Metody redukce bitového toku při přenosu obrazového signálu

Josef Pelikán
UK Praha, matematicko-fyzikální fakulta*

Cílem tohoto příspěvku je seznámit čtenáře se základními postupy a algoritmy používanými pro úsporné uložení nebo přenos digitálních obrazových dat. Komprese obrazu vychází z obecných metod komprimace dat založených na statistickém nebo slovníkovém principu, využívá však některých speciálních vlastností obrazových dat. Často se po ní nevyžaduje 100% věrnost, naopak vždy je žádoucí maximální kompresní poměr, řádově 10:1 až 100:1 při vyhovující kvalitě přenosu.

V tomto materiálu je velká pozornost věnována matematickým základům kompresních algoritmů, zejména ortogonálním systémům funkcí a odpovídajícím transformacím používaným v blokově-kvantovacích metodách. Jsou zavedeny a popsány: spojitá a diskrétní Fourierova transformace a jejich vztah k Fourierovým řadám, diskrétní cosinová a sinová transformace a některé další negoniometrické systémy funkcí: Haarovy, Walshovy, Hadamardovy, aj. Další část příspěvku je věnována některým moderním algoritmům používaným pro kompresi jednotlivých obrázků (standard JPEG) nebo animačních sekvencí (standardy ISO H.261 "px64" a MPEG).

V závěru materiálu je uveden seznam literatury, která může posloužit k podrobnějšímu studiu všech popisovaných metod. V textu jsou podle možností uváděny odkazy na tyto prameny.

Obsah:

1. Analogový a digitální signál
   1.1. Vzorkování
   1.2. Kvantování
2. Základní metody komprese
   2.1. Prediktivní metody
   2.2. Kódování entropie
   2.3. Slovníkové metody komprese
3. Speciální metody komprese obrazu
   3.1. Transformační metody
   3.2. Hybridní metody
   3.3. Interpolační metody
   3.4. Metody druhé generace
4. Ortogonální transformace
   4.1. Ortogonální systémy funkcí
   4.2. Fourierova analýza
   4.3. Fourierova transformace
   4.4. Diskrétní Fourierova transformace
   4.5. Další goniometrické transformace
   4.6. Pravoúhlé transformace
5. Moderní standardy komprese obrazu
   5.1. JPEG
   5.2. Doporučení CCITT H.261
   5.3. MPEG
6. Literatura

* KSVI MFF UK, Malostranské nám. 25, 11800 Praha 1, tel: (02)24510286, fax: (02)535222, e-mail: Josef.Pelikan@mff.cuni.cz

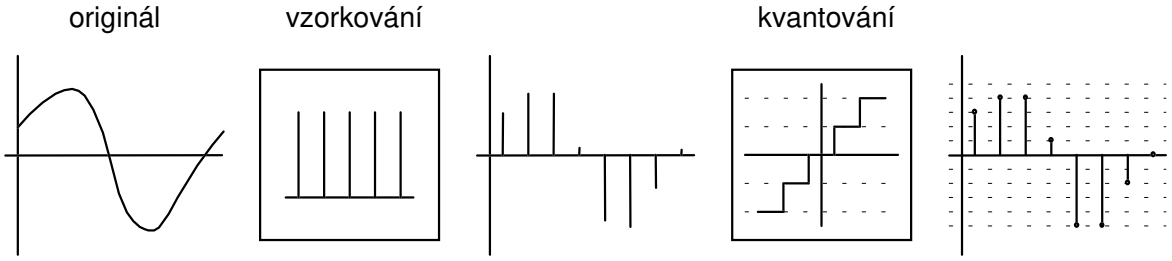
1. Analogový a digitální signál

Univerzální metody komprese dat pracují s digitálními daty, proto se budeme nejprve zabývat metodami diskretizace spojitého (analogového) signálu. **Analogový signál** budeme formálně zapisovat jako spojitou reálnou funkci jedné proměnné vyjadřující závislost nějaké veličiny (např. napětí, proudu) na čase: *g(t)*. Protože nám jde o praktické interpretace matematických výsledků, můžeme se omezit na funkce, které mají na nějakém intervalu $[t_1,t_2]$ **spojitý průběh** s max. konečným počtem nespojitostí, omezený **rozsah hodnot** a celkovou **omezenou energii**. Podmínku omezené energie lze zapsat matematicky takto:

$$E = \lim_{T\to\infty} \int_{-T}^{T} g^2(t)\,dt = \int_{-\infty}^{\infty} g^2(t)\,dt \quad < \infty$$

Obvod, který převádí spojité hodnoty signálu *g(t)* na konečně mnoho **diskrétních hodnot**, budeme nazývat **kvantovač** (quantizer) a proces převodu **kvantování** (quantization). Kvantováním signálu *g(t)* tady dostaneme funkci $g^*(t)$, která již může nabývat pouze omezeného počtu hodnot.

Za účelem digitálního zpracování však musíme udělat ještě jeden krok - místo spojité časové závislosti vybereme posloupnost určitých časových okamžiků, ve kterých hodnotu funkce $g^*(t)$ změříme a uložíme. Tomuto procesu se říká **vzorkování** (sampling). Místo spojité funkce spojité proměnné tedy dostaneme posloupnost diskrétních hodnot, kterou již můžeme zpracovávat digitálními metodami nebo po vhodném zakódování přenášet digitálním kanálem. Celému postupu (vzorkování - kvantování - kódování) se říká **PCM** - pulsní kódová modulace (pulse-code modulation):



1.1. Vzorkování

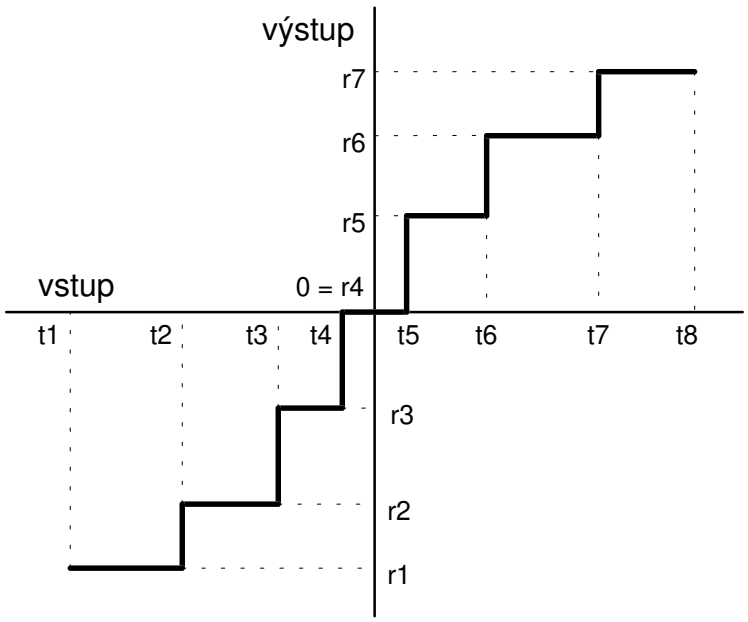
Je zřejmé, že jakost přeneseného signálu je ovlivněna tzv. **vzorkovacím kmitočtem** - počtem vzorků sejmutých za jednotku času (1 sec). Obecně lze říci, že čím větší je vzorkovací kmitočet, tím je možno po zpracování rekonstruovat původní signál s větší přesností. Při digitalizaci frekvenčně omezených signálů však nemá smysl zvyšovat vzorkovací kmitočet neomezeně. Platí totiž tzv. **Shannonova věta** o vzorkování [Haykin89], [Foley90]:

Signál s omezenou energií a kmitočtovým pásmem neobsahujícím kmitočty vyšší než ***W*** [Hz] je možno beze zbytku rekonstruovat ze vzorků sejmutých s frekvencí ***2W*** [Hz].

Mezní frekvence ***2W*** [Hz] je tedy plně postačující pro vzorkování takového signálu a nazývá se **Nyquistova frekvence**.

1.2. Kvantování

Obecně lze L-hodnotový kvantovač popsat posloupností **rozhodovacích hodnot** { t_i, $i=1,...,L+1$, $t_i<t_{i+1}$ } a **rekonstrukčních hodnot** { r_i, $i=1,...,L+1$ }. Při kvantování se pak vstupní hodnota z intervalu $(t_i, t_{i+1}]$ převede na příslušnou výstupní hodnotu r_i. Převodní funkce typického kvantovače je znázorněna na následujícím obrázku:



Při návrhu kvantovače je cílem určit rozhodovací a rekonstrukční hodnoty tak, aby bylo vzniklé zkreslení co nejmenší. Často se používají tzv. **lineární kvantovače**, které mají rozhodovací a rekonstrukční hodnoty rovnoměrně rozdělené. Pro jejich hodnoty platí: $t_k=t_1+(k-1)q$, $r_k=t_k+q/2$, kde $q=(t_{L+1}-t_1)/L$. Lineární kvantovač je vhodný pro zpracování signálu s rovnoměrným rozložením pravděpodobností všech hodnot.

Jestliže má kvantovaný signál nerovnoměrné rozložení hodnot (některé hodnoty se vyskytují častěji než jiné), je lepší použít nelineární kvantovač. Návrh optimálního kvantovače podle **Lloyda-Maxe** (viz např. [Jain81]) vychází ze znalosti rozložení pravděpodobnosti vstupního signálu: označíme ***p(x)*** hustotu pravděpodobnosti signálu a ***q(x)*** převodní funkci kvantovače. Naší snahou bude minimalizovat kvadratickou chybu kvantování:

$$Err = \int_{t_1}^{t_{T+1}} \left[x - q(x)\right]^2 p(x)\, dx$$

Minimalizací této chyby dostaneme soustavu podmínek pro rozhodovací a rekonstrukční hodnoty. Každá rozhodovací hodnota by měla ležet v polovině mezi sousedními rekonstrukčními hodnotami a ty zase musí ležet ve středech hustot rozhodovacích intervalů:

$$t_k = \frac{1}{2}(r_k + r_{k-1})$$

$$r_k = \int_{t_k}^{t_{k+1}} x\, p(x)\, dx \Big/ \int_{t_k}^{t_{k+1}} p(x)\, dx$$

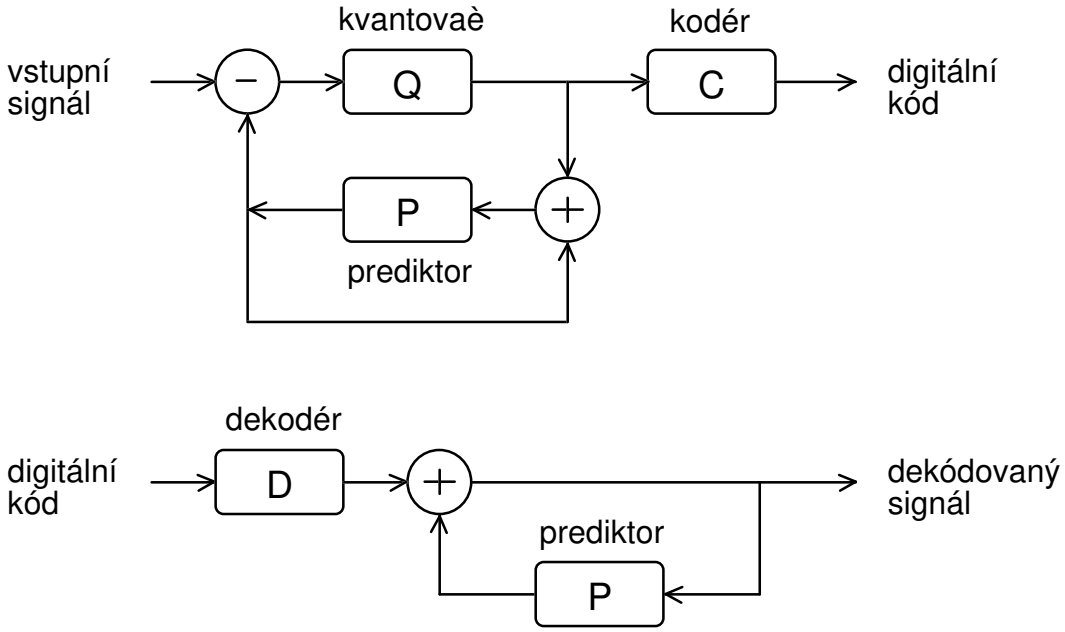
Řešením této soustavy pak dostaneme optimální kvantovač s minimální chybou *Err*.

