

Přednáška kurzu BZVS

Zobrazovací média, formáty pro zpracování a archivaci digitálního obrazu a videa

Ing. P. Petyovský (email: petyovsky@feec.vutbr.cz) ,
kancelář SD3.152, tel. 6434, Technická 12, VUT v Brně

Reprezentace grafiky v paměti počítače

Kompresce rastrového obrazu

Princip komprese RLE

Princip komprese LZ77 / LZW

Princip komprese pomocí Huffmanova kódování

Princip komprese pomocí aritmetického kódování

Princip komprese DCT

Příklady rastrových formátů

Popis formátu PCX

Popis formátu GIF

Popis formátu PNG

Popis formátu JPEG

Popis formátu TIFF

Kompresce digitálního videa

Standardy digitálního videa

MPEG-1

MPEG-2

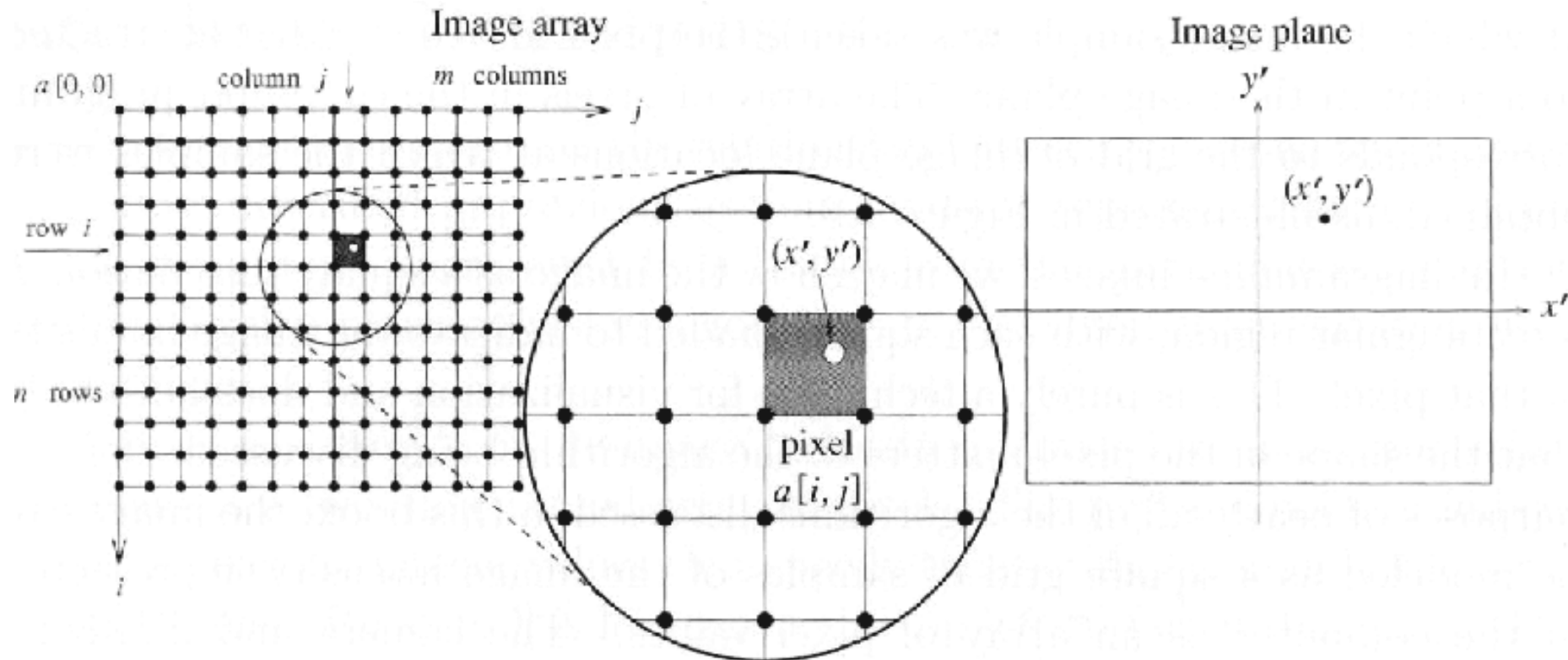
MPEG-4

Příklady formátů digitálního videa

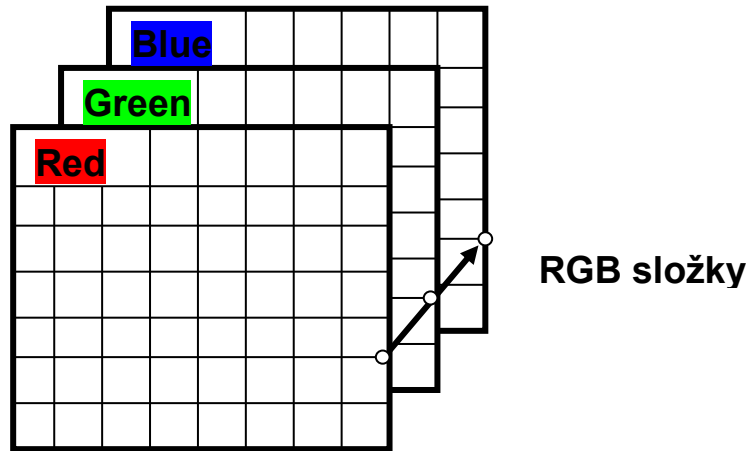
DivX, XVID, MJPG, ASF, QuickTime, DVD, VCD, SVCD

Reprezentace grafiky v paměti počítače

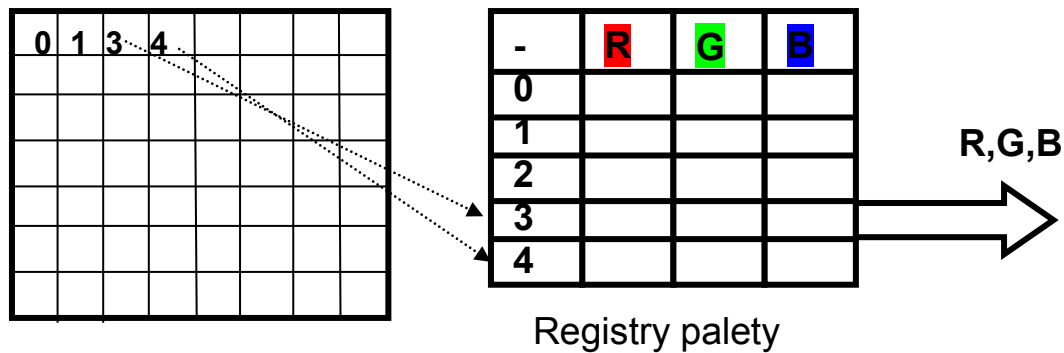
Rastrové zobrazení:



