
JPEG1 standard

© 1997-2001 Josef Pelikán
KSVI MFF UK Praha

e-mail: Josef.Pelikan@mff.cuni.cz

WWW: <http://cgg.ms.mff.cuni.cz/~pepca/>

Cíle JPEG1 standardu

- ◆ **Joint Photographic Experts Group ('86-'90)**
 - multioborová komise ISO/CCITT
- ➔ **špičkový kompresní poměr při dobré kvalitě obrazu**
 - kvalita musí být nastavitelná v široké škále
- ➔ **univerzální použití na libovolné rastrové obrázky se spojitou škálou barevných odstínů**
 - bez ohledu na rozlišení, barevnost nebo statistické vlastnosti dat

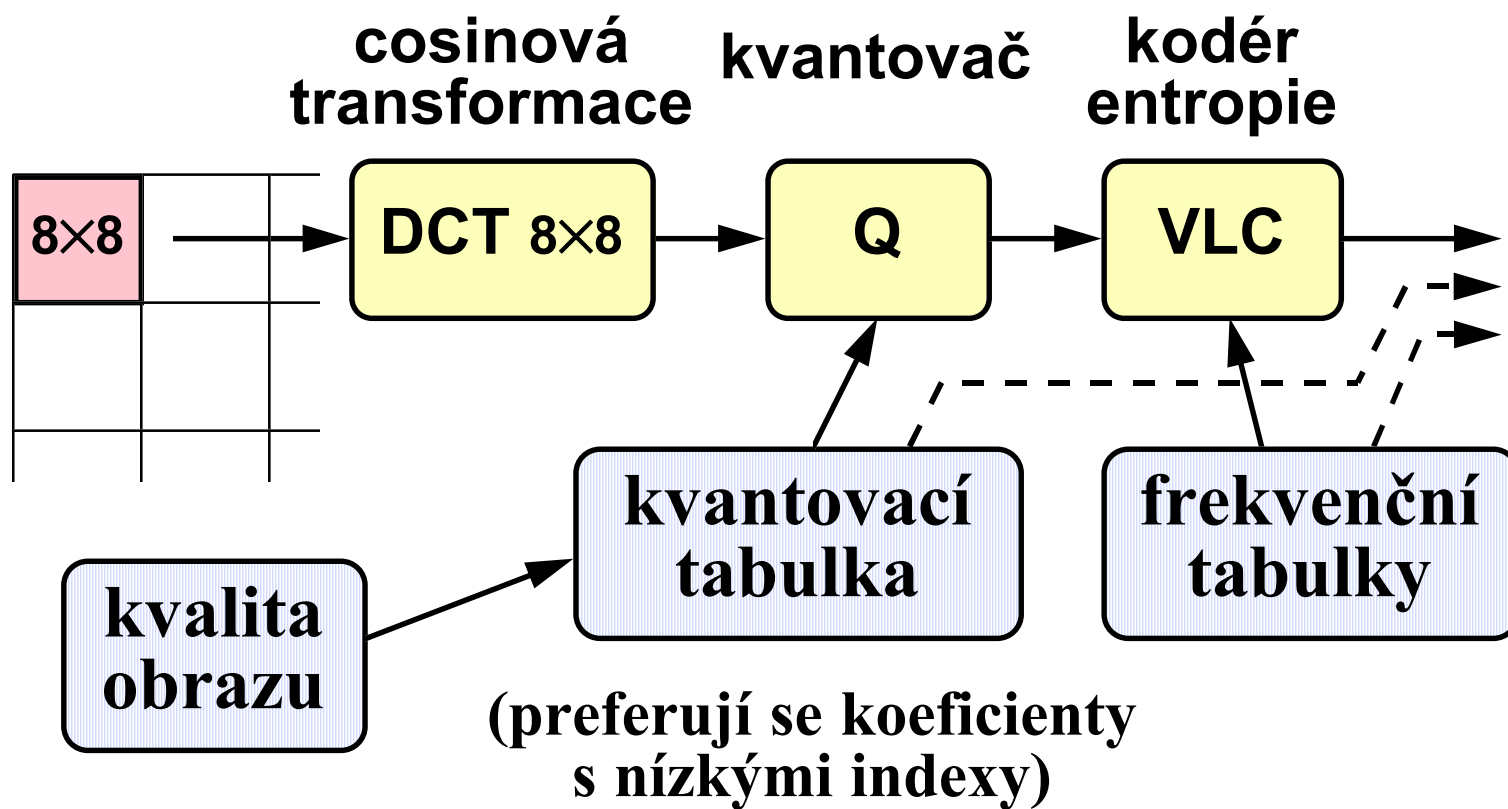
Cíle JPEG1 standardu

- ➔ **rozumná výpočetní složitost** pro SW i HW implementace
- ➔ **čtyři režimy kódování:**
 - **bezeztrátová komprese** - menší kompresní poměr (např. pro vědecká data)
 - **sekvenční kódování** - jediný průchod daty pomocí řádkového rozkladu shora dolů a zleva doprava
 - **progresivní kódování** - několik průchodů s postupným zlepšováním kvality obrazu
 - **hierarchické kódování** - několik různých rozlišení obrazu, která lze samostatně dekódovat

