

Komprese obrazů

Václav Hlaváč

České vysoké učení technické v Praze

Český institut informatiky, robotiky a kybernetiky

160 00 Praha 6, Jugoslávských partyzánů 1580/3

<http://people.ciirc.cvut.cz/hlavac>, vaclav.hlavac@cvut.cz

také z Centra strojového vnímání, <http://cmp.felk.cvut.cz>

Poděkování: Tomáš Svoboda, Jan Kybic

Osnova přednášky:

- ◆ Redundance a irelevance; 1D a 2D.
- ◆ Kroky komprese obrazu.
- ◆ Entropie a komprese.
- ◆ Optimální kódování.
- ◆ Komprese segmentovaných obrazů.
- ◆ Ztrátová komprese v prostoru obrazů.
- ◆ Komprese v transformovaném prostoru. Např. JPEG, Wavelets komprese.

Kompresa obrazů, úvod

- ◆ **Cíl:** Zmenšení množství dat potřebných k reprezentaci obrazu. Spotřebované množství paměti se měří například v bitech.
 - ◆ **Použití:** Pro přenos a uchování dat.
 - ◆ **Proč se liší komprese 2D obrazů od komprese 1D dat?**
-
- ◆ Digitalizovaný obraz se chápe jako dvojrozměrná struktura (matice) náhodných vzorků.
 - ◆ Matematicky řečeno: Cílem je převést matici jasů (nebo 3 matice s barevnými složkami) do jiné reprezentace, v níž jsou obrazová data méně statisticky závislá (zhruba řečeno, méně korelovaná).

Proč je možná komprese obrazů?

- ◆ Obrazy nejsou náhodné. Nejsou většinou pouze náhodným šumem.
- ◆ V obrazech je nadbytečnost (redundance). Hodnotu obrazové funkce lze z hodnot v okolí předpovídat.
- ◆ Hodnoty obrazové funkce nejsou rozděleny rovnoměrně.
- ◆ Mezi hodnotami obrazové jsou plošné závislosti (korelace), velmi často se projevují lokálně, někdy i globálně (např. u některých textur).
- ◆ U barevných obrazů jsou mezi jejich jednotlivými barevnými kanály statistické závislosti.
- ◆ Ne všechny detaily jsou v obrazech potřebné, když je bude pozorovat člověk. Vizuální subsystém člověka mnohé detaily není schopen vnímat.

Možné čtení

- ◆ Anil Jain: “Fundamentals of Digital Image Processing”, 1989.
- ◆ Milan Šonka, Václav Hlaváč, Roger Boyle: “Image Processing, Analysis, and Machine Vision”, 2015, 4th edition.
- ◆ Tomáš Svoboda, Jan Kybic, Václav Hlaváč: “Image Processing, Analysis, and Machine Vision, A MATLAB Companion”, 2007. <http://visionbook.felk.cvut.cz>
- ◆ Petr Vysoký: C. E. Shannon - průkopník informačního věku, Vesmír, svazek 83, srpen 2004, str. 472-473.

Podvzorkování, motivace pro kompresi obrázků

- ◆ Snížení prostorového rozlišení obrazu.
- ◆ Jednoduchá, ztrátová metoda komprese, která se obvykle hodí pro web nebo zobrazování na monitoru s omezeným prostorovým rozlišením.
- ◆ Lze vylepšit pokročilejší interpolací, např. pomocí B-splajnů.

Snížení rozlišení, příklad (1)



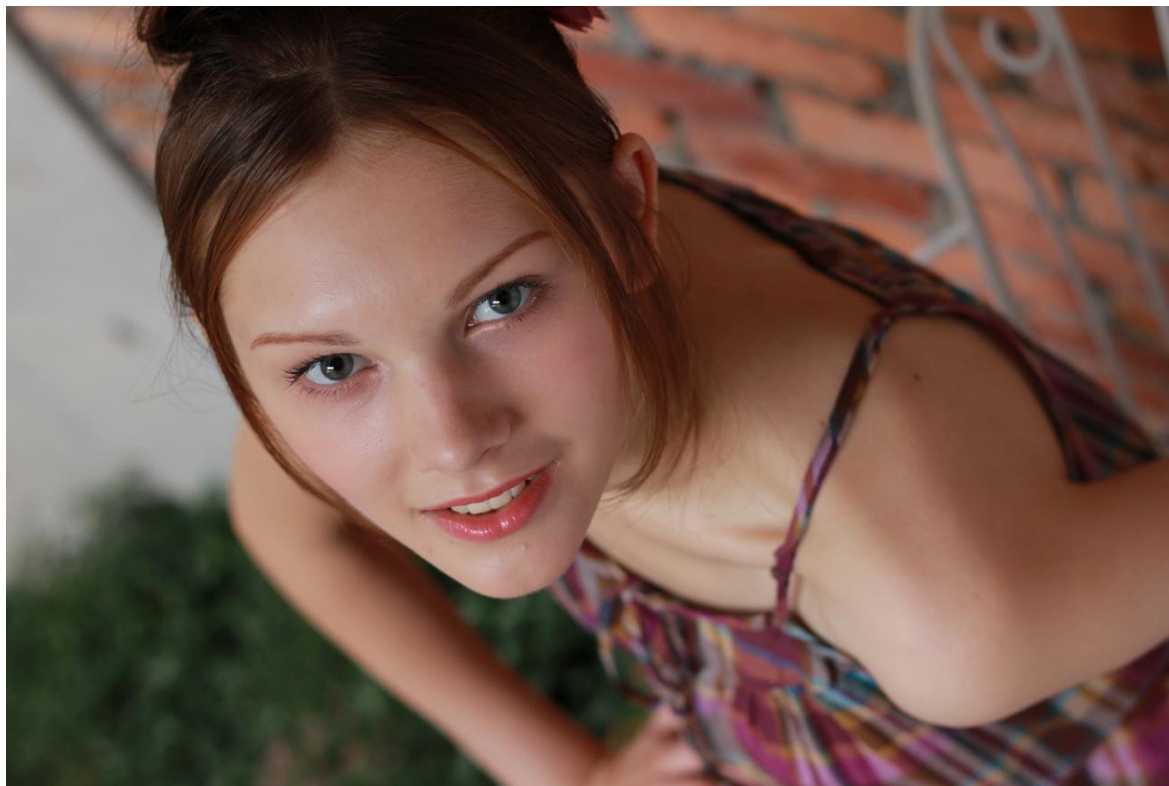
Original size, 3456×5184 , 859 kB (stored as JPEG with quality 75).

Snížení rozlišení, příklad (2)



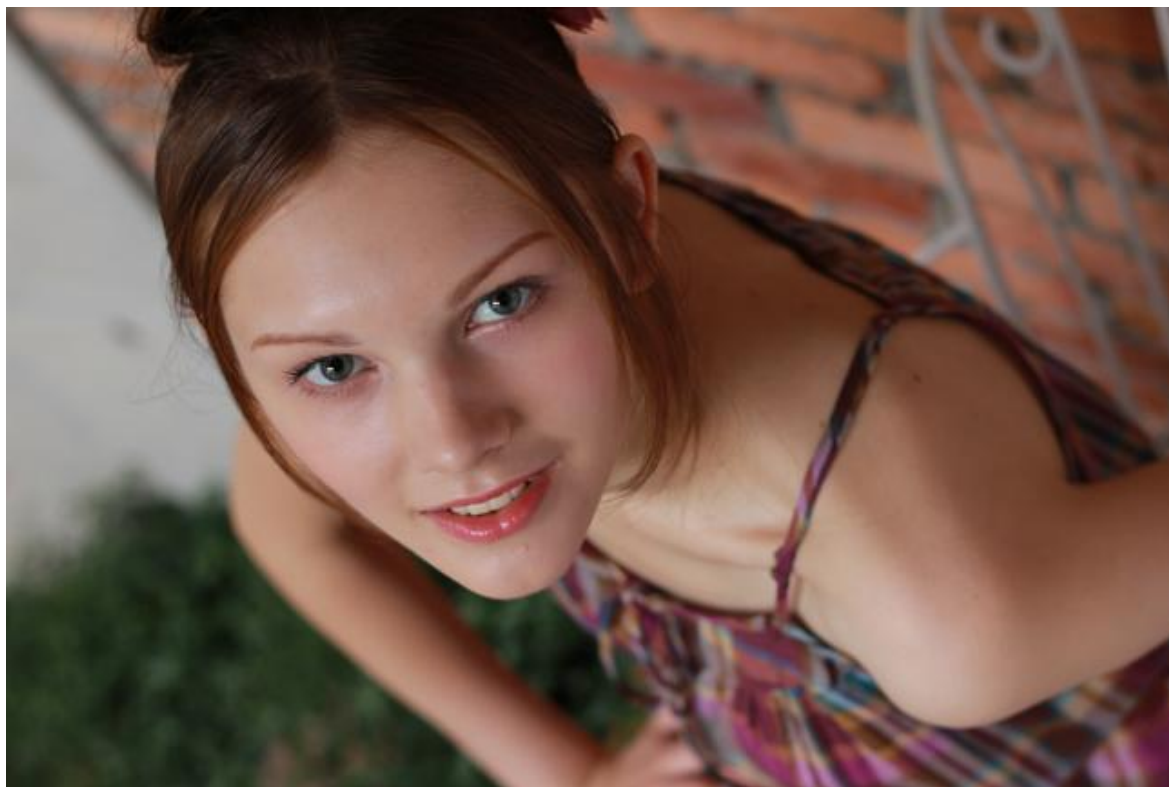
Downsampled $2\times$, 1728×2592 , 237 kB.