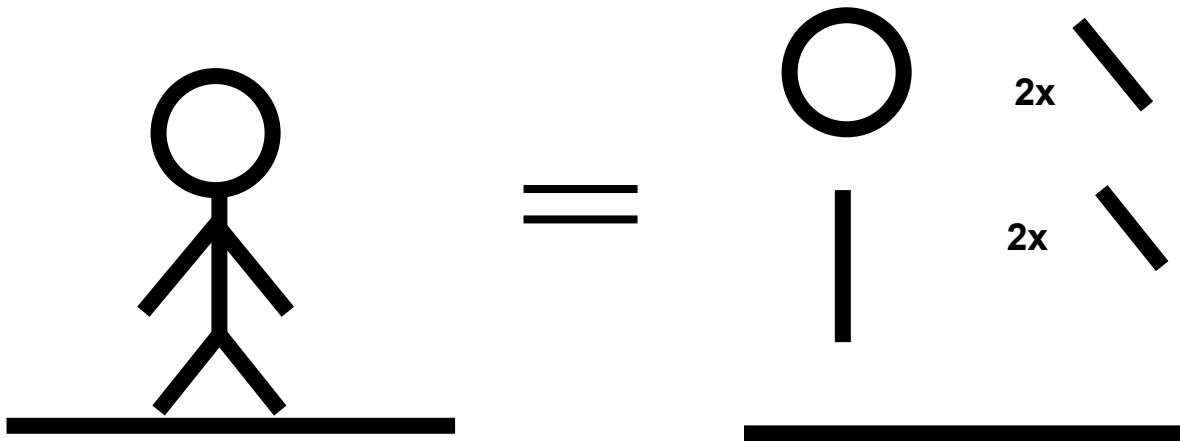
Uspořádání barevného rastrového obrazu v paměti počítače do tzv. bitplánů (Bit Planes) nejčastěji dle RGB.



Uspořádání barevného rastrového obrazu v paměti počítače s využitím paletových registrů.



Vektorové zobrazení:



Vektorový popis obsahuje pouze parametry základních grafických primitiv (křivek, přímek, kružnic, elips atd.) popsanych pomocí strukturovaného zápisu.

```
<svg xmlns="http://www.w3.org/2000/svg"
  xmlns:xlink="http://www.w3.org/1999/xlink">
  <circle cx="100" cy="100" r="30" style="stroke:#000000;stroke-width:7px;fill:#ffffff;fill-
    opacity:0.0;"/>
  <line x1="100" y1="130" x2="100" y2="200" style="stroke:#000000;stroke-width:7px;"/>
  ....
</svg>
```

Mezi které patří:

- umístění, zkosení, rotace
- parametry grafických primitiv (umístění, poloměr, poloosy atd.)
- barva, průhlednost
- tloušťka čar

Úvod do komprese dat

Definice

„Komprese dat (také komprimace dat) je speciální postup při ukládání nebo transportu dat. Úkolem komprese dat je zmenšit datový tok nebo zmenšit potřebu zdrojů při ukládání informací.“

„Komprimace dat je postup snažící se postihnout charakter dat a eliminovat jejich datovou redundanci.“

„Redundance obecně označuje takový stav nebo vlastnost, kdy je použito větší množství prvků než je obvyklé nebo nezbytně nutné.“

zdroj wikipedia

Pojem kompresní metoda a kompresní princip (kódování)

Existuje mnoho různých kompresních metod využívajících stejných kompresních (kódovacích) principů. Často je v literatuře uvedeno, že vznikla nová kompresní metoda (která, ale většinou využívá známé kompresní principy). Jen velmi zřídka, vznikne nový kompresní princip (využití markovovských modelů u metody PPMd/LZMA).

Kompresní principy se nezabývají implementačními detaily. Tuto část problému řeší až kompresní metoda, která je velmi často spojena s nějakým datovým formátem např.: .zip, .arj, .rar, apod.

Rozdělení kompresních principů

Základní dělení:

- Ztrátové - zaokrouhlení výsledků z DCT (u JPEG), fraktální komprese, mezisnímkové kódování (u MPEG)
- Neztrátové
 - Primární - RLE, LZW, LZ77
 - Sekundární - Huffmanovo, Aritmetické kódování

Dělení dle výpočetní náročnosti:

- Symetrické
- Asymetrické

Dělení dle počtu průchodů:

- Jedno-průchodové
- Více-průchodové

Kódování RLE

(bezztrátové, primární, jedno-průchodové, takřka symetrické)

Jednoduchá a pro velkou třídu dat i efektivní metoda, vychází z předpokladu, že ve vstupních datech, se opakují „sousední“ hodnoty. Tehdy do souboru zapíšeme informaci o počtu opakujících se hodnot a poté hodnotu samotnou.

Odtud také název metody: Run-Length Encoding. Metoda pro kódování délky runu (skupiny dat majících stejnou hodnotu).

Příklad:

Vstupní data:

"AAAhooooj"

Výstupní data:

"<3A>h<4o>j"

Pozn.:

V praxi je navíc nutné odlišit běžná data od odkazů (počet, opakovaná hodnota), abychom při dekompresi byli schopni data správně dekodovat.

Kompresní metoda RLE

Aplikovaná v datovém formátu `.pcx`.