2. Základní metody komprese

Zde se budeme zabývat obecnými metodami komprese dat, které se uplatňují při kompresi obrazu. Speciální metody vyvinuté pro kompresi obrazových dat budou popsány v kapitole 3.

2.1. Prediktivní metody

Prediktivní metody jsou založeny na tom, že sousední vzorky kódovaného signálu nejsou úplně nezávislé. Hodnotu jednoho vzorku lze proto přibližně odhadnout ze vzorků předcházejících a zakódovat pak jen chybu vzniklou tímto odhadem. Obecné schéma kódování a dekódování signálu prediktivní metodou je znázorněno na obrázku:



Podle počtu úrovní kvantovače rozlišujeme tzv. **delta modulaci** (DM - delta modulation) pouze s dvouhodnotovým kódem a **rozdílovou delta modulaci** (DPCM - differential pulse code modulation) s více kvantovacími hladinami.

Delta modulace se používá především v komunikační technice při kódování zvuku a není příliš vhodná ke kódování obrazového signálu. Základními nevýhodami DM je omezení při kódování vysokých frekvencí projevující se tzv. přetížením sklonu a dále zrnitost při téměř konstantním průběhu signálu. Zrnitost lze odstranit zavedením třetího stavu kvantovače odpovídajícího nulové změně. Výstup DM se často dále komprimuje klasickými metodami jako jsou RLE nebo entropické kódování - viz další oddíly.

Rozdílové delta modulátory mohou být lineární nebo nelineární, statické nebo adaptivní podle charakteru predikce. Jako příklad může posloužit lineární jednorozměrný DCPM pracující podle následujících vztahů (***g'(t)*** je predikce signálu g(t), ***g*(t)*** je kvantovaný signál a ***e(t)*** kódovaný rozdíl):

$$g'(t) = \sum_{i=1}^{p} a_i \cdot g^*(t-i) \qquad e(t) = g(t) - g'(t).$$

Při kódování obrazu se dosahuje lepších výsledků použitím dvojrozměrného prediktoru, např:

$$g'(t,u) = A \cdot g^*(t-1,u) + B \cdot g^*(t,u-1) + C \cdot g^*(t-1,u-1),$$

kde *A*, *B*, *C* jsou vhodně zvolené konstanty (*A+B+C=1*). Důmyslnější metody používají složitější predikční funkce (např. polynomiální extrapolaci), případně je adaptivně mění v průběhu kódování. Při kódování obrazu se dosahuje u statických DPCM kompresního poměru cca 2.5:1, u dynamických variant až 3.5:1.

Další algoritmus využívající závislosti sousedních vzorků je tzv. **kódování běhů - RLE** (run-length encoding). Jedná se o velmi jednoduchou metodu, která nahrazuje posloupnost stejných vzorků ("běh") zvláštním kódem označujícím jaká hodnota se má opakovat a kolikrát. Tohoto principu se používá při kompresi obrazové informace tam, kde záleží na velké rychlosti a jednoduchosti implementace kodéru/dekodéru (viz např. formát PCX, Targa, JFIF). Příklad použití zvláštní trojice znaků **ESC,z,n** k zakódování **n**-tice opakujících se hodnot **z**:

. . 1 5 6 6 6 6 6 2 2 7 1 5 5 5 5 5 5 5 5 5 6 6 7 8 8 8 8 8 . .

se zakóduje jako

. . 1 5 ESC 6 5 2 2 7 1 ESC 5 10 6 6 7 ESC 8 5 . .

2.2. Kódování entropie

Množství informace obsažené v nějakém signálu nebo zprávě lze formálně popsat pomocí pojmu **entropie**. Čím má signál větší entropii, tím obsahuje více informace a k jeho přenesení musí být použito více **bitů** (bit je zde používán jako jednotka informace reprezentující dvouhodnotovou veličinu - 0/1). Poměr entropie zprávy a velikosti jejího skutečného uložení udává tzv. **redundanci** - čím úsporněji je zpráva uložena (zakódována), tím má menší redundanci.

Jestliže rozdělíme vstupní signál na posloupnost nedělitelných **symbolů** s_i (to mohou být např. znaky textu nebo hodnoty jasu jednotlivých pixelů obrazu) a známe jejich pravděpodobnosti výskytu p_i, můžeme spočítat entropii jednotlivého symbolu s_i v **bitech**

$$H(s_i) = -\log_2 p_i$$

a průměrnou entropii (informační hodnotu) jednoznakové zprávy jako vážený průměr

$$\overline{H} = \sum_i -p_i \cdot \log_2 p_i .$$

Z Shannonovy teorie kódování plyne, že při nezávislém kódování jednotlivých symbolů je optimální zakódovat symbol s_i pomocí $H(s_i)$ bitů (viz [Shannon48]). V roce 1952 navrhnul O. Huffman konstrukci kódu blížícího se této teoretické hranici. Lze dokonce dokázat, že mezi všemi kódy používajícími celočíselný počet bitů k reprezentaci jednoho symbolu je **Huffmanův kód** ten nejkratší. Huffmanovu konstrukci zde nebudeme podrobně popisovat (viz např. [Huffman52], [Adámek89]), omezíme se jen na konstatování, že častěji se vyskytujícím symbolům přiřazuje kratší kódy a naopak. Kód jako celek je sestavován tak, aby jej bylo možné jednoznačně dekódovat. Postupuje se tak, že se sdružují symboly s nejmenšími pravděpodobnostmi a přiřazuje se jim společný prefix. Následující příklad ukazuje Huffmanův kód pro abecedu pěti symbolů A, B, C, D, E s různými pravděpodobnostmi výskytu. Podtržením je vyznačen společný prefix symbolů D a E:

symbol	pravděpodobnost	kód

A	0.42	1
B	0.28	00
C	0.15	011
D	0.08	0100
E	0.07	0101

Ještě lepších výsledků se dosahuje použitím tzv. **aritmetického kódování**, které již není omezeno celočíselným počtem bitů, protože se najednou kódují celé posloupnosti symbolů. Zpráva skládající se z N symbolů s_1 až s_N s pravděpodobnostmi p_1 až p_N je zakódována pomocí maximálně

$\left\lceil \sum_{i=1}^{N} -\log_2 p_i \right\rceil$ bitů, kde $\lceil x \rceil$ označuje zaokrouhlení nahoru.

Huffmanův kód stejné zprávy má délku maximálně $\sum_{i=1}^{N} \lceil -\log_2 p_i \rceil$ bitů.

Nevýhodou aritmetického kódování je složitější algoritmus, při jehož realizaci vzniká mnoho problémů s přesností výpočtů. Existují však i robustní a rychlé implementace používající pouze celočíselnou aritmetiku - viz např. [Frydrych93].

Pro Huffmanovo i aritmetické kódování existují dynamické - adaptivní - varianty, které místo předpokládaných pravděpodobností symbolů počítají jejich skutečné četnosti a podle nich mění kód v průběhu kódování. Výsledkem je v obecném případě lepší kompresní poměr, jejich velkou výhodou je i to, že není třeba předpokládat (a přenášet) jednotlivé pravděpodobnosti.

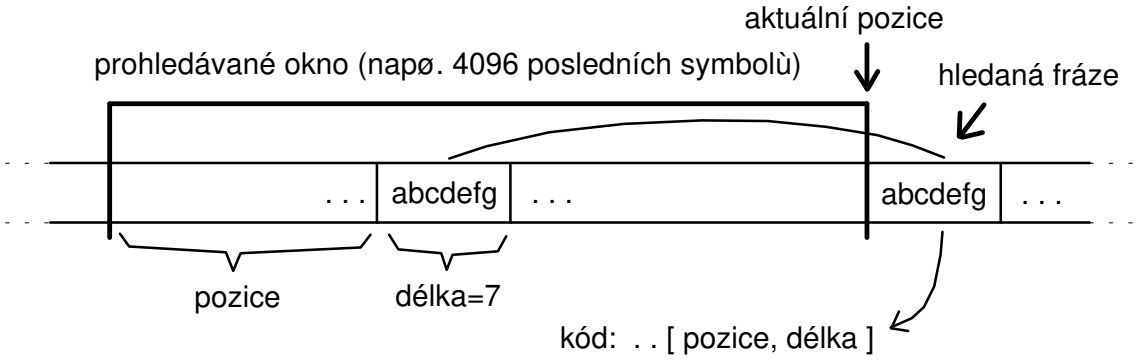
Huffmanovo nebo aritmetické kódování se často používá v algoritmech komprese obrazu. Zařazuje se většinou na konec kódovacího procesu, jako např. v metodách a formátech MLP, JPEG nebo MPEG (viz [Frydrych93], [Wallace91] a [LeGall91]).

2.3. Slovníkové metody komprese

Tzv. **slovníkové kompresní metody** dosahují obecně lepších výsledků než entropické kódování, protože využívají závislosti v delších posloupnostech vstupních symbolů. V některých situacích se však nedají použít pro větší složitost a časovou nebo paměťovou náročnost.

Základy slovníkových metod položili v letech 1976-1978 A. Lempel a J. Ziv, když popsali dva základní kompresní algoritmy nazývané později **LZ77** a **LZ78** (viz [Ziv77], [Ziv78] nebo [Bell89]). V obou metodách se využívá pozorování, že ve zprávách obsahujících mnoho redundance se opakovaně vyskytují delší řetězce symbolů. Vyhledáváním takových frází a jejich úsporným kódováním se pak dosahuje kompresních poměrů 2:1 až 4:1 (pro texty nebo rastrové obrázky). Schémata komprese LZ77 a LZ78 byla později mnohokrát zdokonalena a dnes se již původní varianty algoritmů nepoužívají.

