

Metody zdrojového kódování digitálního signálu

The Methods of the Source
Coding the Digital Signal

Gloser Jan

Bakalářská práce
2005



Univerzita Tomáše Bati ve Zlíně
Fakulta Technologická

Oficiální znění zadání BC

Prohlašuji, že jsem na celé bakalářské práci pracoval samostatně a použitou literaturu jsem citoval.

Ve Zlíně, 31. 05. 2005

.....

jméno diplomanta

Abstrakt

S rozvojem počítačových technologií začaly pronikat ve velké míře tyto technologie i do oblasti průmyslu komerční bezpečnosti.

Tato práce je zaměřena na metody zdrojového kódování digitálního signálu. Tyto metody je možno použít např. při přenosu obrazu v rámci CCTV (Closed Circuit Television). V práci je nastín, přes jaké sítě je možno přenášet digitální signál, dále jsou zde zmíněny možnosti komprimace, způsob zpracování obrazového signálu a metody komprese. V metodách komprese se zaměřuji na bezztrátové a ztrátové komprimační algoritmy.

Developing of computer technologies stimulates the production of new technologies applied into the industry of the commercial security.

This work is focused to the source coding methods of the digital signals. These methods are possible to use in the image transmission in CCTV (Closed Circuit Television). This work describes some technologies of the digital signal transmission, some compression techniques and the possibilities of the digital image processing and the compression too. In a part of the compression methods lossless and lossy compression algorithms are presented.

OBSAH

ÚVOD.....	3
1 PARAMETRY TELEKOMUNIKAČNÍ SÍTĚ	5
1.1 DRUHY SÍTÍ.....	6
2 KOMPRIMACE	7
2.1 DRUHY KOMPRIMACE.....	8
2.2 HLAVNÍ PARAMETRY VÝKONU	8
2.2.1 Rychlost komprese	8
2.2.2 Rychlost dekomprese.....	8
2.2.3 Kompresní (komprimační) poměr	8
3 ZPRACOVÁNÍ OBRAZOVÉHO SIGNÁLU.....	9
3.1 ZÁKLADNÍ SCHÉMA	10
3.2 ZÁKLADNÍ POJMY	11
3.2.1 Dělení digitálních statických obrazů	13
3.2.2 Typy obrazu.....	14
3.2.3 Kompresní standardy	15
3.3 KOMPRIMACE OBRAZU.....	15
3.4 KÓDOVÁNÍ OBRAZOVÝCH SIGNÁLŮ.....	16
3.4.1 Cíle kódování.....	16
3.4.2 Efektivní kód	17
3.5 METODY KÓDOVÁNÍ.....	18
3.5.1 Zdrojové kódování.....	19
3.5.2 Kanálové kódování	19
3.5.3 Reverzibilní (bezztrátová) lossless compression	19
3.5.4 Ireverzibilní (ztrátová komprese) lossy compression	19
3.6 REPREZENTACE OBRAZU	20
3.6.1 Heuristická reprezentace obrazu.....	21
3.6.2 Hierarchická reprezentace obrazů.....	22
3.6.3 Množinová reprezentace obrazů	23
4 METODY KOMPRESY.....	24
4.1 BEZZTRÁTOVÁ KOMPRESY (LOSSLESS COMPRESSION)	24
4.1.1 Huffmanovo kódování.....	24
4.1.2 Aritmetické kódování	27
4.1.3 Shannonovo-Fanovo kódování	31
4.1.4 RLE (Run-length Encoding).....	32
4.1.5 LZW (Lempel-Ziv-Welch algoritmus).....	36
4.2 ZTRÁTOVÁ KOMPRESY	37
4.2.1 JPEG (Joint Photographic Experts Group).....	38
4.2.2 MPEG (Moving Picture Expert Group).....	42
ZÁVĚR.....	48
SEZNAM POUŽITÝCH ZDROJŮ	49
SEZNAM OBRÁZKŮ.....	50
SEZNAM TABULEK	51

ÚVOD

S rozvojem číslicové elektroniky a telekomunikační techniky vznikli nové možnosti při zpracování a přenosu datových, zvukových, především však grafických a obrazových informací. Velký důraz je tu kladen na přenos a archivaci statických a dynamických obrazů.

U moderních a v dnešní době i cenově přístupných počítačů se s pojmem multimédia setkáváme velice často. Počítač se začal chápat jako zprostředkovací činitel pro více oblastí.

Co vůbec slovo multimédia vyjadřuje? Ve slovníku cizích slov bychom našli vysvětlení:

Multi – mnoho

Medium – technické prostředí, látka nebo i zprostředkovací činitel či osoba.

Dnešní programy na multimediálním základě mají celou řadu výhod. Můžeme zde pracovat nejen s textem, ale už i s auditem, grafikou, animacemi a s videem. Při jejich použití je umožněno, aby uživatel měl možnost zasáhnout do průběhu tohoto multimediálního programu.

Proto je cílem této práce přiblížit dnešní multimediální svět, jeho využití v různých oblastech, poukázat na požadavky při tvorbě multimediálních aplikací a také přiblížit základní metody číslicového zpracování obrazu. Na jeho přenos a zpracování se kladou vysoké nároky, jednak z hlediska jeho množství, ale i kvality. Obrazové informace zpracovávají obrazové systémy, které můžeme rozdělit na analogové a číslicové. Pod obrazovými systémy rozumíme všeobecné metody a prostředky pro vytváření, zpracování, přenos a reprodukci obrazové informace při respektování vlastností zraku.

Rozvoj metod a prostředků číslicového zpracování obrazové informace vedl ke vzniku číslicových obrazových komunikačních systémů (ČOKS), jejichž úkolem je přenos textových a grafických resp. obrazových zpráv, které využívají existující televizní a telefonní sítě. Později vznikly faksimilní systémy, které umožňují přenášet textové a grafické zprávy po telefonních linkách.

Přenos obrazové informace bývá realizován telekomunikačními sítěmi, přičemž informace je přenášena k uživateli s dostatečnou kvalitou a popisuje přesně vlastnosti

zkoumaného objektu. Významnými parametry telekomunikačních sítí jsou šířka přenášeného pásma a přenosová rychlost.

Především přenosová rychlost je ovlivňujícím faktorem pro požadovanou kvalitu obrazových a zvukových informací na straně příjmu. Zvuk, obrazové informace a video vyžadují rozsáhlé množství dat.

1 PARAMETRY TELEKOMUNIKAČNÍ SÍTĚ

Jak jsem se již zmínil, významnými parametry jsou přenosová rychlost a šířka přenášeného pásma. Ovlivňujícím faktorem pro kvalitu obrazových a zvukových informací na straně příjmu je především přenosová rychlost. V následující tabulce bych chtěl ilustrovat obvyklé nároky na prostor pro různé typy dat jako jsou text, obrázky, zvuk, animace a video.

Pro komprimaci údajů v současných informačních a komunikačních systémech mluví tři základní důvody:

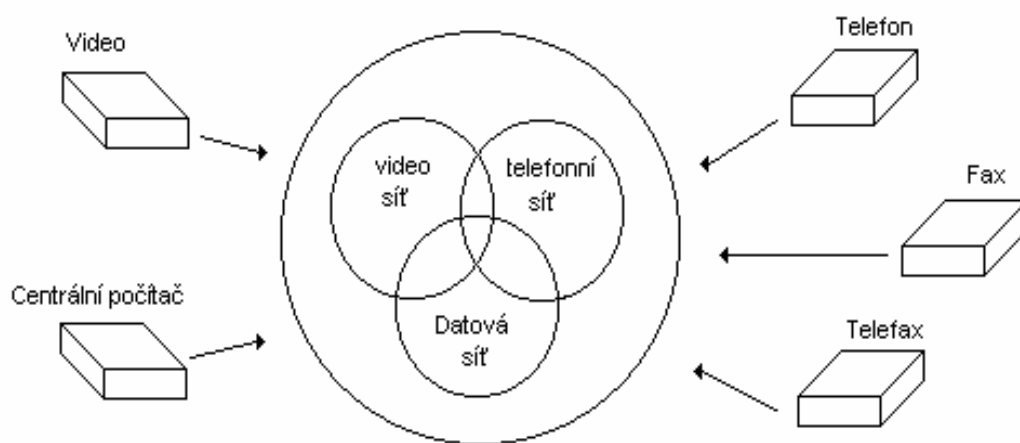
- rozsáhlé paměťové nároky multimediálních dat,
- relativně pomalá paměťová zařízení, která neumožňují přehrávání multimediálních dat (zejména videa) v reálném čase,
- šířka pásma současných přenosových sítí, která neumožňuje přenos videa a často ani zvuku v reálné čase.

Tabulka I. Paměťové a přenosové nároky jednotlivých typů médií

	Typ objektů	Rozeř a šířka požadovaného přenosového pásma
Text	ASCII EBCDIC	2 kB na stránku
Obraz	Rastrová grafika, obrázky, fax	Jednoduché: 64 kB na obrázek Detailní (barevné): 7,5 kB na obr.
Zvuk	Nekódovaný proud digitálního zvuku nebo hlasu	Hlas/Telefon: 8 kHz /8 bitů (mono) 6 – 44 kB/s Audio CD: 44,1 kHz /16 bitů (stereo) 176 kB/s
Animace	Synchronizovaný proud údajů o obrázku a zvuku 15 – 20 rámců / s	320 x 480 x 24 pixelů / rámeč 16 bitové rozlišení barev 16 rámců /s 2–5MB/s
Video	Analogový TV signál nebo digitální obraz se synchronizovaným zvukem 24 – 30rámců / s	640 x 480 x 24 pixelů / rámeč 24 bitové rozlišení barev 30 rámců /s 2707MB/s

1.1 Druhy sítí

Generace přenosových systémů se rozvíjí jako součást sítě ISDN, které integruje přenos zvukových signálů (telefonní síť), textových a grafických zpráv (faximile, videotext, telex), přenos dat (počítačové sítě) a přenos obrazu (video síť) jak je to uvedeno na obr. 1.



Obr. 1

Obr. 1. Struktura přenosových systémů a sítí.

2 KOMPRIMACE

Komprimace umožňuje efektivní digitální reprezentaci zdrojových signálů, jako je text, zvuk, obraz nebo video, použitím redukovaného počtu prvků digitálních informací (bitů), než má originál. Pokud má být použití komprimace efektivní, musí umožňovat reprodukci komprimované informace v požadované kvalitě.

Většinou je možné předpokládat, zda bude datový soubor obsahovat dlouhé bloky stejných dat (např. v některých grafických souborech). V takovém případě lze použít algoritmy vyhledávající tyto dlouhé bloky dat, které jsou posléze nahrazeny jediným symbolem.

Mnohdy jsou zase známy například pravděpodobnosti, se kterými se v datech vyskytují jednotlivé symboly (je známa frekvence jednotlivých znaků v abecedách různých jazyků). Jednou z možností pak je použít s úspěchem frekvenčně závislé kódování založené na přepisu častěji používaných prvků abecedy na kratší bitové posloupnosti a řidčeji se vyskytující prvky na delší bitové posloupnosti (použití tzv. nerovnoměrných kódů).

Většina používaných metod komprimace funguje tzv. proudovou metodou (streaming mode), při které je programem načten jeden nebo několik bitů informace, ty jsou zpracovány (zakódovány) a tímto postupem se pokračuje až do konce komprimovaných dat (obvykle např. konce souboru).

Některé metody používají při kódování informace o celém bloku sousedících bajtů, označujeme je jako blokové metody (block mode), které načítají vstupní informaci blok po bloku a kódují každý blok samostatně.

Většina metod komprimace funguje na fyzické úrovni vstupních dat, tj. nahrazuje vstupní proud bajtů nebo bitů a nestará se o jejich informační obsah a souvislosti. Takovéto metody transformují vstupní proud informací na jiný, pokud možno kratší výstupní proud podle kódovacího algoritmu, bez jakéhokoliv využití sémantických (významových) vědomostí o složení a možné následnosti kódovaných informací. Metody komprimace na logické úrovni tyto znalosti umí využívat a díky tomu mohou dosahovat vyšších poměrů komprimace. Obvykle jsou výpočetně náročnější a jsou použitelné jen pro jistý specifický typ dat.