Při kódování pixelů v obraze definovaných jedním bytem rozlišíme příznak opakování hodnotou nejvyššího bitu:

1	čítač	hodnota	hodnota se opakuje $(1+\text{čítač})$ krát
0	hodnota		Výstup 7bitové neopakující se hodnoty
10000000	hodnota		Výstup pokud <i>hodnota</i> obsahuje MSB=1

Popis algoritmu:

1. Vynuluj *čítač*
2. Do proměnné *BarvaA* přiřaď hodnotu pixelu.
3. Dokud jsou na vstupu pixely, opakuj:
 - 3.1 Do proměnné *BarvaB* přiřaď hodnotu dalšího pixelu
 - 3.2 Pokud $\text{BarvaA} == \text{BarvaB}$ a zároveň hodnota *čítače* nepřesáhla maximální hranici (danou velikostí bytu, slova), potom $\text{čítač} = \text{čítač} + 1$
 Jinak:
 - Zapiš *čítač* a *BarvaA* na výstup
 - Do proměnné *BarvaA* přiřaď hodnotu *BarvaB*
 - Vynuluj *čítač*
4. Zapiš *čítač* a *BarvaA* na výstup

V případě, že kódovaný obrázek obsahuje neopakující se hodnoty v sousedních pixelech, dochází u metody RLE k **záporné kompresi**. Kompresi RLE je vhodná pro obrázky kreslené "od ruky" nebo tzv. "Cartoons" – ilustrace s velkými stejnobarevnými plochami.

Kódování LZ77

(neztrátové, primární, jedno-průchodové, asymetrické)

Původní myšlenka pochází z roku 1977 A. Lempel, J. Ziv. Metoda LZ77 bývá někdy v literatuře také označovaná jako LZSS (i když např. wikipedia označuje LZSS jako modifikaci LZ77).

Metoda se snaží nalézt ve vstupních datech opakující se sekvence symbolů a ty na výstup předat ve formě odkazu na předchozí (již odeslaná) data.

Odkazem je myšlen údaj o pozici opakující se sekvence v předchozích datech. Druhým důležitým parametrem odkazu je délka opakující se sekvence.

Příklad:

Vstupní data:

```
"the_rain_in_Spain_falls_mainly_in_the_plain"  
".....ain....."  
".....ain....."
```

Výstupní data: (V příkladu lze ušetřit ještě jeden byte, zjistěte kde?)

```
"the_rain_<3,3>Sp<9,4>falls_m<11,3>ly_<16,3><34,4>p1<15,3>"
```

V praxi je navíc nutné odlišit běžná data od odkazů, abychom při dekompresi byli schopni data správně dekódovat.

Možností je několik, běžně používanou je rozšíření symbolu o další bit, který rozhodne, zda jde o data nebo odkaz: <pozice,délka>. (Vhodné v případě následného využití některé z metod sekundární komprese.) Metoda je jedno-průchodová, což je výhodná vlastnost zvláště při kompresi velkého objemu dat na hardwarové úrovni. Metoda je silně asymetrická, výpočetní náročnost při kompresi je možné snížit použitím binárního vyhledávacího stromu.

Kódování LZW

(neztrátové, primární, jedno-průchodové, asymetrické)

Metoda LZ77 modifikována v roce 1984 T. Welchem. Vzniklá metoda LZW bývá někdy označovaná jako *dictionary based encoding (slovníkově orientované kódování)*. Metoda je patentována.

Princip vyhledání dříve se opakujících sekvencí zůstala zachována, modifikován je způsob kódování.

Metoda používá místo odkazů $\langle \text{pozice}, \text{délka} \rangle$ odkaz na položku slovníku, který je při kompresi vytvářen a který je neustále doplňován a jeho délka a počet položek tím narůstá.

Základní velikost slovníku je 512 položek a maximální velikost slovníku bývá nejčastěji 4096.

Po té je slovník smazán a metoda opět začíná se slovníkem velikosti 512 položek. Modifikací metody je systematické promazávání nejdéle nepoužitých slov/položek ve slovníku namísto smazání celého slovníku.

Kompresní princip má v případě varianty s pravidelným mazáním celého slovníku, menší kompresní poměr než LZ77.

Huffmanovo kódování

(sekundární, neztrátové, mírně asymetrické, nejčastěji dvou-průchodové)

Myšlenka kódování pochází z roku 1952 od D. Huffmana. Je založena na použití různě dlouhých bitových kódů pro symboly s různou frekvencí výskytu. V praxi používána dávno před r. 1952 (např. Morseovka). (Nejčastěji používané vstupní symboly mají nejkratší výstupní kód.)

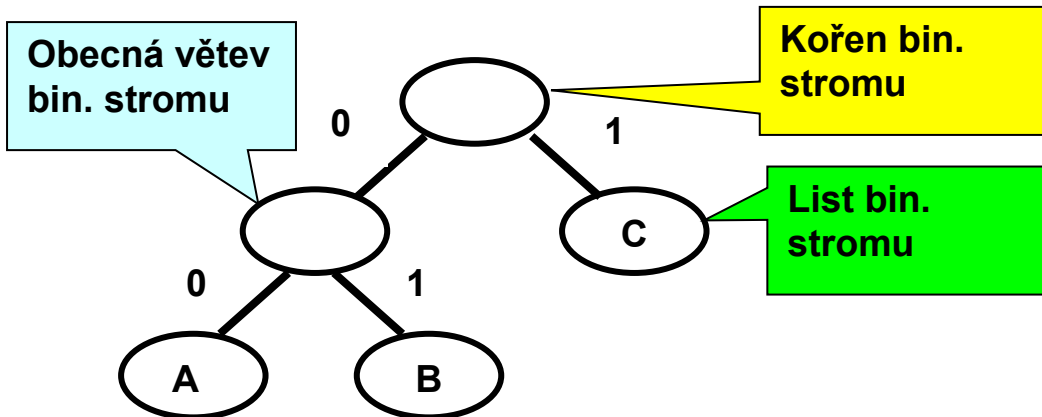
Principem je vytvoření binárního stromu na základě znalosti frekvencí výskytu jednotlivých symbolů ve vstupním souboru.

Příklad:

Vstupní data:

"CABACBBBCCCC"

Vstupní symbol	Počet výskytů
A	2
B	4
C	6



Vstupní symbol	Délka výstupního řetězce	Výstupní řetězec
A	2	00
B	2	01
C	1	1

Výstupní data (binárně):

"100010010101011111"

Pro úspěšnou dekompresi je nutné znát tvar binárního stromu, proto je nutné:

- **Strom ke komprimovaným datům připojit**
- **Strom je fixně dán a je znám dopředu.**
- **Lze využít tzv. dynamickou tvorbu binárního stromu, není nutné jej přenášet, ale menší kompresní poměr.**

Otázka:

Jaký tvar bude mít binární strom v případě těchto frekvencí výskytu vstupních symbolů?

