

Komprese a kompresní algoritmy.

Dělení kompresních algoritmů. Metody jednoduché, statistické, slovníkové komprese. Transformační komprese.

Tomáš Bayer | bayertom@fsv.cvut.cz

Katedra geomatiky, fakulta stavební ČVUT.

Obsah přednášky

- 1 Barevné modely
- 2 Komprese a jejich rozdělení
- 3 Jednoduché kompresní algoritmy
- 4 Statistická komprese
- 5 Slovníková komprese
- 6 Transformační metody komprese

2. Barevné modely a vztahy mezi nimi

Rozdělení barevných modelů:

- *Modely založené na fyziologii oka*

Míchání trojice spektrálních barev ve zvoleném poměru. RGB, CMYK modely.

- *Modely kolorimetrické*

Míchání trojice spektrálních barev ve zvoleném poměru se zavedením vah. CIE model.

- *Modely komplementární*

Kombinace základních a komplementárních barev.

- *Modely psychofyzikální*

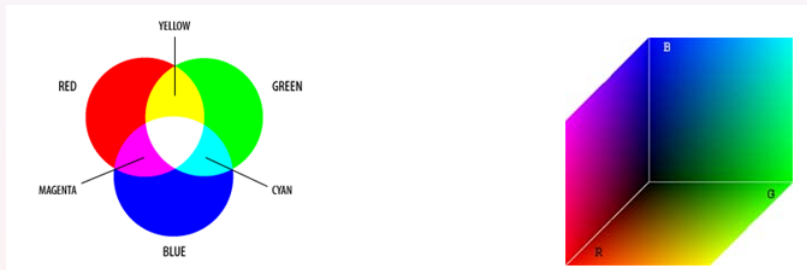
Kombinace Hue (barevný tón), Saturation (Sytost), Value (Jas). HSV model.

- *Modely fyzikální*

Přenos TV signálu. YUV model.

3. RGB model

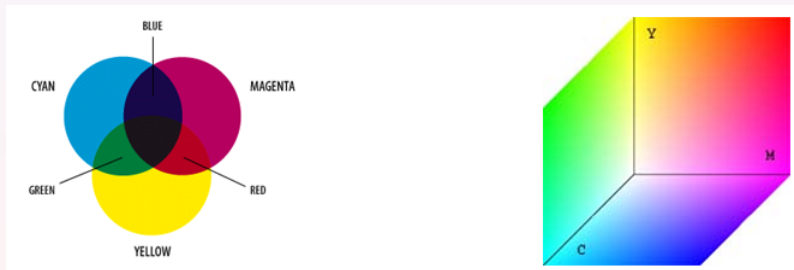
Znázornění modelu RGB za použití diagramu a jednotkové krychle.



Výsledná barva vzniká skládáním červené, zelené a modré barvy ve vhodných poměrech.

4. CMY(K) model

Znázornění modelu CMYK za použití diagramu a jednotkové krychle.



Výsledná barva vzniká skládáním žluté, azurové a purpurové barvy ve vhodných poměrech.

5. YUV model

Tvořen třemi složkami:

- jednou jasovou složkou Y ,
- dvěma barevnými (*chrominačními*) složkami C_B , C_R .

Vztah mezi RGB a $YC_B C_R$ modelem:

$$\begin{pmatrix} Y \\ C_B \\ C_R \end{pmatrix} = \begin{pmatrix} 0.2990 & 0.5870 & 0.1140 \\ -0.1687 & -0.3313 & 0.5000 \\ 0.5000 & -0.4187 & -0.0813 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} + \begin{pmatrix} 0 \\ 128 \\ 128 \end{pmatrix} \quad (1)$$

Vztah mezi $YC_B C_R$ a RGB modelem:

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1.0000 & 0.0000 & 1.4020 \\ 1.0000 & -0.3441 & -0.7141 \\ 1.0000 & 1.7720 & -0.0001 \end{pmatrix} \begin{pmatrix} Y \\ C_B - 128 \\ C_R - 128 \end{pmatrix} \quad (2)$$

YUV model používán při přenosu TV signálu.

6. Komprese dat a její hodnocení

=proces kódování dat, při kterém dochází k odstraňování nadbytečných informací s cílem umožnit efektivnější reprezentaci původních dat.

Kompresní poměr p .

Kompresní poměr p definujeme jako poměr velikosti souboru před kompresí a velikosti souboru po kompresi v bajtech (B). $p < 1 \Rightarrow$ *negativní* komprese.

$$p = \frac{s_v}{s_k}. \quad (3)$$

Kompresní faktor f .