2.1 Druhy komprimace

Komprimační algoritmy lze podle náročnosti (zejména časové) komprimačního a nekomprimačního procesu rozdělit na :

- symetrické – doba potřebná pro kompresi i dekompresi dat je přibližně stejná,
- asymetrické – zde je kratší dekomprimace než komprimace dat. Můžeme používat při zálohování dat, které ve většině případů vůbec nepotřebujeme.

2.2 Hlavní parametry výkonu

Hlavními parametry výkonu (anglicky je seznam takových parametrů označován jako benchmarks) komprimačních algoritmů jsou především:

- rychlost komprese,
- rychlost dekomprese,
- kompresní (komprimační) poměr.

2.2.1 Rychlost komprese

Čas potřebný pro sbalení souboru do komprimované podoby, může být také velice důležitým faktorem. Čas potřebný pro kompresi různými komprimačními programy se u rozsáhlých souborů liší (řádově až o celé minuty).

2.2.2 Rychlost dekomprese

Čas potřebný k rozbalení komprimovaného souboru do původní podoby, je ze všech tří zmíněných parametrů asi nejméně důležitý. Doba potřebná pro dekomprimaci bývá téměř u všech komprimačních programů kratší než doba potřebná ke komprimaci, a proto tento parametr nebývá tím omezujícím nebo kritickým faktorem při výběru vhodného komprimačního programu.

2.2.3 Kompresní (komprimační) poměr

Kompresní poměr vyjadřuje poměr mezi velikostí (vyjádřené počtem bitů) komprimovaného a původního souboru.

3 ZPRACOVÁNÍ OBRAZOVÉHO SIGNÁLU

Porovnání a zpracování obrazu se vyvíjeli jako samostatné disciplíny, které navzájem úzce souvisí. Oblast působnosti zpracování obrazu je nejen kódování, filtrace, zvýrazňování a obnovení, ale i analýza a rozpoznávání obrazu. To znamená rozpracovat teoreticky a prakticky problémy analýzy a rozpracování obrazu. Oblast rozpoznávání obrazu je nejen odběr a výběr příznaků, primitivů a jejich klasifikace, ale i předzpracování a analýza obrazu.

Rozpracování obrazu je tedy spojeno s analýzou a klasifikací objektu v obrazech. Spočívá v řešení třech základních problémů :

- a) Získání obrazu (problém vhodného senzoru, digitalizace, předzpracování)
- b) Analýza obrazu (výběr příznaků a nebo primitivů, segmentace obrazu)
- c) Klasifikace – analýza obrazu (přiřazení k určité třídě, detekce objektů, analýza scén)

Zpracování obrazu souvisí v současnosti s prudkým rozvojem polovodičové a zejména číslicové techniky, přičemž dominantní postavení zaujímají číslicové metody zpracování obrazu. Obraz potom vystupuje ve tvaru matice $N \times N$, jejichž prvky jsou X_{ij} , kde $j, i = 1, 2, \dots, N$ a určují jas příslušného bodu v definovaném rastru.

S rozvojem výpočetní techniky se ukázalo velmi důležité a efektivní archivovat data v paměti počítače. Spojením výpočetní a telekomunikační techniky vznikají nové možnosti zpracování a přenosu obrazu.

Existuje mnoho kódovacích metod, které se liší mezi sebou zejména dosaženým kompresním poměrem, rychlostí kódování a dekodování kvalitou rekonstrukce a výpočtovou náročností algoritmů.

Kompresní poměr lze obecně vyjádřit jako :

Poměr komprimace = rozměr vstupního proudu / rozměr výstupního proudu

Na kompresi obrazových údajů se používají transformační, predikční a hybridní metody. Z ortogonálních transformací je z hlediska střední kvadratické chyby optimální Karhunenova – Loèveova transformace (KLT).

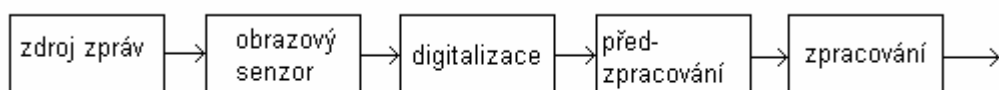
Jednou z progresivních metod v oblasti kódování a přenosu statických šedých (tzv. víceúrovňových) a barevných obrazů a obrazových video sekvencí je fraktálová komprese.

Fraktálově založené obrazové kódování se také někdy nazývá fraktálové kódování obrazu, které patří mezi nové metody obrazové komprese. Základem většiny fraktálově založených kompresních metod obrazu je metoda iterujících funkčních systémů *IFS*. Technika fraktálově založeného obrazového kódování je také rozšířena na kódování obrazových video sekvencí. Fraktálové kódování je zajímavé zejména s ohledem na velké kompresní poměry. Patří ke ztrátové formě obrazového kódování. Fraktálové kódování je typickým představitelem ztrátového kódování s postupnou rekonstrukcí na straně příjemce. Jde o asymetrickou metodu ztrátového kódování z hlediska porovnání doby kódovacího a dekódovacího procesu. Fraktálová komprese obrazu z hlediska svého vzniku úzce souvisí s problematikou telekomunikačních sítí a je jednou z moderních kompresních metod. Fraktálová komprese obrazu dává výsledky komprese porovnatelné s JPEG standardem (založeném na použití diskrétní kosinové transformace – DCT, resp. diskrétní waveletové transformace – DWT) a její předností je výpočetně jednoduché dekódování. Prostřednictvím fraktálové metody kódování statického obrázku můžeme kódovat video sekvence digitálního zobrazení údajů (video). I když je fraktálové kódování 2D obrazů je slibné, obvykle se požaduje vyšší efektivnost kódování dat 3D bloků (video). Ve většině případů 3D data představují digitální video. Fraktálové kódování obrazových sekvencí představuje jen jednu z mnoha metod kódování video sekvencí.

3.1 Základní schéma

Pod pojmem číslicové zpracování obrazové informace se rozumí převod obrazové zprávy do číslicového tvaru, zpracování v číslicovém tvaru a přenos resp. další zpracování obrazové informace.

Všeobecný model číslicového zpracování obrazové informace možno znázornit blokovým schématem na obr. 2



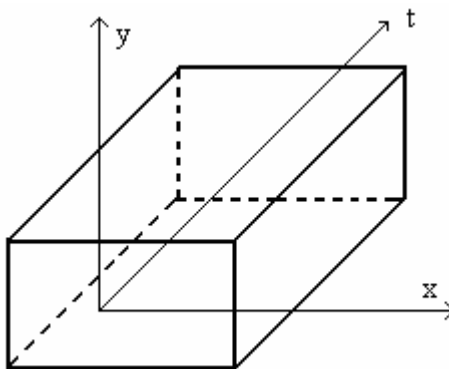
Obr. 2. Všeobecný model číslicového zpracování obrazové informace.

- a) Zdroj zprávy – produkuje jednotlivé obrazové zprávy a je možno jej považovat za zdroj náhodného procesu, ve kterém se jednotlivé zprávy (obrazy) vyskytují s různou pravděpodobností a statistickou závislostí.
- b) Obrazový senzor – transformuje obrazovou zprávu nejčastěji pomocí fotoelektrické konverze na analogový elektrický signál. V bloku digitalizace se tento signál blokuje, kvantuje a číslicově kóduje. Výsledkem procesu digitalizace je digitální obraz, který je reprezentovaný maticí čísel reprezentujících jas obrazu.
- c) Předzpracování – obrazové informace spočívá v odstranění různých skreslení, šumů, kterými byl obraz znehodnocen v procesu snímání a digitalizace. Součástí předzpracování je někdy také úprava obrazových dat (obrazu), která zvyšuje efektivnost dalšího zpracování, např. kódování.
- d) Zpracování – obrazové informace závisí na cílech číslicového zpracování. Může to být zejména :
 - *archivace obrazu* – je efektivní uložení obrazu na paměťových médiích tak, aby se dosáhlo co největšího efektu komprese dat,
 - *analýza obrazu* – má za cíl výběr určitých charakteristických vlastností (příznaků) obrazu, které mohou sloužit např. pro rozpoznání obrazu,
 - *přenos obrazu* – má za cíl efektivní a věrný přenos obrazu, které se musí na straně příjemce co nejméně lišit od původní obrazové zprávy

Číslicové obrazové komunikační systémy dovolují komunikaci s obrazem (videem), přičemž v tomto systému prioritní informací je dynamický barevný obraz a jako přídatné informace jsou řeč, počítačové údaje, grafika a statický binární obraz.

3.2 Základní pojmy

Vizuální zprávy je možno si představit jako obrazový tok $B(x,y,t)$, kde funkce $B(x,y,t)$ vyjadřuje plošné rozdělení světelné veličiny probíhající v čase t , přitom se pro názornost předpokládá omezený geometrický rozměr obrazu v osách x,y (obr. 3).



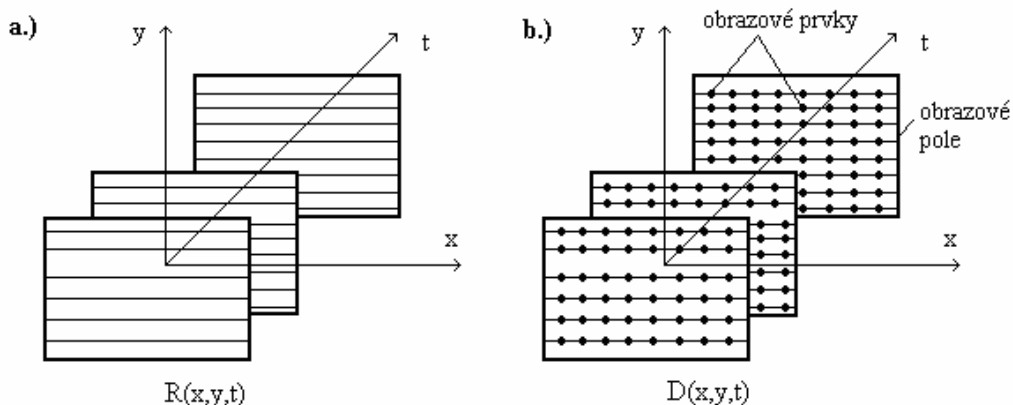
Obr. 3. Geometrický rozměr obrazu.

Pokud je výchozí obraz bez vnitřního pohybu, jde o tzv. **statický obraz**, který můžeme popsat funkcí $B(x,y)$, tj. obrazový tok je závislý jen na prostorových souřadnicích a nemění se v závislosti na čase. Naopak funkce $B(x,y,t)$ popisuje **dynamický (pohyblivý) obraz**.

Zatím co původní obraz je daný trojrozměrnou spojitou funkcí $B(x,y,t)$ obrazová informace na výstupu senzoru je dána prostorovým a časovým rozkladem původního obrazu. Rozklad obrazu v čase je možné charakterizovat odpovídající funkcí $D(x,y,t)$, která je posloupností Diracových funkcí (x,y,t) . Výsledkem prostorového rozkladu je to, že jednotlivé řádky se nahradí posloupností vzorků informace $g(x,y,t)$, pro které platí :

$$g(x,y,t) = B(x,y,t) * D(x,y,t)$$

Jednotlivé vzorky obrazové informace se nazývají obrazové prvky (obr. 4b).



Obr. 4. Obrazové pole a prvky.

Výsledkem časové a prostorové diskretizace je tedy množina obrazových prvků, které v prostorovém uspořádání vytvářejí obrazové pole resp. obrazovou matici, ve které je každý prvek obrazovým prvkem. Rozměr obrazového pole resp. obrazové matice se nazývá rastr obrazu. V procesu digitalizace se vzorky obrazové informace kvantují.