Vstupní symbol	Počet výskytů
A	2
B	2
C	3
D	3
E	4 (7)

Aritmetické kódování

(sekundární, neztrátové, mírně asymetrické, běžně dvou-průchodové, výpočetně náročné)

Myšlenka kódování je založena na úvaze, že bit není dál nedělitelné množství informace. Pracuje s četností výskytu vstupních symbolů jako s desetinnými čísly, proto velká výpočetní náročnost. Běžně se zde pracuje s desetinnými čísly majícími stovky a více bitů.

Zatímco u Huffmanova kódování měl vždy každý konkrétní vstupní symbol přiřazen svůj konkrétní počet bitů na výstupu. Aritmetické kódování pracuje vždy se skupinou několika vstupních znaků, které kóduje jako určitý počet výstupních bitů. Tím je zajištěn ještě vyšší kompresní poměr, než je tomu u Huffmanova kódování.

Zjednodušená představa je, že bychom v případě Huffmanova kódování (pomocí binárního stromu) definovali několik různých binárních stromů, které bychom průběžně střídali.

Kompresní princip je bohužel patentován, proto je i přes výhody vysokého kompresního poměru prakticky minimálně nepoužíván.

Reprezentace desetinného čísla v binární soustavě

Příklad:

$$(5.75)_{\text{dec}} = (101,11)_{\text{bin}}$$

$$(5 \cdot 10^0 + 7 \cdot 10^{-1} + 5 \cdot 10^{-2})_{\text{dec}} = (1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2})_{\text{bin}}$$

Pozn.:

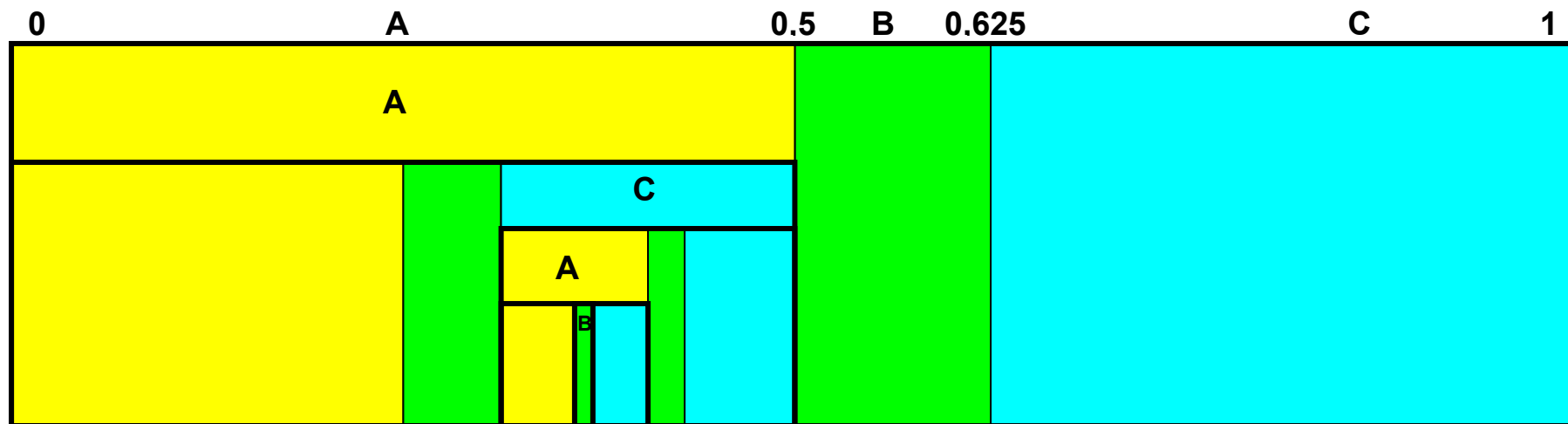
Díky nesoudělnému základu desítkové a dvojkové soustavy platí, že desetinné číslo mající v jedné soustavě vyjádření s konečným počtem cifer, může být v druhé soustavě vyjádřené desetinným číslem s opakující se periodou.

Aritmetické kódování příklad

Vstupní data:

"ACABACAC"

Vstupní symbol	Počet výskytů	Četnost výskytů
A	4	$4/8 = 0,5$
B	1	$1/8 = 0,125$
C	3	$3/8 = 0,375$



Jakékoliv reálné číslo z tohoto intervalu kóduje sekvenci vstupních dat: ACAB

Aritmetické kódování numerická podoba příkladu

Počáteční hranice intervalů pro vstupní symboly (A,B,C) jsou:

$$\left\langle \frac{0}{8}; \frac{4}{8}; \frac{5}{8}; \frac{8}{8} \right\rangle.$$

Úprava hranic intervalů po symbolu A:

$$\left(\left\langle \frac{0}{8}; \frac{4}{8}; \frac{5}{8}; \frac{8}{8} \right\rangle - \frac{0}{8} \right) \cdot \frac{1}{2} + \frac{0}{8} = \left\langle \frac{0}{16}; \frac{4}{16}; \frac{5}{16}; \frac{8}{16} \right\rangle.$$

Úprava hranic intervalů po symbolu C:

$$\left(\left\langle \frac{0}{16}; \frac{4}{16}; \frac{5}{16}; \frac{8}{16} \right\rangle - \frac{0}{16} \right) \cdot \frac{3}{8} + \frac{5}{16} = \left\langle \frac{40}{128}; \frac{52}{128}; \frac{55}{128}; \frac{64}{128} \right\rangle.$$

Úprava hranic intervalů po symbolu A:

$$\left(\left\langle \frac{40}{128}; \frac{52}{128}; \frac{55}{128}; \frac{64}{128} \right\rangle - \frac{40}{128} \right) \cdot \frac{1}{2} + \frac{40}{128} = \left\langle \frac{80}{256}; \frac{92}{256}; \frac{95}{256}; \frac{104}{256} \right\rangle.$$

Poslední symbol je B, hledáme tedy takové číslo z prostředního intervalu, které dokážeme reprezentovat co nejlépe (tj. jako co nejkratším desetinným číslem v dané soustavě):

$$\left\langle \frac{92}{256}; \frac{95}{256} \right\rangle \rightarrow \frac{92}{256} = \frac{23}{64} = \frac{1}{4} + \frac{1}{16} + \frac{1}{32} + \frac{1}{64} = \frac{1}{2^2} + \frac{1}{2^4} + \frac{1}{2^5} + \frac{1}{2^6} = 0.010111_{bin}.$$

Výsledek je tedy „010111“. Pro úspěšnou dekompresi je stejně jako u Huffmanova kódování nutné znát četnosti výskytů vstupních symbolů.