Převrácená hodnota kompresního poměru.

$$f = \frac{1}{p} = \frac{s_k}{s_v} \quad (4)$$

Faktor relativní komprese r .

Účinnost kompresního algoritmu vzhledem k jinému algoritmu představujícímu etalon: logaritmus poměru velikosti souboru komprimovaného etalonem a velikosti souboru komprimovaného posuzovaným algoritmem.

$$r = 100 \cdot \ln \left(\frac{s_e}{s_k} \right) \quad (5)$$

7. Dělení kompresních algoritmů

3 základní kritéria dělení kompresních algoritmů:

- 1 *Podle vztahu ke komprimovaným datům*
Ztrátové komprese.
Bezztrátové komprese.
- 2 *Podle doby komprese*
Symetrické komprese.
Asymetrické komprese.
- 3 *Podle počtu průchodů*
Jednoprůchodové komprese.
Víceprůchodové komprese.

Jednotlivé skupiny často kombinovány s cílem dosažení co nejvyššího kompresního poměru.

8. Metody komprese

Dělení do do čtyř skupin:

- *Jednoduché metody komprese*
RLE komprese...
- *Statistické metody komprese*
Aritmetické kódování, Huffmanovo kódování...
- *Slovníkové metody komprese*
LZW, LZ-77, LZ-78...
- *Transformační metody komprese*
JPEG, JPEG 2000, fraktálová komprese.

Některé z kompresních algoritmů chráněny patentem (RLE).
Jiné nikoliv (lze je volně použít).

9. Metody jednoduché komprese

Charakteristika:

- Založeny na triviálních principech.
- Nepracují se složitějším matematickým aparátem.
- Symetrické, jednopřechodové.
- Bezztrátové.
- Nevýhodou nižší kompresní poměr.
- Nahrazení opakujících se posloupností znaků ve vstupním souboru 1 nebo > zástupnými znaky ve výstupním souboru.

Použití:

Rastrová data (formát CIT), textové dokumenty.

Zástupce:

Run Length Encoding (RLE).

10. RLE komprese

Nahrazování posloupnosti opakujících se znaků 1 / 2 zástupnými znaky.
 Pozor na negativní kompresi: počet opakování znaků >3 .
 Zpravidla používána pro nejméně čtyři opakující se znaky.

Efektivní pro černo-bílé rastry.
 Na RLE kompresi založen formát CIT.

Princip komprese:

- V souboru hledáme pravidelně se opakující posloupnosti znaků.
- Nahradíme jejich výskyt dvojicí znaků:
 - první znak: l_c identifikátor komprese
 V souboru se taková hodnota nesmí vyskytnout.
 - druhý znak: počet opakujících se znaků n .

Vstupní data:

172 126 126 126 126 126 126 126 126 195 195 21 21 21 21

Výstupní data:

172 256 126 8 195 195 256 21 4

11. RLE dekomprese

Modifikace RLE komprese:

Dosahuje vyšší účinnosti.

Do souboru přidáváme pouze jeden znak.

Zarážku l_c zvýšenou o počet opakování znaku n .

172 255+8 126 195 255+4 21 -> 172 263 126 195 259 21

Princip dekomprese:

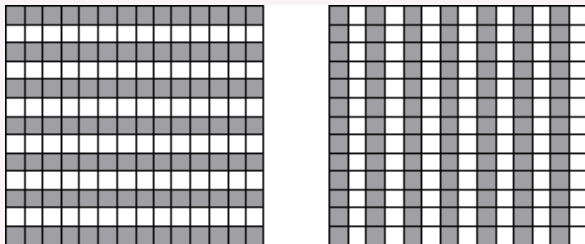
- 1 Načtení znaku c .
- 2 Test, zda je hodnota prvku větší než l_c :
 - Pokud $c < l_c$: přidáme c do dekomprimovaného souboru a načítáme další znak .
 - Jinak:
 - 1 Nastavíme příznak provádění komprese.
 - 2 Do výstupního souboru přidáme n opakujících se hodnot c .
 - 3 Zrušíme příznak dekomprese.

12. Účinnost RLE komprese

- + Nejlepší komprimační poměr u binárních rastrů.
- + Lepších výsledků dosahuje u dlouhých souborů.
- + Možnost existence delších sekvencí opakujících se znaků.

Důležitou roli hraje i orientace vzoru.

- Komprimace vodorovných linií efektivnější než komprimace svislých.



13. Metody statistické komprese

Charakteristika:

Založeny na statistické analýze četnosti znaků vstupního souboru.

Asymetrické, jednopřechodové.