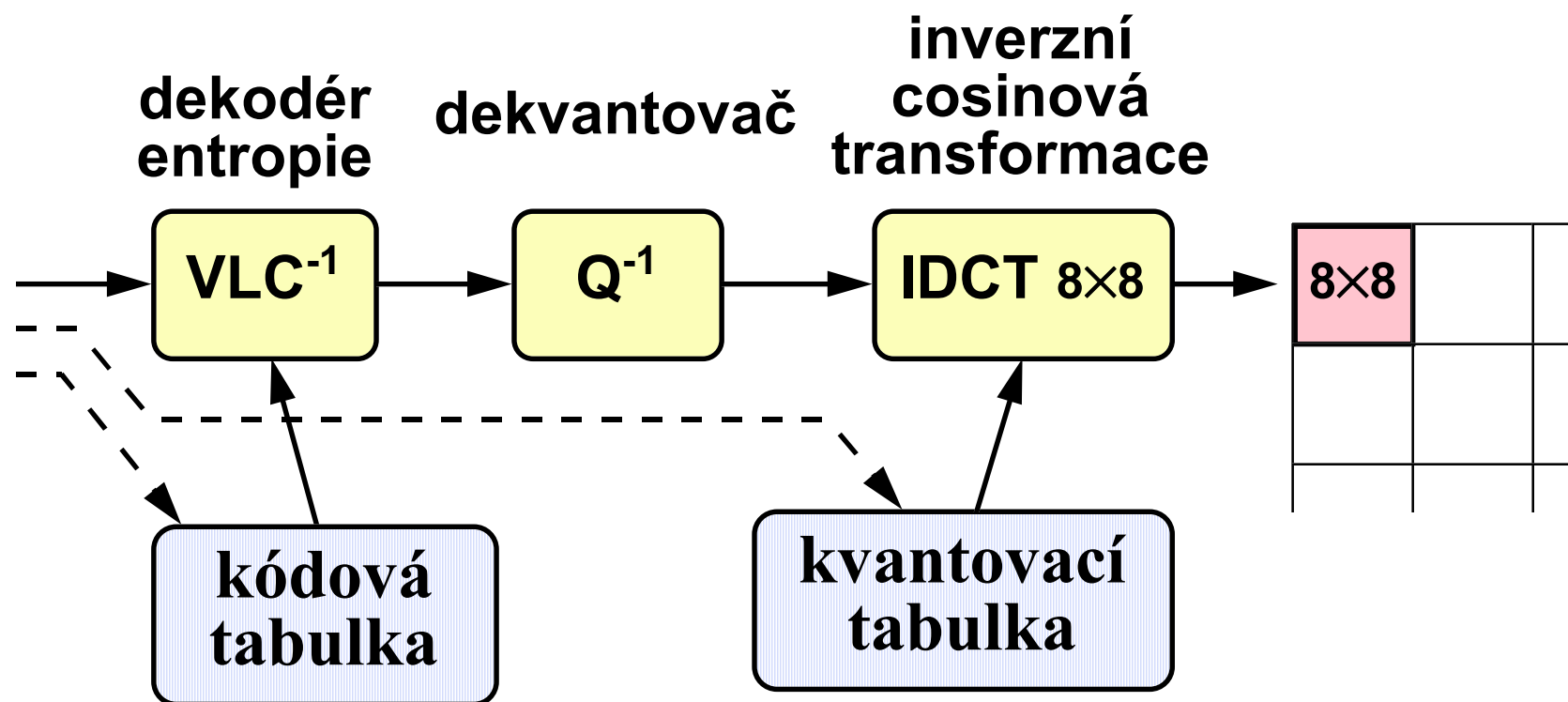
Výběr kompresních metod

- ◆ **široké výběrové řízení (1987-88)**
 - na začátku bylo zkoumáno 12 metod
 - zúžení na 3 nejlepší kandidáty
 - ⇒ výběr **blokové kvantizace** založené na DCT s blokem velikosti **8×8 pixelů**
- ➡ ke **kvantování koeficientů** se používají fyziologicky optimalizované tabulky
- ➡ pro konečné **kódování** slouží Huffmanův nebo aritmetický kód

Ztrátový kodér



Ztrátový dekodér



Diskrétní cosinová transformace

Před transformací se provádí **posun hodnot** všech pixelů z intervalu $[0, 2^p - 1]$ do $[-2^{p-1}, 2^{p-1} - 1]$.

p=8 nebo **p=12**

— 2D diskrétní cosinová transformace (DCT 8×8) —

$$F(u, v) = C_u C_v \sum_{x, y=0}^7 f(x, y) \cdot \cos \frac{\pi(2x+1)u}{16} \cos \frac{\pi(2y+1)v}{16}$$

$$C_u = \begin{cases} 1/2\sqrt{2} & \text{pro } u = 0 \\ 1/2 & \text{jinak} \end{cases}$$

Inverzní cosinová transformace