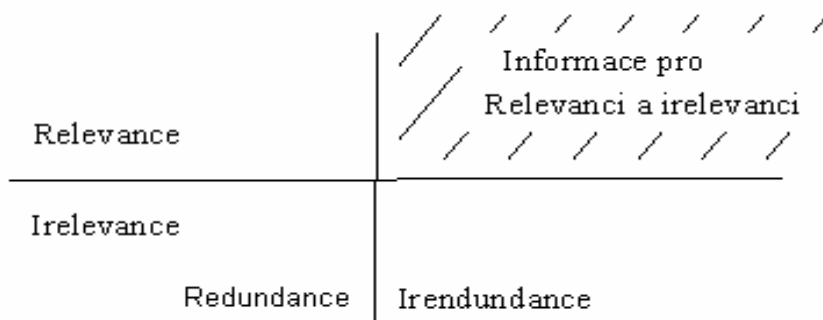
3.2.1 Dělení digitálních statických obrazů

Digitální statické obrazy lze podle charakteru obrazových informací rozdělit na:

- binární resp. dvouúrovňové obrazy,
- víceúrovňové resp. šedé obrazy,
- barevné obrazy.

Informační obsah obrazu můžeme znázornit pomocí informační roviny obrazu, kterou dělíme na:

- horizontální,
- vertikální.



Obr. 5. Informační obsah obrazu.

O rozdělení na relevanci a irelevanci rozhodne uživatel (příjemce) informace, resp. druh aplikace. Redundance je skryta ve statistických vlastnostech obrazu (korelaci, nelinearita rozdělení pravděpodobnosti obrazových prvků, atd.) a představuje nadbytečné množství informací z hlediska kódování. Pro příjemce je žádoucí průnik relevance a iredundance (1. kvadrant). Vysílání je jen informace z 1. kvadrantu. V procesu dekódování se obnoví i redundantní část, čímž získáme celou relevantní polovinu.

Kvadr na obr. 3 představuje informační tok (obraz) a jakýkoliv příčný řez tímto kvádrem představuje jednotlivé snímky. Každý bod tohoto kvádrů můžeme vyjádřit pomocí třech souřadnic v, h, t (vertikální, horizontální, časová). Informační tok (obraz) popisujeme jasovou funkcí.

Pro statický obraz : jasová funkce : x

Bod : (v, h)

Jas bodu : $x(v, h)$

Pro dynamický obraz : jasová funkce : x

Bod : (v, h, t)

Jas bodu : $x(v, h, t)$

Z hlediska vlastnosti zraku můžeme odstranit určitou irelevanci. Zrakový systém má konečnou prostorovou rozlišovací schopnost v horizontálním i vertikálním směru. Potom snímek můžeme vyjádřit konečným počtem bodů. Zrakový systém má také omezenou pohybovou (časovou) rozlišovací schopnost. Potom spojitý tok můžeme nahradit diskrétním tokem vybraných snímků reprezentovaných diskrétními body. Zbytek představuje irelevanci.

Zrakový systém (příjemce) vnímá scénu integrálně a pro zrakový systém platí :

$$\rho_1 * \rho_2 * \rho_3 = konst.$$

kde ρ_1 – je prostorová rozlišovací schopnost,

ρ_2 – je gradační rozlišovací schopnost,

ρ_3 – časová rozlišovací schopnost.

Když se jeden z parametrů ρ_i mění, tak to automaticky vyvolá změnu druhých parametrů, aby se zachovala kvalita obrazu.

3.2.2 Typy obrazu

- a) Binární obrazy – jas každého obrazového prvku je reprezentován jen jedním bitem („0“ nebo „1“), takže máme dvě jasové úrovně

- b) Statické obrazy – je to obraz bez vnitřního pohybu (fotografie). Jas každého obrazového prvku je reprezentován větším počtem bitů než jeden, přitom získáváme větší počet jasových úrovní než dvě.
- c) Dynamické (pohybové) obrazy – představují sekvence – video. Je to posloupnost statických snímků.

3.2.3 Kompresní standardy

- a) JPEG – tento standard je výsledkem práce skupiny expertů označované jako Joint Photographic Experts Group (JPEG), která vznikla s cílem vytvoření mezinárodního standardu pro kompresi a dekompresi spojitě tónovaného (víceúrovňového) nepohyblivého obrazu jak „černo-bílého“ (tzv. šedého) a také barevného.
- b) MPEG – tento standard představuje prostředek pro kompresi a dekompresi pohyblivého obrazového signálu a s ním spojeného zvukového signálu, uložených na digitálních paměťových médiích jakými jsou např. CD-ROM, DAT, pevné disky, optické disky, jako i přenos těchto signálů po různých komunikačních kanálech např. ISDN, MAN, LAN apod. Byl vytvořen mezinárodní skupinou expertů označovanou právě MPEG –Moving Picture Experts Group. Standard využívá mnohé prvky a metody kompresních standardů, především standardu H.261 CCIT, pro videotelefonii a standardu JPEG pro kompresi a dekompresi statických obrazů.

3.3 Komprimace obrazu

Ze zkušeností víme, že obrázek uložený ve formátu BMP (prostá bitová mapa) zabírá na diskovém médiu více místa než stejný obrázek uložený např. formátem JPEG, PCX, GIF a další. Formáty těchto obrázků v sobě již zahrnují kompresi dat. To je umožněno faktem, že obrazová data mají svá specifika.

Většina původních obrázků převedených do digitální formy obsahuje nadbytečné informace, které lidské oko není schopné postihnout. Lidské oko má omezenou rozlišovací schopnost jak ve vztahu k barevné hloubce, tak k obrysovým detailům.

Ačkoli číselné vyjádření jednotlivých pixelů obrázku bývá zcela odlišné, komprimační formáty obrázků se snaží v těchto číselných kombinacích najít řád, který jim

umožní zmenšit náročnost obrázku na kapacitu diskového média. Ačkoli tato komprimace vede ke ztrátě dat, kvalita obrázku měřena subjektivním lidským vjemem zůstává stále stejná.

3.4 Kódování obrazových signálů

Kódování je úprava vnitřní struktury zprávy z hlediska způsobu uložení informačního obsahu tak, aby tato struktura vyhovovala požadavkům informačního procesu. Obecně je kódování předpis, který každému prvku konečné množiny A přiřazuje jedno slovo v konečné množině B , přitom množina A se nazývá zdrojová abeceda a její prvky jsou zdrojové znaky a množina B se nazývá kódová abeceda a její prvky jsou kódové znaky.

3.4.1 Cíle kódování

Cíle kódování digitálních signálů jsou:

- a) přizpůsobení formy uložení informačního obsahu z hlediska technického uspořádání přenosového kanálu (např. číslicové přenosy potřebují binární vyjádření, člověk dekadické),
- b) zvýšení efektivnosti zprávy, tj. odstranění nežádoucí redundance,
- c) cílevědomé zvyšování redundance zprávy, které je zaměřeno na zvýšení odolnosti vůči poruchám. Toto (tzv. kanálové) kódování zavádí záměrně do procesu kódování redundanci, která umožňuje odhalit a někdy i opravit chyby při přenosu (detekční a korekční kódy)

Kódy lze rozdělit z více hledisek, přičemž rozdělujeme kódy :

- rovnoměrné (blokové) – kódy, ve kterých blokové znaky mají stejnou délku, tj. stejný počet symbolů.
- nerovnoměrné – kódy, ve kterých kódové znaky mají různou délku.

Kódovací postupy jsou :

- kombinační – pokud na kódování stačí znalost zdrojového znaku,

- sekvenční – pokud kódování využívá kromě znalosti daného zdrojového znaku také informace z předchozího procesu kódování.

3.4.2 Efektivní kód

Efektivní kód je takový kód, který používá na vyjádření kódového znaku nejmenší počet symbolů (znaků). Při zvyšování efektivnosti kódování je důležitý požadavek, aby kódové znaky neobsahovali prázdné symboly. Kódování bez prázdných symbolů je zaručeno prefixovými kódy. Jsou to také kódy, ve kterých žádný kódový znak větší délky neobsahuje ve svém začátku kódový znak jiného zdrojového znaku.

Při vytváření efektivních kódů se nejčastěji používají tyto postupy :

- a) kódy typu Shannon – Fan,
- b) kódy typu Hoffman,
- c) aritmetické kódy.

Jde o nerovnoměrné prefixové kódy, ve kterých pro zdrojové znaky s vysokou pravděpodobností výskytu se volí krátké kódové znaky. Pro znaky s nízkou pravděpodobností výskytu se volí delší kódové znaky.

Z hlediska přenosu informace v přenosovém kanálu s poruchami možno proces kódování, resp. dekódování znázornit podle obr. 6.

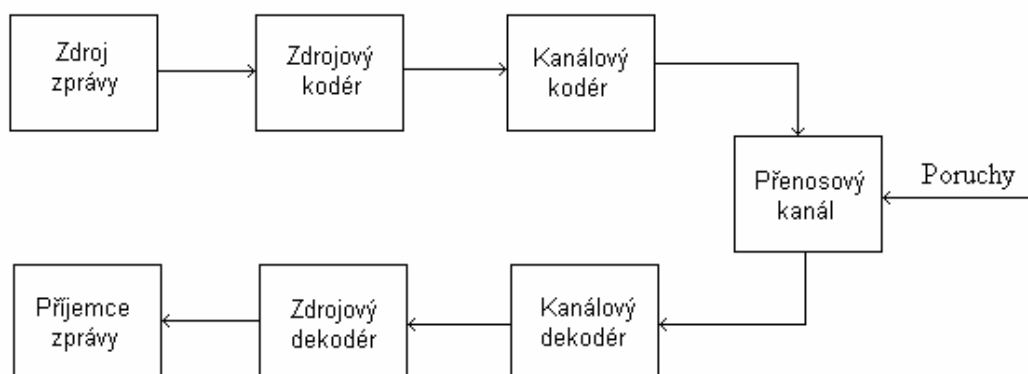
Na vyjádření účinnosti zdrojového kódování se používá parametr, který vyjadřuje efekt zhušťování (komprese) dat a možno ho vyjádřit ve tvaru :

$$Cr = N / N_k \quad (1)$$

kde N – je počet bitů nezakódované zprávy,

N_k – je počet bitů zakódované zprávy.

Pro efektivní kódování musí platit, že $Cr > 1$.



Obr. 6. Proces kódování a dekódování.

3.5 Metody kódování

Kódování a přenos statických obrazů je základní úlohou v telematických systémech, kde hlavními úkoly jsou :

- zkrácení doby přenosu dokumentů – dosahuje se použitím efektivních metod zdrojového kódování, které poskytují větší efekt komprese dat,
- zvýšení spolehlivosti přenosu – možno dosáhnout metodami automatické korekce přijaté informace, resp. vhodnými metodami kanálového kódování,
- zvýšení kvality přenášených obrazů – dosahuje se vyšší hustotou vzorkování originálního obrazu.

Speciální oblast zpracování statických obrazů jsou obrazové data banky, jejich budování si vyžaduje řešení dvou základních problémů :

- efektivní uložení obrazových dat – je potřeba hledat efektivní metody kódování statických obrazů ve smyslu komprese dat,
- speciální přístup k obrazovým datům – problém dálkového listování v obrazových bankách (telebrowsing).

3.5.1 Zdrojové kódování

Umožňuje zvýšit bitovou rychlost na základě redundance a irelevance. Redukce irelevance je proces nevratný (ireverzibilní proces), který způsobuje vždy určitou ztrátu informačního obsahu. Kritériem redukce irelevance jsou vlastnosti příjemce. Redukce redundance je proces vratný (reverzibilní proces) a může se obnovit. Tento proces potlačení redundance je nezávislý na příjemci. Kódování se přizpůsobuje zdroji obrazu za účelem maximálního snížení bitové rychlosti. Optimalizuje se vzhledem k vlastnosti zdroje obrazu.

3.5.2 Kanálové kódování

Optimálně přizpůsobuje kódovaný obraz vlastnostem přenosového kanálu. Spočívá v zavedení takové redundance, aby rušení v přenosovém kanálu způsobilo co nejmenší zkreslení údajů. Efektivní metody digitalizace signálů známe jako metody efektivního (úsporného) číslicového kódování signálů, dosahují kompresy údajů, resp. redukci rychlosti přenosu údajů, při zachování kvality kódování. Kompresi údajů se požaduje za účelem vysoké efektivnosti číslicového zpracování (nepodstatné složky) a redundance (nadbytečné složky) signálu.