Snížení rozlišení, příklad (3)



Downsampled $4\times$, 864×1296 , 75 kB.

Snížení rozlišení, příklad (4)



Downsampled $8\times$, 432×648 , 27 kB.

Snížení rozlišení, příklad (5)



Downsampled $16\times$, 216×324 , 10 kB.

Snížení rozlišení, příklad (6)



Downsampled $16\times$, 216×324 , 10 kB, bicubic interpolation.

Snížení rozlišení, příklad (7)



Downsampled $32\times$, 108×162 , 4.2 kB.

Snížení rozlišení, příklad (8)



Downsampled $32\times$, 108×162 , 4.2 kB, bicubic interpolation.

Redundance, irrelevance

◆ Redundance v kódování

- Základní princip: častěji se opakující symbol se kóduje kratším kódovým slovem než řídčeji se opakující symbol.
- Optimální kódování: Huffmanovo a aritmetické kódování.

◆ Redundance mezi pixely, modeluje se a jen residuum vůči modelu se kóduje, protože má menší rozptyl. Různé modely, např.:

- Lineární integrální transformace obrazu, např. Fourierova, kosinová či vlnková.
- Prediktivní modely, např. lineární kombinace několika předchozích vzorků.
- Úsporné algoritmy generování obrazu, např. fraktální.

◆ Irrelevance z hlediska vnímání člověkem

- Nezobrazit např. všechny jasové úrovně, barvy nebo frekvence (obvykle vysoké).

Rozdělení metod komprese obrazů

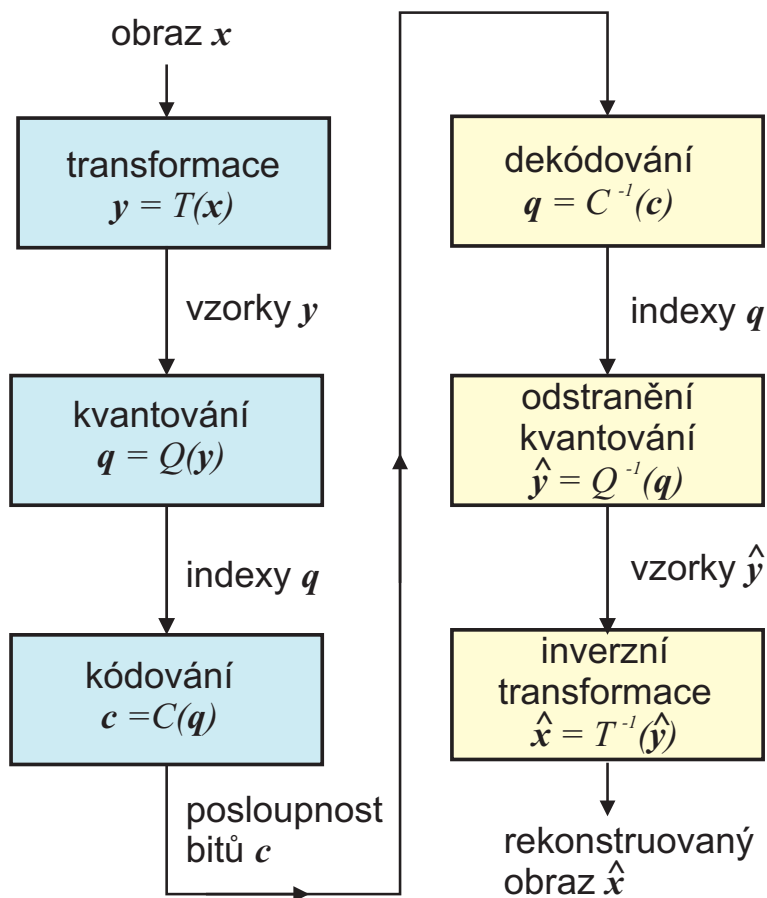
1. **Založené na interpretaci dat** $\Rightarrow$ je potřebná segmentace obrazu.
 - ◆ Metody jsou závislé na sémantice dat.
 - ◆ Dosahuje se nejvyšších kompresních poměrů.
 - ◆ Dekomprese nezíská původní obraz zcela.
2. **Bez interpretace dat** $\Rightarrow$ odstranění redundantní a irelevantní informace.
 - ◆ Lze použít na libovolný obraz, a to bez ohledu na jeho sémantiku.
 - ◆ Využívá se statistických nadbytečností v obraze a (případně) irelevance některé informace pro lidské vnímání.

Dvě třídy metod bez interpretace – ztrátové a bezeztrátové

Dvě velké třídy používaných postupů:

1. **Bezeztrátové metody**, odstraňují redundanci.
 - ◆ Odstraňují pouze statistickou nadbytečnost.
 - ◆ Umožňují úplnou rekonstrukci výchozího signálu/obrazu.
2. **Ztrátové metody**, odstraňují irelevantní informaci.
 - ◆ Nevratně odstraňují část informace, která má v daném kontextu malý význam (např. vysoké frekvence, jasové detaily nepostřehnutelné lidským okem).
 - ◆ Umožňují pouze částečnou rekonstrukci výchozího signálu/obrazu.

Kompresa obrazu a jeho zpětná rekonstrukce



- ◆ Transformace T odstraňuje redundanci a je obvykle invertovatelná.
Např.: kosínová transformace, kódování úseky řádků (RLE).
- ◆ Kvantování Q odstraňuje irelevanci a není invertovatelné.
Např.: Zanedbání koeficientů kosínové transformace odpovídajících vysokým frekvencím.
- ◆ Kódování C a dekódování C^{-1} je invertovatelné a bezztrátové.

Teorie informace a redundance

- ◆ **Entropie ve fyzice** je měrou energie soustavy, která není k dispozici k vykonání práce. Jelikož práci lze získat “z řádu soustavy”, je entropie měrou neuspořádanosti soustavy. Souvisí s druhou termodynamickou větou.
- ◆ Pojem zavedl v roce 1850 německý fyzik Rudolf Clausius (1822-1888, jeden ze zakladatelů termodynamiky).

Entropie v teorii informace, Claude Shannon, 1948

$$H_e = - \sum_i p_i \log_2 p_i \quad [\text{bitů}] ,$$

kde p_i je pravděpodobnost i -tého symbolu ve zprávě.

Entropie, dva příklady

Nechť jsou ve zprávě jen dva znaky a, b.

Příklad 1 $p(a) = p(b) = \frac{1}{2}$

$$H = - \left(\frac{1}{2} \log_2 \frac{1}{2} + \frac{1}{2} \log_2 \frac{1}{2} \right) = \left(\frac{1}{2} \cdot 1 + \frac{1}{2} \cdot 1 \right) = 1$$

Příklad 2 $p(a) = 0,99; p(b) = 0,01$

$$\begin{aligned} H &= - (0,99 \log_2 0,99 + 0,01 \log_2 0,01) \\ &= - (0,99 \cdot (-0,0145) + 0,01 \cdot (-6,6439)) \\ &= 0,0144 + 0,0664 = 0,0808 \end{aligned}$$

Entropie pro šedotónový obraz

Nechť obraz má G jasových úrovní, $k = 0 \dots G - 1$ s pravděpodobnostmi $P(k)$.

Entropie $H_e = - \sum_k P(k) \log_2 P(k) \quad [\text{bitů}] ,$

Nechť b je nejmenší počet bitů, kterým lze reprezentovat počet kvantizačních úrovní.

Informační redundance $r = b - H_e .$

Odhad entropie z histogramu obrazu

Nechť $h(k)$, $0 \leq k \leq 2^b - 1$ a M , N jsou rozměry obrazu.

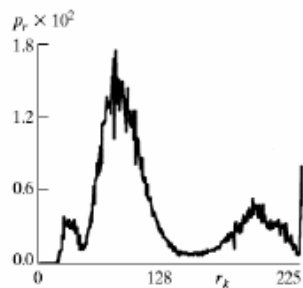
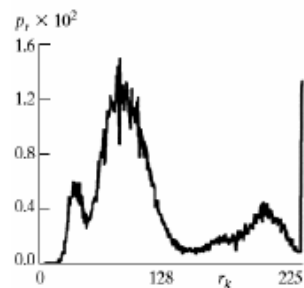
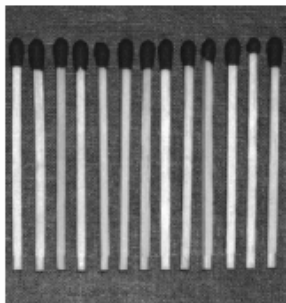
Odhad pravděpodobnosti $\hat{P} = \frac{h(k)}{M N}$.