Výrazně vyšší kompresní poměr než jednoduché metody.

Použití:

Rastrová data, textové dokumenty, obecná data (ZIP, RAR).

Zástupci:

- Huffmanovo kódování
- Shannon-Fanovo kódování

Historická metoda statistické komprese:

Morseova abeceda.

14. Shannon-Fanovo kódování

Využívá *prefixový kód* vzniklý analýzou vstupního souboru.

Prefixový kód:

Kód s minimální délkou, žádný kód znaku nezačíná kódem (tj. není prefixem) znaku jiného.

Princip Shannon-Fanova kódování:

Znaky původního souboru reprezentovány různě dlouhým kódem (posloupnost bitů) \Rightarrow úspora místa.

Znakům s větší pravděpodobností p_i přiřazuje kratší kódy k_i .

Znakům s menší pravděpodobností p_i přiřazuje delší kódy k_i .

Vstupní data:

172 126 126 126 126 126 126 126 126 195 195 21 21 21 21

Výstupní data:

111 0 0 0 0 0 0 0 0 110 110 10 10 10

15. Shannon-Fanovo kódování: komprese

Princip komprese:

- 1 Analýza četností znaků vstupního souboru: každému znaku přiřadíme hodnotu absolutní četnosti.
- 2 Uspořádané dvojice znak-četnost seřadíme sestupně podle hodnot četnosti.
- 3 Seznam rozdělíme na 2 intervaly:
Rozdíly sum četností p_i v obou rozdělených intervalech jsou minimální.
 - Prvky ležící v prvním intervalu: kód 0.
 - Prvky ležící v druhém intervalu: kód 1.
- 4 Body 2,3 rekurzivně aplikujeme na intervaly obsahující více než jeden prvek.

Postup tvorby Shannon–Fanova kódu.

Hodnota	p_i	S-F kód
126	8	8
21	4	1
195	2	1
172	1	1

Hodnota	p_i	S-F kód
126	8	0
21	4	10
195	2	11
172	1	11

Hodnota	p_i	S-F kód
126	8	0
21	4	10
95	2	110
172	1	111

16. Huffmanovo kódování

Charakteristika:

Podobný princip jako Shannon-Fanův kód.

Huffmanův kód lze reprezentovat Huffmanovým stromem.

Listy tvoří prvky s nejnižší četností výskytu.

Listy mají nejdelší kód (cesta od kořene k listům).

Varianty Huffmanova kódování:

- statické Huffmanovo kódování,
- adaptivní Huffmanovo kódování.

Výsledkem Huffmanova kódování *optimální prefixový kód*.

Vstupní data:

172 126 126 126 126 126 126 126 126 195 195 21 21 21 21

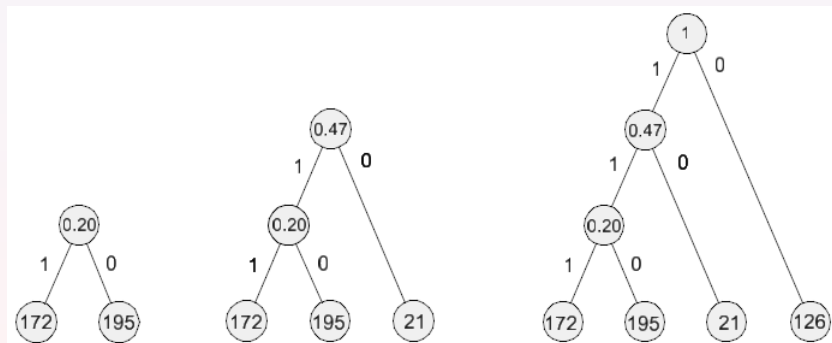
17. Huffmanovo kódování

Princip komprese:

- 1 Nalezneme 2 uzly představované znaky s nejnižšími relativní četnostmi. Budou tvořit listy nově vytvářeného Huffmanova stromu.
- 2 Vytvoříme předka této dvojice uzlů (kořen). Ohodnocení předka představuje součet ohodnocení potomků.
- 3 Nalezneme dvojici uzlů s nejnižším ohodnocením. Může být tvořena kombinacemi:
 - 1 uzel-list,
 - 2 list-list.
- 4 Vytvoříme předka této dvojice (nový kořen). Jeho ohodnocení je rovno součtu ohodnocení potomků.
- 5 Opakujeme body 3) a 4) pro všechny znaky. Tj. dokud nedojde ke spojení dílčích stromů.

18. Huffmanovo kódování: ukázka

Postup tvorby Huffmanova prefixového stromu.