2D inverzní diskrétní cosinová transformace (IDCT 8×8)

$$f(\mathbf{x}, \mathbf{y}) = \sum_{u,v=0}^7 C_u C_v \cdot F(u, v) \cdot \cos \frac{\pi(2x+1)u}{16} \cos \frac{\pi(2y+1)v}{16}$$

$$C_u = \begin{cases} 1/2\sqrt{2} & \text{pro } u = 0 \\ 1/2 & \text{jinak} \end{cases}$$

Kvantování DCT koeficientů

- ◆ **základní princip ztrátové koprese JPEG**
 - nezachovává se původní (spočítaná) přesnost DCT koeficientů
- ➔ **přesnost kódování jednotlivých koeficientů závisí na jejich důležitosti**
 - koeficienty s malým významem pro vizuální kvalitu rekonstruovaného obrazu jsou přenášeny s menší přesností nebo zcela potlačeny
- ➔ **lineární kvantovače řízené tzv. kvantovacími tabulkami** (menší hodnota \Leftrightarrow větší důležitost)

Kvantování DCT koeficientů

Celočíselná kvantovací tabulka: $[\underline{Q(u,v)}]_{u,v=0}^7$

menší hodnota $Q(u,v) \Leftrightarrow$ větší důležitost $F(u,v)$

Kvantování:

$$F^Q(u,v) = \text{round}\left(\frac{F(u,v)}{Q(u,v)}\right)$$

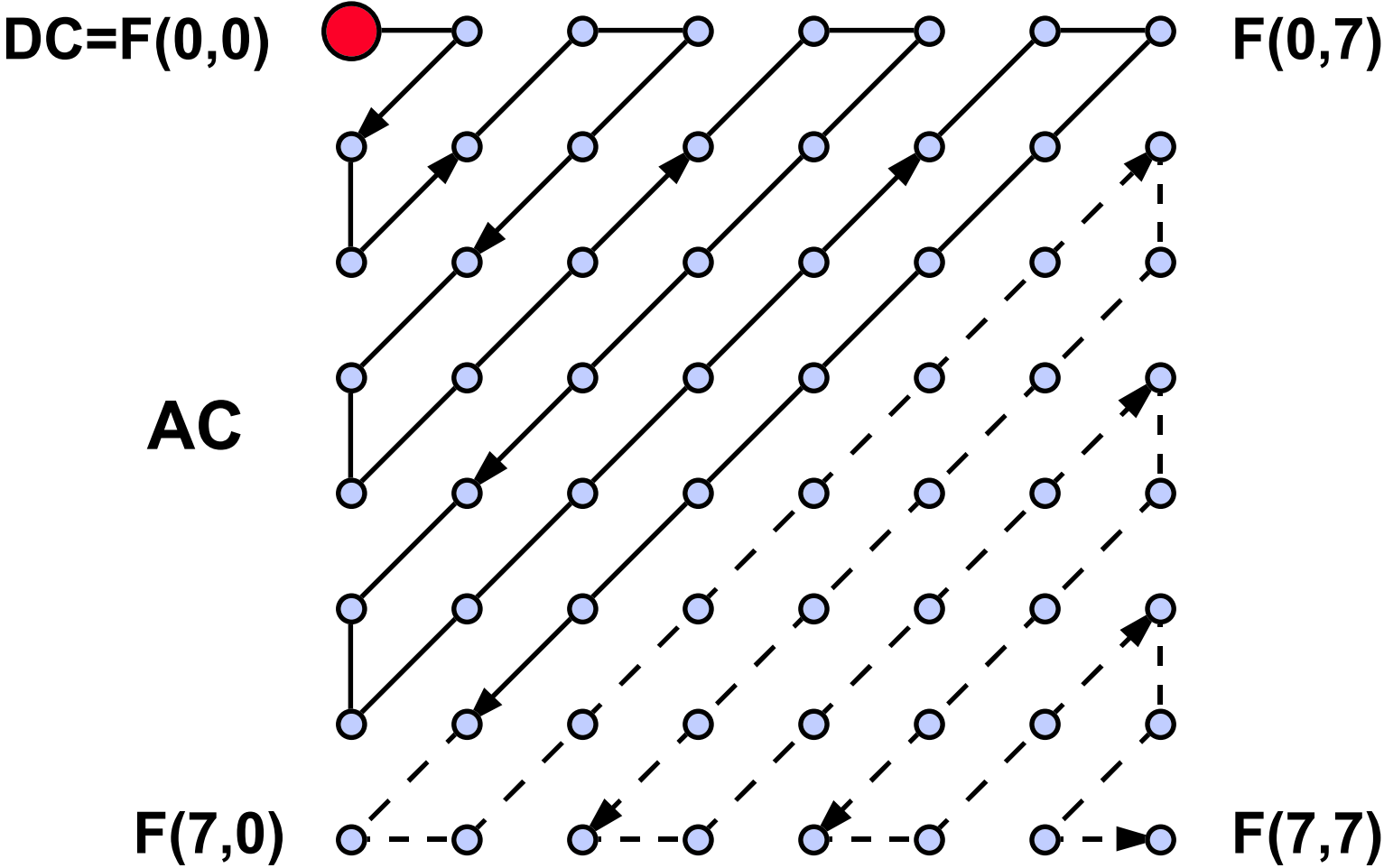
Dekvantování:
(rekonstrukce)

$$F^\bullet(u,v) = F^Q(u,v) \cdot Q(u,v)$$

Kódování kvantovaných koef.

- ◆ **stejnoseměrná složka** (DC, $F(\mathbf{0},\mathbf{0})$) je kódována odlišným způsobem
 - predikce z předchozího bloku, přenáší se jen rozdíl
- ◆ **střídavé složky** (AC, $F(\mathbf{u},\mathbf{v})$ pro $\mathbf{u}>\mathbf{0} \vee \mathbf{v}>\mathbf{0}$) jsou uspořádány do tzv. “cik-cak” posloupnosti
 - úsporné přeskokování nulových hodnot (RLE)
 - nenulové hodnoty se kódují pomocí **VLI** (celé číslo s proměnným počtem bitů)
- ➔ vše se nakonec komprimuje **kodečem entropie**
 - statický Huffmanův nebo dynamic. aritmetický kód

Cik-cak posloupnost



Mezikód

- ◆ **DC: přesnost (S .. počet bitů), hodnota (VLI)**
 - symbol S je kódován entropickým kódem
 - vlastní hodnota je uložena ve formátu **VLI** (není entropicky kódována)
- ◆ **AC: [počet nul, přesnost] (S), hodnota (VLI)**
 - symbol S obsahuje počet přeskočených nulových AC koeficientů a počet bitů kódovaného koeficientu
 - hodnota koeficientu je přenášena jako **VLI**
 - symbol S se může několikrát opakovat (pro víc nul)
 - speciální symbol **[0,0]** označuje konec bloku (EOB)

Kód VLI (“Variable Length Integer”)

Přesnost v bitech	Kódované hodnoty
1	-1, 1
2	-3 .. -2, 2 .. 3
3	-7 .. -4, 4 .. 7
4	-15 .. -8, 8 .. 15
5	-31 .. -16, 16 .. 31
6	-63 .. -32, 32 .. 63
7	-127 .. -64, 64 .. 127
8	-255 .. -128, 128 .. 255
9	-511 .. -256, 256 .. 511
10	-1023 .. -512, 512 .. 1023
11	-2047 .. -1024, 1024 .. 2047