Princip metody **LZ77** spočívá v tom, že se kódovaná fráze (řetězec vstupních symbolů) hledá v již dříve zakódované části zprávy. Kvůli omezení časové složitosti se hledání omezuje na "okno" dané velikosti - typicky několik tisíc posledních symbolů. V případě úspěchu se fráze zakóduje dvěma údaji: délkou a relativní pozicí v prohledávaném okně. Schéma práce metody LZ77 je na následujícím obrázku (fráze 'abcdefg' byla v okně úspěšně nalezena):



Metoda **LZ78** je založena na trochu jiném principu: během komprese se vytváří seznam nejčastěji se vyskytujících frází - tzv. **slovník**. Každá vstupní fráze se nejprve hledá ve slovníku - je-li nalezena, zakóduje se pomocí pořadového čísla ve slovníku, není-li nalezena, zakóduje se jiným způsobem (např. po jednotlivých symbolech) a do slovníku se přidá. Existuje mnoho variant algoritmu LZ78 lišících se reprezentací slovníku, kódováním nenalezené fráze, přidáváním nových frází, atd. Při kompresi obrazu se nejčastěji používá algoritmus **LZW** s proměnlivou délkou kódu (viz [Welch84], [GIF90]) a následujícím schématem výpočtu:

1. Všechny symboly se zařadí do slovníku jako jednoznakové fráze
   Nastaví se počáteční délka kódu **n** (např. 9 bitů při 256 vstupních symbolech)
   [prefix] = '' {prázdná fráze}
2. Read(**K**) {čtení symbolu ze vstupu}
   Je-li konec vstupu,
   Write(**kód[prefix]**), {zápis kódu fráze na výstup}
   konec
3. Je **[prefix]K** ve slovníku?
   Ano: **[prefix]** = **[prefix]K**
   Ne: přidání **[prefix]K** do slovníku,
   Write(**kód[prefix]**), {zápis kódu fráze na výstup}
   [prefix] = **K**
4. Jdi na bod 2.

Přidávání do slovníku (bod 3.):
Je-li slovník velikosti 2^{n} zaplněn,
n = **n**+1,
Je-li **n**>12, slovník se vyprázdní a znova nastaví jako v bodě 1.

Při dekódování LZW kódu se staví slovník frází zcela analogicky, takže není problém rekonstruovat původní zprávu. Dekodér musí pouze znát počáteční délku kódu **n**.

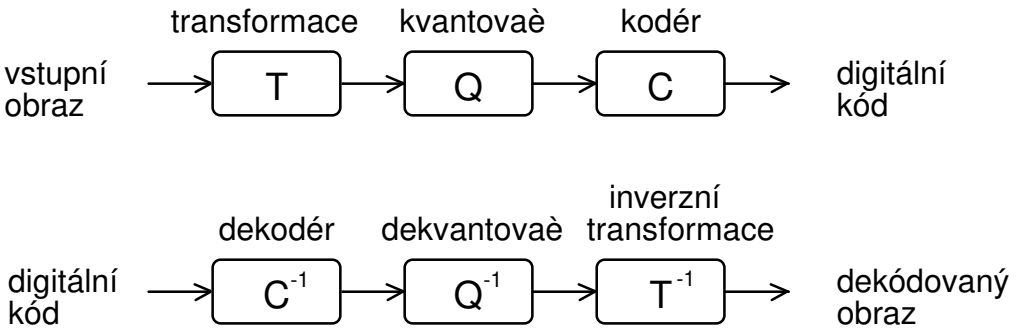
Popsaný algoritmus se mj. používá při uložení obrazu ve formátech GIF, TIFF nebo Targa.

3. Speciální metody komprese obrazu

Mezi metody vyvinuté speciálně pro kompresi obrazových dat lze zahrnout prediktivní metody se zvláštními dvojrozměrnými predikčními funkcemi - příklad takové metody byl uveden v oddíle 2.1. Dalším speciálním algoritmům je věnována tato kapitola, postupně bude stručně popsáno několik základních principů redukce obrazových dat.

3.1. Transformační metody

Transformační metody jsou někdy nazývány **blokově-kvantovacími** metodami, protože se snaží kvantovat najednou celý blok (vektor) vzorků obrazu. Snahou je maximálně potlačit závislost mezi jednotlivými vzorky dat, a tak je použita nějaká vhodná transformace, aby se blok vzorků převedl na blok "méně závislých" koeficientů. Ty se potom kvantují a kódují nezávisle jeden po druhém. K rekonstrukci původního obrazu se používá dekvantizace a inverzní transformace, jak je naznačeno v obecném schématu transformační metody:



Kompresní poměr a míra zkreslení celé transformační metody je nejvíce závislá na zvolené transformaci **T** a na parametrech použitého kvantovače. Transformace pracuje s blokem obrazu a převádí ho na blok koeficientů, typicky se používá čtvercových bloků velikosti 4x4 až 32x32. Na transformaci **T** jsou kladeny následující požadavky:

- většina důležitých informací o bloku hodnot by měla být soustředěna do několika málo koeficientů, aby bylo možné docílit velkého kompresního poměru vynecháním dalších koeficientů
- zkreslení vzniklé odstraněním méně podstatných koeficientů a kvantováním by mělo být co nejmenší
- snadnost výpočtu transformace i inverzní transformace
- koeficienty po transformaci by měly být co nejméně závislé (nejlépe nezávislé)

Karhunen a Loeve (KL) navrhli optimální lineární transformaci **T** za předpokladu, že vstupní hodnoty v bloku odpovídají Gaussovskému náhodnému rozdělení (s obecnou kovarianční maticí ***R***). Cílem bylo minimalizovat střední kvadratickou chybu přenosu. Odvození optimální transformační matice lze najít např. v [Jain81], pro nás bude důležité jen to, že KL-transformační matice se počítá obtížně - je nutné mj. určit vlastní vektory kovarianční matice ***R***. Proto se v praxi místo KL-transformace používají jiné suboptimální transformace vyhovující dobře v případě stacionárních pravděpodobností hodnot v transformovaném bloku. Pro tyto alternativní transformace existují rychlé algoritmy výpočtu, které je možné implementovat i hardwarově.

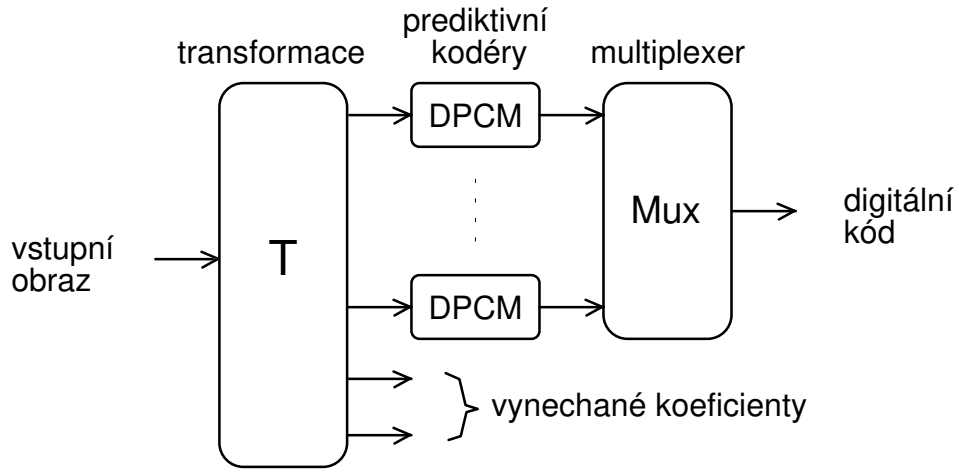
Jedná se hlavně o goniometrické transformace: diskrétní Fourierova, cosinová a sinová transformace nebo pravoúhlé transformace Hadamardova, Haarova, Welshova aj. Protože mají tyto ortogonální transformace stále větší význam v moderních metodách komprese obrazu, věnujeme jim celou kapitolu 4.

Pro minimalizaci zkreslení by se v transformační metodě měl použít optimální Lloydův-Maxův kvantovač, často se však používá jiných - jednodušších - kvantovačů, zejména lineárních.

3.2. Hybridní metody

Hybridní metody spojují výhody prediktivních a transformačních metod a využívají závislosti mezi odpovídajícími transformovanými koeficienty sousedních bloků. Schéma práce

obecného hybridního kodéru dostaneme tak, že v transformační metodě použijeme DPCM modulů ke kódování jednotlivých koeficientů:



Při volbě vhodné transformace (např. diskrétní cosinová) a vynechání méně významných koeficientů (až 50%) lze dosáhnout kompresních poměrů 5:1 až 15:1 při dostačující kvalitě rekonstruovaného obrazu (viz [Jain81]).

3.3. Interpolační metody

Další metody komprese obrazu jsou založeny na přenosu redukovaného počtu vzorků a interpolaci chybějících hodnot při dekódování. Kvalita přenosu závisí na počtu a rozmístění kódovaných pixelů a na metodě interpolace, kompresní poměr bývá u čisté neadaptivní metody až 4:1. Často se používá lineární interpolace nebo vážené průměrování, někdy i polynomiální interpolace vyššího stupně. Ukazuje se však, že lineární interpolace je ve většině případů dostatečně kvalitní.

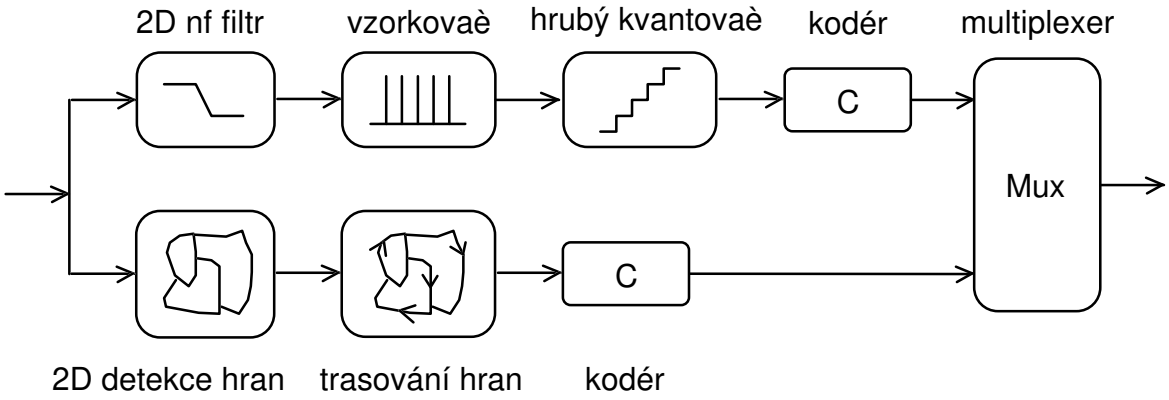
Interpolační metody se používají např. ve standardu JPEG (při ztrátovém i hierarchickém kódování), příklady dalších metod lze najít v [Frydrych93].