Výstupní data:

111 0 0 0 0 0 0 0 0 0 110 110 10 10 10

19. Huffmanovo kódování: dekomprese

Načítání komprimovaného souboru po bitech.

Procházíme prefixový strom shora dolů (od kořene k listům):

- 1 Načteme znak.
- 2 Je-li roven 0, přejdeme do pravého podstromu.
- 3 Je-li roven 1, přejdeme do levého podstromu.
- 4 Postup opakujeme tak dlouho, dokud nedorazíme do listu.
Huffmanův kód nahradíme listem.

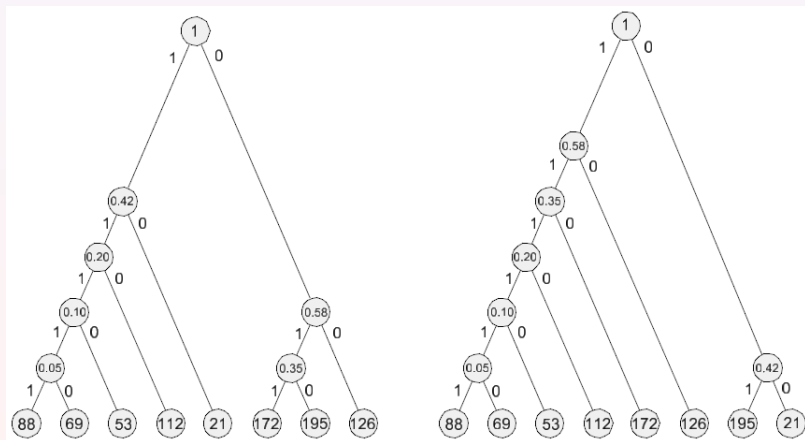
Body opakujeme, dokud nedosáhneme konce souboru.

Praktické použití:

Kompresi JPEG, MPEG či ZIP (Huffmanovo kódování + LZW), ARJ.
Není patentově chráněno.

20. Ukázka Huffmanova stromu

Různé varianty Huffmanova prefixového stromu pro stejná data.



21. Metody slovníkové komprese

Založeny na vyhledávání opakujících se posloupností znaků.
Posloupnost v každém budoucím výskytu nahrazujeme kratším kódem.

Nemají k dispozici statistické údaje o prvcích komprimovaného souboru.

Soubor není na rozdíl od statistických metod předem analyzován.

Dělení podle metodiky komprese:

- Metoda posuvného okna.
- Metoda rostoucího slovníku.

Dělení podle způsobu tvorby slovníku:

- Metoda statické slovníkové komprese.
- Metoda adaptivní slovníkové komprese.

22. Dělení metod slovníkové komprese

Metoda posuvného okna

Vyskytovala se posloupnost znaků určité délky ve vstupních datech?

Pokud ano, je místo ní umístěn odkaz na první (popř. předchozí) místo výskytu.

LZ-77, LZB, LZH.

Metoda rostoucího slovníku

Ze vstupních dat je dynamicky vytvářen slovník.

Nachází se posloupnost znaků nenachází ve slovníku?

Pokud ano, je na jeho místo umístěn kód, pod kterým se ve slovníku nachází.

LZ-78 a LZW.

Metoda statické slovníkové komprese

Slovník je vytvořen před počátkem komprese.

V průběhu komprese se nemění (součástí komprimovaného souboru).

Metoda adaptivní slovníkové komprese

Slovník je vytvářen v průběhu komprese souboru na základě vstupních dat.

Není součástí komprimovaných dat, algoritmus ho umí zpětně z komprimovaných dat vytvořit.

LZW.

23. Algoritmus LZ-77

Název akronymem příjmení tvůrců Abrahama Lempela a Jakoba Ziva (Izrael) + roku objevení.

Charakteristika LZ-77:

- Založena na nahrazování opakujících se posloupností znaků (tzv. předpon) kódy odkazujícími na jejich výskyt v předchozím textu.
- Nalezneme –li takovou posloupnost znaků, nahradíme jejich výskyt trojicí znaků.
- První znak představuje vzdálenost počátku předpony od aktuálně prohledávané pozice.
- Druhý znak délku předpony.
- Třetí znak první následující znak za předponou.