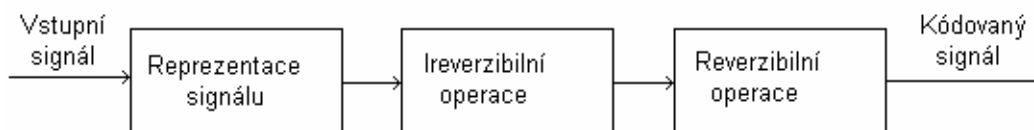
3.5.3 Reverzibilní (bezztrátová) lossless compression

Vychází se z předpokladu, že pro většinu souborů je nepřijatelná jakákoli ztráta dat v důsledku komprimačního a dekomprimačního procesu. Používá se hlavně u binárních a textových souborů, při kódování biomedicínských obrazů a pod.

3.5.4 Ireverzibilní (ztrátová komprese) lossy compression

Používá se tehdy, když chceme dosáhnout dobrého kompresního poměru a nevádí nám, že o určitá data přijdeme. V některých případech ani nepoznáme, že byla ztrátová komprese použita. V tomto případě dokonalá zpětná rekonstrukce původního signálu není možná. Tyto metody se používají často pro paměťově náročnější média, jedná se především o komprimaci obrazu a zvuku.

Všeobecný model efektivního kódování je znázorněn na obr. 7.



Obr. 7. Efektivní kódování signálu.

Pro uvedené problémy je potřeba řešit metody kódování s ohledem na přenos obrazu. Jak vyplývá z doporučení JPEG (Joint Photographic Experts Group), zpracování obrazové informace pro potřeby obrazové databáze by mělo probíhat ve více režimech. Jsou to :

- a) **sekvenční kódování** – obraz se kóduje zleva doprava a směrem shora dolů,
- b) **postupné kódování** – obraz se kóduje a přenáší postupně tak, aby se na stranu příjmu přenesla co nejdříve hrubá informace o obrazu a potom se informace upřesňuje,
- c) **exaktní kódování** – kódování umožňuje po dekódování exaktní rekonstrukci originálního obrazu,
- d) **hierarchické kódování** – kódováním je možno měnit rozlišovací schopnost přijatého obrazu.

Výběr vhodného kódu je závislý především na dalším číslicovém zpracování postupnosti kódových slov.

3.6 Reprezentace obrazu

Vzhledem k cílům číslicového zpracování obrazu je kódování pouze jednou z dalších operací v tomto zpracování. Proto se z tohoto důvodu zavedl pojem **reprezentace obrazu**, které má širší význam než kódování. Kódování, resp. efektivní kódování je speciálním případem reprezentace obrazu.

3.6.1 Heuristická reprezentace obrazu

Využívají zejména statistických vlastností obrazů a využívají základní poznatky z teorie informací. Kódování vizuální informace si vyžaduje vzhledem na vysoký objem vstupních dat odstranit její nadbytečnou část, tj. redundanci, která je v této informaci značně vysoká.

Ve vizuální informaci je možno rozlišit dvě formy redundance :

- statistická redundance,
- subjektivní redundance.

Při heuristické reprezentaci obrazu se využívá blokové kódování binárních a víceúrovňových obrazů a využívá se i aproximační blokové kódování.

Podle velikosti bloku a způsobu jejich dělení rozlišujeme :

- autoadaptivní jednorozměrné kódování,
- autoadaptivní dvourozměrné kódování s jednorozměrným dělením,
- autoadaptivní dvourozměrné kódování s dvourozměrným dělením.

Při heuristické reprezentaci obrazu se využívá kódování délek úseků řádku.

Využívají se :

- Jednorozměrné RL kódy – aplikují se na jeden řádek obrazu, tedy kódují se jednotlivé úseky řádku bílé resp. černé barvy.
- Dvourozměrné RL kódy – aplikují se na více řádků, tedy využívají i vertikální korelaci mezi obrazovými prvky.

Jednorozměrné RL kódy můžeme rozdělit do dvou skupin :

- lineární kódy,
- logaritmické kódy.

Do skupiny jednorozměrných RL kódů patří také tyto kódy :

- zkrácený Huffmanův kód,
- modifikovaný Huffmanův kód.

Dvojrozměrné RL kódy můžeme rozdělit do dvou základních skupin :

- současné kódování n řádků se aplikuje na skupinu n řádků, přičemž počet řádků se volí v rozmezí 2 až 3,
- sekvenční kódování řádek po řádku využívá obsah předcházejícího již zakódovaného řádku a aplikuje se postupně řádek po řádku.

Metody sekvenčního kódování řádek po řádku možno rozdělit na :

- predikční kódování,
- kódování hraničních bodů.

3.6.2 Hierarchická reprezentace obrazů

Využívají topologické vlastnosti a hierarchickou strukturu obrazu. Zakládají se na principu rekurzivní dekompozice datového prostoru, přitom je možno je rozdělit z těchto hledisek :

- dimenze datového prostoru,
- způsob dekompozice (regulérní a neregulérní, adaptivní a neadaptivní),
- adaptivnost dekompozice.

Při hierarchické reprezentaci obrazu se využívá :

- Kvadrantový strom obrazu (KSO)** – slouží k reprezentaci binárních obrazů, které mají tvar segmentu, přičemž pod pojmem segment se rozumí uzavřený obrazec, tvořen černými obrazovými prvky (úroveň „1“).

Kvantový strom obrazu můžeme reprezentovat dvěma základními způsoby :

- Směrníkovou (pointrovou) reprezentací,
- Bezsměrníkovou (bezpointrovou) reprezentací.

Známe tyto reprezentace KSO :

- lineární kvadrantový strom,
- QC reprezentace,
- reprezentace KSO pomocí DF výrazu (DF algoritmus, reprezentace víceúrovňových obrazů DF),
- reprezentace KSO pomocí BF výrazu,
- reprezentace KSO pomocí CD výrazů.

b) Pyramidální reprezentace – pokud máme originální obraz ve čtvercovém rastru $2^N \times 2^N$, potom pyramida je posloupnost $\{P_i\}$, přičemž P_{i+1} představuje obraz, jehož rozměr (rastr) je nějakou částí rozměru obrazu P_i .

Pyramidy můžeme rozdělit podle těchto hledisek :

- typ operace,
- překrývatelnost (překrývající a nepřekrývající se pyramida),
- velikost okna.

Některé nejpoužívanější typy pyramid :

- nepřekrývající se pyramida 2×2 ,
- překrývající se pyramida 4×4 ,
- Gaussova pyramida,
- Laplaceova pyramida.

3.6.3 Množinová reprezentace obrazů

Množinové reprezentace obrazů jsou založeny na poznatcích matematické morfologie a využívají množinové operace s obrazem. Matematická morfologie je matematický aparát na opis struktury obrazu pomocí vhodných transformací.

Jsou to zobrazení přiřazující každé podmnožině nějaké základní množiny jinou podmnožinu té množiny, která jednodušší strukturu (méně prvků). Tuto podmnožinu můžeme potom efektivně kódovat resp. použít na další zpracování.

4 METODY KOMPRESY

4.1 Bezztrátová komprese (lossless compression)

Vychází se z předpokladu, že pro většinu souborů je nepřijatelná jakákoli ztráta dat v důsledku komprimačního a dekomprimačního procesu.

4.1.1 Huffmanovo kódování

Huffmanovo kódování je nejznámějším zástupcem skupiny algoritmů, které pracují na základě různých četností znaků v kódovaných datech. Huffmanovo kódování je komprimační technika, která je určena především pro komprimaci textových souborů.

Při komprimaci se postupuje tak, že nejprve komprimační algoritmus zjistí pravděpodobnosti výskytů jednotlivých znaků (příp. jejich kombinací) a každému znaku (kombinaci znaků) přiřadí jedinečný kód. Takovéto kódy se liší svou bitovou délkou. Tato část algoritmu je nejdůležitější a musí být navržena tak, aby přiřazení kódů znakům respektovalo požadavek na přiřazení bitově nejkratších kódů znakům s častějším výskytem a bitově delších kódů znakům s méně častým výskytem. Pak již jen algoritmus postupně načítá znaky vstupního souboru, nachází odpovídající předem přiřazené kódy a tyto kódy zapisuje na výstup.

Vytvoření binárního stromu si vysvětlíme na konkrétním příkladu. Předpokládejme, že potřebujeme zkomprimovat soubor, který obsahuje pouze znaky **A**, **8**, **0**, **K** a **R**. Dále předpokládejme, že proběhla analýza pravděpodobnosti výskytu těchto znaků v souboru určeném pro komprimaci s následujícím výsledkem:

znak **A** se v souboru vyskytuje s pravděpodobností 50 %,

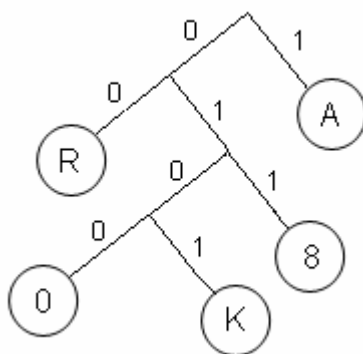
znak **8** se v souboru vyskytuje s pravděpodobností 12,5 %,

znak **0** se v souboru vyskytuje s pravděpodobností 6,25 %,

znak **K** se v souboru vyskytuje s pravděpodobností 6,25 %,

znak **R** se v souboru vyskytuje s pravděpodobností 25 %.

Binární strom podle zásad Huffmanova algoritmu potom tvoříme od koncových „listů“ stromu, tj. od znaků s nejmenší pravděpodobností (viz obr. 8). Vybereme nejprve znaky s nejmenší pravděpodobností výskytu a uděláme z nich dva koncové listy stromu. Potom vybereme znak s nejmenší pravděpodobností z dosud nezpracovaných znaků, a tak postupujeme až ke kořeni stromu.



Obr. 8. Příklad binárního stromu Huffmanova kódování.

Vytvořením binárního stromu jsou definovány kódy přiřazené podle četnosti výskytu jednotlivým znakům souboru. Tyto znaky se přiřadí jednotlivým znakům tak, že ke každému znaku se postupuje stromovou strukturou. Při pohybu od uzlu k uzlu zaznamenáme jedničku vždy při cestě vpravo dolů a nulu při cestě vlevo dolů. Pokud se budeme držet binárního stromu vytvořeného na obr. 8, budou kódy přiřazené jednotlivým znakům vypadat takto:

A: 1
8: 011
0: 0100
K: 0101
R: 00

Komprese na základě vytvořeného binárního stromu probíhá tak, že komprimační algoritmus čte sekvenčně znaky ze vstupního souboru a do výstupního pak zapisuje jim příslušné sekvence bitů.

Dekomprese sice vyžaduje znalost původního binárního stromu, ale je také velice jednoduchá. Pokud jsou kódy ve zkomprimované podobě souboru uloženy bezprostředně za sebou, bude již dekomprimační algoritmus schopen rozpoznat znak příslušející ke kódu. Algoritmus bude prostě číst kód (o kterém ještě není známo, jak je dlouhý) bit po bitu a podle hodnoty bitu bude postupovat po binárním stromě od kořene až k „listu“ stromu představujícímu znak. Tento znak je již původním znakem, který byl při komprimaci zakódován.

Binární strom lze bez problémů sestavit pouze v případě, že pravděpodobnosti výskytu všech znaků vstupního souboru jsou mocninou čísla $1/2$. Pouze v tom případě každé vyšší patro vytvářeného stromu obsahuje uzel nebo list stromu s pravděpodobností výskytu dvojnásobnou oproti listům či uzlům ležícím o patro níž.

Je zřejmé, že ideálního kompresního poměru může být dosaženo pouze při dodržení této zásady. V praxi však s něčím takovým nelze počítat vlastně nikdy, a proto je zapotřebí zaokrouhlovat pravděpodobnosti výskytu jednotlivých znaků.

Zaokrouhlování ovšem musí být prováděno při dodržení dvou základních zásad:

- součet všech procentuálních hodnot musí samozřejmě dávat hodnotu 100% ,
- zároveň je nutné dbát na to, že v každém patře vytvářeného stromu může být maximálně určitý počet listů a uzlů. Tento počet je dále omezen již vytvořenými patry stromu.