Odhad entropie $\hat{H}_e = - \sum_k^{2^b-1} \hat{P}(k) \log_2 \hat{P}(k)$ [bitů]

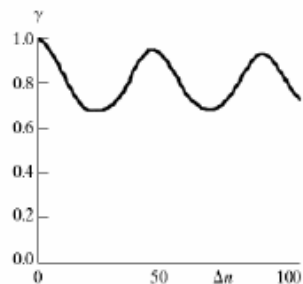
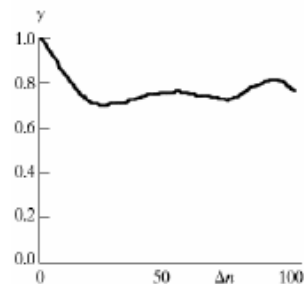
Poznámka:

Odhad entropie je příliš optimistický. Ve skutečnosti je entropie nižší, protože mezi jasy jednotlivých pixelů obrazu existují statistické závislosti (redundance).

Ilustrace redundance mezi pixely



Histogram



Normalizovaná
autokorelace

Tři definice kompresního poměru

1. Na základě **redundance** (měřené entropií) $K = \frac{b}{\hat{H}_e}$

2. Na základě **úspory paměti**

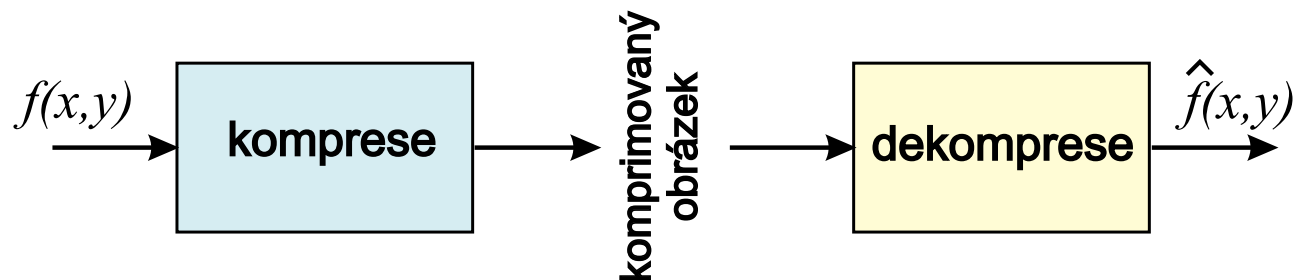
$$\kappa = \frac{n_1}{n_2} = \frac{\text{délka zprávy před kompresí}}{\text{délka zprávy po kompresi}}$$

3. **Relativní úspora paměti** $R = 1 - \frac{1}{\kappa}$

Příklad 1: $n_1 = n_2 \Rightarrow \kappa = 1, R = 0$.

Příklad 2: $n_1 : n_2 = 10 : 1 \Rightarrow \kappa = 10, R = 0,9 = 90\%$.

Ohodnocování věrnosti komprese



- ◆ Rekonstruovaný obraz (odhad) $\hat{f}(x, y) = f(x, y) + e(x, y)$, kde $f(x, y)$ je výchozí obrázek a $e(x, y)$ je chyba rekonstrukce (také residuum) po kompresi.
- ◆ Zde se zkoumá otázka: Jak blízko jsou si $f(x, y)$ a $\hat{f}(x, y)$?
- ◆ Kritéria pro ohodnocování věrnosti komprese:
 - Subjektivní: závisí na lidském pozorovateli, používané v televizi; snadná a prakticky používaná metoda je diferenční obrázek.
 - Objektivní: počítají se matematicky. Snaha nahradit subjektivní metody.

Měření ztrátovosti komprese

Nechť $u_1, \dots, u_n$ je výchozí posloupnost a $u'_1, \dots, u'_n$ ztrátově komprimovaná posloupnost

- ◆ Středně kvadratická chyba (Mean Square Error, MSE)

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n \left(u_i - u'_i \right)^2$$

- ◆ Poměr signálu k šumu (Signal to Noise Ratio, SNR)

$$\text{SNR} = 10 \log_{10} \frac{P^2}{\text{MSE}^2} [\text{dB}] ,$$

kde P je interval hodnot vstupní posloupnosti,

$$P = \max\{u_1, \dots, u_n\} - \min\{u_1, \dots, u_n\}.$$

Měření ztrátovosti komprese (2)

- ◆ Špičkový poměr signálu k šumu (Peak-Signal to Noise Ratio, PSNR)

$$\text{PSNR} = 10 \log_{10} \frac{M^2}{\text{MSE}^2},$$

kde M je maximální interval hodnot vstupní posloupnosti, tj. 256 pro osmibitový vstup a 65356 pro šestnáctibitový vstup.

V praxi se používá zejména SNR a PSNR. Výraz MSE slouží jako pomocný pro definici SNR a PSNR.

Huffmanovo kódování, z roku 1952

- ◆ Vstup: symboly s pravděpodobností jejich výskytu; zpráva.
- ◆ Výstup: optimálně zakódovaná zpráva.
- ◆ Prefixový kód, tj. žádné kódové slovo nemůže být prefixem žádného jiného kódového slova. Umožňuje dekódování, aniž by se znala délka jednotlivých slov.
- ◆ Postup: podle pravděpodobností výskytu symbolů se zdola nahoru se vytváří binární (Huffmanův) strom. Tento strom potom slouží ke generování zakódované zprávy.
- ◆ Celočíselný počet bitů na symbol.
- ◆ Nechť b je průměrný počet bitů na symbol. Potom průměrná délka kódového slova je L ,

$$H(b) \leq L \leq H(b) + 1$$

Huffmanovo kódování, příklad

Vstup: zprávy obsahují osm symbolů s pravděpodobnostmi výskytu

s_0	s_1	s_2	s_3	s_4	s_5	s_6	s_7
0.12	0.26	0.3	0.15	0.1	0.03	0.02	0.02

$$s_2 = 2 \quad \underline{0.3}$$

$$s_1 = 1 \quad \underline{0.26}$$

$$s_3 = 3 \quad \underline{0.15}$$

$$s_0 = 0 \quad \underline{0.12}$$

$$s_4 = 4 \quad \underline{0.1}$$

$$s_5 = 5 \quad \underline{0.03}$$

$$s_6 = 6 \quad \underline{0.02}$$

$$s_7 = 7 \quad \underline{0.02}$$

Huffmanovo kódování, příklad

Vstup: zprávy obsahují osm symbolů s pravděpodobnostmi výskytu

s_0	s_1	s_2	s_3	s_4	s_5	s_6	s_7
0.12	0.26	0.3	0.15	0.1	0.03	0.02	0.02

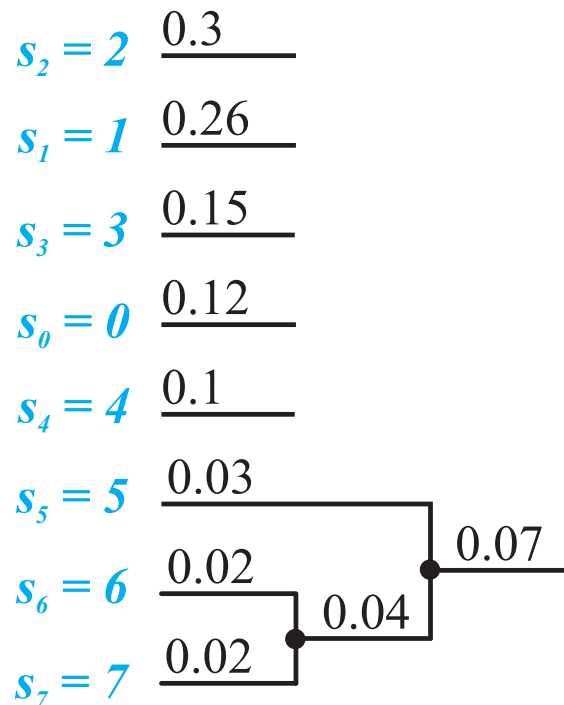
$s_2 = 2$ 0.3
 $s_1 = 1$ 0.26
 $s_3 = 3$ 0.15
 $s_0 = 0$ 0.12
 $s_4 = 4$ 0.1
 $s_5 = 5$ 0.03
 $s_6 = 6$ 0.02
 $s_7 = 7$ 0.02