Příklad:

Vstupní data: "KOKOS"

Výstupní data: "K022S"

Předpona "K0" začíná o 2 znaky vlevo vzhledem k aktuální pozici

24. Algoritmus LZ-77: komprese (1/2)

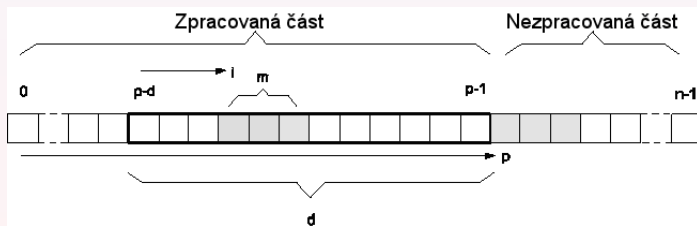
LZ-77 využívá posuvného okna, *sliding window*.

Pohybuje nad vstupním souborem ve směru zleva do prava.

Různá délka, závisí na způsobu implementace.

Nejčastěji se používá hodnota 215 bajtů, tj. 32 768 znaků.

Délka okna d , počet znaků v souboru n .



Hledáme řetězec s maximální délkou m (=počet shodných znaků), který je předponou dosud nezpracované části souboru.

Aktuální pozice v textu p (=počátek dosud nezpracované části souboru).

25. Algoritmus LZ-77: komprese (2/2)

Pozice počátku shodného řetězce vzhledem k počátku okna i .

Aktuální poloha okna dána souřadnicemi $\{p - d, p - 1\}$,

Dosud nezpracovaná část souboru dána souřadnicemi $\{p, n - 1\}$.

Relativní pozici dp počátku shodného řetězce vzhledem k p

$$dp = d - i + 1. \quad (6)$$

Hodnota z =poslední zpracovaný znak.

- V případě nalezení shody ($m > 0$) ho lze interpretovat jako první „*neshodný*“ znak. V dalším kroku bude z představovat poslední znak sliding window

$$z = p + m. \quad (7)$$

- Pokud nebyla nalezena žádná shoda mezi nezpracovanou částí textu a sliding window, $dp = 0$, $z = p$, $m = 0$, sliding window posuneme o jeden znak vpravo.
- Pokud byla nalezena shoda v m znacích, posuneme sliding window o $m + 1$ znaků.

Novou aktuální pozici p určíme

$$p = z + 1 = p + m + 1. \quad (8)$$

Výstup LZ-77: uspořádaná posloupnost tří údajů: hodnota dp , hodnota m a znak z .

26. Algoritmus LZ-77: Ukázka komprese

Vstupní data: Leteljelenjetelem

Výstupní data 1 e t 2 1 l j 3 2 e n 1 1 a d j 6 1 t 2 1 l 4 1 m

Text																									dp	m	z
1	e	t	e	l	j	e	l	e	n	n	a	d	j	e	t	e	l	e	m	0	0	l					
1	e	t	e	l	j	e	l	e	n	n	a	d	j	e	t	e	l	e	m	0	0	e					
1	e	t	e	l	j	e	l	e	n	n	a	d	j	e	t	e	l	e	m	0	0	t					
1	e	t	e	l	j	e	l	e	n	n	a	d	j	e	t	e	l	e	m	2	1	l					
1	e	t	e	l	j	e	l	e	n	n	a	d	j	e	t	e	l	e	m	0	0	j					
1	e	t	e	l	j	e	l	e	n	n	a	d	j	e	t	e	l	e	m	3	2	e					
1	e	t	e	l	j	e	l	e	n	n	a	d	j	e	t	e	l	e	m	0	0	n					
1	e	t	e	l	j	e	l	e	n	n	a	d	j	e	t	e	l	e	m	1	1	a					
1	e	t	e	l	j	e	l	e	n	n	a	d	j	e	t	e	l	e	m	0	0	d					
1	e	t	e	l	j	e	l	e	n	n	a	d	j	e	t	e	l	e	m	0	0	j					
1	e	t	e	l	j	e	l	e	n	n	a	d	j	e	t	e	l	e	m	6	1	t					
1	e	t	e	l	j	e	l	e	n	n	a	d	j	e	t	e	l	e	m	2	1	l					
1	e	t	e	l	j	e	l	e	n	n	a	d	j	e	t	e	l	e	m	4	1	m					

27. Použití algoritmu LZ-77

Metoda “Deflate”:

LZ-77 v kombinaci s Huffmanovým kódováním označujeme jako metodu *deflate*. Je základem komprimačních programů ZIP a PKZIP a představuje jeden z nejčastěji používaným komprimačních algoritmů vůbec.

Použití LZ-77:

LZ-77 je používán i v oblasti počítačové grafiky.

Dobré kompresní poměry pravidelně se opakujících posloupností znaků (tj. pravidelné vzory).

Součástí formátu PNG (grafická data v prostředí Internetu).