Nejfrekventovanější S-symboly

Symbol	Huffman	Symbol	Huffman
[0,1]	00	[0,6]	1111000
[0,2]	01	[1,3]	1111001
[0,3]	100	[5,1]	1111010
[EOB]	1010	[6,1]	1111011
[0,4]	1011	[0,7]	11111000
[1,1]	1100	[2,2]	11111001
[0,5]	11010	[7,1]	11111010
[1,2]	11011	[1,4]	111110110
[2,1]	11100	[3,2]	111110111
[3,1]	111010	[8,1]	1111110000
[4,1]	111011	[9,1]	1111110001

Příklad

Originál
(512 bitů):

139	144	149	153	155	155	155	155
144	151	153	156	159	156	156	156
150	155	160	163	158	156	156	156
159	161	162	160	160	159	159	159
159	160	161	162	162	155	155	155
161	161	161	161	160	157	157	157
162	162	161	163	162	157	157	157
162	162	161	161	163	158	158	158

DCT
koeficienty:

235.6	-1.0	-12.1	-5.2	2.1	-1.7	-2.7	1.3
-22.6	-17.5	-6.2	-3.2	-2.9	-0.1	0.4	-1.2
-10.9	-9.3	-1.6	1.5	0.2	-0.9	-0.6	-0.1
-7.1	-1.9	0.2	1.5	0.9	-0.1	0.0	0.3
-0.6	-0.8	1.5	1.6	-0.1	-0.7	0.6	1.3
1.8	-0.2	1.6	-0.3	-0.8	1.5	1.0	-1.0
-1.3	-0.4	-0.3	-1.5	-0.5	1.7	1.1	-0.8
-2.6	1.6	-3.8	-1.8	1.9	1.2	-0.6	-0.4

Příklad

Mezikód: (2)(3) - (1,2)(-2) - (0,1)(-1) - (0,1)(-1) - (0,1)(-1) -
- (2,1)(-1) - (0,0)

Huffman: (011)(11) - (11011)(01) - (00)(0) - (00)(0) - (00)(0) -
- (11100)(0) - (1010)

Výsledek:
(31 bitů -
16.5 : 1)

011111101101000000001110001010

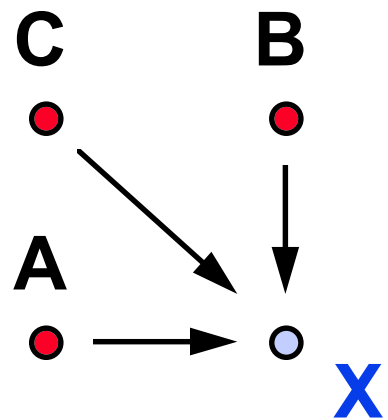
Chyba
rekonstrukce:

5	2	0	-1	-1	1	1	1
4	-1	-1	-2	-3	0	0	0
5	1	-3	-5	0	1	0	-1
1	0	-1	2	1	0	-2	-4
4	3	3	1	0	5	3	1
2	3	3	3	2	3	1	0
-2	-1	1	-1	0	4	2	1
-4	-3	0	0	-1	3	1	0

Bezeztrátová metoda JPEG

- ◆ založena na **jiném principu** (prediktivní kodér)
 - bezeztrátová varianta s DCT by se definovala velmi obtížně
- ◆ vhodná i pro **přesná data** (2-16 bitů/pixel)
- ➔ **1-2D prediktivní metoda**
 - výběr z 7 netriviálních prediktorů
- ➔ **chyby predikce** se reprezentují pomocí **VLI**
 - přesnost (v bitech) se kóduje opět Huffmanovým kódem

Prediktivní metoda



Prediktor

X

0

0

1

A

2

B

3

C

4

A + B - C

5

A + $\frac{1}{2}$ (B - C)

6

B + $\frac{1}{2}$ (A - C)

7

$\frac{1}{2}$ (A + B)