0.04

Huffmanovo kódování, příklad

Vstup: zprávy obsahují osm symbolů s pravděpodobnostmi výskytu

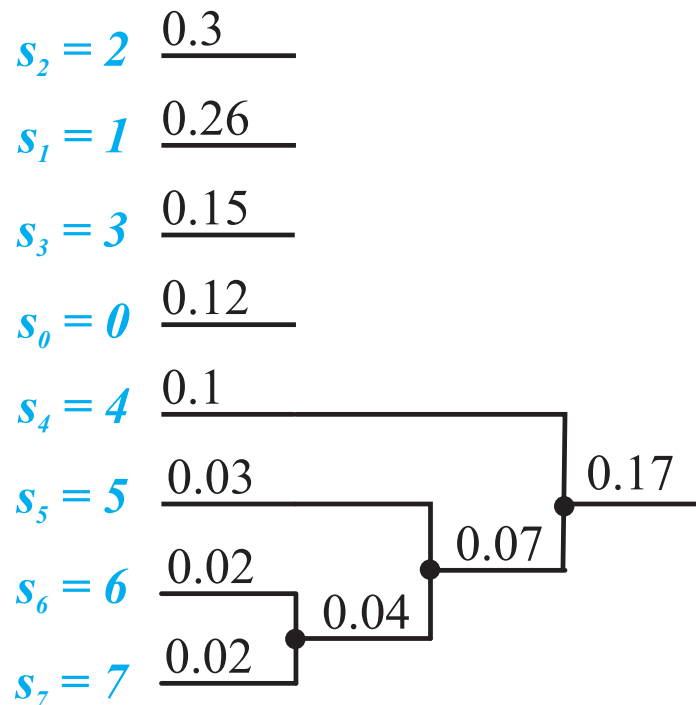
s_0	s_1	s_2	s_3	s_4	s_5	s_6	s_7
0.12	0.26	0.3	0.15	0.1	0.03	0.02	0.02



Huffmanovo kódování, příklad

Vstup: zprávy obsahují osm symbolů s pravděpodobnostmi výskytu

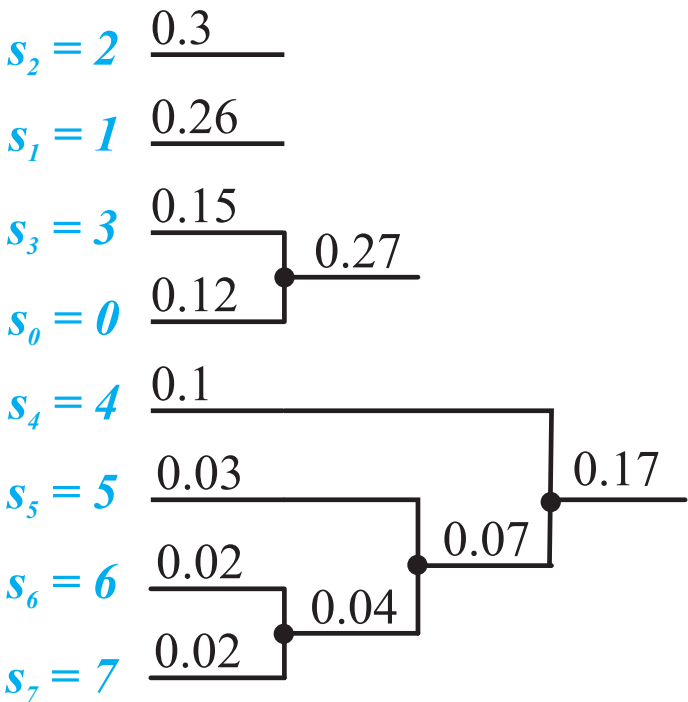
s_0	s_1	s_2	s_3	s_4	s_5	s_6	s_7
0.12	0.26	0.3	0.15	0.1	0.03	0.02	0.02



Huffmanovo kódování, příklad

Vstup: zprávy obsahují osm symbolů s pravděpodobnostmi výskytu

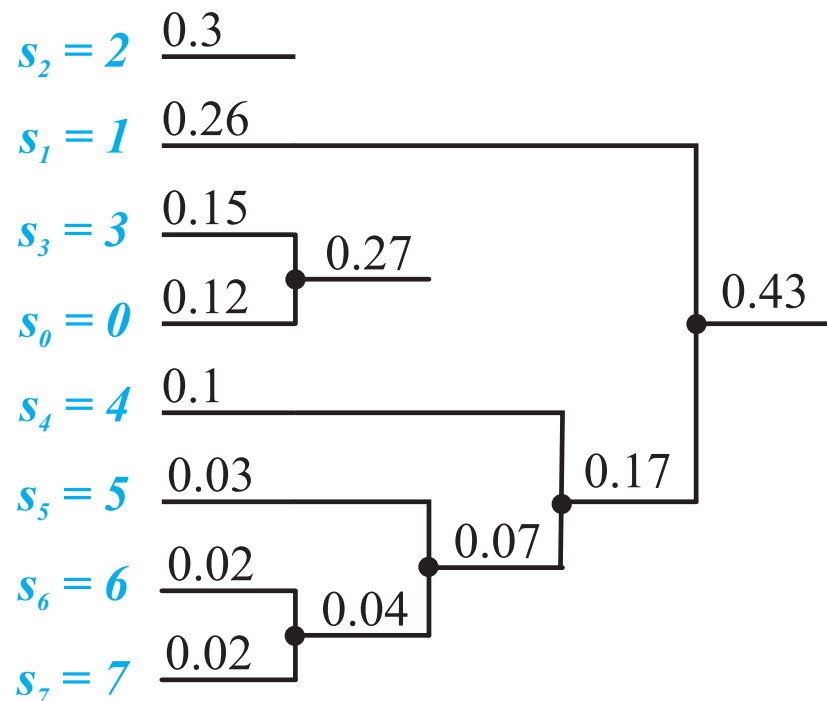
s_0	s_1	s_2	s_3	s_4	s_5	s_6	s_7
0.12	0.26	0.3	0.15	0.1	0.03	0.02	0.02



Huffmanovo kódování, příklad

Vstup: zprávy obsahují osm symbolů s pravděpodobnostmi výskytu

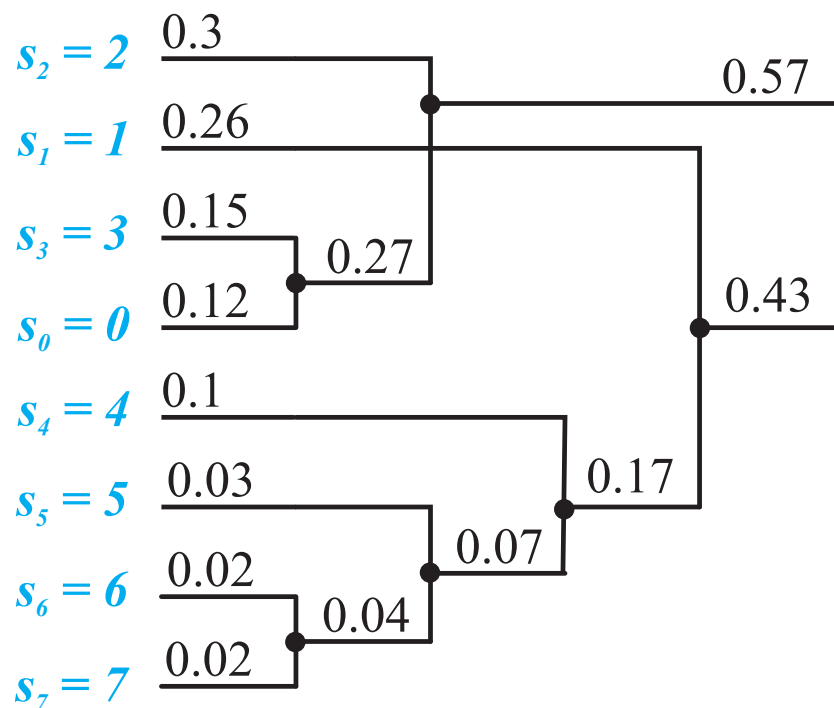
s_0	s_1	s_2	s_3	s_4	s_5	s_6	s_7
0.12	0.26	0.3	0.15	0.1	0.03	0.02	0.02



Huffmanovo kódování, příklad

Vstup: zprávy obsahují osm symbolů s pravděpodobnostmi výskytu

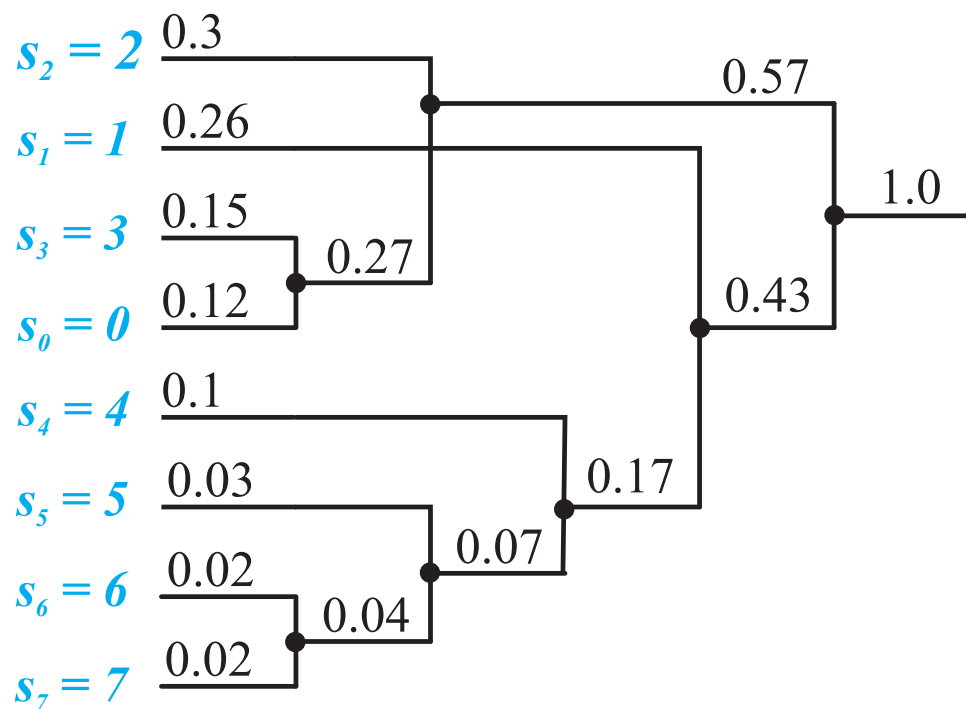
s_0	s_1	s_2	s_3	s_4	s_5	s_6	s_7
0.12	0.26	0.3	0.15	0.1	0.03	0.02	0.02