Výhoda oproti jiným formátům (např. JPEG): i přes velký kompresní poměr LZ-77 bezztrátová, nedochází k vizuální degradaci rastrových dat.

Účinnost komprese:

Ovlivněna délkou okna.

Dlouhé okno=vyšší kompresní poměr, značně vzroste doba komprese.

Krátké okno=doba komprese se výrazně zkrátí, dosažený kompresní poměr nižší.

28. Transformační metody komprese

Využívají *geometrické transformace* obrazových dat.
Obraz při kompresi aproximován vhodnou funkcí.
Při dekompresi zpětně matematicky rekonstruován.

Často víceprůchodové, ztrátové.
Kombinace řady kompresních technik.
Mnohem vyšší kompresní poměr, až 500:1.

Zástupci: JPEG komprese, vlnková komprese, fraktální komprese.

Princip transformačních metod:

Využívají faktu, že lidské oko není příliš citlivé.
Lze ho snadno “oklamat”.
Oko si nevšimne malých odchylek ve změnách barvy či jasu v rastru.
Vyšší citlivost vůči změně jasu.

29. JPEG komprese

Charakteristika JPEG komprese:

- nejpoužívanější (nikoliv však nejlepší) technika komprese rastrů,
- ztrátový algoritmus kombinující více technik,
- dosahuje vyšší kompresní poměr než bezztrátové algoritmy, až 15:1,
- vhodná pro kompresi přirozených rastrů (fotografie),
- nevhodná pro kompresi rastrů s ostrými barevnými přechody.

Faktor komprese q .

- nastavení úrovně komprimace: ovlivňuje množství informací, které budou při kompresi „zanedbány“,
- nabývá hodnoty v intervalu $\langle 0, 100 \rangle$, odpovídá procentuálnímu vyjádření.

30. Podstata JPEG komprese

Malé změny barvy je lidské oko méně citlivé než na malé změny jasu.

Nevýznamné změny barev jsou odstraňovány.

Změny jasu jsou naopak s co největší přesností uchovávány.

Charakteristika:

- Nepracuje s rastroem jako s celkem
Dochází k rozdělení obrazu na submatice 8x8.
- Postup zpravidla po řadách.
Z levého horního do pravého dolního rohu rastru.
- Kombinuje několik různých postupů pro zvýšení kompresního poměru.
- Využívá poměrně robustní matematický aparát.
Diskrétní kosinová transformace (DCT).
- JPEG komprese má „rozumnou“ výpočetní složitost i na průměrně výkonném hardware.

31. Postup JPEG komprese (1/2)

- *Separace obrazu na R, G, B složky*

Separace R, G, B složek ze vstupního obrazu.
Následná komprese probíhá po složkách.

- *Rozdělení na submatice*

Rozdělení barevných složek na submatice o rozměrech 8x8.
Dále zpracovávány samostatně bez ohledu na “okolní” submatice.

- *Transformace RGB do $Y C_B C_R$*

Cílem co nejmenší poškození jasové složky.
Ta bude komprimována méně než obě složky chrominační.

$$\begin{aligned} Y &= 0,2990 \cdot R + 0,5870 \cdot G + 0,1140 \cdot B, \\ C_B &= -0,1687 \cdot R - 0,3313 \cdot G + 0,5000 \cdot B + 128, \\ C_R &= 0,5000 \cdot R - 0,4187 \cdot G - 0,0813 \cdot B + 128. \end{aligned} \quad (9)$$

32. Postup JPEG komprese (2/2)

- *Transformace intervalu*

Transformace hodnot barevných složek Y , C_B , C_R z intervalu $\langle 0, 255 \rangle$ na interval $\langle -255, 255 \rangle$ (označení Y' , C'_B , C'_R).

$$\begin{aligned} Y' &= 2 \cdot Y - 255 \\ C'_B &= 2 \cdot C_B - 255 \\ C'_R &= 2 \cdot C_R - 255 \end{aligned} \tag{10}$$

- *Převzorkování (resampling) rastru*

První ztrátový krok komprese, cílem snížení počtu barevných odstínů. Převzorkování průměrováním submatic 2*1, 2*2, 3*3, 4*4 pixely.

- *Diskrétní kosinová transformace (DCT)*

Matematický základ JPEG komprese.

33. Diskrétní kosinová transformace (DCT)

= výpočetně nejnáročnějším prvek JPEG komprese.

Převádí obrazový signál závislý na čase posloupnost signálů.

Odlišují se amplitudou a frekvencí, tvoří *frekvenční spektrum*.

Rastr: největší množství informací soustředěny v oblastech s nižšími frekvencemi.