Vícesložkový obraz

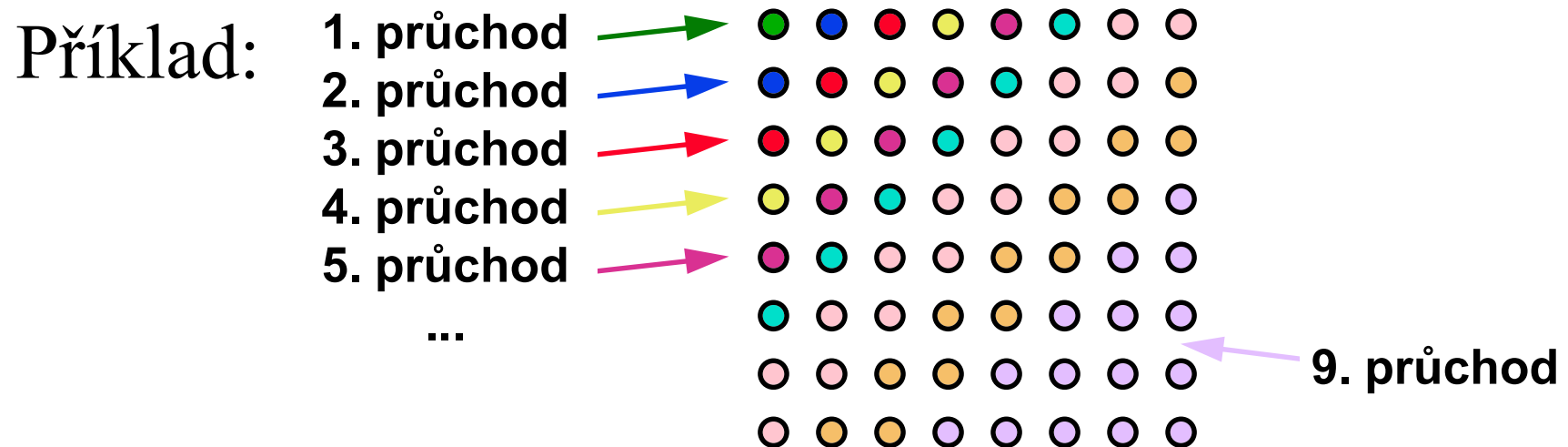
- ◆ JPEG může kódovat až **255 složek obrazu**
 - všechny složky musí mít stejný počet bitů/pixel (8 nebo 12 pro DCT, 2 až 16 pro bezeztrát. metodu)
- ◆ jednotlivé složky mohou mít **různá rozlišení**
 - celočíselné poměry 4 : 1 až 1 : 4
- ◆ uživatelsky definované **prokládání**
- ◆ možnost **přepínání kvantizačních a frekvenčních tabulek**
 - až 4 kvantizační a 4 frekvenční tabulky

Progresivní ztrátový režim

- ◆ obraz je kódován v **několika průchodech** - postupně se zlepšuje jeho kvalita
 - výhodné pro on-line aplikace (BBS, Internet)
 - přenos lze brzo přerušit v případě, že obrázek neodpovídá požadavkům)
- ◆ kodér i dekodér potřebují **buffer** pro F^Q koeficienty **celého obrázku**
 - o 3 bity větší přesnost než mají původní data
- ➔ **dvě progresivní metody**: spektrální výběr a postupná aproximace