Huffmanovo kódování, příklad

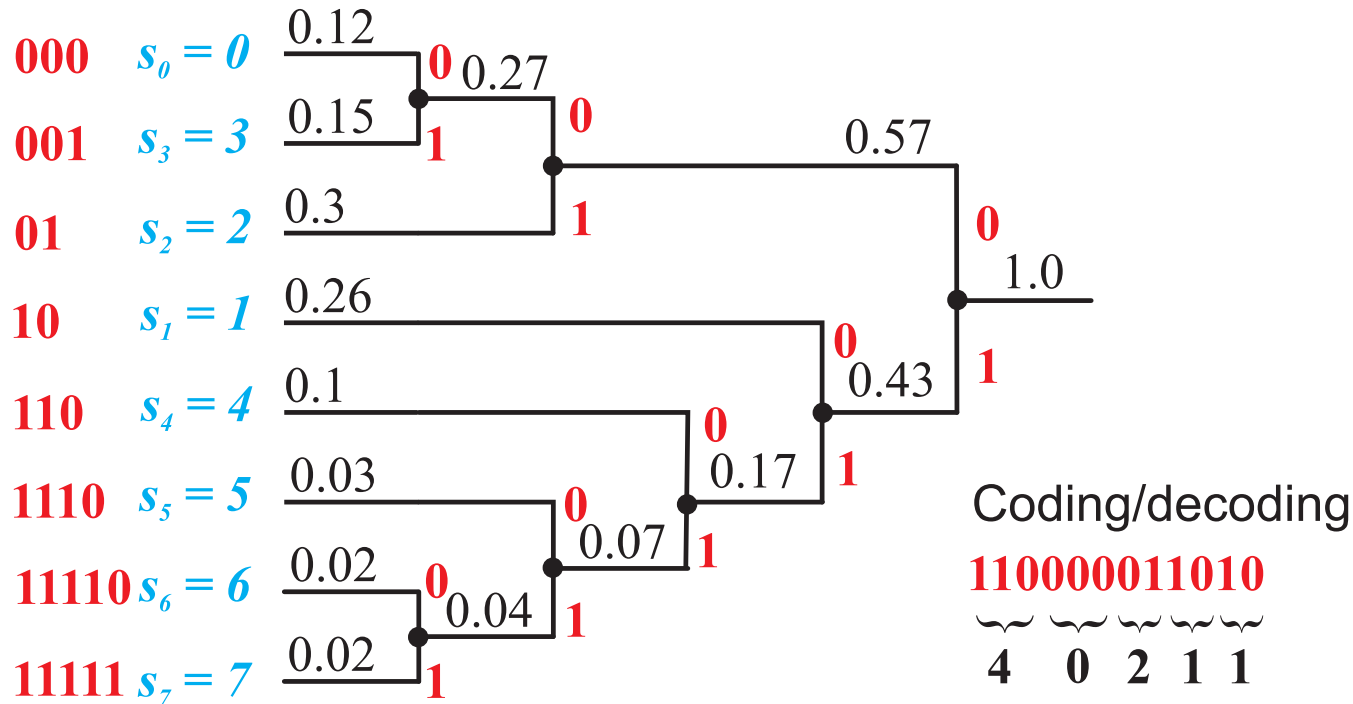
Vstup: zprávy obsahují osm symbolů s pravděpodobnostmi výskytu

s_0	s_1	s_2	s_3	s_4	s_5	s_6	s_7
0.12	0.26	0.3	0.15	0.1	0.03	0.02	0.02



Huffmanovo kódování, příklad přeuspořádání stromu

- ◆ Důvod: aby se větve stromu nekřížily.
- ◆ Kódování: u větvení stromu 0, 1.



Komprese segmentovaných (interpretovaných) obrazových dat



37/59

Tytéž metody se používají pro binární obrazy, protože je lze chápat jako výsledek segmentace. Jedničky v obraze odpovídají objektům a nuly pozadí (nebo opačně).

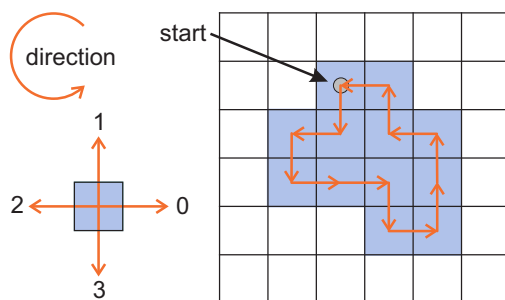
Rozdělení metod:

- ◆ **Reprezentace hranice oblasti řetězovým kódem**, zvláštní případ polygonální aproximace hranice (bezeztrátová komprese).
- ◆ **Aproximace hranice oblasti polygonální křivkou**, také nazýváno vektorizace hranice nebo křivky (ztrátová komprese).
- ◆ **Kódování oblasti úseky řádků** segmentovaných oblastí (bezeztrátová komprese).
- ◆ **Koutková komprese** (bezeztrátová), dovoluje realizovat množinové operace a několik dalších přímo v komprimovaném tvaru. Navrhl M.I. Schlesinger v 1986, V. Hlaváč má o metodě samostatnou přednášku.

Řetězový kód hranice oblasti

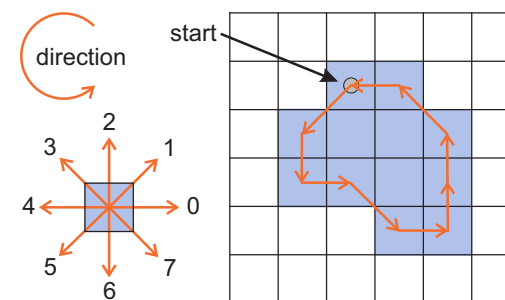
- ◆ Řetězový kód (H. Freeman 1961) je zvláštní případ náhrady hranice oblasti polygonem. Jednotlivé úsečky odpovídající 4, 8, 6-sousedům mají délku 1.
- ◆ Je zadán počáteční bod hranice, např. levý horní pixel.
- ◆ Předpokládá se směr obcházení hranice proti směru hodinových ručiček.
- ◆ Rychlá implementace: vyjádří se 3×3 okolí okamžitého pixelu a řetězový kód se najde ve vyhledávací tabulce.
- ◆ Nevýhoda: řetězový kód závisí na zvoleném počátečním bodě.

4-okolí



Řetězový kód: 3 2 3 0 0 3 0 1 1 2 1 2

8-okolí

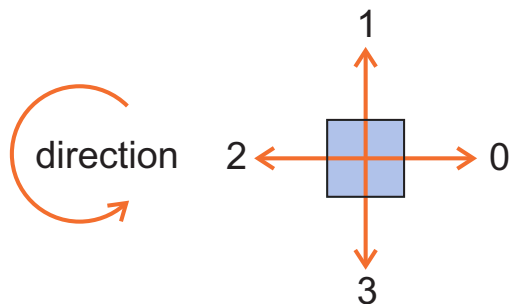


Řetězový kód: 5 6 0 7 0 2 2 3 4

Derivace dd řetězového kódu

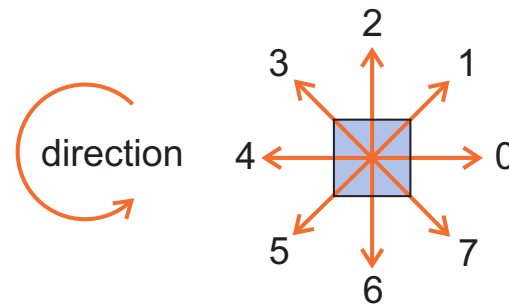
- Derivace dd (také první difference) řetězového kódu je invariantní vůči pootočení o 90° pro 4-okolí nebo o 45° pro 8-okolí.
- Derivace $dd =$ počtu “kroků” pootočení proti směru hodinových ručiček pro natočení z předchozího směru d_{old} do nového směru d_{new} .

