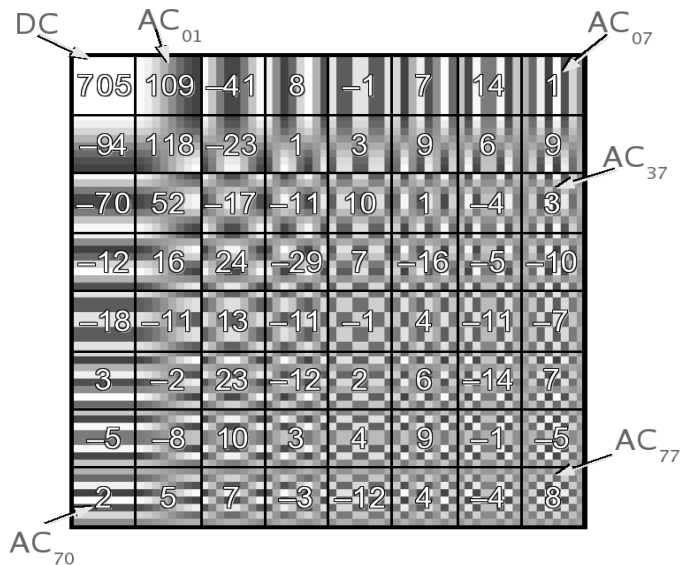


$$\frac{1}{4} C(i) \cdot C(j) \sum_{x=0}^7 \sum_{y=0}^7 p(x, y) \cdot \cos \left[\frac{(2x+1)i\pi}{16} \right] \cdot \cos \left[\frac{(2y+1)j\pi}{16} \right]$$

Výsledek DCT pro náš blok I.

$$\text{DCT}_Y = \begin{pmatrix} 705 & 109 & -41 & 8 & -1 & 7 & 14 & 1 \\ -94 & 118 & -23 & 1 & 3 & 9 & 6 & 9 \\ -70 & 52 & -17 & -11 & 10 & 1 & -4 & 3 \\ -12 & 16 & 24 & -29 & 7 & -16 & -5 & -10 \\ -18 & -11 & 13 & -11 & -1 & 4 & -11 & -7 \\ 3 & -2 & 23 & -12 & 2 & 6 & -14 & 7 \\ -5 & -8 & 10 & 3 & 4 & 9 & -1 & -5 \\ 2 & 5 & 7 & -3 & -12 & 4 & -4 & 8 \end{pmatrix}$$

Výsledek DCT pro náš blok II.

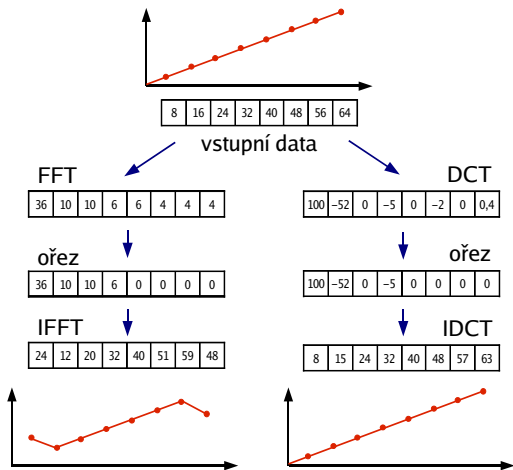


Obrazový blok je lineární kombinací obrazců DCT II.

Obrázek uložený s kvalitou 3%, rozdělení do bloků a stupeň ztrátové komprese originál značně znehodnotily.



Proč použít DCT a ne třeba FFT



DCT je podobná FFT, ale umí mnohem lépe aproximovat lineární signály za pomoci méně koeficientů.

Kvantizační tabulka pro luminační složku

$$Y_Q = \begin{pmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{pmatrix}$$

Koeficienty kvantizační tabulky, zde pro kvalitu 50 %, mohou být násobeny konstantou α , která určuje míru ztrátovosti.

Kvantizační matice navržena na základě psychologických výzkumů dr. Loeschellera.

Kvalita Q :

$$\begin{aligned} 1 \leq Q \leq 50, & \quad \alpha = \frac{50}{Q}, \\ 51 \leq Q \leq 100, & \quad \alpha = 2 - \frac{Q}{50} \end{aligned}$$

Kvantizační tabulka pro chrominační složku

$$\mathbb{C}_Q = \begin{pmatrix} 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{pmatrix}$$

Kvantizační tabulka $\mathbb{C}_Q$ je vůči barvám mnohem agresivnější. V případě potřeby formát JPEG umožňuje definovat vlastní tabulky, které mohou být vůči danému typu obrazu šetrnější.