Spektrální výběr

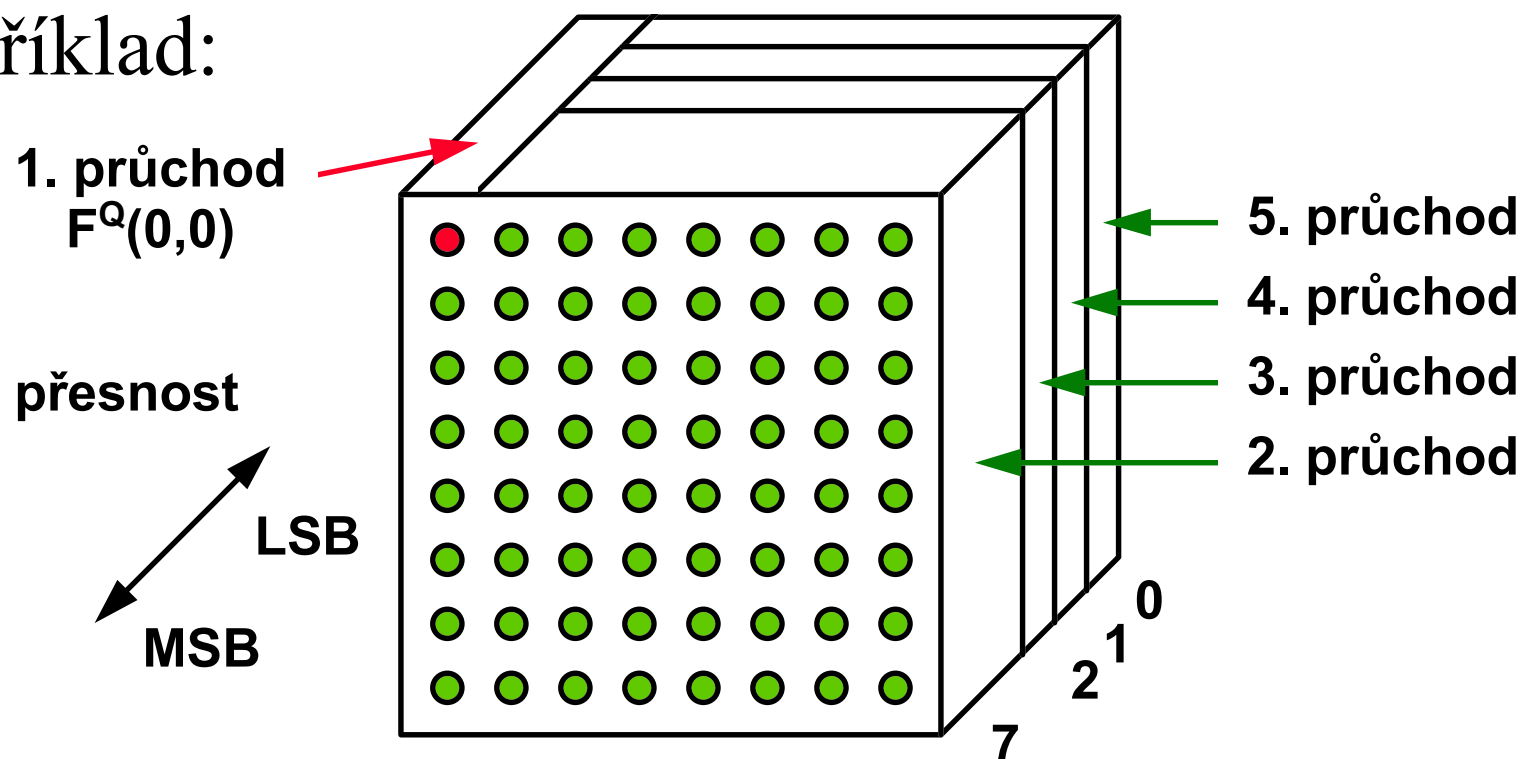
- ◆ v každém průchodu se přenese jen **několik DCT koeficientů**
 - od nejdůležitějších k méně důležitým
 - koeficienty se přenášejí v plné přesnosti



Postupná aproximace

- ◆ přenášejí se všechny DCT koeficienty, postupně se zpřesňují

Příklad:



Hierarchický režim

- ◆ **pyramidální kódování** - obraz je postupně přenášen v několika rozlišeních
 - o stupeň nižší rozlišení slouží po interpolaci jako predikce pro vyšší stupeň
 - kódují se pouze rozdíly
- ➔ jednotlivé stupně (snímky) se kódují **libovolnou metodou JPEG**
 - základní DCT, progresivní kódování nebo bezeztrátová metoda
 - poslední stupeň bezeztrát. \Rightarrow bezeztrát. hierarchie

JPEG File Interchange Format

- ◆ jednoduchá aplikace standardu komprese JPEG
 - formát JFIF: soubory s příponou **.JPG**
 - komplikovanější a bohatší je např. TIFF JPEG
- ➔ přenositelnost PC ↔ Mac ↔ UNIX
- ➔ standardní barevný systém **YCbCr** (CCIR 601)
- ➔ zjednodušený obraz - “thumbnail” (JPEG)
- ➔ podpora **různého rozlišení** jednotlivých barevných složek
 - barevné složky se mohou kódovat úsporněji

Barevný systém YC_bC_r

- ◆ podle doporučení **CCIR č. 601**
 - používá se i v barevném TV vysílání
- ◆ **8 bitů** na každou složku:
 - **Y: jasová složka** (odstín šedi nahrazující barvu)
 - **C_b resp. C_r : barevné rozdílové složky** (reprezentují převážně **B** resp. **R**)

$$Y = 0.299 R + 0.587 G + 0.114 B$$
$$C_b = -0.1687 R - 0.3313 G + 0.5 B + 128$$
$$C_r = 0.5 R - 0.4187 G - 0.0813 B + 128$$