3.4. Metody druhé generace

Tak se někdy nazývají nejmodernější metody komprese obrazu, které využívají poznatků fyziologických a neuro-psychologických výzkumů systému lidského vidění. Se znalostí mechanismu zpracování vizuální informace v oku a mozkovém centru vidění lze zkonstruovat algoritmy dosahující většího kompresního poměru (až 100:1) při zachování stejné subjektivní kvality výsledku jako u předchozích metod - metod první generace.

Hranice mezi první a druhou generace není ostrá, některé metody leží na pomezí tohoto dělení. Obecně lze říci, že neuro-psychologické výsledky se začínaly v kompresi obrazu používat na začátku 80. let. V tomto oddíle si ukážeme jen několik základních principů komprese druhé generace, podrobnější a úplnější přehled lze najít v [Kunt85] nebo [Frydrych93].

Metoda **'Synthetic High System'** (SHS) využívá odlišného zpracování nízkofrekvenční a vysokofrekvenční složky na sítnici lidského oka a signál kóduje podobným způsobem - odděleně nf a vf složku. Dosahuje kompresního poměru 8:1 a stala se základem mnoha jiných kompresních metod. Schéma práce kodéru SHS:



Pyramidální kódování se též opírá o rozklad obrazu na frekvenční pásma, čímž napodobuje práci zrakové soustavy člověka. Používá hierarchického rozkladu obrazového signálu na několik pásem, která jsou odděleně zpracován a vzorkováním a kvantováním.

Od originálního obrazu g se nejprve nízkofrekvenčním filtrem s mezním kmitočtem f_1 oddělí nf složka g_1, která bude sloužit jako predikce obrazu g. Spočítá se chyba predikce $e_1=g-g_1$ a ta se potom dále kóduje obdobným způsobem: filtrem f_2 se z ní oddělí nf složka g_2, spočítá se další chyba predikce $e_2=g_1-g_2$, atd. Jednotlivé chyby e_1, e_2 až e_N se pak vzorkují, kvantují a přenášejí; nižší frekvence jsou vzorkovány v hrubším rastru, proto se metoda nazývá pyramidální. Kompresního poměru asi 10:1 je dosaženo hrubším vzorkováním nízkých frekvencí a omezením přesnosti vysokých frekvencí (na něž je lidské oko méně citlivé).

Metoda **VPIC** (Visual Pattern Image Coding) používá ke kompresi obrazu tzv. **vzorků** (patterns) velikosti 4x4 pixely. Vstupní obraz se rozdělí na bloky 4x4, v každém se spočítá průměrná hodnota a odečte se od hodnot pixelů bloku. V takto vzniklých datech se hledají a kódují jednoduché vzorky. Kompresního poměru 20:1 je dosaženo zanedbáním nebo nepřesným kódováním tvarů méně důležitých pro vizuální vjem (detaily viz [Chen90] a [Chen92]).

Metody založené na **segmentaci** se obecně snaží o dekompozici obrazu na disjunktní segmenty popsané obrysem a texturou (barvou, vzorkem). Kódováním údajů o segmentech lze dosáhnout kompresních poměrů až 70:1.

Mezi tyto metody patří např. metoda **narůstání oblastí** (region growing), **rozděl a spoj** (split and merge) nebo **směrová dekompozice** (directional decomposition). Podrobnější popis je možné najít v [Kunt85].

4. Ortogonální transformace

V oddílu 3.1 byl popsán základní princip transformačních metod komprese obrazu včetně požadavků kladených na funkci **T**. Optimální KL-transformace se ukázala jako nepraktická pro velkou výpočetní složitost, proto se nyní budeme podrobněji zabývat efektivnějšími a používanějšími suboptimálními transformačními funkcemi. Vzhledem k požadavku nezávislosti výsledných koeficientů se nejčastěji používají transformace založené na projekci do nějakého ortogonálního systému funkcí.

4.1. Ortogonální systémy funkcí

Definice: Systém spojitých funkcí reálné proměnné { $u_n(t)$, n = 0, 1, ... } nazveme **ortogonálním** na nějakém intervalu T, jestliže platí

$$\int_T u_m(t)\, u_n(t)\, dt = \begin{cases} c > 0 & \text{pro} \quad m = n \\ 0 & \text{pro} \quad m \neq n \end{cases} . \qquad (1)$$

Jestliže je navíc $c = 1$, nazýváme takový systém **ortonormálním**.

Předpokládejme, že máme nějaký signál $g(t)$ definovaný na intervalu T a že jej lze vyjádřit pomocí systému { $u_n(t)$ } řadou

$$g(t) = \sum_{n=0}^{\infty} a_n \cdot u_n(t) . \qquad (2)$$

Potom můžeme koeficienty a_n spočítat integrálem

$$a_n = \frac{1}{c} \int_T g(t)\, u_n(t)\, dt, \quad \text{pro} \quad n = 0, 1, \ldots$$

Definice: Ortogonální systém funkcí { $u_n(t)$ } splňující nerovnost

$$\int_T u_n^{\,2}(t)\, dt \; < \; \infty$$

nazveme **úplným**, jestliže platí jedna z následujících ekvivalentních podmínek:

1. neexistuje signál $g(t)$ vyhovující podmínce

   $$\int_T g^2(t)\, dt \; < \; \infty , \qquad (3)$$

   který by byl ortogonální na celý systém { $u_n(t)$ }, tj:

   $$\int_T g(t)\, u_n(t)\, dt \; = \; 0, \quad \text{pro} \quad n = 0, 1, \ldots$$

2. pro libovolný po částech spojitý signál $g(t)$ splňující (3) a pro libovolně malé $\varepsilon > 0$ existuje číslo N takové, že součet konečné řady

   $$\hat{g}(t) = \sum_{n=0}^{N-1} a_n \cdot u_n(t)$$

   aproximuje funkci $g(t)$ s kvadratickou chybou menší než ε: $\int_T \left| g(t) - \hat{g}(t) \right|^2 dt \; < \; \varepsilon$.

Ze zavedených pojmů a tvrzení plyne, že průmět do systému ortogonálních funkcí dává možnost vyjádřit libovolný signál ve tvaru nekonečné řady koeficientů { a_0, a_1, a_2, ... }. Navíc, je-li systém funkcí úplný, lze takovou reprezentaci zredukovat na konečnou posloupnost koeficientů { a_0, a_1, ... a_{N-1} }. Koeficienty a_n jsou přitom navzájem nezávislé, což plyne z ortogonality systému funkcí podle podmínky (1).

Fyzikální význam uvedených pojmů dostaneme, když umocníme na druhou rovnici (2) a spočítáme střední hodnotu takové veličiny na intervalu **T**. Dostaneme vztah

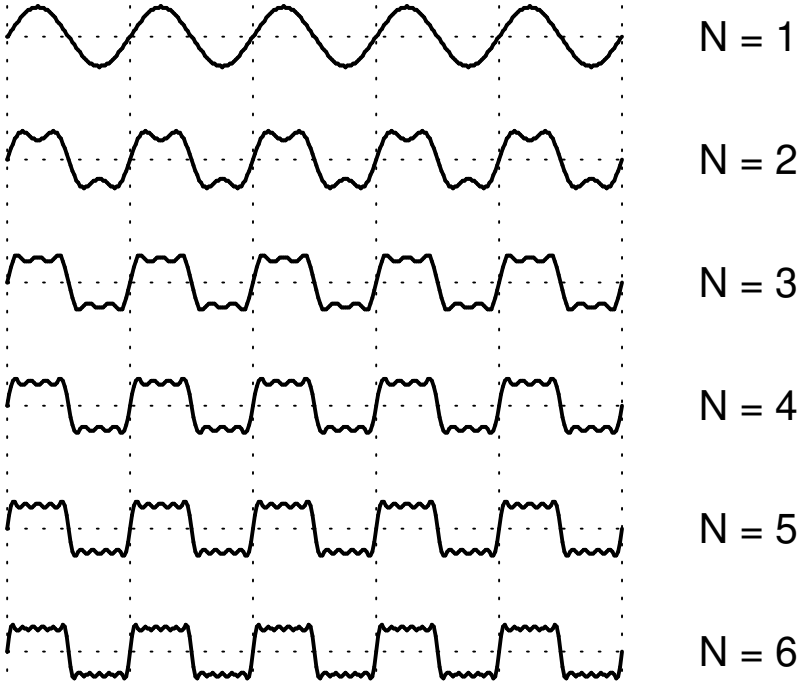
$$\frac{1}{|T|} \int_T g^2(t)\, dt \; = \; \frac{c}{|T|} \sum_{n=0}^{\infty} a_n^{\,2} .$$

Jestliže chápeme funkci $g(t)$ jako závislost napětí nebo proudu na čase, vyjadřuje tato rovnice průměrný ztrátový výkon signálu g na odporu 1Ω.

Příklad: signál s **obdélníkovým průběhem** a periodou 2π lze reprezentovat nekonečnou řadou

$$g(t) = \sum_{n=0}^{\infty} \frac{1}{2n+1} \cdot \sin\left[(2n+1)t\right], \qquad (4)$$

{ **sin[(2*n*+1)*t*]** } je zde ortogonální systém funkcí. Průběh několika prvních aproximací (částečných součtů řady (4)) je znázorněn na obrázku:



4.2. Fourierova analýza

Rozklad periodického signálu ***g(t)*** s periodou $\mathbf{T_0}$ na tzv. **Fourierovu řadu** můžeme napsat jako

$$g(t) = a_0 + 2 \cdot \sum_{n=1}^{\infty} \left[a_n \cdot \cos\left(\frac{2\pi n\, t}{T_0}\right) + b_n \cdot \sin\left(\frac{2\pi n\, t}{T_0}\right) \right].$$

Ortogonální systém se zde skládá z funkcí

$$\left\{ \quad 1, \quad \cos\left(\frac{2\pi n\, t}{T_0}\right), \quad \sin\left(\frac{2\pi n\, t}{T_0}\right) \quad \text{pro } n = 1, 2, \cdots \quad \right\}$$

a koeficienty a_n, b_n lze spočítat podle vztahů

$$a_0 = \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} g(t)\, dt$$

$$a_n = \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} g(t) \cdot \cos\left(\frac{2\pi n t}{T_0}\right) dt \quad \text{pro } n = 1, 2, \ldots$$

$$b_n = \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} g(t) \cdot \sin\left(\frac{2\pi n t}{T_0}\right) dt \quad \text{pro } n = 1, 2, \ldots$$

Při použití komplexní aritmetiky se zápis Fourierovy řady zjednoduší - viz [Haykin89]. Platí totiž

$$\exp(it) = \cos(t) + i \cdot \sin(t), \quad \text{kde } i = \sqrt{-1},$$

a tak můžeme reálné koeficienty a_n, b_n nahradit komplexními c_n a celou řadu přepsat do tvaru

$$g(t) = \sum_{n=-\infty}^{\infty} c_n \cdot \exp\left(\frac{2\pi n t i}{T_0}\right), \quad \text{kde}$$

$$c_n = \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} g(t) \cdot \exp\left(-\frac{2\pi n t i}{T_0}\right) dt \quad \text{pro } n = 0, \pm 1, \pm 2, \ldots$$

Komplexním koeficientům c_n se říká **diskrétní spektrum** periodického signálu ***g(t)*** a často se ještě rozkládají na absolutní hodnotu (amplitudu) a fázovou složku.