Kódování pomocí DCT (discrete cosine transform)

(primární, obecně neztrátové, symetrické, běžně jedno-průchodové, výpočetně náročné)

Diskrétní kosinová transformace je formou diskrétní Fourierovy transformace. Obrazová data jsou považována za barevné vzorky spojitých barevných signálů naměřené v diskrétní síti pixelů. Výsledkem DCT je pak nalezení sady parametrů kosinových funkcí, jejichž složením lze rekonstruovat původní obraz. Rozklad na jednotlivé složky je výsledkem dopředné transformace (FDCT), která je vlastně převodem do tzv. oblasti prostorových frekvencí. Zpětná transformace (IDCT) je potom rekonstrukcí z frekvenční oblasti prostorových frekvencí.

Postup:

Obrazová data jsou rozdělena do čtverců 8x8. Každý čtverec je podroben samostatně DCT. Výsledkem je 8x8 koeficientů $F(u,v)$ získaných výpočtem z hodnot pixelů $f(x,y)$ podle vztahu:

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

pro tzv. DC člen, (tj.: $u,v=0$):

$$C(u), C(v) = \frac{1}{\sqrt{2}} ,$$

všude jinde AC členy platí, že:

$$C(u), C(v) = 1 .$$

Zpětná (inverzní) transformace má tvar:

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

Pozn.:

Hodnoty získané dopřednou transformací celočíselných vstupních dat jsou neceločíselné!!! (bude důležité pro JPEG).

Vlastnosti formátu „.PCX“

Autor: Zsoft PaintBrush

Komprese: RLE nebo bez komprese.

Jeden z nejstarších formátů, stále značně rozšířen pro svou jednoduchost. Umožňuje ukládat obrázky v různých barevných hloubkách od černobílých až po 8 nebo 24 bitů/pixel. Bitplánové uspořádání.

Vlastnosti formátu „.TIFF“

Autor: CCITT

Komprese: Všechny běžně používané komprese.

Velice univerzální, velmi složitý formát. Dnešní verze TIFF 6.0 - supplement 2. Formát využívaný pro profesionální práci s rastrovými snímky. Pochází z roku 1984, poslední aktualizace 2002.

Ačkoliv je jasně a dobře dokumentovaný, mezi běžnými uživateli panuje nedůvěra ke kompatibilitě souboru TIFF a to i případě, kdy se jedná o shodné verze. Tento fakt je způsoben tím, že škála možností zápisu formátu je tak široká, že jen málokterý program dokáže interpretovat všechny povolené varianty. Tvrdí se, že: *”Každý umí zapsat svůj TIFF, ale málokdo umí číst jiný, než právě ten svůj”*.

Vlastnosti formátu „.GIF“

Autor: CompuServe

Komprese: LZW

Formát z doby prvních VGA videokaret, specializovaný na obrázky s paletou, případně s jedním bitem na pixel. Původně určen k přenosu obrázků po telefonních linkách. Existuje ve dvou verzích *G87a*, *G89a* (vylepšení pro WWW).

Základní charakteristiky:

- Možnost tvořit animované sekvence (bannery).
- Možnost prokládání řádků.
- Definice průhledné barvy.
- Uložení dalších doplňkových údajů.

Omezení na max. počet 256 barev je však kritickým faktorem pro jeho použití v budoucnu (rozšíření na více jak 256 barev sice existuje, není však většinou podporováno), nahrazen formátem .PNG.

Vlastnosti formátu „.PNG“

Autor: ISO (Portable Network Graphics), Otevřený standard

Komprese: LZ77

Všechny bitové hloubky. Zavádí pojem předzpracování pixelu před samotnou neztrátovou kompresí. Má všechny vlastnosti formátu „.GIF“ a mnohé další (např. dokáže ukládat obrázky v barevném modelu RGBA). V budoucnu pravděpodobně zcela nahradí „.GIF“. Varianta APNG a MNG podporuje i animace jako .GIF.

Vlastnosti formátu „JPEG“

Autor: Joint Photographic Experts Group

Komprese: DCT, Kvantizace (nevratná ztráta informace), Huffmanova / Aritmetická

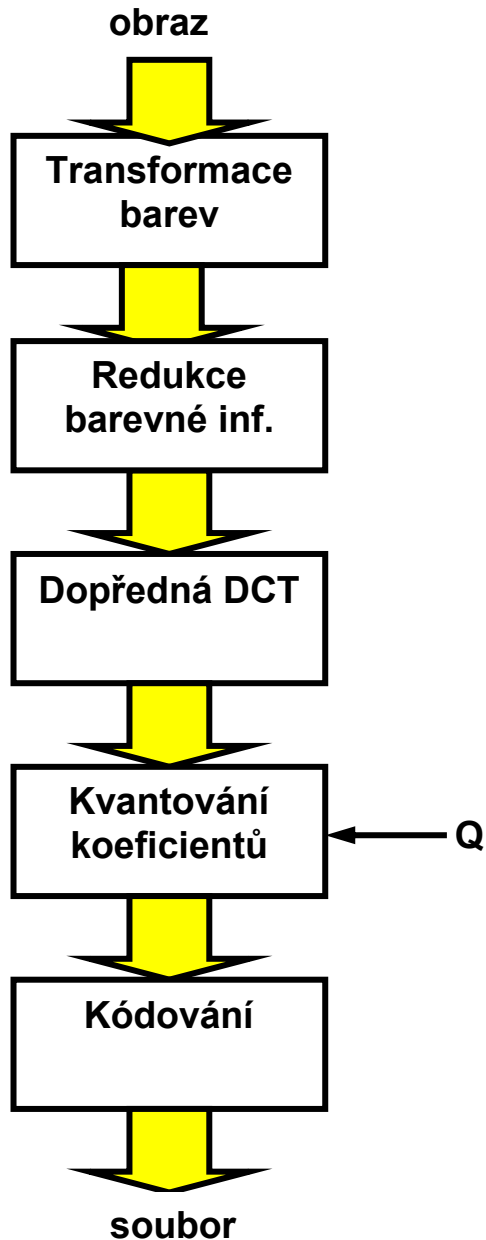
Formát s vynikajícím kompresním poměrem 25:1, vhodný pouze pro snímky s větší bitovou hloubkou (více než 8 bitů na pixel). I když vznikl již v roce 1987, neustále využíván a rozšiřován (poslední rozšíření: ISO/IEC 10918-6:2013).