Př. Předpokládejme, že máme opět stejný soubor znaků jako v předcházejícím případě, tentokrát ovšem s jinou pravděpodobnostní distribucí:

znak **A** se v souboru vyskytuje s pravděpodobností 46% ,

znak **B** se v souboru vyskytuje s pravděpodobností 22% ,

znak **O** se v souboru vyskytuje s pravděpodobností 5% ,

znak **K** se v souboru vyskytuje s pravděpodobností 5% ,

znak **R** se v souboru vyskytuje s pravděpodobností 22% .

Při vytváření binárního stromu podle zásad Huffmanova algoritmu nalezneme nejprve dva znaky s nejnižší pravděpodobností (**O** a **K**).

Jejich pravděpodobnosti zaokrouhlíme na 6,25 %, což je mocnina $1/2$. Následují znaky **8** a **R** se stejnou pravděpodobností. Zaokrouhlení obou příslušných pravděpodobností na 25 % je nepřipustné, protože na jediný zbývajících znak by zbyla pravděpodobnost 37,5 % – ta je ovšem v binárním stromě nepřipustná. Je tedy nutné pravděpodobnost jednoho z těchto znaků zaokrouhlit na 12,5 % a pravděpodobnost druhého zaokrouhlit na 25 %. To znamená, že ačkoli oba znaky se ve vstupním souboru vyskytují se stejnou pravděpodobností, jeden z nich bude v komprimované formě reprezentován třemi bity a druhý dvěma bity.

4.1.2 Aritmetické kódování

Jak bylo vysvětleno výše, Huffmanovo kódování dosahovalo optimálních výsledků pouze v případě, kdy pravděpodobnosti jednotlivých znaků byly celočíselnými mocninami $1/2$. V ostatních případech by bylo optimální na zakódování jednoho znaku použít neceločíselný počet bitů, což však samozřejmě binární soustava neumožňuje.

Aritmetické kódování tento nedostatek odstraňuje velice zajímavým způsobem – kóduje celou zprávu jako jediné kódové slovo.

Základní myšlenku použitou v této komprimační metodě lze popsat ve stručnosti takto: aritmetické kódování reprezentuje celou zprávu jako číslo z intervalu $<0,1)$. Na začátku kódování uvažujeme celý tento interval. Jak se zpráva prodlužuje, zužujeme postupně interval – přibližují se k sobě i horní a dolní mez nově vytvářeného intervalu. Na konec stačí zapsat libovolné číslo z výsledného intervalu – to samo o sobě reprezentuje celou zprávu.

Algoritmus komprese lze nastínit jako následující sekvenci kroků:

- zjištění pravděpodobností výskytu jednotlivých znaků ve zdrojovém souboru,
- rozdělení intervalu $<0,1)$ na podinterval, jejichž vzájemný poměr velikostí odpovídá poměru pravděpodobností jednotlivých znaků (seřazených podle abecedy),
- uložení tohoto základního rozdělení intervalu $<0,1)$
- vlastní komprese víceznakové zprávy.

Kompresi víceznakové zprávy bude probíhat tak, že se nejprve vybere první znak vstupního souboru. Ten zúží interval $\langle 0, 1 \rangle$ na podinterval příslušející tomuto znaku tak, jak mu byl přidělen v druhém bodě celkového algoritmu aritmetického kódování. Tento podinterval bude rozdělen stejným způsobem jako dříve celý interval $\langle 0, 1 \rangle$. Po načtení dalšího znaku bude podinterval dále zúžen podle načteného znaku. Tak to půjde dále až do načtení posledního znaku zprávy.

Posledním bodem je vybrání kteréhokoli zlomku náležejícího do výsledného nejjemnějšího podintervalu a jeho převedení do binární formy.

Čím je kódovaný znak pravděpodobnější, tím se interval zúží méně.

Nejjednodušeji lze tento algoritmus pochopit na příkladu. Naším úkolem bude za pomoci aritmetického kódování zakódovat tříbajtovou zprávu obsahující znaky „XXY“. V souladu s prvním bodem algoritmu aritmetického kódování zjistíme, že zpráva obsahuje pouze dva znaky:

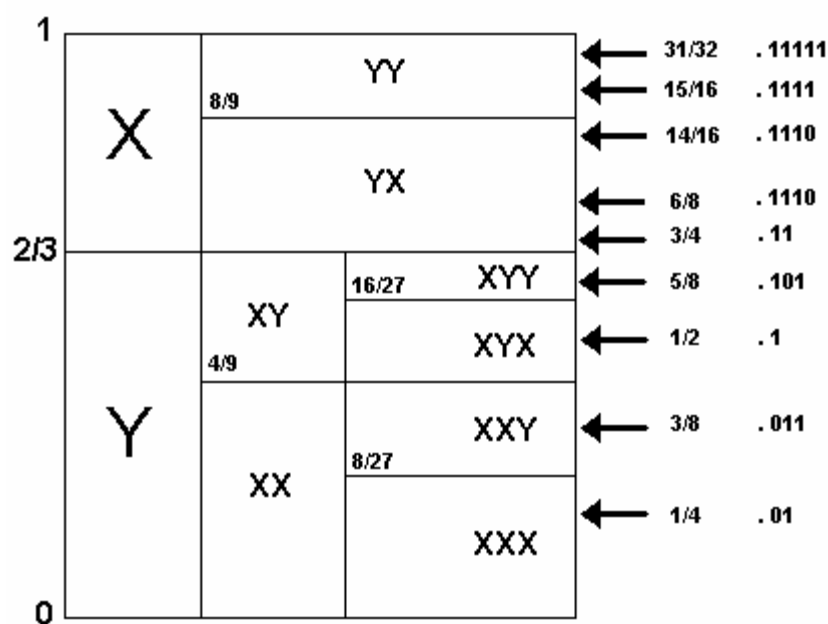
pravděpodobnost výskytu znaku **X** je $2/3$,

pravděpodobnost výskytu znaku **Y** je $1/3$.

V dalším kroku rozdělíme v tomto poměru (v poměru 2:1) interval $\langle 0, 1 \rangle$ a rozdělení si zapamatujeme. Nyní již přistoupíme ke kódování vstupního souboru. Načteme první znak souboru, kterým je písmeno **X** a zjemníme původní interval na podinterval $\langle 0, 2/3 \rangle$. Tento interval opět rozdělíme v poměru 2:1. Dělicím bodem bude v tomto případě bod $4/9$. Načteme další znak – je jím opět písmeno **X**, které nám podinterval opět zúží na $\langle 0, 4/9 \rangle$.

Tento interval opět rozdělíme v poměru 2:1 – dělicím bodem je v tomto případě zlomek $8/27$. Následujícím znakem zprávy je písmeno **Y**, jež opět zúží interval, o který se budeme zajímat, na $\langle 8/27, 4/9 \rangle$. Vzhledem k tomu, že třetí znak je v našem vstupním souboru také znakem konečným, je konečný i tento interval. Zbývá již jen vybrat libovolnou hodnotu z tohoto intervalu, která převedena do binární formy bude reprezentovat celou vstupní sekvenci. Vyberme číslo $3/8$.

Na obr. 9 jsou zachyceny všechny možnosti kódování tříbajtové zprávy sestávající ze znaků **X** a **Y**, jejichž poměr pravděpodobností výskytu je 2:1.



Obr. 9. Diagram všech možností kódování tříbajtové zprávy sestávající ze znaků X a Y, jejichž poměr pravděpodobností výskytu je 2:1.

Dekomprese zakódované zprávy do původního tvaru je prostá. Načteme rozdělení intervalu $<0, 1)$ na podintervaly podle poměru pravděpodobností výskytu jednotlivých znaků a počet znaků původní zprávy. Dále budeme postupovat podle následujícího velice jednoduchého návodu:

- vybereme podinterval $<K(i), K(i+1))$ intervalu $<0, 1)$ tak, aby kód náležel do tohoto intervalu,
- na výstup zapíšeme *znak* náležející svou pravděpodobností do tohoto intervalu $read(kód)$,
- while (není obnovena celá zpráva),
 $\{ kód = (kód - K(i)) / P(znak) \}$.

$P(znak)$ reprezentuje pravděpodobnost výskytu příslušného znaku.

Pokud se vrátíme k našemu příkladu, bude dekomprimace vypadat následovně:

Načteme kód $3/8$, který reprezentuje vstupní posloupnost znaků. Tento zlomek náleží do intervalu $<0, 2/3)$, který odpovídá znaku **X**. Zapíšeme tento znak na výstup. Nová hodnota kódu bude

$$(3/8 - 0) / (2/3) = 4/9$$

Zlomek $4/9$ náleží opět do podintervalu $<0, 2/3)$. Jako druhý znak zapíšeme na výstup opět **X**. Úprava kódu bude nyní následující:

$$(4/9 - 0) / (2/3) = 2/3.$$

Zlomek $2/3$ náleží do podintervalu $<2/3, 1)$. Tento interval odpovídá znaku **Y**, který je tudíž třetím znakem původní zprávy. Víme, že originální sekvence se skládala právě ze tří znaků, a proto dekomprimaci ukončíme.

Tím jste se seznámili se základní myšlenkou komprimace obecně známé pod pojmem *arithmetic coding* (aritmetické kódování). Při převádění této chytré a jednoduché myšlenky do počítačové praxe se však setkáváme s celou řadou problémů – nejzávažnější je asi potřeba konverze reálných čísel na bitové vyjádření. Výsledný kód reprezentující vstupní posloupnost je zapotřebí převést do formy vyjádřitelné ve dvojkové reprezentaci. Navíc s každým reálným číslem rozdělujícím základní interval $<0, 1)$ je nezbytné pracovat s vyloučením jakéhokoli zaokrouhlování.

Při vyjádření výsledného zlomku v bitovém vyjádření je důležitý fakt, že jmenovatel musí být mocninou čísla 2. Jedině tak je možné jednoznačně stanovit délku výsledného bitového vyjádření celého zlomku. Počet bitů bitového vyjádření je takový, aby pomocí nich bylo možné vyjádřit všechny hodnoty v intervalu $<0, \text{jmenovatel}-1>$. Nyní tedy několik jednoduchých příkladů, jak se zlomky konvertují na svá bitová vyjádření (nejnižší bit je uveden vpravo):

Zlomek Bitové vyjádření

$1/2$	1
$1/4$	01
$3/4$	11
$5/8$	101
$31/32$	11111

Huffmanovo kódování a Shannonovo-Fanovo kódování tvoří rodinu příbuzných komprimačních algoritmů založených na využití pravděpodobností výskytu jednotlivých znaků nebo jejich posloupností v komprimovaných datech. Žádná z těchto metod se příliš nepoužívá samostatně, ale dosahuje velmi dobrých výsledků v kombinaci se slovníkovou metodou (LZW). Samostatně má toto kódování praktický význam především při použití na textových souborech.

Výhody:

- velmi dobré kompresní poměry ve spojení se slovníkovou metodou,
- poměrně jednoduchý algoritmus dekomprimace.

Nevýhody:

- Huffmanovo a Shannonovo-Fanovo kódování je nejefektivnější při hodnotách pravděpodobnosti výskytů, jež jsou celočíselnou mocninou čísla $\frac{1}{2}$.

4.1.4 RLE (Run-length Encoding)

Do češtiny je toto kódování překládané někdy jako proudové kódování. Jedná se o komprimační algoritmus, který lze použít na jakýkoli druh dat. Na charakteru těchto dat ovšem velice silně závisí dosažený kompresní poměr. U této metody není dosaženo příliš vysokého komprimačního poměru.

Výhodou této komprese je jeho snadná instalace, jednoduchost celého algoritmu a s tím související poměrně vysoké rychlosti komprimace a dekomprimace.

Základním principem komprimace metodou RLE je zhuštění opakovaných znaků, které se v souboru vyskytují hned za sebou. Řetězec opakujících se znaků se nazývá proud (odtud proudové kódování). Tento proud znaků je vždy zkomprimován do formy jednoho paketu RLE. Paket musí vždy obsahovat dvě informace:

- proudové číslo udávající počet znaků proudu snížený o jedničku,
- následující proudovou hodnotu, jež se shoduje s hodnotou opakujícího se znaku v proudu.

Př. Pokud soubor obsahuje následující sekvenci znaků, **AAAkkRRRRRm** zakóduje se tato sekvence do podoby:

3A1k4R0m

V udané posloupnosti znaků rozpozná komprimační algoritmus čtyři proudy a přiřadí jim čtyři pakety.