Efektivní uložení nejvýznamnější informací do poměrně malého množství koeficientů.

Rovnice DCT:

$$F(u, v) = \frac{1}{4} C(u) \cdot C(v) \left[\sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \cdot \cos \frac{(2x+1)u\pi}{16} \cdot \cos \frac{(2y+1)v\pi}{16} \right] \quad (11)$$

$$C(u) = \begin{cases} \frac{\sqrt{2}}{2}, & u = 0, \\ 1, & u \neq 0. \end{cases}, \quad C(v) = \begin{cases} \frac{\sqrt{2}}{2}, & v = 0, \\ 1, & v \neq 0. \end{cases}$$

34. Diskrétní kosinová transformace

$x, y \dots$ řádkový / sloupcový index ve zdrojovém rastru.

$f(x, y) \dots$ hodnota barevné / jasové složky ve zdrojovém rastru.

$u, v \dots$ řádkový / sloupcový index v cílovém rastru.

$F(u, v) \dots$ transformovaná hodnota barevné / jasové složky v cílovém rastru.

Vlastnosti matice F :

Největší hodnotu má prvek $F[0][0] \Rightarrow$ *DC člen*.

Nejvíce ovlivňuje kvalitu obrazu.

AC členy:

Další prvky s vyššími hodnotami jsou soustředěny v levém horním rohu.

Čím má prvek větší význam, tím je jeho hodnota vyšší.

Prvky s malou hodnotou mohou být zanedbány.

Řada prvků má podobné hodnoty.

Pokud by měly stejnou hodnotou, komprese by byla efektivnější.

35. Kvantizace DCT koeficientů (1/2)

Nejvíce ztrátová část JPEG komprese.

Provádí zaokrouhlení DCT koeficientů.

Cílem je vypuštění “nepodstatných” koeficientů (koeficientů s malou hodnotou).

Výsledkem kvantizace matice F_Q , u které bude řada AC koeficientů stejných.

Postup kvantizace:

Kvantizace provádí dělení matice $F(u, v)$ “po prvcích” vhodnou kvantizační maticí $Q(u, v)$.

$$F_Q(u, v) = \frac{F(u, v)}{Q(u, v)} \quad (12)$$

Kvantizační matice $Q(u, v)$ bude z důvodu různého stupně komprese jiná pro složku Y a složky C_B, C_R .

$$Q(u, v)_{50}^Y = \begin{pmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 87 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 26 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{pmatrix}$$

36. Kvantizace DCT koeficientů (2/2)

$$Q(u, v)_{50}^C = \begin{pmatrix} 17 & 18 & 24 & 47 & 66 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 69 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{pmatrix}$$

Kvantizační matice se ukládá do JPEG souboru.

Prvky matice $Q(u, v)$ lze určit z (13).

Faktor komprese q uváděn v procentech

$$Q(u, v) = \frac{50 \cdot Q(u, v)_{50}}{q}. \quad (13)$$

Kvantizace citlivější ke složce Y , prvky $F(u, v)$ dělíme “menšími hodnotami” $Q(u, v)$.

Kvantizovaná matice je maticí řádkou, trojúhelníkovou, dominantní prvek v levém horním rohu.

Spousta členů je rovná nule.

Můžeme dosáhnout vyššího kompresního poměru než při kompresi matice nekvantizované.

37. Ukázka matice před kvantizací a po kvantizaci

$$F(u, v) = \begin{pmatrix} 1270 & 125 & 20 & 19 & 6 & 5 & -15 & 10 \\ -3 & -80 & 35 & 23 & 40 & 32 & 14 & -8 \\ 104 & -15 & -51 & -32 & -56 & 8 & -4 & -3 \\ -18 & 11 & 11 & 42 & -3 & -11 & -2 & 1 \\ -2 & -15 & -13 & -15 & -8 & 5 & 2 & -2 \\ -16 & 4 & -3 & 11 & -1 & -8 & -4 & 0 \\ 12 & 3 & -1 & -3 & 13 & 3 & 5 & 0 \\ -7 & 8 & 2 & -2 & -5 & -5 & -1 & 1 \end{pmatrix}$$

$$F_Q(u, v)^Y = \begin{pmatrix} 79 & 11 & 2 & 1 & 0 & 0 & 0 & 0 \\ 0 & -7 & 3 & 1 & 2 & 1 & 0 & 0 \\ 7 & -1 & -1 & -1 & -1 & 0 & 0 & 0 \\ -1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

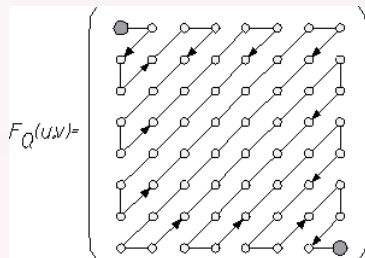
38. Uspořádání do CLK-CAK sekvencí+kompresce

Cílem vytvoření posloupnosti koeficientů, kterou lze efektivně komprimovat.