4.3. Fourierova transformace

Fourierovy řady jsou tedy prostředkem pro reprezentaci spojitých periodických signálů. Můžeme je však dále zobecnit na tzv. **Fourierovu transformaci**, která slouží k reprezentaci spojitých **neperiodických** signálů. Odvození lze najít např. v [Haykin89] a jeho výsledkem jsou dva transformační postupy. K funkci ***g(t)*** se definuje její Fourierův obraz - funkce ***G(f)***, která je jejím reprezentantem ve spektrální oblasti (vyjadřuje složení frekvencí, ze kterých lze funkci ***g(t)*** poskládat). Funkce ***g(t)*** určuje jednoznačně funkci ***G(f)*** a naopak - těmto převodům se říká **přímá a zpětná Fourierova transformace**:

$$G(f) = \int_{-\infty}^{\infty} g(t) \cdot \exp(-2\pi f t i)\, dt = \int_{-\infty}^{\infty} g(t) \cdot \left[\cos(2\pi f t) - i \cdot \sin(2\pi f t)\right] dx \qquad (5)$$

$$g(t) = \int_{-\infty}^{\infty} G(f) \cdot \exp(2\pi f t i)\, df = \int_{-\infty}^{\infty} G(f) \cdot \left[\cos(2\pi f t) + i \cdot \sin(2\pi f t)\right] df \qquad (6)$$

Aby byla funkce ***g(t)*** Fourierovsky transformovatelná, musí být splněny tzv. **Dirichletovy podmínky**:

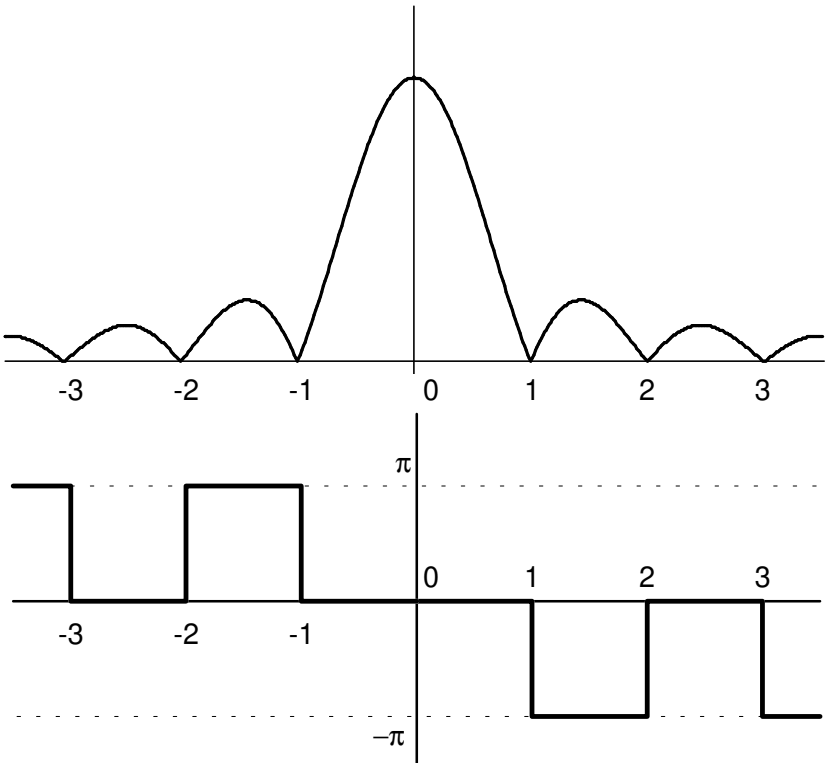
1. funkce může mít maximálně konečný počet nespojitostí, lokálních maxim a minim na nějakém konečném intervalu
2. funkce musí být absolutně integrovatelná na intervalu (-∞,∞), tento integrál musí být konečný

Každý signál s omezenou energií (zavedený v kapitole 1) tyto podmínky naštěstí splňuje.

Příklad: **obdélníkový impuls** nabývá hodnoty **1** na intervalu (-1/2, 1/2), jinde je jeho hodnota nulová. Definujme funkci **sinc(*x*)** jako

$$\mathrm{sinc}(x) = \frac{\sin(\pi \mathrm{x})}{\pi \mathrm{x}},$$

potom můžeme napsat Fourierův obraz impulsu jako funkci $G(f) = \mathrm{sinc}(f)$. Průběh obou složek (amplitudy i fáze) funkce ***G(f)*** je znázorněn na obrázku:



4.4. Diskrétní Fourierova transformace

Dosud jsme se zabývali spojitými funkcemi (signály) a transformacemi. Ty mají velký význam v analogové telekomunikační technice, při digitálním přenosu a zpracování obrazu však potřebujeme častěji pracovat s diskrétními funkcemi definovanými pouze v určitých časových okamžicích. Takové signály jsme také již zaváděli v kapitole 1: pracovali jsme se signálem reprezentovaným posloupností hodnot **- vzorků**.

Dále budeme pracovat s diskrétními signály definovanými v ***m*** bodech: budeme je označovat ***f(x)*** nebo ***g(x)*** pro ***x*=0**, **1**, ..., ***m*-1**. Ze vztahů (5) a (6) lze odvodit analogické předpisy, kterým se říká přímá a inverzní **diskrétní Fourierova transformace (DFT)**:

$$G(u) = \frac{1}{m} \sum_{x=0}^{m-1} g(x) \cdot \exp\left(\frac{-2\pi u x i}{m}\right) \qquad \text{pro } u = 0, 1, ..., m\text{-}1 \qquad (7)$$

$$g(x) = \sum_{u=0}^{m-1} G(u) \cdot \exp\left(\frac{2\pi u x i}{m}\right) \qquad \text{pro } x = 0, 1, ..., m\text{-}1. \qquad (8)$$

DFT má velký význam nejen při kompresi signálu, ale i při návrhu digitálních filtrů a spektrální analýze číslicových signálů. Nyní si všimneme základních vlastností Fourierových koeficientů ***G***: dosadíme-li za $u = 0$, vychází z rovnice (7):

$$G(0) = \frac{1}{m} \sum_{x=0}^{m-1} g(x).$$

To znamená, že nultý koeficient reprezentuje **stejnosměrnou složku** signálu - průměrnou hodnotu všech vzorků ***g***. Naopak, koeficienty ***G*(1)** až ***G*(*m*-1)** reprezentují **střídavé složky** signálu s délkami period ***m*** až ***m*/(*m*-1)**.

Diskrétní Fourierova transformace má několik dalších velmi důležitých vlastností:

- DFT je **lineární zobrazení**, tj. máme-li tři signály ***f*(*x*)**, ***g*(*x*)** a ***h*(*x*)** s příslušnými Fourierovými obrazy ***F*(*u*)**, ***G*(*u*)** a ***H*(*u*)** takovými, že
  $h(x) = a \cdot f(x) + b \cdot g(x)$, pak platí i $H(u) = a \cdot F(u) + b \cdot G(u)$.

- věta o **komplexně sdružených koeficientech**: je-li ***m*** sudé číslo, pak pro Fourierovy koeficienty funkce ***g*** platí
  $G\left(\frac{m}{2} + l\right) = \overline{G}\left(\frac{m}{2} - l\right)$ pro všechna ***l*** = 0, 1, ..., *m*/2.

- věta o **posunutí**: máme dva signály časově posunuté o ***l***: $f(x) = g(x + l)$. Potom pro jejich Fourierovy koeficienty platí
  $$F(u) = G(u) \cdot \exp\left(\frac{-2\pi l\, u\, i}{m}\right).$$

- věta o **konvoluci**: máme dva signály ***f*(*x*)** a ***g*(*x*)** a jejich konvoluci ***h*(*x*)** definovanou jako
  $h(x) = \frac{1}{m} \sum_{y=0}^{m-1} f(y) \cdot g(x - y)$. Pak pro koeficienty ***H*(*u*)** platí $H(u) = F(u) \cdot G(u)$.

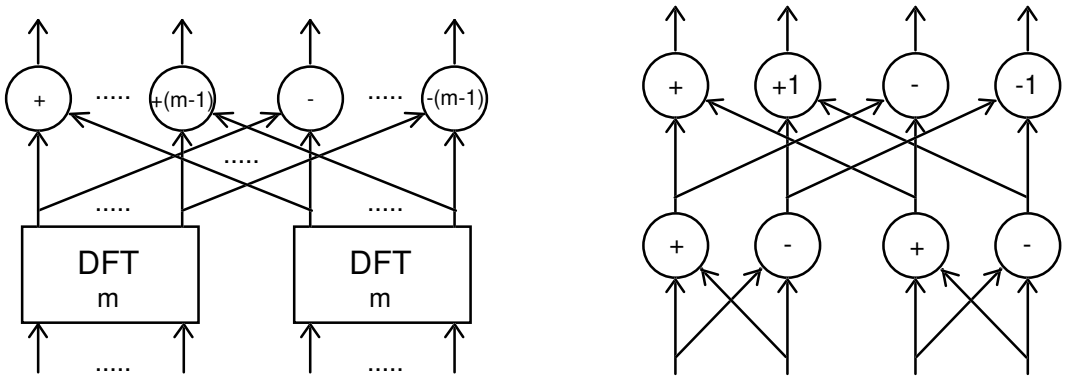
Používání DFT by nemohlo být tak rozšířené, kdyby neexistoval rychlý algoritmus jejího výpočtu. Podle definice (7) by se muselo provést $O(m^2)$ operací při výpočtu vektoru koeficientů ***G*(*u*)**. Již od roku 1965 ([Cooley65]) je známa metoda rychlého výpočtu přímé i inverzní diskrétní Fourierovy transformace (FDFT a IDFT) pracující řádově v čase $O(m \log m)$, při paralelních implementacích pomocí obvodů VLSI se dosahuje dokonce složitosti $O(\log m)$. Podrobné odvození je možno nalézt např. v [Kučera89] nebo [Ahmed75], zde se omezíme pouze na schematický popis metody.

Základní myšlenka rychlého algoritmu spočívá v principu "**rozděl a panuj**" - je-li číslo ***m*** sudé, lze rozdělit koeficienty ***G*(*u*)** na dvě stejně velké skupiny - sudé a liché. K výpočtu obou skupin pak potřebujeme spočítat dvě DFT velikosti ***m*/2**, kombinaci výsledků provedeme v konstantním čase.

Schéma implementace rychlé DFT naznačují následující obrázky: vlevo je nakreslen obvod počítající DFT velikosti **2*m***; používá k tomu dvou paralelních obvodů poloviční velikosti ***m***. Kombinační člen znázorněný kroužkem s legendou **±*n*** realizuje operaci

$$x \pm y \cdot \exp\left(\frac{-2\pi n\, i}{m}\right).$$

Na pravém schématu je obvod realizující DFT velikosti **4**; v horní vrstvě je třeba do předchozího vztahu dosadit ***m* = 2**.