Tyto pakety jsou:

- 3A,
- 1k,
- 4R,
- 0m.

Prvním údajem v paketu je vždy proudové číslo (počet znaků proudu snížený o jedničku). Tento údaj je následován proudovou hodnotou znaku. Vidíme, že zmenšení velikosti souboru bylo dosaženo výhradně díky proudům znaků obsahujícím více než dva znaky. Pokud se proud skládá pouze z jediného znaku, komprimovaný paket RLE je dokonce delší než původní proud.

Metoda RLE není vhodná pro komprimaci textových souborů. Kompresní poměr by byl v takovém případě velice špatný. Stejně nevýhodná by byla komprimace většiny binárních souborů. Proto tato metoda nachází uplatnění při komprimaci digitálních obrazů.

Metoda RLE je vhodná pro komprimaci jednoduchých obrázků s malou barevnou hloubkou (maximálně 256 barev). Takové obrázky se vytvářejí například v programu Paintbrush (Malování). Tyto obrázky mají často jednobarevné pozadí, které zaručuje dostatečnou velikost jednotlivých proudů. U těchto obrázků je možné dosáhnout kompresního poměru až 70 %. Naopak pro komprimaci fotografií se komprimační metoda RLE nehodí. Fotografie totiž nesplňují podmínku dlouhých proudů.

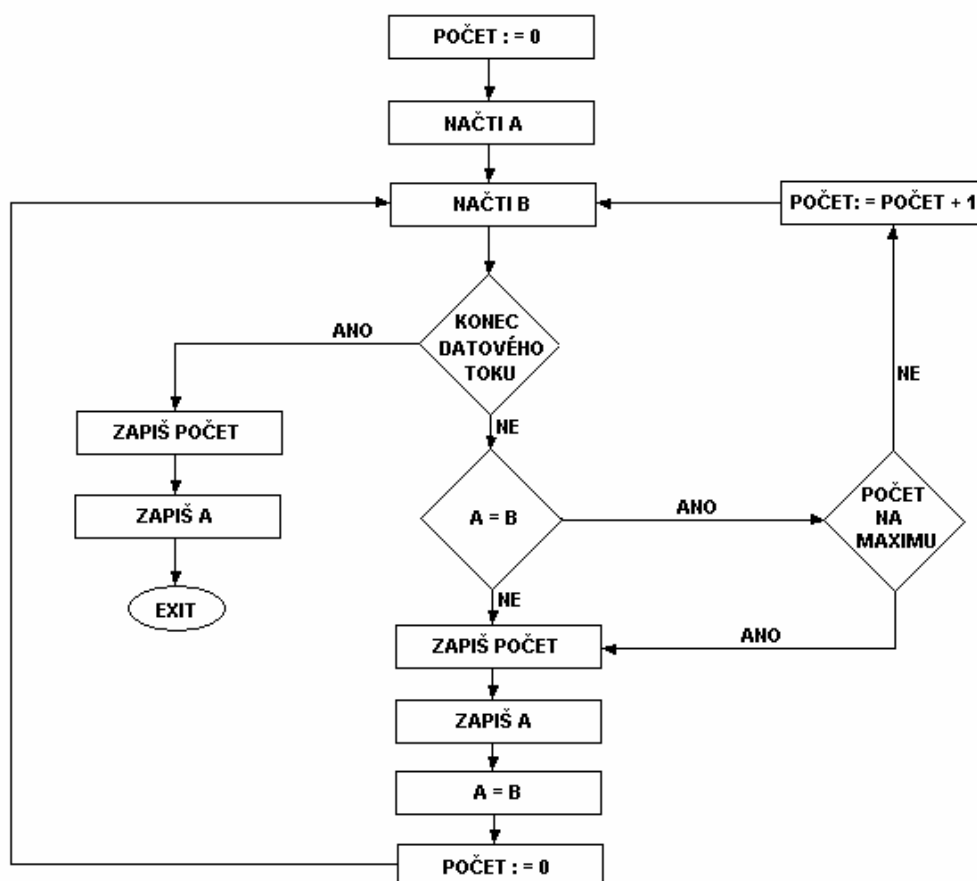
Existují tři základní přístupy k praktické realizaci algoritmu RLE ve vztahu k bitovým mapám jednoduchých obrázků. Ve všech těchto přístupech jsou obvykle původní data předlohy proudově kódována sekvenčním postupem, který přistupuje k datům jako k jednorozměrnému poli dat a nikoli jako k dvourozměrné mapě dat.

Bitová mapa se začíná číst a komprimovat od levého horního rohu a postupuje se buď po řádcích nebo sloupcích jednotlivých pixelů až do pravého dolního rohu.

Při komprimaci je však vhodné udržovat informaci o koncích jednotlivých řádků (resp.sloupců) bitové mapy. Tyto značky lze potom s výhodou použít, pokud je potřeba rozbalit pouze určitou část obrázku (např. 101. až 180. řádek původní předlohy). V takovém případě dekomprimační algoritmus nemusí dekomprimovat prvních 100 řádků obrázku pouze proto, aby našel požadovaný začátek výřezu. Dekomprimace může začít přímo na 101. řádku.

Bohužel některé programy podporující grafické formáty s kompresí RLE informaci o konci řádků či sloupců do komprimované podoby obrázků neukládají.

Základní struktura algoritmu RLE je zobrazena na obr. 11.



Obr. 11. Základní schéma algoritmu RLE.

Praktické přístupy ke kódování bitových map se dělí podle typu základních datových prvků tvořících proudy.

Podle typů těchto základních prvků existují tři praktické modifikace:

- A) bitová úroveň
- B) bajtová úroveň
- C) pixelová úroveň

- ad A) bitová úroveň – kódování metodou RLE, rozeznává pouze dva proudy znaků – proud jedniček a proud nul. Taková metoda je účinná prakticky pouze na monochromatické jednobitové obrázky. Tato data obsahují dostatečně dlouhé proudy jedniček i nul k tomu, aby metoda RLE v této modifikaci dosáhla dobrých kompresních poměrů.
- ad B) bajtová úroveň – tato modifikace kóduje proudy opakujících se bajtových hodnot a naopak si nevšímá dělení na bity nebo hranic šestnáctibitových (případně 32 bitových) slov. Paket RLE se v tomto případě skládá ze dvou bajtů. První bajt udává proudové číslo v rozsahu 0-255 a druhý bajt proudovou hodnotu. Její rozsah je samozřejmě opět 0-255. Zde tedy může velikost jediného proudu, kterému bude odpovídat jeden RLE paket, dosahovat délky až 256 znaků. Metoda RLE na úrovni bajtů je vhodná pro obrázky, které jsou uloženy ve formě jeden bajt na pixel.
- ad C) pixelová úroveň – tato metoda se používá u složitějších obrázků, kdy je jediný pixel reprezentován více než jedním bajtem dat. Velikost paketu RLE se v tomto případě liší podle počtu bajtů na pixel. Pokud např. obrázek podporuje schéma dat „3 bytes per pixel“ (3 bajty na pixel), bude paket RLE čtyřbajtový – první bajt jako proudové číslo a následující tři bajty jediné proudové hodnoty. Kódovací metoda zůstává stejná jako u jednobajtové úrovně a opět lze využít modifikace s indikátorem přímého paketu.

Výhody:

- jednoduchost algoritmu a jeho snadné použití vedoucí k vysoké komprimační i nekomprimační rychlosti.

Nevýhody:

- úzká oblast dat, na kterých tato metoda dosahuje dobré kompresní poměry,
- komprimace RLE je vhodná pouze pro jednoduché obrázky s malou barevnou hloubkou (např. obrázky z aplikace Paintbrush). Nehodí se ani pro složitější obrázky ani pro textové nebo binární soubory.

4.1.5 LZW (Lempel-Ziv-Welch algoritmus)

Komprimační algoritmus Lempel-Ziv-Welch (LZW) je jednou z nejrozšířenějších komprimačních metod, kterou používají (v různé formě) jak kompresní programy (např. ARJ, PKZIP, ZOO, LHA atd.), tak i různé grafické formáty obrázků. Jedná se o tzv. substituční (adaptivní slovníkovou) metodu. Základním principem tohoto kompresního algoritmu je vyhledávání stejných posloupností bajtů v originálním souboru. Pomocí odkazů na tyto posloupnosti dat algoritmus buduje datový slovník.

Slovníková komprese LZW je založena na postupném rozpoznávání algoritmu a ukládá do tabulky řetězce znaků a tyto řetězce nahrazuje ve výstupním textu přirozenými čísly z předem definovaného intervalu. Definice intervalu je závislá na charakteru komprimovaných dat.

Komprimace probíhá podle následujícího schématu:

- pokud se posloupnost bajtů (řetězec) ve vytvářeném slovníku nevyskytuje, je tato posloupnost přidána do slovníku a v nezměněné formě zapsána do komprimovaného výstupního toku dat.
- pokud se vstupní posloupnost bajtů ve slovníku již nachází, zapíše se do výstupního toku dat pouze zástupná slovníková hodnota odpovídající nalezené vstupní posloupnosti.

Díky tomu, že zástupná hodnota je vždy menší než čtená posloupnost, dochází ke kompresi dat.

Dekomprimace je inverzním procesem, při kterém algoritmus čte komprimovaný tok dat a stejným způsobem vytváří datový slovník posloupností (řetězců) bajtů. Dekompresor tedy postupně čte komprimované kódy, zapisuje na výstup příslušné řetězce a přidává nové řetězce do slovníku.

Jednou z důležitých předností, kterou je třeba vyzdvihnout, je využitelnost algoritmů LZW u takových zařízení, kterými data pouze procházejí a kde je žádoucí data zpracovat rychle. Takovými zařízeními jsou např. modemy. Zde je totiž třeba rychlé zpracování dat a vyslání po lince k příjemci.

Výhody:

- velmi dobrý kompresní poměr (zvláště při aplikaci některé statistické metody na výsledný kód),
- rychlá komprese i dekomprese a nevelké nároky na paměť,
- adaptivní metoda vytvářející dynamický substituční slovník. Tento slovník není nutné ukládat pro potřeby dekomprese – dekompresní část algoritmu vytvoří duplicitní slovník přímo při dekompresi,
- možnost kontinuálního vysílání zkomprimovaných dat (např. modemem) bez nutnosti čekání na dokončení komprimace celého datového bloku.

Nevýhody:

- rychlý nárůst slovníkových kódů odkazujících na řetězce původního souboru vede v některých případech k zaplnění paměti určené pro slovníkové kódy. Z toho pak vyplývá smazání aktuálního slovníku a jeho nové vytváření bez možnosti použití „zapomenutých“ odkazů. V případě větších slovníků zase narůstá doba vyhledání řetězce v tabulce.

4.2 Ztrátová komprese

Ztrátové komprimační algoritmy kódují data způsobem, který nejen že nezaručuje stoprocentní obnovu dat po dekomprimaci, ale dokonce schválně některé informace při kompresi ztrácí. Je zřejmé, že taková kompresní schémata lze použít pouze tam, kde existuje určitá redundance (nadbytečnost) vstupních dat a kde ztrátou některých původních informací datový soubor neutrpí žádnou újmu. Takovými daty mohou být statické obrázky, zvuková data, data uchovávající video sekvence atp.

Výše uvedená data jsou redundantní ve vztahu k nedokonalosti lidských smyslů, kterými tato data vnímáme. U obrázků existují detaily, které lidské oko prakticky nemůže postihnout – především malé změny barev sousedních bodů. Stejně tak ve zvukových a videosekvenčních datových souborech je možné nalézt redundanci založenou na podobném principu.

4.2.1 JPEG (Joint Photographic Experts Group)

Je jednou z nejpokrokovějších metod ve ztrátové kompresi redundantních grafických předloh. Joint Photographic Experts Group – komise pro standardy pracující pod International Standard Organization (ISO). Na rozdíl od ostatních kompresních schémat popsaných dosud není JPEG samotným, přesně definovaným algoritmem. Jedná se spíše o sadu kompresních metod, které mohou být přizpůsobeny konkrétním požadavkům uživatele. Z toho je zřejmé, že JPEG není ani přesně definovaným grafickým formátem.