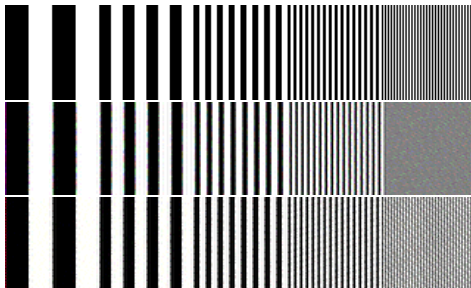
DCT informace po kvantizaci s 50 % kvalitou

$$\text{DCT}_{Y./Y_Q} = \begin{pmatrix} 705 & 109 & -41 & 8 & \dots \\ -94 & 118 & -23 & 1 & \dots \\ -70 & 52 & -17 & -11 & \dots \\ -12 & 16 & 24 & -29 & \dots \\ -18 & -11 & 13 & -11 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} ./ \begin{pmatrix} 16 & 11 & 10 & 16 & \dots \\ 12 & 12 & 14 & 19 & \dots \\ 14 & 13 & 16 & 24 & \dots \\ 14 & 17 & 22 & 29 & \dots \\ 18 & 22 & 37 & 56 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

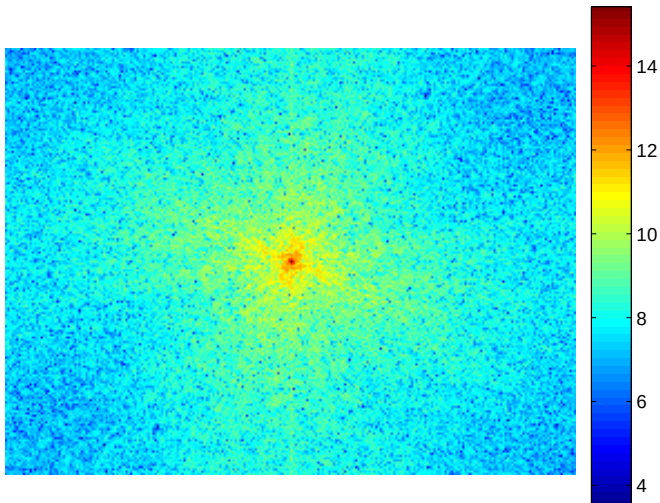
$$\text{DCT}_{Y./Q} = \begin{pmatrix} 44 & 10 & -4 & 1 & 0 & 0 & 0 & 0 & 0 \\ -8 & 10 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ -5 & 4 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Proč se při DCT zkreslí ostré barevné přechody

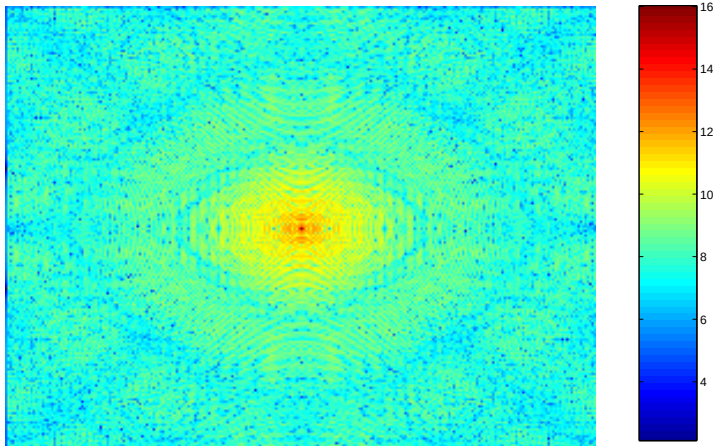
- ▶ Ostré barevné přechody, hrany a šum zastupují v obraze vysokofrekvenční složku.
- ▶ Pozvolné barevné přechody zastupují nízkofrekvenční složku.
- ▶ Oko a mozek si nízkofrekvenční informaci doplní bez problému, s vysokofrekvenční má problém.
- ▶ Kvantizační matice potlačují vysokofrekvenční složky!