4-okolí



if $d_{\text{new}} \geq d_{\text{old}}$ then $dd = d_{\text{new}} - d_{\text{old}}$
 if $d_{\text{new}} < d_{\text{old}}$ then $dd = 4 + d_{\text{new}} - d_{\text{old}}$

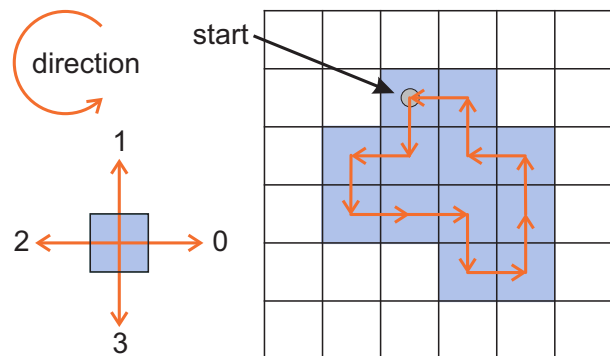
8-okolí



if $d_{\text{new}} \geq d_{\text{old}}$ then $dd = d_{\text{new}} - d_{\text{old}}$
 if $d_{\text{new}} < d_{\text{old}}$ then $dd = 8 + d_{\text{new}} - d_{\text{old}}$

Derivace řetězového kódu *dd*, příklad

4-okolí

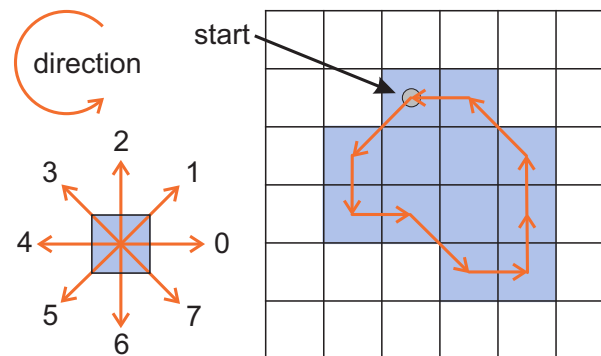


if $d_{\text{new}} \geq d_{\text{old}}$ then $dd = d_{\text{new}} - d_{\text{old}}$
 if $d_{\text{new}} < d_{\text{old}}$ then $dd = 4 + d_{\text{new}} - d_{\text{old}}$

Řetězový kód: 3 2 3 0 0 3 0 1 1 2 1 2

Derivace *dd*: 3 1 1 0 3 1 1 0 1 3 1 1

8-okolí



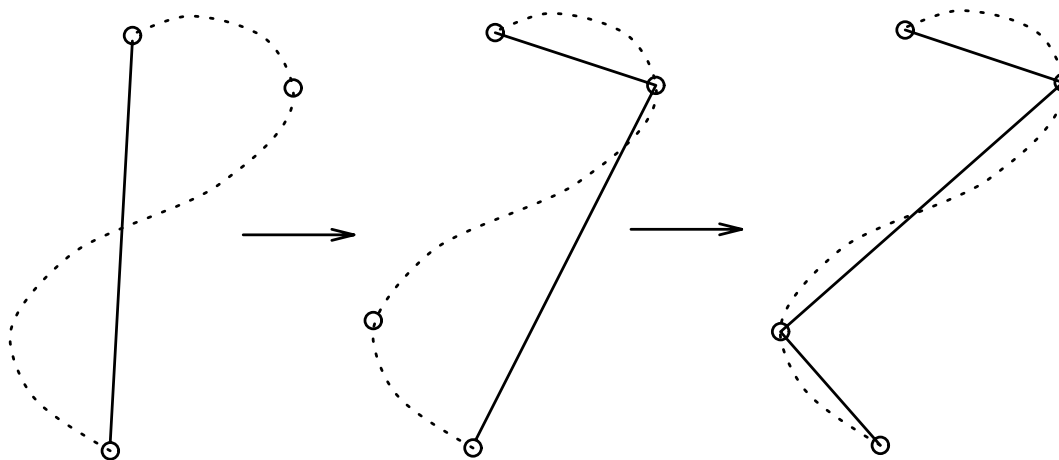
if $d_{\text{new}} \geq d_{\text{old}}$ then $dd = d_{\text{new}} - d_{\text{old}}$
 if $d_{\text{new}} < d_{\text{old}}$ then $dd = 8 + d_{\text{new}} - d_{\text{old}}$

Řetězový kód: 5 6 0 7 0 2 2 3 4

Derivace *dd*: 1 2 7 1 2 0 1 1 1

Aproximace hranice oblasti polygonální křivkou

Ramer (1972), Douglas-Peuckerův (1973) rekurzivní algoritmus



Urs Ramer: An iterative procedure for the polygonal approximation of plane curves. In Computer Graphics and Image Processing. Volume 1, Issue 3, pp. 244-256, 1972.

David Douglas, Thomas Peucker, Algorithms for the reduction of the number of points required to represent a digitized line or its caricature, The Canadian Cartographer 10(2), 112-122 (1973)

Kódování oblastí úseky řádků

- ◆ Angl. Run Length Encoding, RLE. Kódem je seznam seznamů.
- ◆ Vnější seznam obsahuje několik vnitřních seznamů. Každý z nich popisuje jeden řádek obrazu obsahující černé pixely.
- ◆ Každý vnitřní seznam popisuje jeden řádek, ale jen ty obsahující černé pixely. První prvek v seznamu je číslo řádku (v našem příkladě je **modré**).
- ◆ Zbývající prvky vnitřního seznamu jsou dvojice čísel. První číslo dvojice je číslo sloupce, v němž začíná spojitý úsek pixelů. Druhé číslo ve dvojici je číslo sloupce, v němž spojitý úsek končí.
- ◆ RLE používá FAX (norma CCITT Group 3).

Příklad:

	0	1	2	3	4	5	6
0							
1		■			■		
2		■	■	■	■		
3							
4							
5			■	■		■	
6							

Kódování RLE:

((**1**1144)(**2**14)(**5**2355))

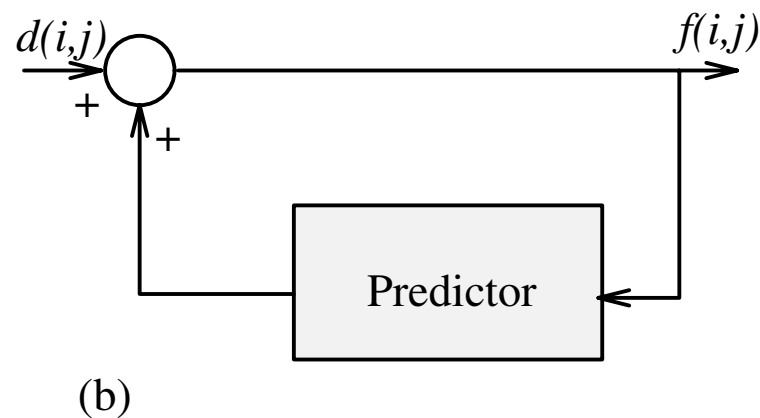
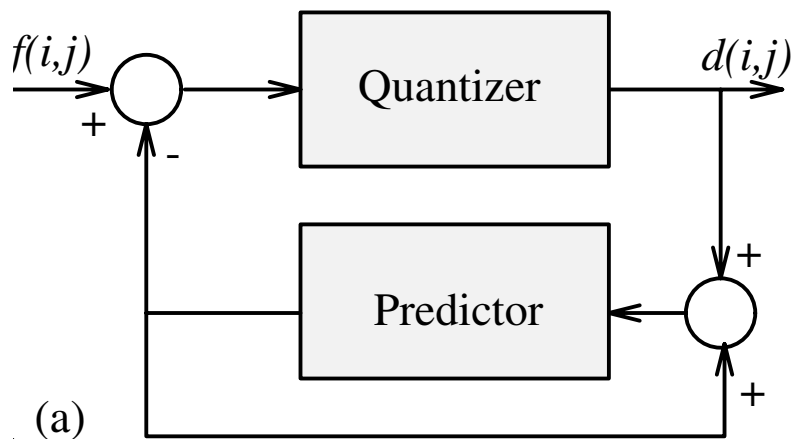
Ztrátová komprese, rozdělení přístupů

Tři hlavní přístupy ke ztrátové kompresi obrazů:

1. **Komprese dat v prostoru obrazů**, tj. odstranění mnoha pixelů v obraze a vyplňování vzniklých děr pomocí doplňování **angl. inpainting**.
2. Použití **prediktoru**, který aproximuje hodnotu pixelu z několika "minulých" vzorků pixelů. Metoda bude bezztrátová, když se s obrazem ukládá/přenáší celé reziduum $\hat{e}$. Když se celé reziduum neuchovává nebo nepřenáší. Komprese je ztrátová. Příklad: Digitální pulsně kódová modulace.
3. **Komprese dat v po transformaci**, např. diskrétní kosínová transformace nebo vlnková (wavelet) transformace. Data zbývající po kompresi se použijí pro rekonstrukci výchozího obrazu.

Prediktivní komprese – myšlenka

- ◆ Myšlenkou je najít matematický model dovolující predikovat hodnotu pixelu na základě několika pixelů v malém okolí.
- ◆ Ukládán/přenášen je pouze rozdíl mezi skutečnou a predikovanou hodnotou (predikční chyba) pro každý pixel a parametry modelu pro celý obrázek.
- ◆ Ke kompresi dochází, protože rozdílová data mají menší statistický rozptyl než původní data.