Při zpracování dvojrozměrného signálu (obrazu) se používá **2D DFT** definované analogicky pro diskrétní funkce dvou proměnných (matice ***m*×*n*** prvků):

$$G(u,v) = \frac{1}{m \cdot n} \sum_{y=0}^{n-1} \sum_{x=0}^{m-1} g(x,y) \cdot \exp\left(\frac{-2\pi u x i}{m}\right) \cdot \exp\left(\frac{-2\pi v y i}{n}\right),$$

pro ***u*** = **0**, **1**, ..., ***n*-1**, ***v*** = **0**, **1**, ..., ***m*-1**. Pro výpočet 2D DFT je možné použít 1D algoritmů, protože lze snadno vytknout

$$G(u,v) = \frac{1}{n} \sum_{y=0}^{n-1} \exp\left(\frac{-2\pi v y i}{n}\right) \cdot \frac{1}{m} \sum_{x=0}^{m-1} g(x,y) \cdot \exp\left(\frac{-2\pi u x i}{m}\right). \qquad (9)$$

Podle předchozí rovnice lze spočítat 2D DFT takto: nejprve se spočítá 1D DFT na sloupcích matice ***g*(*x*,*y*)** a mezivýsledek se ještě jednou ztransformuje 1D DFT po řádcích.

4.5. Další goniometrické transformace

Při kódování obrazu se používají i další goniometrické transformace spojené s jinými systémy ortogonálních funkcí. Vezmeme-li za základ funkci cosinus, dostaneme **diskrétní cosinovou transformaci (DCT)**, pouze s funkcemi sinus zase **diskrétní sinovou transformaci (DST)**. Obě transformace se využívají při kompresi obrazu jako alternativy KL-transformace (viz oddíl 3.1) - DCT je v obecném případě nejlepší aproximací KLT, DST je v některých speciálních případech identická s KLT (viz [Skarbek93]). DCT byla zvolena po pečlivém výběru jako základ mnoha standardů pro kompresi obrazu (JPEG, ISO H.261, CD-I, MPEG - viz další kapitolu).

Diskrétní sinová transformace používá ortogonální systém funkcí

$$\left\{ s_n(x) = \sin\left[\frac{(n+1)(x+1)\pi}{m+1}\right], \quad \text{pro } x, n = 0, 1, \ldots m-1 \right\}.$$

Obraz funkce ***g*(*x*)** pomocí DST se spočítá podle vztahu

$$G(u) = \sqrt{\frac{2}{m+1}} \cdot \sum_{x=0}^{m-1} g(x) \cdot \sin\left[\frac{(u+1)(x+1)\pi}{m+1}\right].$$

Rychlý algoritmus výpočtu DST je uveden v [Skarbek93].

Diskrétní cosinová transformace se používá častěji, proto se jí budeme věnovat podrobně. Příslušný ortogonální systém funkcí:

$$\left\{ \frac{1}{\sqrt{m}},\quad c_n(x) = \frac{2}{\sqrt{m}} \cdot \cos\left[\frac{(2x+1)\,n\,\pi}{2\,m}\right],\quad \text{pro } x = 0,1,\ldots m-1,\quad n = 1,2,\ldots m-1 \right\}$$

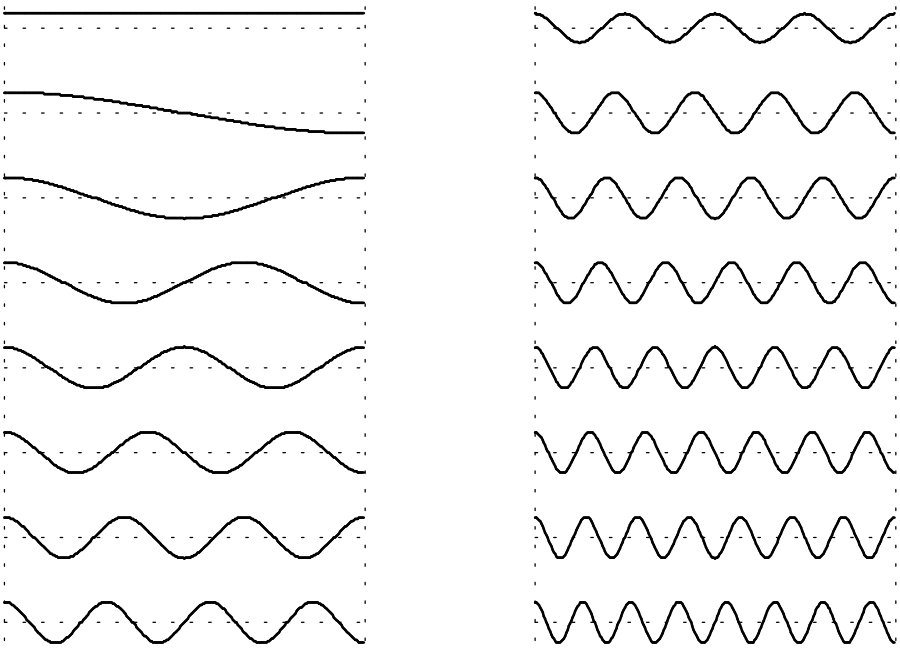
Přímá a zpětná DCT se počítají podle vztahů

$$G(u) = \sqrt{\frac{2}{m}} \cdot C(u) \cdot \sum_{x=0}^{m-1} g(x) \cdot \cos\frac{(2x+1)\,u\,\pi}{2\,m}$$

$$g(x) = \sqrt{\frac{2}{m}} \cdot \sum_{u=0}^{m-1} C(u) \cdot G(u) \cdot \cos\frac{(2x+1)\,u\,\pi}{2\,m},$$

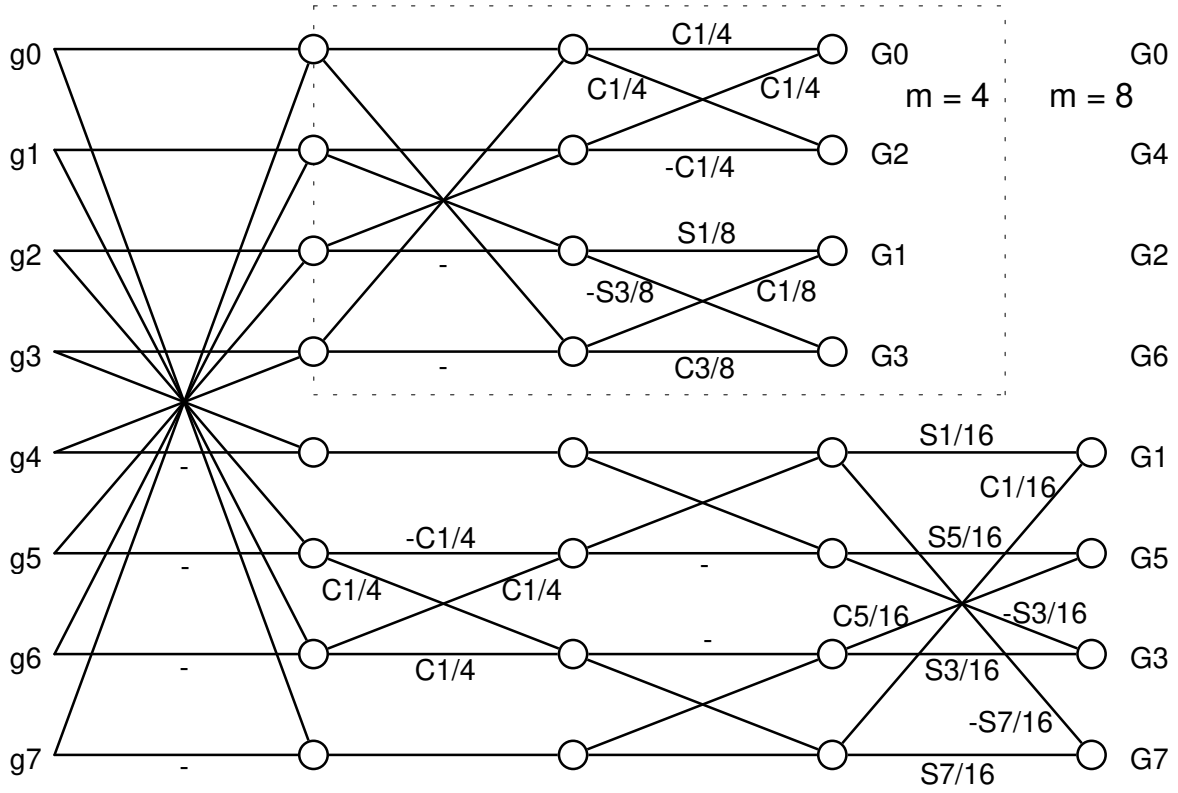
kde $C(i) = \begin{cases} \frac{1}{\sqrt{2}} & \text{pro } i = 0 \\ 1 & \text{jinak} \end{cases}$

Podobně jako u diskrétní Fourierovy transformace, i u DCT reprezentuje nultý koeficient ***G*(0)** stejnosměrnou složku signálu. Ostatní koeficienty odpovídají střídavým složkám se zvyšujícími se frekvencemi. Na obrázku je ukázka bázických funkcí DCT pro ***m* = 16**:



Je známo několik rychlých algoritmů pro výpočet DCT (např. [Chen77], [Loeffler89], [Rao-90]) pracujících v čase ***O*(*m* log *m*)**, při ***m* = 8** nejrychlejší z nich potřebuje pouze 11 násobení a 29 sčítání/odečítání.

Na obrázku je schéma výpočtu rychlé DCT podle [Chen77] pro ***m* = 4** a ***m* = 8**. Kroužky realizují operaci sčítání, hrany diagramu označené **±C*x*** respektive **±S*x*** reprezentují vynásobení hodnoty konstantou **±cos(*x*π)** respektive **±sin(*x*π)**:



4.6. Pravoúhlé transformace

Spojité systémy funkcí zavedené v předchozích oddílech mají mnohé dobré vlastnosti, ale k výpočtu příslušných transformací je zapotřebí přesné aritmetiky včetně operace násobení. V některých případech je naší snahou počítat potřebné transformace ještě rychleji, respektive pouze s operacemi sčítání a odčítání. Tehdy můžeme využít pravoúhlých transformací, z nichž některé mají vlastnosti srovnatelné s goniometrickými transformacemi. Zájemci o podrobnější studium mohou využít [Ahmed75], [Jain81] nebo [Netravali80].