FFT spektrum obrázku papouška

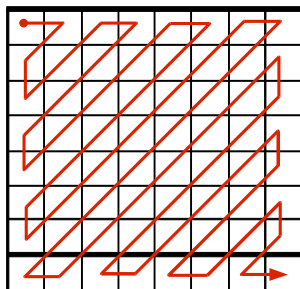


FFT spektrum obrázku usměvavého ksichtíku



Převedení matice na posloupnost

$$\begin{pmatrix} 44 & 10 & -4 & 1 & 0 & 0 & 0 & 0 \\ -8 & 10 & -2 & 0 & 0 & 0 & 0 & 0 \\ -5 & 4 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$



Díky „cik-cak“ přechtení se nenulové členy dostanou na první pozice a zbytek vyplní nepotřebné nuly:

44, 10, -8, -5, 10, -4, 1, -2, 4, -1, -1, 1, -1, 0, 0, 0, 0, 0, 1,
0, 0, 0, 0, 0, -1, 0, 0, ...

Rozdělení posloupnosti do kategorií

Posloupnost je potom rozčleněna do kategorií:

(počet předchozích nul / kategorie / vlastní hodnota).

Pro jasové AC členy, je potom 0 kategorie 0, pro ostatní celočíselné hodnoty číslo kategorie odpovídá bitové délce hodnoty. Pro nejčastěji zastoupené AC hodnoty $\{-1, +1\}$ je to 1, vyjádřeno 0 nebo 1, pro $\{-3, -2, +2, +3\}$ délka 2, což je vyjádřeno jako $\{00, 01, 10, 11\}$, pro $\{-7, \dots, -4, +4, \dots, +7\}$ délka 3, atd.

- ▶ velký počet nul omezí RLE kódování
- ▶ od určitého místa, kde jsou zastoupeny samé nuly se vloží EOB (end of block).
- ▶ DC členy, jasové i chrominační členy mají každý své kódování.

Výsledek rozdělení do kategorií

44, 10, -8, -5, 10, -4, 1, -2, 4, -1, -1, 1, -1, 0, 0,
0, 0, 0, 1, 0, 0, 0, 0, 0, -1, 0, 0, ...

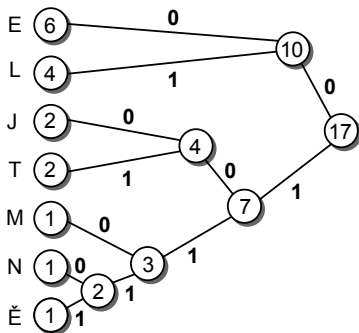
↓↓↓

44; (0/4/10), (0/4/-8), (0/3/-5), (0/4/10),
(0/3/-4), (0/1/1), (0/2/-2), (0/3/4), (0/1/-1),
(0/1/-1), (0/1/1), (0/1/-1), (5/1/1), (5/1/-1),
EOB

Kompresa pomocí Huffmanova kódování — úvod

Elementům souboru, které se opakují nejčastěji přiřazeny krátká kódová slova, ty které méně často delší.

JELEN JETELEM LETĚL (v 8 bitovém ASCII 136 bitů)



Znak	Kód
E	00
L	01
J	100
T	101
M	110
N	1110
Ě	1111

1000001001110 0100101111101 10000101000100110

Komprese pomocí Huffmanova kódování — výsledek

- ▶ $(0/4/10)$, $(0/4/-8)$, $(0/3/-5)$, $(0/4/10)$, $(0/3/-4)$,
 $(0/1/1)$, $(0/2/-2)$, $(0/3/4)$, $(0/1/-1)$, $(0/1/-1)$, $(0/1/1)$,
 $(0/1/-1)$, $(5/1/1)$, $(5/1/-1)$, **EOB**
- ▶ Výsledný kód závisí na počtu předcházejících nul a bitové délce.
- ▶ Př. $0/1$ (žádná nula/délka 1) má 00, $0/2$ 01, $0/3$ 100, $1/1$ (1 nula/délka 1) 100, $5/1$ 1111010, atd.
- ▶ Výsledek:
1011 1100 1011 0111 100 010 1011 1100 100 011 00 1 01
01 100 100 00 0 00 0 00 1 00 0 1111010 1 1111010 0 1010
- ▶ Původní blok 64 Bytů zmenšen na 9 Bytů.

- ▶ <http://wikipedia.org/> heslo *Discrete cosine transform*
- ▶ Vít, Vladimír Ing.: **Televizní technika – barevné přenosové soustavy**. BEN technická literatura, 1997.
- ▶ Norma *ITU T.81 JPEG compression*, ITU 1992
<http://www.w3.org/Graphics/JPEG/itu-t81.pdf>
- ▶ *Smith, Steven W.*: **The Scientists and Engineer's Guide to Digital Signal Processing**. California Technical Publishing, 1999.
www.dspguide.com