Digitální pulsně kódová modulace (1)

- ◆ Mějme obraz $f(i, j)$. Odhadněme statistické závislosti v obraze pomocí autokorelační funkce $R(i, j, k, l) = \mathcal{E}(f(i, j) f(k, l)) = f f^\top$.
- ◆ Hledáme matematický model prediktoru $\hat{f}(i, j)$.
- ◆ Rozdíl $d(i, j) = \hat{f}(i, j) - f(i, j)$.
- ◆ Předpokládejme např. lineární prediktor 3. řádu

$$\hat{f}(i, j) = a_1 f(i, j - 1) + a_2 f(i - 1, j - 1) + a_3 f(i - 1, j) ,$$

kde a_1, a_2, a_3 jsou parametry prediktivního modelu.

$f(i, j-1)$	$f(i, j)$
$f(i-1, j-1)$	$f(i-1, j)$

Digitální pulsně kódová modulace (2)

- ♦ Jak se odhadnou parametry prediktivního modelu a_1, a_2, a_3 ?
- ♦ Vyřešením statistické optimalizační úlohy. Předpokládá se stacionární náhodný proces f s nulovou střední hodnotou.

$$e = \mathcal{E} \left([\tilde{f}(i, j) - f(i, j)]^2 \right) .$$

a prediktor třetího řádu

$$a_1 R(0, 0) + a_2 R(0, 1) + a_3 R(1, 1) = R(1, 0)$$

$$a_1 R(0, 1) + a_2 R(0, 0) + a_3 R(1, 0) = R(1, 1)$$

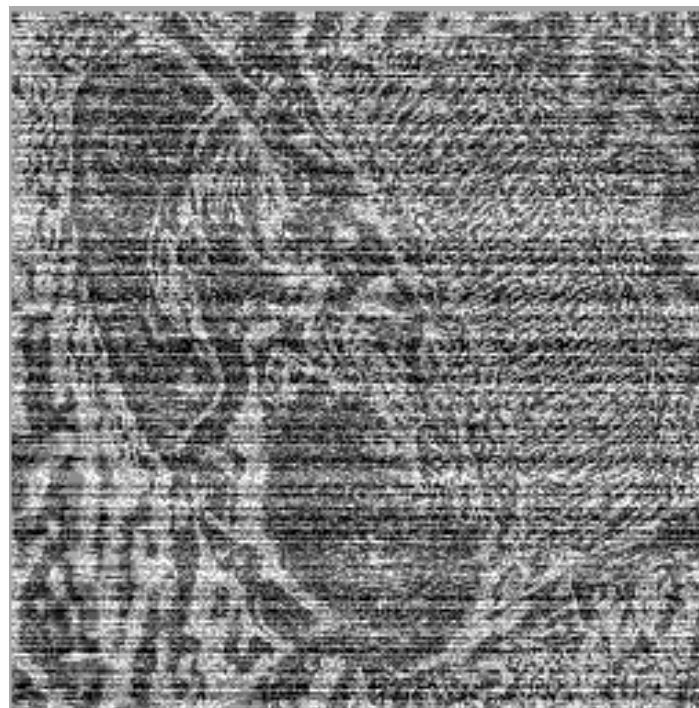
$$a_1 R(1, 1) + a_2 R(1, 0) + a_3 R(0, 0) = R(0, 1)$$

kde $R(m, n)$ je autokorelační funkce speciálního tvaru $R(\alpha, \beta) = R(0, 0) \exp(-c_1 \alpha - c_2 \beta)$.

DPCM – Příklad, $K = 3.8$



Po rekonstrukci $K = 3.8$.

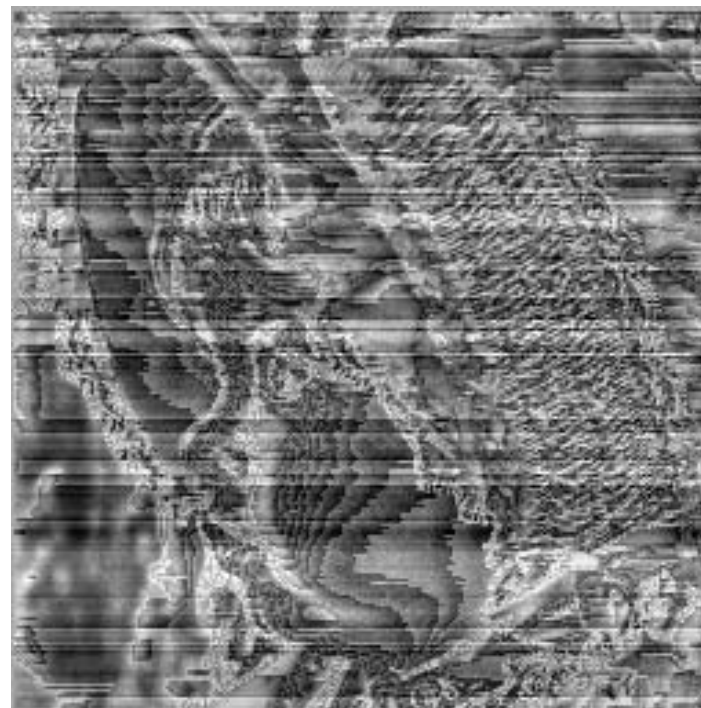


Rozdílový snímek.

DPCM – Příklad, $K = 6.2$



Po rekonstrukci $K = 6.2$.



Rozdílový snímek.

JPEG, úvod

- ◆ Joint Photographic Expert Group. Standartizováno v roce 1992.
- ◆ Používá se na šedotónové i barevné obrázky. Barevné se nejdříve převedou z barevného prostoru RGB do prostoru YUV, kde lze matice U , V reprezentovat v polovičním rozlišení než matici Y ($\approx$ intenzita).
- ◆ Existuje jako ztrátová i bezetrátová komprese.
- ◆ První generace (.jpg) z 1992 používá DCT (diskrétní kosínovou transformaci) pro odstranění redundance a irelevance. Pro optimální kódování se použije převod koeficientů DCT do 1D vektoru, kódování úseky řádků a symboly kóduje Huffmanovým kódováním.
- ◆ Druhá generace JPEG2000 (.jp2) z roku 2000 odstraňuje redundanci a irelevanci pomocí vlnkové transformace. Potom kóduje v jednotlivých bitových rovinách a symboly kóduje aritmetickým kódováním.

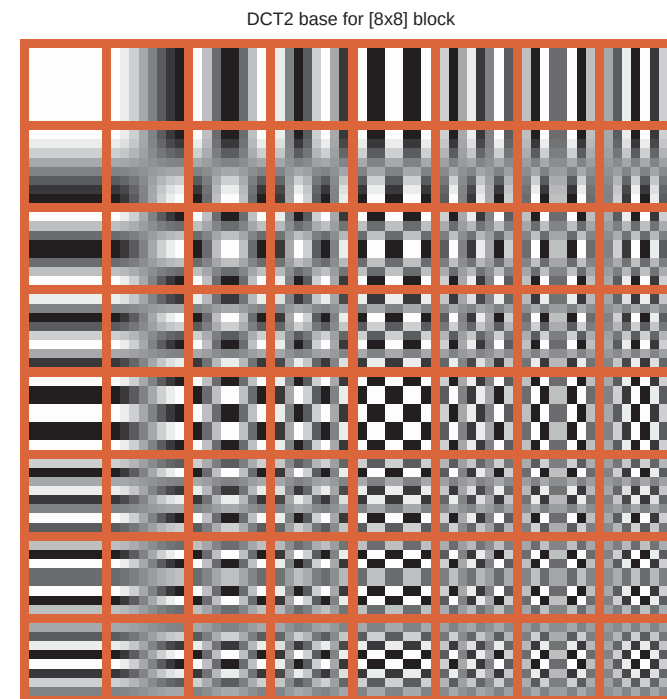
Proč bylo DCT zvoleno pro JPEG?

- ◆ DCT je implicitně periodická. Nejsou takové problémy s nespojitostí.
- ◆ DCT dobře aproximuje PCA (metoda hlavních směrů, Principal Component Analysis, Karhunen-Loeve rozvoj), která je optimální z hlediska středněkvadratické chyby (energie).
- ◆ Na rozdíl od PCA, DCT má pevně zvolené báze funkce.
- ◆ Aby se ušetřil výpočetní čas, je obraz rozdělen na bloky 8×8 , které jsou komprimovány nezávisle na sobě.

DCT, báзовé funkce

Pevně zvolených 64 báзовých funkcí.

- ◆ Každý blok obrazu 8×8 lze vyjádřit jako lineární kombinaci báзовých funkcí.
- ◆ Při DCT kompresi se najde pro každý blok 64 vah lineární kombinace.
- ◆ Váhy jsou prahovány. Velikost prahu ovlivňuje míru komprese, tj. volí se požadovaná irrelevance.