Jednoduchý systém pravoúhlých funkcí zavedl roku 1922 H. **Rademacher -** systém je ortonormální na intervalu $[0,1]$, ale není úplný. Rademacherovy funkce můžeme definovat rekurzivně:

$$\mathrm{rad}(1,t) = \mathrm{sgn}\sin(2\pi t)$$

$$\mathrm{rad}(m,t) = \mathrm{rad}(1,2^{m-1}t).$$

A. **Haar** zavedl roku 1910 ortonormální úplný systém pravoúhlých funkcí definovaných na intervalu $[0,1]$. Haarovy funkce se definují takto:

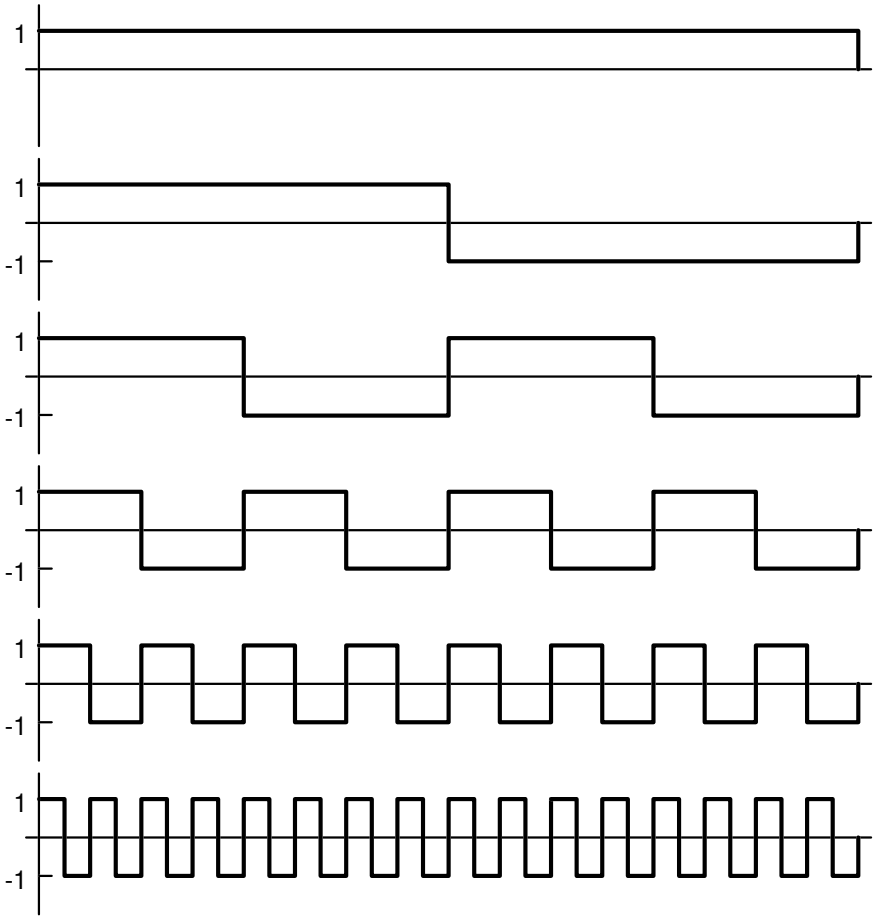
$$\mathrm{har}(0,0,t) = 1$$

$$\mathrm{har}(r,m,t) = \begin{cases} 2^{r/2} & \text{pro} \quad \dfrac{m-1}{2^r} \le t < \dfrac{m-1/2}{2^r} \\ -2^{r/2} & \text{pro} \quad \dfrac{m-1/2}{2^r} \le t < \dfrac{m}{2^r} \\ 0 & \text{jinak} \end{cases}$$

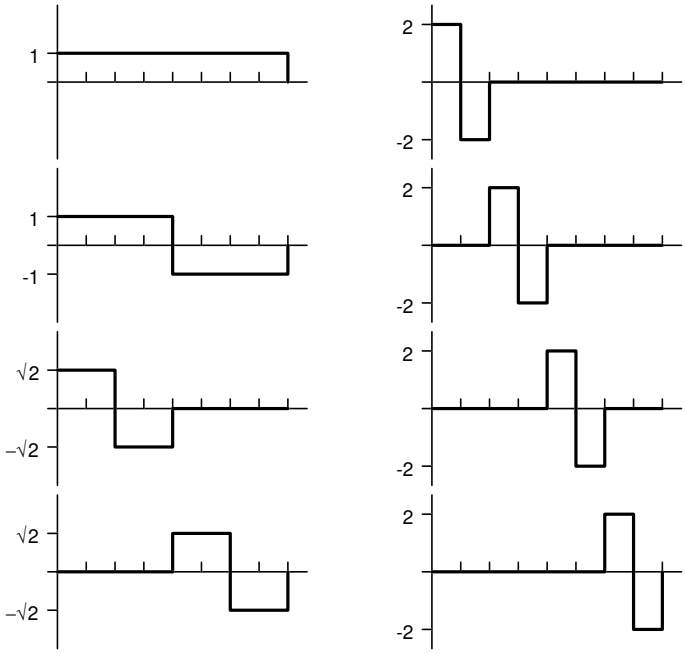
kde $\mathbf{0 \le r < \log_2 N}$ a $\mathbf{1 \le m < 2^r}$.

Ortonormální systém osmi Haarových funkcí se složí z funkcí **har(0,0,*t*)**, **har(0,1,*t*)**, **har(1,1,*t*)**, **har(1,2,*t*)**, **har(2,1,*t*)**, **har(2,2,*t*)**, **har(2,3,*t*)** a **har(2,4,*t*)**. Obdobně by se sestavovaly větší systémy.

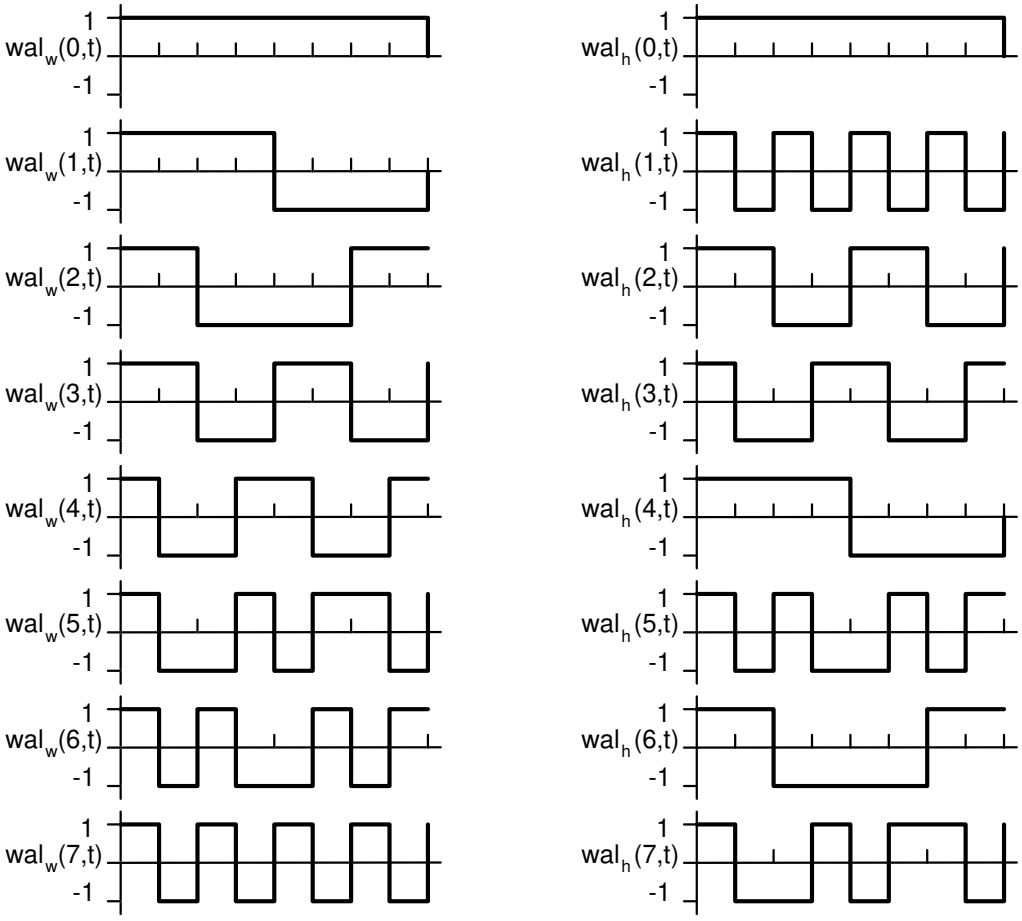
Na prvním obrázku jsou znázorněny průběhy prvních šesti Rademacherových funkcí:



Na dalším obrázku je ortonormální systém osmi Haarových funkcí:

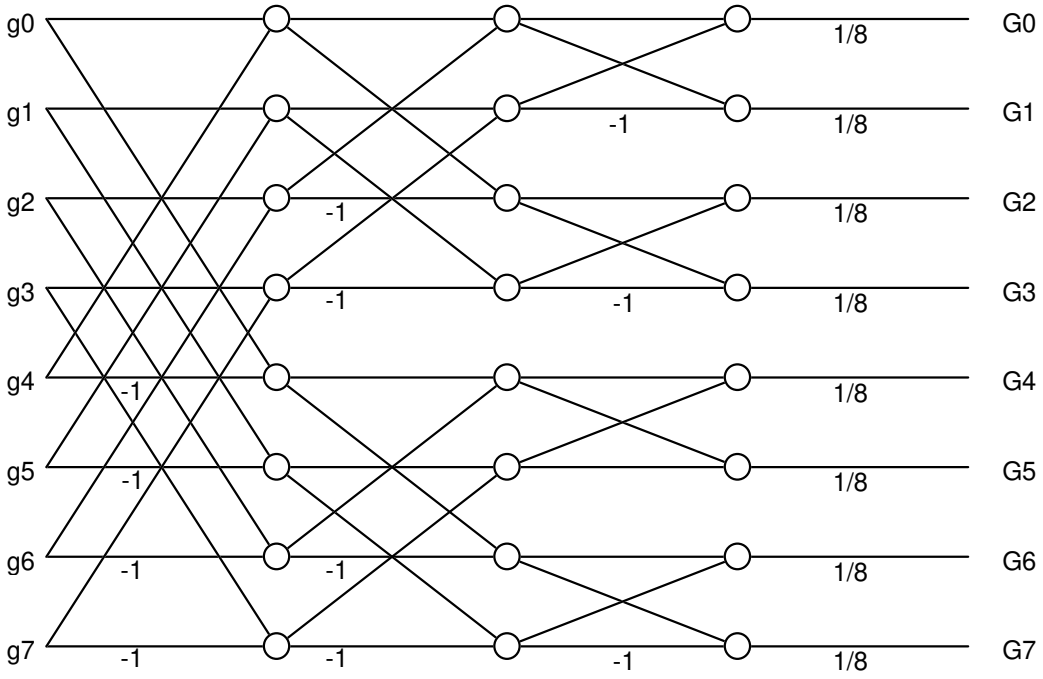


V roce 1923 J. **Walsh** doplnil pravoúhlý systém Rademacherových funkcí tak, aby byl úplný a uspořádal funkce podle rostoucího kmitočtu. Později se začalo používat ještě jiné uspořádání podle Hadamarda - proto budeme tyto dvě varianty odlišovat malým pravým indexem **w** nebo **h**. Na obrázku jsou znázorněny obě uspořádání - vlevo Walshovo a vpravo Hadamardovo:



Obou uspořádání Walshových funkcí se používá k diskrétní transformaci signálu. Podle uspořádání se mluví o **Walshově transformaci** nebo **Hadamardově transformaci** (viz např. [Ahmed75]). Tyto transformace mají skoro tak dobré vlastnosti jako např. DCT, počítají se však rychleji, protože nepoužívají operaci násobení.