JPEG byl navržen pro práci s barevnými předlohami i s předlohami obsahujícími pouze různé stupně šedé. JPEG je velice vhodný pro kompresi fotografií s velkou barevnou hloubkou. Horších výsledků dosahuje při práci s vektorovými obrázky a jednoduchou grafikou.

Celá komprimační technologie je založena na faktu, že lidské oko dokáže hůře rozeznat malé změny barev blízkých bodů než rozdíly v intenzitě a jasu. Z toho důvodu je také tento standard vhodnější pro barevné obrázky než pro černobílé předlohy skládající se pouze z mnoha odstínů šedé.

Kompresní standard JPEG je tedy vhodný pro kompresi složitých barevných grafických předloh (fotografií) s velkou barevnou hloubkou – typicky 24 bitů na pixel (tj. 16 milionů barev). Kompresní poměr běžně dosahuje u takových grafických předloh hodnot 20:1 bez vizuální újmy na kvalitě obrázku.

Je třeba zdůraznit, že uvedený kompresní poměr se opravdu týká digitálních fotografií a vůbec obrázků s velice rozmanitou skladbou, a to jak barevnou, tak i co do drobných detailů.

JPEG vyžaduje alespoň 16 odstínů šedé (4 bity na pixel). Poznamenejme však, že např. GIF je bezztrátový až do 256 odstínů šedé, zatímco JPEG nikoli. Naopak barevná fotografie exportovaná např. do formátu GIF tedy nejprve ztratí většinu své barevné hloubky, přičemž se dosáhne kompresního poměru 3:1. Grafický formát GIF dále používá bezztrátovou kompresní metodu Lempel-Ziv-Welsh (LZW), která zvýší kompresní poměr v průměru na konečnou hodnotu 5:1. Ve srovnání s kompresním poměrem 20:1 u metody JPEG je to velice málo uspokojivý výsledek, dosažený navíc za cenu ztráty většiny barevné informace.

Základní verze kompresního schématu JPEG funguje tak, že celý obrázek je dekomprimován z formátu JPEG během jediného skenu od horního k dolnímu okraji obrázku.

Progresivní JPEG

Je určen speciálně pro publikování na stránkách WWW. Při použití progresivního JPEGu se celý obrázek zobrazuje v několika po sobě následujících skenovacích přebězích, během nichž se zlepšuje kvalita obrázku. Po prvním skenu se zobrazí pouze hrubé kontury obrázku atd. Kvalita obrázku se tak progresivně zlepšuje. Výhodou tohoto přístupu je fakt, že k prvnímu náhledu na celý obrázek postačí zpracování malého zlomku celkových dat kódujících původní grafickou předlohu.

Progresivní JPEG je ohromnou výhodou při používání pomalého dial-up (po telefonní lince) připojení k serveru WWW. Již po přenesení malého množství dat uživatel vidí základní kontury celého obrázku. Při použití tzv. základního JPEGu musí uživatel počkat na přenesení celého obrázku, aby zjistil, co grafická data zobrazují. Určitou nevýhodou tohoto druhu kódování je fakt, že při každém jednotlivém skenovacím přeběhu musí počítač provést prakticky stejné množství matematických operací jako při jediném přeběhu „základního“ JPEGu. Při dnešních výkonech počítačů je však limitujícím faktorem právě přenosová rychlost modemů. Pokud se data na vstupu dekodéru objevují dostatečně rychle, lze dekodér upravit tak, že přeskakuje některé zobrazovací přeběhy. To znamená, že ani majitelé rychlých připojení k Internetu nebudou trpět větší matematickou náročností při dekódování předlohy.

Standard ztrátové komprese JFIF (JPEG File Interchange Format)

Jelikož označení *JPEG* není ani přesným kompresním algoritmem ani konkrétním grafickým formátem, bylo nutno standardizovat některý formát založený na ztrátové kompresi. Tímto standardem se stal *JFIF*. Tento formát byl upraven speciálně pro potřeby Internetu a obecně se o něm mluví jako o „formátu *JPEG*“.

Standardní soubor *JFIF* začíná čtveřicí bajtů – hexadecimálně *FF D8 FF E0*, následují dva bajty, jejichž hodnota se liší soubor od souboru (většinou jsou to ale hexadecimální hodnoty *00* a *10*), dalšími čtyřmi znaky je řetězec '*JFIF*'. Tyto sekvence znaků jsou tedy identifikátorem standardu pojmenovaného jako *JFIF*.

Kompresní algoritmus JPEG

Komprimační schéma JPEG se skládá z následujících částí:

- Transformace barev – snímky (pokud používají jiný barevný model, např. *RGB*) se převedou do barevného modelu *YUV* (nebo *YCbCr* atd.), kde je samostatně uchována jasová (*Y*) a dvě barvonosné složky (*U* a *V*). Tento mód je pro další zpracování daleko vhodnější než obvykle v počítači užívaný mód *RGB*. Důvodem je již výše zmíněná vlastnost lidského oka lépe rozeznávat změny v intenzitě (jasu) a nikoli v malé změně barvy.
- Redukce barev – pro zmenšení objemu obrazových dat se používá tzv. podvzorkování barev. Jasové složky obrazu se zachovávají všechny beze změny, ale barevné složky jsou redukovány v poměru *2:1* v horizontálním a *2:1* nebo *1:1* (beze změny) ve vertikálním směru. Snížení počtu barev je prováděno průměrováním na menší počet hodnot. Podvzorkování barev je vhodné především u složitých barevných předloh, jak bylo vysvětlen dříve. Další transformace se již provádějí odděleně a nezávisle pro jasovou i obě barevné složky.
- Diskrétní kosinová transformace – obraz se rozdělí do čtvercových oblastí 8×8 bodů a na každou takovou oblast se aplikuje diskrétní kosinová transformace. Tato transformace v podstatě převede amplitudovou informaci na informaci frekvenční, u které lze jednoduše rozlišit, které části informace jsou pro danou oblast obrazu dominantní (nižší frekvence), a které části popisují pouze jemné detaily (vyšší frekvence). Získáme tak opět matice 8×8 .

Na rozdíl od původních matic, kde byla obrazová informace rozložena rovnoměrně (pro každý pixel jsme znali jeho barevnou hodnotu), po aplikování transformace se informace s největším vlivem dostaly do levého horního rohu a se stoupající vzdáleností od něj důležitost informací klesá. Tím se dominantní a důležitější složky obrazu reprezentované nižšími frekvencemi oddělí od drobných detailů (vyšších frekvencí) a obraz se tak připraví k další redukci vypuštěním těchto drobných detailů.

- Kvantizace – až do této fáze se v úpravách obrazu jednalo o vratné operace a nedocházelo (kromě podvzorkování barev) ke ztrátě informací. Nyní přichází fundamentální krok: provedením kvantifikace dochází k nevratným změnám, proto je komprese JPEG nazývána kompresí ztrátovou.

Každá ze 64 frekvencí jednotlivých bloků po diskrétní kosinové transformaci se vydělí kvantizačními koeficienty (které jsou součástí předem experimentálně připravených kvantizačních matic) a zaokrouhlí na celá čísla. Podle volby kvantizační matice můžeme ovlivnit stupeň komprese a tím i výslednou kvalitu obrázku.

- Kódování – po provedení kvantizace dostaneme řídkou matici, která kromě několika koeficientů v levém horním rohu obsahuje samé nuly. Tuto matici převedeme do posloupnosti čísel, přičemž postupujeme po diagonálách od levého horního k pravému dolnímu rohu (tzv. cik-cak algoritmus). Takto získaná posloupnost se zakóduje Huffmanovým kódováním. Právě tato část způsobí významnou kompresi obrazových dat.
- Konečná fáze – pod pojmem konečné fáze se rozumí přidání hlavičky ke komprimovaným datům. Aby dekompresor mohl správně rekonstruovat původní grafická data, je nutné zahrnout do hlavičky kvantizační tabulku a tabulku Huffmanových kódů.
- Dekomprese – při dekompresi JPEG se používá inverzního postupu. Vzhledem k tomu, že inverzní postup není z matematického hlediska o moc jednodušší, nebývá doba potřebná pro dekompresi o mnoho kratší než při původním kódovacím procesu.

Výhody:

- mnohem lepší kompresní poměr než u bezztrátové komprese,
- díky výbornému kompresnímu poměru velice vhodný formát pro publikaci obrázků na stránkách WWW. Progresivní forma standardu JPEG je vhodná i pro uživatele s pomalým připojením přes modem.

Nevýhody:

- nevhodný pro jednoduché barevné předlohy s rovnými ostrými barevnými hranami a velkými jednobarevnými plochami,
- poměrně značná složitost komprimačního i dekomprimačního algoritmu promítající se do poměrně dlouhé doby zpracování.

4.2.2 MPEG (Moving Picture Expert Group)

MPEG je zkratka pro Moving Picture Expert Group, komisi, která se zabývá vývojem standardů pro komprimaci videa a připojeného audiosignálu. Skupina pracuje pod organizací ISO – International Standards Organization.

Existuje několik norem pro kompresi videa MPEG, které vyhovují různým oblastem využití.

MPEG-1

Tato nejpoužívanější norma je navržena s ohledem na technologii CD tak, že nejvyšší datový tok je až 1,5 Mb/s. Běžným uživatelům nejdostupnější komprimační a přenosový standard pro přenos videa a s ním souvisejícího audiosignálu. V principu je norma MPEG-1 definována až do velikosti obrázku 4095×4095×60 (60 snímků za sekundu). Norma MPEG-1 sestává ze čtyř částí:

- IS 11172-1 popisuje synchronizaci a multiplexaci videa i zvukového signálu,
- IS 11172-2 popisuje kompresi neprokládaného videosignálu,
- IS 11172-3 popisuje kompresi připojeného audiosignálu,
- IS 11172-4 popisuje testování shody přenesených dat s původními.

Kompresce obrazové informace, je velice podobná kompresnímu schématu JPEG. Každý snímek je rozdělen do bloků 8×8 pixelů, na tyto bloky je použita diskretní kosinová transformace. Výsledné koeficienty transformace jsou dále kvantovány. (Kvantizace vlastně znamená dělení čísla jinou hodnotou. Tím se dosahuje snížení počtu bitů nesoucích příslušnou informaci na úkor informace obsažené v nejméně významných bitech.)

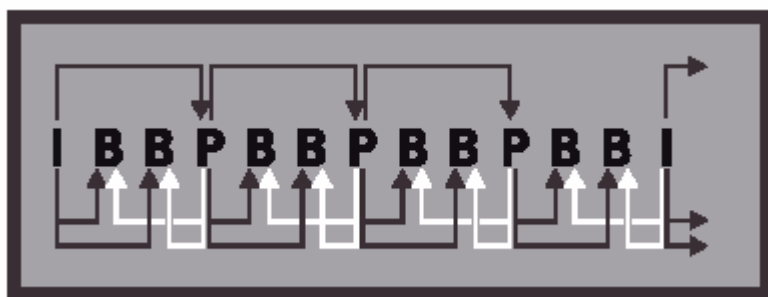
Posledním krokem je opět bezztrátová komprimační část zastoupená Huffmanovým kódováním. Tato poslední část je použita jak na kvantované koeficienty pocházející z diskretní kosinové transformace, tak i na tzv. pohybové vektory, které jsou ve schématu MPEG (ve srovnání s kódováním statických obrázků) zcela nové.

Oproti kódování jednoduchého statického obrázku musí být do videosekvence zahrnuta právě navazující sekvenčnost jednotlivých snímků. Základní myšlenka je založena na předpovědi „pohybu“ jednotlivých částí obrázku od snímku ke snímku.

Pro vyvážení kvality přehrávaného videa spolu s co možná nejlepšími kompresními poměry používá norma *MPEG-1* trochu komplikované schéma používající tři typy snímků.