Prvky se stejnými hodnotami budou za sebou.

První prvek $F_Q[1][1]$, poslední prvek $F_Q[8][8]$.

Prvky v cik-cak sekvenci řazeny ve směru diagonál matice F_Q .



Ukázka:

79 11 07 -7 2 13 -1 -1 0 1 -3 1 0 0 2 -1 1 -1 -1 0 0 0 1 -1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 ...

Následně komprese cik-cak sekvence *Huffmanovým kódováním*.

39. JPEG dekomprese

- *Dekomprese Huffmanova kódu*

Dekomprese binárních dat komprimovaných Huffmanovým kódem.

Výsledkem je posloupnost dat představující CIK-CAK sekvenci.

- *Naplnění submatice $F_Q(u, v)$*

Převod CIK-CAK sekvence na submatici $F_Q(u, v)$, rozměr (8x8).

- *Dekvantizace koeficientů*

Inverzní operace ke kvantizaci.

Není ztrátová, nedochází k zaokrouhlování hodnot vypočtených koeficientů.

$$F(u, v) = F_Q(u, v) \cdot Q(u, v). \quad (14)$$

- *Inverzní diskretní kosinová transformace (IDCT)*

Výpočetně nejnáročnější krok dekomprese.

39. Inverzní diskrétní kosinová transformace

Převod prostorové frekvence do prostorových souřadnic (x, y, barva).
Převod frekvenčního spektra na časové.

$$f(x, y) = \frac{1}{4} \left[\sum_{u=0}^7 \sum_{v=0}^7 C(u) \cdot C(v) F(u, v) \cdot \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16} \right], \quad (15)$$

$$C(u) = \begin{cases} \frac{\sqrt{2}}{2}, & u = 0, \\ 1, & u \neq 0. \end{cases}, \quad C(v) = \begin{cases} \frac{\sqrt{2}}{2}, & v = 0, \\ 1, & v \neq 0. \end{cases}$$

x, y...řádkový / sloupcový index v cílovém rastru.

f(x, y)...transformovaná hodnota barevné/jasové složky v cílovém rastru.

u, v... řádkový / sloupcový index ve zdrojovém rastru.

F(u, v)... hodnota barevné/jasové složky ve zdrojovém rastru.

40. JPEG dekomprese

- Zpětná transformace intervalu*

Zpětná transformace intervalu $\langle -255, 255 \rangle$ na interval $\langle 0, 255 \rangle$

$$\begin{aligned} Y &= 0,5 \cdot (Y' + 255), \\ C_B &= 0,5 \cdot (C'_B + 255), \\ C_E &= 0,5 \cdot (C'_R + 255). \end{aligned} \quad (16)$$

- Převod $YC_B C_R \Rightarrow RGB$*

Zpětná konverze mezi modely.

Převod modelu $YC_B C_R$ do modelu RGB.

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1.0000 & 0.0000 & 1.4020 \\ 1.0000 & -0.3441 & -0.7141 \\ 1.0000 & 1.7720 & -0.0001 \end{pmatrix} \begin{pmatrix} Y \\ C_B - 128 \\ C_R - 128 \end{pmatrix} \quad (17)$$

41. Vhodnost použití JPEG komprese (1/2)

Nepřirozené dělení vstupního rastru na submatice (8x8).
Negativně ovlivňuje výslednou kvalitu dekomprimovaného obrazu.

Při vyšším faktoru komprese viditelné „artefakty“ těchto submatic.
Pravidelné plochy pixelů podobných barevných odstínů.
Ztráta detailů: rastr vyhlazen (zprůměrován).
Roste s větším zaokrouhlením koeficientů $F(u, v)$.

Vliv faktoru komprese na kvalitu přirozeného rastru.
Vlevo originál, uprostřed $q = 10\%$, vpravo $q = 90\%$.



42. Vhodnost použití JPEG komprese (2/2)

Nevhodnost formátu JPG pro ostré barevné přechody.

Dochází k vizuální degradaci dat, moiré efekty.

Pro tento typ dat by JPEG komprese neměla být vůbec používána.

JPEG komprese rastru v 256 barvách obsahujícího ostré barevné přechody.

Vizuální degradace obrazu pro $q = 90\%$.