Využíván všude tam, kde je třeba dosáhnout velkého kompresního poměru, a nevadí rušivé artefakty vzniklé ztrátovou kompresí.

Existují další varianty a rozšíření tohoto formátu, např. JPEG2000 (wavelets), JPEG XR (HDR) , BPG (HEVC).

V roce 2010 představil konkurenci JPEG formátu označenou WebP. Jedná se o výsledek spojení společností On2 Tech a Google a práci na VP8 (kompresní formát pro video), původní verze formátu umožňovala pouze ztrátovou kompresi, nyní umožňuje používat i neztrátový typ komprese. Do současnosti ale formát WebP, nijak „neohrozil“ postavení formátu .JPEG.

Postup komprese:



Převod do barevného modelu:
 $Y C_B C_R$. Jednotlivé složky se dále
zpracovávají samostatně

Složky $C_B C_R$ jsou průměrovány (tzv.
decimovány) podle pravidla 2h1v nebo
2h2v.
(nevratná ztráta informace)

Použití diskrétní kosinové
transformace na oblasti velké 8x8
pixelů.

Vynásobení a zaokrouhlení koeficientů
(neceločíslných) získaných pomocí
DCT váhovou maticí určenou na
základě požadované kvality snímku Q .

Kódování DC členů, kódování AC členů
Huffmanova / Aritmetická komprese.

Kompresie digitálního videa

Video v sobě zahrnuje informaci o obrazu a zvuku, pro které se využívají naprosto odlišné metody digitálního zpracování. Přesto je ale nutné zajistit současný transport a prezentaci obou informací ve správném čase. Tento úkol zajišťují tzv. kontejnerové formáty, které mají za úkol zapouzdřit zvukovou a obrazovou informaci do jediného souboru a případně opatřit informacemi pro detekci a opravu chyb při přenosu.

Obrazovou informací se rozumí sekvence snímků, jdoucích v čase dostatečně rychle za sebou. Je-li frekvence vyšší než 16 Hz, není lidské oko schopné snímky od sebe odlišit, čili vzniká iluze pohybu.

Pokud má snímek velikost 320x240 bodů s 24 bitovou barevnou hloubkou, budeme potřebovat pro jeho uložení 225KB ($320*240*3/1024$). Při použití 25 snímků za sekundu bude již potřeba pro uložení jedné sekundy videa přibližně 5,5MB. To je značný objem dat, se kterým je nutné manipulovat. Proto potřebuje datový tok výrazně zmenšit, k čemuž slouží různé druhy kompresních algoritmů (bezeztrátové a ztrátové).

Bezeztrátové kompresní metody

RAW

Video se ukládá naprosto bez komprese, což vede k neúnosným datovým tokům a velkému objemu uložených dat. Při rozlišení obrazu 720x576 (plný PAL), 25 snímcích za vteřinu a 24 bitech barevné informace na pixel, je výsledný datový tok:

29,7MB/s !

HuffYUV

Tento kodek využívá Huffmanova kódování, kdy se často používané symboly nahrazují kratšími kódy. Typicky dosahuje úrovně komprese 2.5:1. Díky své výpočetní nenáročnosti a neztrátovosti je poměrně hojně používán.

Ztrátové kompresní metody

MPEG

Historie MPEG (čte se jako *em-peg*) je zkratkou pro Moving Picture Coding Experts Group, což je označení dané rodiny mezinárodních standardů používaných pro kódování audio-vizuální informace v digitálně komprimovaném formátu. Rodina MPEG standardů obsahuje MPEG-1, MPEG-2 a aktuální MPEG-4, formálně známé jako ISO/IEC-11172, ISO/IEC-13818 a ISO/IEC-14496. MPEG je původně označení dané skupině expertů, která vyvíjela tento standard. Tato skupina byla ustanovena v roce 1988.

Standardy digitálního videa odvozené od digitální TV

MPEG-1

MPEG-1 standard, ustanoven v roce 1992, je navržen k vytváření dostatečné kvalitních snímků a zvuku při nízkém datovém toku. Navržen byl primárně pro práci s videem o rozlišení 352x288 bodů a 25 snímků/s při datovém toku 1.5Mbit/s, (ale umožňuje práci až s rozlišením 4095x4095 pixelů při 30 snímcích / s). Norma MPEG-1 obsahuje čtyři části:

- IS 11172-1 (System): Popis synchronizace a multiplexování obrazu a zvuku.
- IS 11172-2 (Video): Popis komprese neprovázaných video snímků.
- IS 11172-3 (Audio): Popis komprese zvuku.
- IS 11172-4 (Compliance Testing): Popis testování shody přenesených dat s daty původními.

Audio

IS 11172-3 normy MPEG-1 popisuje kompresi audio signálu. Existují tři kódovací schémata nazývaná Layer 1 až 3. Rostoucí číslo "vrstvy" vyjadřuje zvyšující se kvalitu zvuku a tím i větší složitost kódovacího procesu. Každá vrstva má specifikován formát datového toku a dekodér. V souladu se zvyšující se výpočetní náročností a složitostí použitého algoritmu se zvyšuje i číslo vrstvy. Je použita hierarchická

kompatibilita vrstev, což znamená, že dekodér postavený na určitou vrstvu je schopen dekódovat i vrstvu s nižším číslem. MPEG-1 Layer-3 formát je více znám jako MP3.