- Prvním z nich je statický snímek **typu „I“** (intraframe). Tyto snímky představují statické snímky tak, jak je lze vidět, pokud rozvinete klasický film zhotovený chemickou cestou. Tyto snímky neobsahují žádnou informaci o snímcích předcházejících. Existence takových snímků je pochopitelná – např. první snímek celého videa musí být právě typu „I“.
- Druhým typem je **typ „P“** (predicted – předpovězený). K interpretaci tohoto snímku – z pohledu dekompresoru – je nutná znalost předchozího dekódovaného snímku typu „I“ nebo „P“. V obrázku typu „P“ je tedy obsažen odkaz na jeden z předcházejících snímků a snímek sám již obsahuje i pohybové vektory. Tyto pohybové vektory určují, jak se změnily jednotlivé detaily předcházejícího snímku, který je již plně rekonstruován.
- Posledním druhem snímků, které se používají v kompresní a přenosové normě MPEG-1, je **typ „B“** (bidirectional – obousměrný). Jak již název tohoto snímku napovídá, k dekódování takového obrázku je zapotřebí znalost dvou nejbližších snímků typu „I“ nebo „P“ – jednoho, který se v sekvenci snímků nalézá před a jednoho v sekvenci za příslušným snímkem typu „B“.

Při tvorbě snímku typu „B“ (z pohledu kompresního algoritmu) se bere v úvahu „dopředný“ i „zpětný“ pohybový vektor pro každý blok kódovaného snímku a průměrováním příslušných bloků snímku „budoucího“ i „minulého“ se kódovací algoritmus pokouší dobrat nejlepšího výsledku (tzv. interpolační kódování, resp. *interpolace*).



Obr. 12. Sekvence snímků videa formátu MPEG-1.

Dále se zaměříme na kompresi zvuku

Kompresi audio signálu popisuje ISO 11172 - 3 normy MPEG-1. Existují tři kódovací schémata nazývaná Layer 1 až Layer 3. Zvyšující se číslo „vrstvy“ vyjadřuje zvyšující se kvalitu zvuku a tím i složitost kódovacího procesu. Pro každou vrstvu je specifikován formát datového toku a dekodér. V souladu se zvyšující se náročností a složitostí použitého algoritmu se zvyšujícím se číslem vrstvy je použita hierarchická kompatibilita vrstev. To znamená, že dekodér postavený na vrstvu N je schopen dekódovat i vrstvu s nižším číslem.

Celkový datový tok MPEG-1 může být maximálně 1,5 Mb/s – z toho je 1,2 Mb/s rezervováno pro video data a 0,3 Mb/s pro audio data. Pro srovnání: datový tok u CD (stereo, 16 bitů a 44,1 kHz) je 1,4 Mb/s. MPEG podporuje kompresní poměry od 1:2,7 až po 1:24.

Kompresní poměr 1:6 (256 kb/s) se jeví co do kvality při vhodných poslechových podmínkách podle zkušeností jako naprosto nerozeznatelný od originálu. Vyšší kompresní poměry se již mohou projevit na kvalitě reprodukováného zvuku, protože i v případě komprese zvuku se jedná o ztrátovou kompresi.

Kompresi je u tohoto ztrátového modelu založena na stejném principu jako využití redundance informací v grafických datech – kompresor vynechá detaily, jež jsou pro lidské

ucho nepostižitelné (redukce irelevance zvukového signálu). Průměrné lidské ucho je schopno zachytit zvuk přibližně v těchto mezích:

- frekvenční rozsah $16\text{ Hz} - 20\text{ kHz}$
- dynamický rozsah (ticho - hluk) asi 96 dB .

Citlivost je u lidského ucha nelineární a právě toho se využívá.

- První metoda zvaná **Frequency Masking** je založená na tom, že lidské ucho není schopno rozlišit v přítomnosti silného signálu signál slabší, který tak zanikne.
- Druhá metoda zvaná **Temporal Masking** je založena na setrvačnosti vjemu zvuku. Např. když budeme přehrávat signál 1 kHz hlasitostí 60 dB a k němu ještě tón $1,1\text{ kHz}$ hlasitostí 40 dB , bude druhý tón překryt a tudíž bude neslyšitelný. Poté co vypneme silnější signál, bude tón ještě asi 5 ms neslyšitelný. Tuto prodlevu lze taktéž využít ke zjednodušení dat.

Jak již bylo uvedeno, formát MPEG definuje v základu 3 typy vrstev. Data se dělí na tzv. frames (rámce). Každý frame se pak skládá z 384 jednotlivých vzorků.

- Layer 1 – dělicí filtr pracující s kritickým pásmem (critical band - poměr hlasitostí a závislost citlivostí na frekvenci), je aplikován na jednotlivé rámce (frames) a používá stejnou frekvenci. Psychoakustický model zde používá pouze frekvenční maskování (frequency Masking). Datový tok se může pohybovat od 32 kb/s do 448 kb/s
- Layer 2 – oproti Layer 1 dělicí filtr pracuje se třemi rámcemi najednou (předchozí, současný a následující = 1152 vzorků) a je zde použito jednoduché dočasné maskování (temporal masking). Soubory mají koncovku *.mp2 a datový tok je z intervalu od 32 kb/s do 384 kb/s .
- Layer 3 – používá lepší dělicí filtr s proměnnou frekvencí. Psychoakustický model zde používá plně dočasné maskování (temporal masking) a redukce vazeb mezi stereo signály. Soubory mají koncovku *.mp3. Datový tok je z intervalu od 32 kb/s do 320 kb/s .

Samozřejmě je možné kompresní schéma MPEG používat i pro přehrávání a kopírování zvukových CD nosičů. **Je ovšem nutné připomenout, že při tom nesmějí být porušena práva pro šíření a kopírování nahrávek.**

Programům, které dokáží přímo z audio CD uložit skladby do formátu WAV, se obecně říká „grabovací programy“ (CD Rippers). Podmínkou je ovšem dobrá mechanika CD-ROM, která umožňuje načítat audio stopy na CD jako datové soubory.

Soubory MP3 používají kompresi MPEG Audio Layer 3. Když se při kompresi stereo audio souboru typu WAV (stereo, 16 bitů a 44,1 kHz) použijí parametry 128 kb/s a vzorkovací frekvence 44,1 kHz, získá se soubor ve formátu MP3, který se kvalitou blíží audio CD.

Největší výhodou MP3 je velikost souboru – kompresní poměr při CD kvalitě dosahuje 12:1. Na jedno datové CD je tak možné uložit 12 až 13 klasických audio CD. Toto datové CD je pak možné přehrávat v kvalitě, která je nerozeznatelná od originální nahrávky.

MPEG-2

Tato norma je navržena s ohledem na využití v dálkových a satelitních přenosech signálu při zachování televizní kvality. Norma MPEG-2 umožňuje rozlišení až do 16383×16383 bodů, jediné omezení je, že výška i šířka snímku musí být dělitelná 16 pro lepší rozdělení na oblasti při komprimaci.

MPEG-3

Tato norma byla původně myšlena jako podpora HDTV, tedy televize s vysokým rozlišením, ovšem tuto oblast byla po úpravách schopna pokrýt i norma MPEG-2. Od normy MPEG-3 se tedy upustilo a dále se nepoužívá.

MPEG-4

Na rozdíl od rostoucích požadavků na datový tok v předcházejících normách je norma MPEG-4 definována pro přenos videa a připojeného audiosignálu po pomalých linkách s rychlostí od 4800 do 64000 bitů/s, tedy převážně po modemech. Tato rychlost je velice malá a MPEG-4 proto vychází z rozlišení 176×144 bodů při 10 snímcích za sekundu.

Výhody:

- možnost ztrátové komprese představuje velkou výhodu digitalizovaného videa a audia oproti běžným analogovým technologiím. Při kompresi sekvence videosnímků je možné použít tzv. pohybové vektory udávající, jak se změnil snímek oproti jinému snímku sekvence. Není tedy nutné každý snímek videa kódovat jako statický – to je výhoda, která umožňuje dosáhnout ještě lepších kompresních poměrů. U audio CD lze dosáhnout uložení 12 až 13 audio CD na jediné datové CD při prakticky nezměněné kvalitě.

Nevýhody:

- samotná komprimace videa je složitým procesem jak na paměť, tak i na početní výkon počítače. Ačkoli dekodování (přehrávání) je mnohonásobně jednodušší, i zde jsou paměťové nároky a především početní výkon počítače limitující pro čistě softwarové zpracování dat ve formátu MPEG. K dosažení profesionálních výsledků v této oblasti je nezbytné využít přídatné hardwarové zařízení (speciální karty pro přehrávání videa).

ZÁVĚR

Cílem bakalářské práce bylo popsat metody zdrojového kódování digitálních signálů. Metody zdrojového kódování digitálních signálů se prakticky využívají např. při přenosu obrazových video sekvencí v rámci CCTV (Closed Circuit Television). Praktickým ukazatelem zdrojového kódování je komprese signálů. Telekomunikačními linkami lze přenášet větší objem dat v komprimovaném tvaru (za danou časovou jednotku), nebo daný (komprimovaný) signál lze přenášet přes telekomunikační linky v kratším čase.

Komerční praxe stimuluje požadavky přenosu signálů (převážně obrazových) v co největší kvalitě a v co nejkratším čase. Řešením těchto praktických požadavek jsou právě metody zdrojového kódování signálů (tzv. komprese signálů).

Bakalářská práce je členěna na čtyři kapitoly.

První kapitola se zabývá některými pohledy na současné telekomunikační sítě a jejich druhy.

Druhá kapitola popisuje obecné principy zdrojového kódování signálů, představuje základní druhy a parametry komprimace signálů.

Třetí kapitola bakalářské práce se věnuje problematice zpracování obrazových signálů. Popisuje jednotlivé dělení a typy obrazů, způsoby jejich reprezentace a komprese, včetně kompresních standardů.

Čtvrtá kapitola popisuje jednotlivé frekventované kompresní techniky. V rámci bezztrátové komprese popisuje Huffmanovo kódování, Shannonovo-Fanovo kódování a aritmetické kódování digitálních signálů. Metody ztrátového kódování se soustředí zejména na popis JPEG a MPEG standardů.

Problematika zdrojového kódování signálů je problematika interdisciplinární. Standardizace kompresních technik umožnila průnik moderních informačních a komunikačních technologií i do oblasti bezpečnostního průmyslu. Proto použití algoritmů a metod zdrojového kódování signálů lze očekávat i v budoucnosti.

SEZNAM POUŽITÝCH ZDROJŮ

- 1) ADÁMEK, J. *Kódování*. Praha: SNTL, 1989.
- 2) ANSON, L. F. Komprese obrázku - fraktály. In: *Bajt*. č. 10/1993, s. 99 – 103.
- 3) BAHENSKÝ, Z. Huffmanův kód. In: *Elektronika*. č. 1/1990, s. 27 – 28.
- 4) ČAPEK, J., FABIAN, P. *Komprimace dat - principy a praxe*. Praha: Computer Press, 2000-
- 5) JELÍNEK, J. Komprese dat. In: *PC World*. č. 12/1992, s. 28 – 31.
- 6) PROCHÁZKA, T. *Komprimace signálu v ISDN*. Brno: VUT, 2002.
- 7) PTÁČEK, M. *Digitální zpracování a přenos obrazové zprávy*. Praha: Nakladatelství dopravy a spojů, 1983.
- 8) MICROSOFT - <http://www.microsoft.com>

SEZNAM OBRÁZKŮ

Obr. 1. Struktura přenosových systémů a sítí.	6
Obr. 2. Všeobecný model číslicového zpracování obrazové informace.....	10
Obr. 3. Geometrický rozměr obrazu.	12
Obr. 4. Obrazové pole a prvky.....	12
Obr. 5. Informační obsah obrazu.	13
Obr. 6. Proces kódování a dekódování.	18
Obr. 7. Efektivní kódování signálu.....	20
Obr. 8. Příklad binárního stromu Huffmanova kódování.	25
Obr. 9. Diagram všech možností kódování tříbajtové zprávy sestávající ze znaků X a Y, jejichž poměr pravděpodobností výskytu je 2:1.	29
Obr. 10. Příklad binárního stromu vytvořeného Shannonovou-Fanovou metodou.....	31
Obr. 11. Základní schéma algoritmu RLE.....	34
Obr. 12. Sekvence snímků videa formátu MPEG-1.	44

SEZNAM TABULEK

Tabulka I. Paměťové a přenosové nároky jednotlivých typů médií	5
--	---