Složky s různým rozlišením

◆ posun prvního vzorku $[0,0]$:

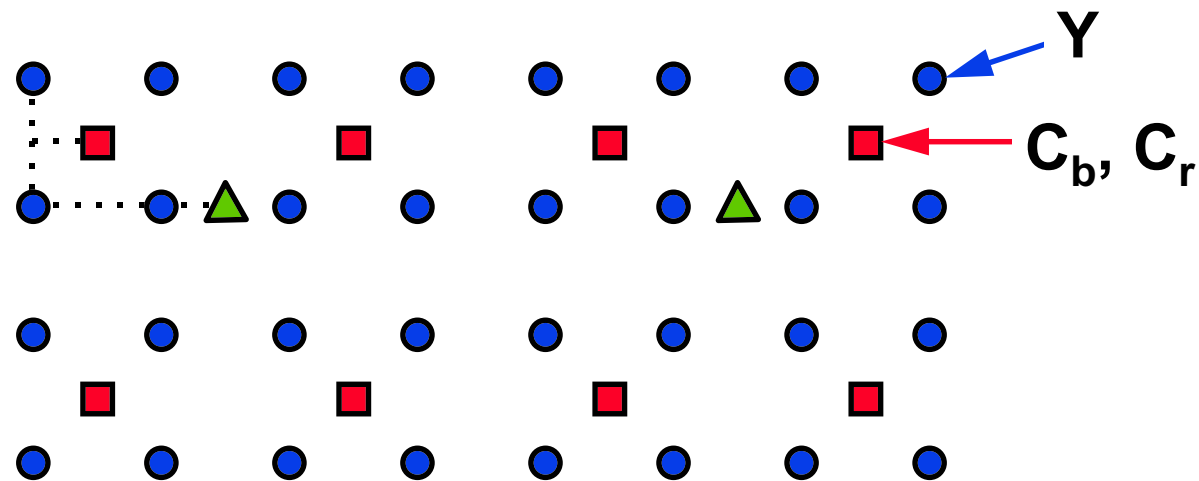
$$\underline{X_{\text{offset}}} = \frac{X_{\text{max}}}{2X_{\text{res}}} - \frac{1}{2} \quad \underline{Y_{\text{offset}}} = \frac{Y_{\text{max}}}{2Y_{\text{res}}} - \frac{1}{2}$$

Příklad:

● 256×288

■ 128×144

▲ 64×96



Příklad komprese JPEG

- 1 **RAYTRACE**: umělý obrázek spočítaný metodou sledování paprsku (1200×900×24 RGB)
- 2 **PAINTING**: impresionistický obraz sejmutý scannerem (600×685×24 RGB)
- 3 **BWPHOTO**: černobílá fotografie (portrét) sejmutá scannerem (624×735×8 gray)
- 4 **COLPHOTO**: barevná fotografie (krajina), velmi kvalitní digitalizace (512×480×24 RGB)

Výsledky komprese

obrázek	①	②	③	④
originál	3240 KB	1233 KB	458 KB	737 KB
srovnání - ARJ	41%	74%	71%	86%
JPEG - 100%	18.6%	31.0%	52.0%	34.0%
JPEG - 90%	7.3%	11.3%	20.0%	14.0%
JPEG - 70%	4.0%	5.1%	10.8%	7.0%
JPEG - 50%	3.0%	3.5%	7.5%	4.7%

Základní literatura

- **ISO/IEC JTC1 CD 10918: *Digital Compression and Coding of continuous-tone still images*, ISO 1993**
- **G. Wallace: *The JPEG Still Picture Compression Standard*, Communications of the ACM, April 1991**
- **E. Hamilton: *JPEG File Interchange Format, version 1.02*, September 1992**

Konec

Další informace:

- **V. Bhaskaran, K. Konstantinides: *Image and Video Compression Standards, Algorithms and Architectures*, Kluwer Academic Publishers, Boston 1995, 129-159**
- **ed. by H.-M. Hang, J. Woods: *Handbook of Visual Communications*, Academic Press, San Diego 1995, 242-246, 366-375**