Video

Formát MPEG používá diskretní kosinovou transformaci (DCT) obdobně jako JPEG. Snímky jsou rozděleny do tří skupin:

- I-frame (intraframe) – běžný snímek
- P-frame (predictive) – rozdílový snímek mezi běžným snímekem a předchozím I nebo P-snímekem
- B-frame (bi-directional) – rozdílový obrázek mezi dvěma nejbližšími I či P-snímky.

Každý snímek je samostatně komprimován pomocí DCT (JPEG), nejmenší komprese se dosahuje u I-snímeků, nejlepší u B-snímeků. Typicky jsou objemy snímků I:P:B po kompresi v poměru 15:5:2. S praktických důvodů jsou snímky kódovány v přesně definovaných posloupnostech s periodou 12 snímků

I B B P B B P B B P B B I

tj. tak, aby se každé 0.4 sekundy objevil jeden I-snímek.

Vzhledem k tomu, že B-snímky jsou vztaženy k okolním snímkům, není při přenosu či zápisu dat do souboru dodrženo uvedené pořadí snímků. Posloupnost:

$I_1 B_2 B_3 P_4 B_5 B_6 P_7 B_8 B_9 P_{10} B_{11} B_{12} I_{13}$

je odeslána/uložena jako:

$I_1 P_4 B_2 B_3 P_7 B_5 B_6 P_{10} B_8 B_9 I_{13} B_{11} B_{12}$,

neboť pro zobrazení mezilehlých snímků B_2 a B_3 je potřeba znát snímek P_4 .

MPEG-2

MPEG-2 standard, ustanoven v roce 1994, byl navrhnout k vytváření vysoce kvalitních snímků při vyšších datových tocích. MPEG-2 není nutně lepší než MPEG-1, dokonce MPEG-2 při nižších datových tocích (odpovídajících MPEG-1) nedává stejnou kvalitu obrazu jako MPEG-1. Na rozdíl od svého předchůdce, ale přináší podporu pro prokládané snímky (půlsnímky) a možnost využít proměnlivý datový tok. MPEG-2 definuje čtyři úrovně komprese (levels), určené maximálním rozlišením snímků, maximálním počtem snímku za sekundu a maximálním datovým tokem:

Level	Max. šířka [pixely]	Max. výška [pixely]	Max. počet snímků za s [Hz]	Max. dat. tok, [Mbit/s]
Low	352	288	30	4
Main	720	576	30	15
High-1440	1440	1152	60	60
High	1920	1152	60	80

MPEG-4

Vývoj MPEG-4 standardu byl zahájen v roce 1995 a dokončen koncem roku 1998 (ISO 14496). Původně byl určen pro velmi nízké datové toky, nyní umožňuje kódování od velmi nízkých datových toků (2 Kbit/s pro zvuk, 5 Kbit/s pro video) do velmi vysokých (5 Mbit/s pro video, 64 Kbit/s na kanál pro audio CD kvalitu). Není to již přesná definice komprese a komprimačních algoritmů, ale množina parametrů a vlastností, které musí kompresní postup splňovat, aby byl MPEG-4 kompatibilní. Norma definuje osm částí, z nichž některé jsou stále ještě ve vývoji.

- ISO/IEC 14496-1 (Systems)
- ISO/IEC 14496-2 (Visual)
- ISO/IEC 14496-3 (Audio)
- ISO/IEC 14496-4 (Conformance)

- **ISO/IEC 14496-5 (Reference Software)**
- **ISO/IEC 14496-6 (Delivery Multimedia Integration Framework)**
- **ISO/IEC 14496-7 (Optimised software for MPEG-4 tools)**
- stále ve vývoji
- **ISO/IEC 14496-8 (4 on IP framework)**
- stále ve vývoji

MPEG-4 popisuje složky sluchové, vizuální a audiovizuální, nazývané "media objects". Tyto mediální objekty mohou mít přírodní nebo umělý původ, čímž se myslí, že mohou být uloženy pomocí kamery či mikrofону, nebo generované počítačem. Spojením těchto objektů vznikají ucelené celky, které popisují audiovizuální scénu. Generování scény může vypadat například takto:

- **umístění mediálních objektů kamkoliv na scénu vymezenou souřadnicovým systémem**
- **aplikace transformací pro změnu geometrických a akustických vlastností objektu**
- **spojení primitivních mediálních objektů a tím i vytvoření ucelenějšího objektu**
- **aplikace datového toku na mediální objekty pro změnu jejich vlastností (například zvuk, pohyblivou texturu náležící objektu, animační parametry řídící umělou tvář atd.)**
- **mění se interaktivně směr pohledu uživatele a poloha zdroje zvuku kdekoliv ve scéně.**

Dále je zabezpečen výběr a synchronizace dat, spojených s těmito objekty. Formát umožňuje navíc různé způsoby interaktivity a obsahuje podporu pro správu a ochranu duševního vlastnictví (DRM).

Příklady dalších formátů digitálního videa

DivX 3.11a Alpha

V roce 1999 vytvořil MS, software *Windows Media Tools 4*, který obsahoval tři kodeky MPEG-4 označené V1, V2, V3. V původní beta verzi bylo možné používat „.AVI“ soubory jako výsledný datový kontejner, ale ve finální verzi však už bylo možné zvolit pouze nový formát ASF. DivX 3.11a Alpha je nelegální a upravená verze kodeku ASF MS-MPEG4v3, kde je opětně povoleno užívání AVI souborů jako cílového kontejneru. Vznikly dva kodeky *Slow motion* a *Fast motion*. Jak již název napovídá, byla pouze změněna distribuce bitové rychlosti (bitrate). U *Fast motion* se u scén s malým pohybem snižoval bitrate daleko více, takže u rychlých scén bylo možné použít větší množství dat a tím i zvýšit celkovou kvalitu.