Příklad, kameraman

image block



image intensities

1	185	187	184	183	189	186	185	186
2	185	184	186	190	187	186	189	191
3	186	187	187	188	190	185	189	191
4	186	189	189	189	193	193	193	195
5	185	190	188	193	199	198	189	184
6	191	187	162	156	116	30	15	14
7	168	102	49	22	15	11	10	10
8	25	19	19	26	17	11	10	10
	1	2	3	4	5	6	7	8

Příklad, kameraman, DCT

image intensities

1	185	187	184	183	189	186	185	186
2	185	184	186	190	187	186	189	191
3	186	187	187	188	190	185	189	191
4	186	189	189	189	193	193	193	195
5	185	190	188	193	199	198	189	184
6	191	187	162	156	116	30	15	14
7	168	102	49	22	15	11	10	10
8	25	19	19	26	17	11	10	10
	1	2	3	4	5	6	7	8

coefficients of the DCT2

1	1117	114	10	7	19	-2	-7	2
2	459	-119	-20	-11	-16	-4	3	0
3	-267	-3	24	8	1	6	4	-1
4	50	107	-9	-1	11	-6	-7	3
5	52	-111	-22	-2	-16	-2	5	-3
6	-38	39	46	19	2	0	4	3
7	-17	39	-46	-26	8	-5	-10	2
8	30	-46	28	22	-9	2	7	-1
	1	2	3	4	5	6	7	8

Příklad, kameraman, 100 % a 50 %

100 % of most significant DCT2 coeffs



50 % of most significant DCT2 coeffs



Příklad, kameraman, 20 % a 5 %

20 % of most significant DCT2 coeffs



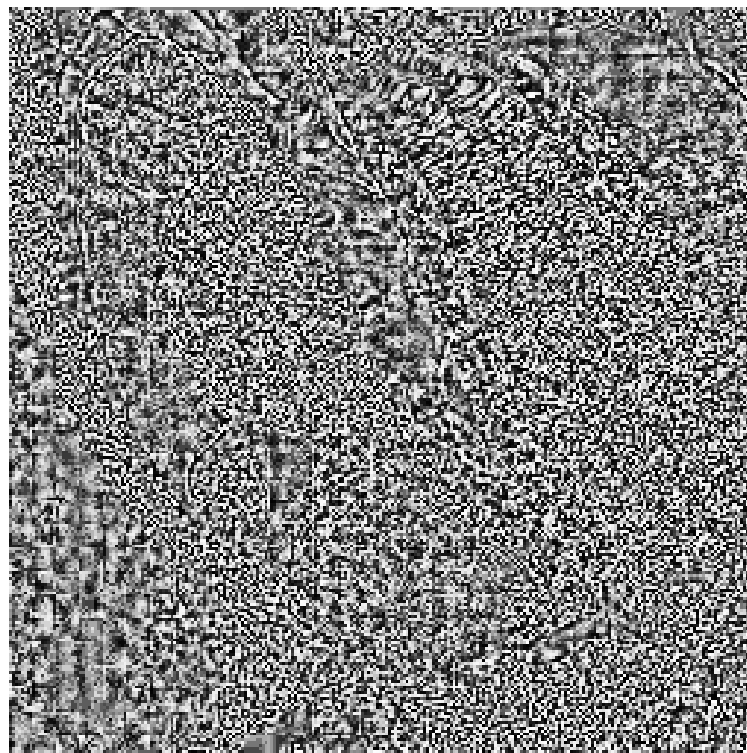
5 % of most significant DCT2 coeffs



JPEG – příklad, $K = 3.8$



Po rekonstrukci $K = 3.8$.

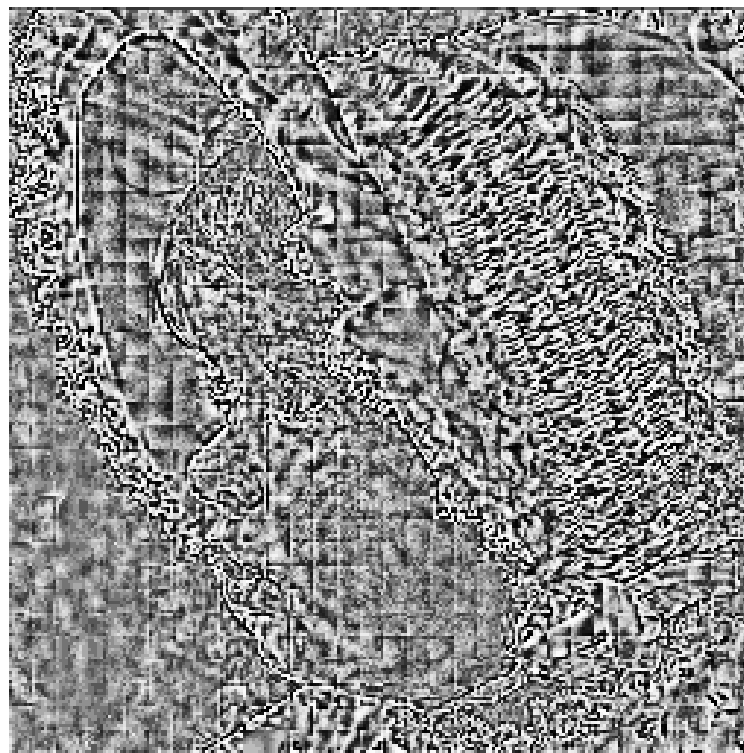


Rozdílový snímek.

JPEG – příklad, $K = 4.2$



Po rekonstrukci $K = 4.2$.

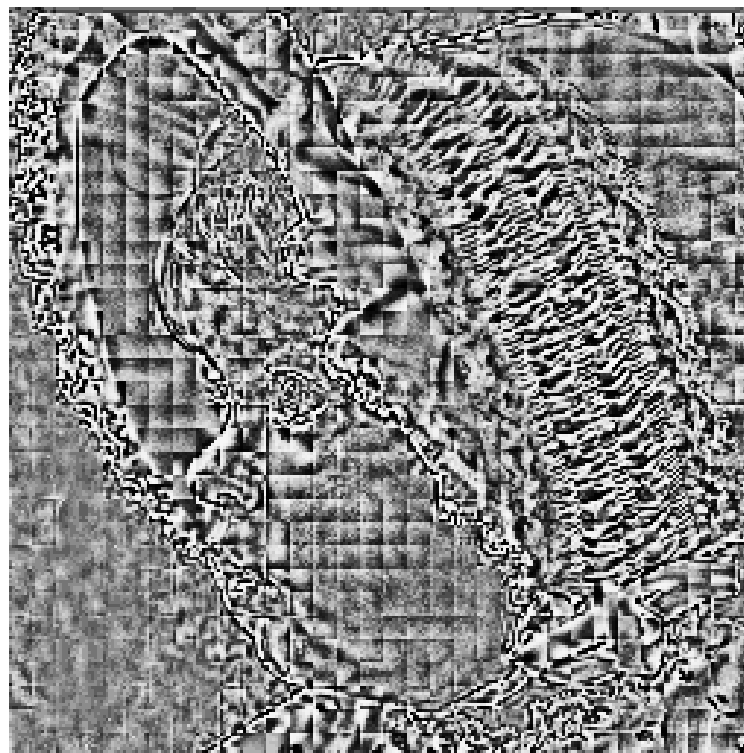


Rozdílový snímek.

JPEG – Příklad, $K = 5.6$



Po rekonstrukci $K = 5.6$.

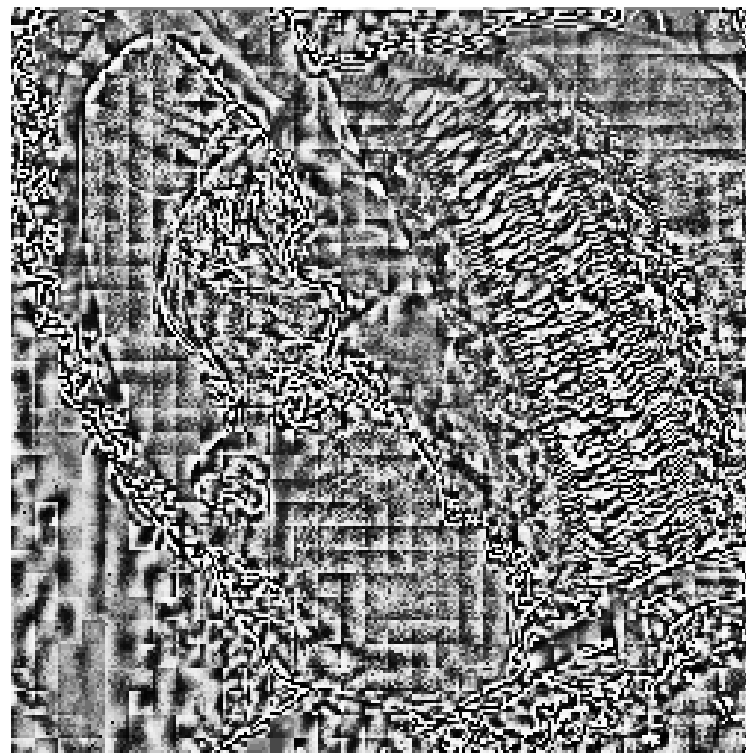


Rozdílový snímek.

JPEG – příklad, $K = 10.2$



Po rekonstrukci $K = 10.2$.



Rozdílový snímek.