DivX 5.x

<http://www.divx.com/>

V roce 2000 původní autor DivX 3.11a s několika dalšími programátory vytvořil projekt OpenDivX (Project Mayo), který měl nahradit dosavadní nelegální DivX a vytvořit kodek MPEG-4, který bude open source a do jehož vývoje se bude moci zapojit kdokoliv. Postupem času tato skupina založila vlastní firmu DivX Networks, jejímž výsledkem byl vlastní kodek DivX 4, založený na OpenDivX, kompatibilním s původním DivX 3, ale se nejednalo již o open source project. Jeho kvalita byla o mírně horší než DivX 3, šlo ale o plně funkční software. Další verze přišly s výrazným zlepšením enkódování a kvalitativně se dostaly až na úroveň DivX3 (a tedy MS MPEG-4 V3), kodeku který byl stále “freeware“.

Začátkem března 2002 pak přišla verze 5.0, která obsahuje další vylepšení v kvalitě, ovšem pouze v placené verzi DivX Pro. Základní verze je stále zdarma, zůstává však kvalitativně na úrovni předchozí, obsahuje ale různé optimalizace pro zvýšení rychlosti a pár vlastností navíc. Novinkou je také nové vnitřní uspořádání informací v souborů DivX, který by měl řešit problémy se synchronizací zvuku v kontejneru AVI.

XviD

<http://www.xvid.org/>, <http://gordianknot.sourceforge.net/>

Open source kodek, původně vycházející z DivX (pozpátku XviD). Jakmile byl projekt OpenDivX převeden do komerční báze, jejímž výsledkem byl kodek DivX4, skupina open source vývojářů se rozhodla opustit původní OpenDivX a vytvořit vlastní kodek, zpětně kompatibilní s DivX.

MJPEG

Na rozdíl od MPEG formátu se jedná pouze o sekvenci snímků popřípadě pulsů, které jsou nezávisle na sobě komprimovány pomocí stejným principem, jako snímky u MPEG-1 (tj. DCT).

AVC/H.264

Advanced Video Coding - Kodek kompatibilní s normou MPEG-4 part 10 též zvané MPEG-4 AVC, defakto nynější standard pro přenos videa na internetu, pozemní digitální vysílání i BlueRay médií.

HEVC/H.265

High Efficiency Video Coding – Nastupující formát postavený na modifikaci H.264 určené pro 4K - 8K cílem je snížit paměťové nároky alespoň na polovinu oproti H.264. Schválen v roce 2013, jako ISO/IEC 23008-2 za jeho použití je spojeno s licenčními poplatky.

Příklady datových kontejnerů pro ukládání digitálního videa

AVI

Nejznámější datový kontejner původně navržen fy MS jako jeden z hlavní kontejnerových formátů pro projekt Video for Windows a MPEG4.

ASF

ASF je zkratka pro Advanced Streaming Format. Jedná se o chráněný formát vyvinutý firmou Microsoft pro streaming videa podporující DRM. Většina ASF souborů je založena na Microsoft MPEG-4 V2 technologii.

QuickTime

Nejedná se jen o formát, ale o vlastní video platformu (založenou na MPEG-4, H.264), vytvořenou firmou Apple. Oblíben na internetu, umožňuje streaming.

VCD (video CD), SVCD (super video CD)

Formáty určené k ukládání videa na CD.

VCD využívá MPEG-1 při rozlišení 352x280a konstantním datovém toku. Maximální délka videa uložená na VCD je cca. 74minut.

SVCD využívá MPEG-2 při rozlišení 480x576 a variabilní datový tok. Maximální délka videa uložená kolísá mezi 35 – 70 minutami. SVCD umožňuje uložit šesti-kanálový zvuk. (5+1)

Oba formáty umožňují definovat interaktivní nabídky (menu).

DVD (digital versatile disc)

Využívá formát MPEG-2 ale navíc definuje strukturu adresářů a další vlastnosti (zvukové a titulkové stopy, menu) kterými je samotné MPEG stream “obalen”. Celé DVD tedy neobsahuje žádný soubor .MPG, ale pouze soubory : .IFO, .VOB, .BUP.

Blue-Ray, HD DVD, VMD

Využití laserového paprsku o kratší vlnové délce, umožnilo navýšit, kapacitu nových datových medií oproti DVD a umožnilo ukládat HD (High Definition) video (1920x1080). Kapacita nových medií dosahuje 25-50Gb. V minulé době jsme byli svědky boje standardů Blue-Ray (Sony), HD DVD (Toshiba). Další alternativou byl VMD (Versatile multilayer disc). Vítězem se stal formát Blue-Ray.

Matroska, WebM

Otevřené kontejnerové formáty. WebM vznikl jako projekt společnosti Google, podporuje omezené množství funkcionality nabízené původním formátem Matroška (.MKV, .MKA, .MK3D, .MKS).

Literatura:

- [1] Žára, J. a kol.: Počítačová grafika - Principy a algoritmy, Grada 1992, ISBN 80-85623-00-5
- [2] Žára, J.; Beneš, B.; Felkel, P.: Moderní počítačová grafika, Computer press 1998, ISBN 80-7226-049-9
- [3] Serba, I.; Zendulka, J.; Sochor, J.: Základy počítačové grafiky, skriptum VUT 1991
- [4] Juříček, T.: SW Modul pro kódování a dekódování video formátů, bak. práce UAMT FEKT VUT 2005
- [5] Tišnovský, P. : <http://www.root.cz/clanky/pcx-prakticky-implementace-komprimace-rle/>
- [6] Wikipedia RLE: <http://cs.wikipedia.org/wiki/RLE>
- [7] Wikipedia LZ77: <http://en.wikipedia.org/wiki/LZ77>
- [8] Wikipedia LZSS: <http://en.wikipedia.org/wiki/LZSS>
- [9] Wikipedia CZ LZW: <http://cs.wikipedia.org/wiki/LZW>
- [10] Wikipedia CZ Huffman: http://cs.wikipedia.org/wiki/Huffmanovo_k%C3%B3dov%C3%A1n%C3%AD
- [11] Wikipedia EN Huffman: http://en.wikipedia.org/wiki/Huffman_coding
- [12] Wikipedia Arithmetic coding: http://en.wikipedia.org/wiki/Arithmetic_coding
- [13] Žára, J.; Beneš, B.; Felkel, P.: Moderní počítačová grafika, Computer press 1998, ISBN 80-7226-049-9
- [14] Biondi, P.; Desclaux, F.: Silver Needle in the Skype, EADS CCR