Existují rychlé verze obou transformačních algoritmů pracující v čase $O(m \log m)$, v případě paralelní hardwarové implementace dokonce $O(\log m)$. Jako ukázku si zde uvedeme schéma výpočtu Hadamardovy transformace pro $m = 8$, kroužky realizují operaci sčítání, hrany diagramu označené čísly reprezentují vynásobení hodnoty příslušnou konstantou (resp. bitový posun):



5. Moderní standardy komprese obrazu

Koncem osmdesátých let začal bouřlivý rozvoj aplikací zpracovávajících obraz. Protože byly vyráběny stále dokonalejší videoadaptéry ve velkých sériích, staly se běžně dostupnými. Obrázky zobrazitelné na monitorech osobních počítačů tak několikanásobně zvětšily svoji velikost ve srovnání s předchozími lety. Díky tomu byl velký zájem odborníků soustředěn na dokonalejší metody komprese obrazu, tato snaha se projevila i v mezinárodním standardizačním úsilí. Byly vytvářeny mezinárodní komise zabývající se výběrem optimálních kódovacích algoritmů pro jednotlivé obrázky i animační sekvence. Tak vznikly postupně standardy **JPEG** (jednotlivé obrázky), **CCITT H.261** a **MPEG** (videosignál).

V této kapitole jsou jen zhruba nastíněny principy práce těchto metod, přesnější popis najde zájemce v [Hamilton92], [LeGall91], [Liou91], [Wallace91].

5.1. JPEG

Standard JPEG navrhovala smíšená komise "**Joint Photographic Experts Group**" sestavená organizacemi ISO a CCITT od roku 1986 do 1991. Cílem bylo sjednotit metody komprese

jednotlivých obrázků pro ukládání a přenos po počítačových sítích i mezi různými platformami. Základní požadavky kladené na kompresní metody byly:

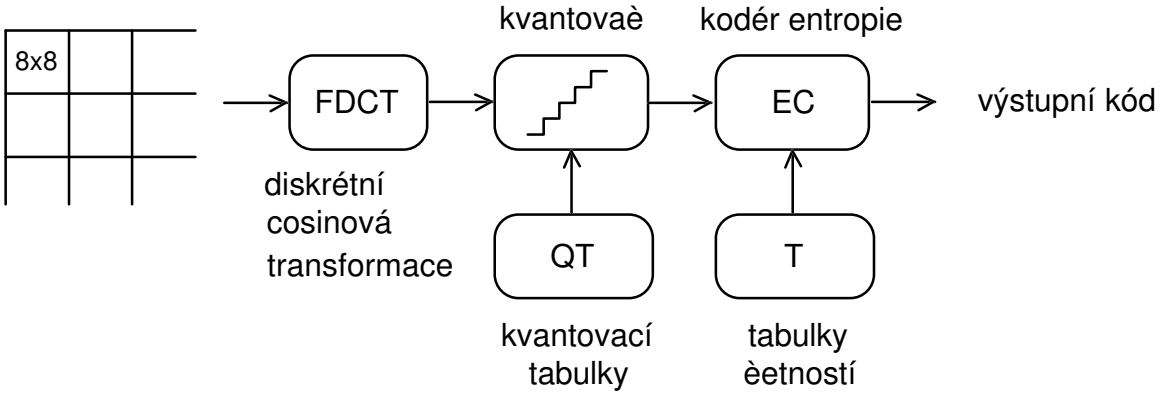
- dosahování špičkového kompresního poměru při zachování dobré kvality přenosu. Uživatel musí mít možnost nastavovat pomocí parametru požadovanou kvalitu v širokém rozsahu

- použitelnost na libovolné rastrové obrázky se spojitou škálou odstínů bez ohledu na barevnost, rozlišení, statistické vlastnosti, atd.

- rozumná výpočetní složitost pro softwarové i hardwarové implementace

- čtyři režimy kódování:
  1. bezeztrátová komprese - lze rekonstruovat originál bez jakéhokoliv zkreslení
  2. sekvenční kódování - obraz se zakóduje jediným průchodem pomocí řádkového rozkladu
  3. progresivní kódování - obraz je kódován v několika průchodech, kdy se postupně zlepšuje jeho kvalita
  4. hierarchické kódování - obraz je kódován v několika různých rozlišeních, která lze samostatně dekódovat

Komise JPEG zkoumala celkem 12 různých principů komprese, nakonec zvolila jako nejlépe vyhovující blokovou kvantovací metodu s **diskrétní cosinovou transformací** aplikovanou na bloky obrazu velikosti 8×8 pixelů. Velkých úspor se dosahuje kvantizací cosinových koeficientů, vynecháním méně podstatných vysokofrekvenčních složek a optimalizovaným entropickým kódováním (Huffmanův kód nebo aritmetické kódování). Stejnosměrné složky se kódují zvlášť pomocí diferenční metody. Blokové schéma kodéru JPEG:



Kvantování cosinových koeficientů je důležitý krok, v němž lze ovlivnit kvalitu výsledného obrazu. Během práce komise JPEG byly provedeny rozsáhlé praktické výzkumy a testy, na jejich konci stojí dvě doporučené kvantovací tabulky, první pro kódování jasové složky obrazu **Y**:

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Druhou tabulkou se kvantují barevné kanály **Cb** a **Cr**:

17	18	24	47	66	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	69	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

Algoritmus kvantování je samozřejmě univerzální, uživatel může dodat jako parametr jakoukoliv jinou kvantovací tabulku např. při nestandardním rozdělení vstupních dat. Kvantovací tabulky jsou také ovlivňovány zvolenou kvalitou kódování - předchozí vzorové tabulky byly určeny pro 50% kvalitu.

Kódování entropie může být realizováno Huffmanovým nebo aritmetickým kodérem, nulové koeficienty se přeskakují pomocí RLE. Huffmanův kodér je řízen tabulkami dodanými uživatelem, zvlášť pro stejnosměrné a střídavé složky a zvlášť pro jasový a barevný signál.

Výsledky komprese metodou JPEG se liší podle zadaného parametru kvality a podle charakteru kódovaného obrazu. Nejlepších výsledků metoda dosahuje u barevných obrázků s jemnými přechody mezi jednotlivými barvami.

Pro barevné 24-bitové obrázky se udává komprese **0.25-0.75** bitu/pixel při střední až dobré kvalitě a **0.75-2** bity/pixel při výtečné kvalitě.

Prakticky ověřené výsledky na čtyřech typických reprezentačních datech:

1. uměle vytvořený obrázek s rozlišením 1200×900×24 bitů RGB. Obsahuje velké konstantní plochy a jemné přechody mezi barvami
2. reprodukce impresionistického malíře digitalizovaná scannerem v rozlišení 600×685×24 bitů RGB
3. černobílá fotografie (portrét) digitalizovaná scannerem v kvalitě 624×735×8 bitů
4. barevná fotografie (podzimní krajina) sejmutá scannerem a předzpracovaná. Rozlišení 512×480×24 bitů RGB. Brilantní snímek, obsahuje velmi jemné detaily a téměř žádný šum

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

5.2. Doporučení CCITT H.261

Doporučení komise **CCITT č. H.261** slouží ke kódování videosignálu ve formátech CIF resp. QCIF (rozlišení jasové složky 360×288 resp. 144×180). Cílem bylo vyvinout tak efektivní postupy, aby bylo možné kódovat a dekódovat videosignál v reálném čase pro potřeby videotelefonie, videokonferencí, atp. Omezením byl bitový tok dosažitelný na kanálech **ISDN -** p× 64 kbitů/s (p = 1, 2, ...). Ve formátu QCIF a při snímkové frekvenci 10 Hz bylo třeba dosáhnout

kompresního poměru **47.5:1** pro přenos po jednoduchém kanálu 64 kbitů/s, což se nakonec autorům podařilo.

Velké **komprese bitového toku** se zde dosahuje několika metodami:

- zmenšováním rozlišení barevné složky (Cb, Cr) na polovinu
- blokovým kvantováním za pomoci **DCT**, bloky mají velikost 8×8 pixelů
- **adaptivním kvantováním** cosinových koeficientů - kvantovač je řízen komunikačním blokem: pokud je výstupní přenosový buffer naplněný, kvantuje se velice hrubě, v opačném případě se kvantovač přepne na jemnější režim s kvalitnějším přenosem
- **kompenzací pohybu** mezi jednotlivými snímky - intra-snímky a predikované snímky

Větší pozornost budeme věnovat jen metodě potlačení časové redundance - využívá se zde podobnosti sousedních snímků. U tzv. **predikovaného snímku** se jeho makrobloky (16×16 pixelů) sestavují z makrobloků předchozího snímku posunutých o vhodné vektory. Případná odchylka se zakóduje pomocí DPCM.

Tzv. **intra-snímky** jsou zcela nezávislé na okolí, používají se na začátku sekvence a po každé podstatnější změně záběru. Kódují se blokově pomocí DCT.

5.3. MPEG

Na standardu MPEG pracovala komise "**Moving Picture Experts Group**" v letech 1989 až 1990. Cílem bylo zkombinovat kompresi obrazu a zvuku tak, aby výsledný tok nepřevyšoval 1.5 Mbitů/s, a tak umožnit uložení multimediálního programu na soudobých běžně dostupných médiích - disku nebo CD. Návrh standardu je univerzální a předpokládá využití v různých aplikacích (videotex, multimedia, hry, reklama, video-mail, video-telefon, video-konference, film, TV).

Základní požadavky kladené na kódovací metody:

- **přímý přístup** k jednotlivým snímkům sekvence
- **rychloposuv s vyhledáváním** vpřed i vzad
- **zpětný chod** běžnou rychlostí
- **synchronizace** audio ↔ video
- **malá citlivost k chybám** přenosu
- **malé zpoždění** při on-line aplikacích (u video-telefonu je únosné zpoždění max. 150 ms)
- možnost **střihu** a editace
- **flexibilita** - nezávislost na rozlišení, tvaru pixelu, snímkové frekvenci
- **snadná implementace** - možnost sériové výroby VLSI hardwaru s technologií roku 1990

Několik základních přístupů potlačuje **prostorovou** a **časovou redundanci**:

- Nezávislé - tzv. **intra-snímky** (**I**) jsou kódovány pomocí blokového kvantování. Použitá transformace je **DCT** na blocích velikosti 8×8 pixelů
- tzv. **predikované snímky** (**P**) jsou závislé na nějakém předchozím - referenčním - snímku. Referenční snímek může být typu I nebo P a predikovaný snímek si z něj může vybírat bloky, které po posunutí poslouží jako **prediktory**. Kóduje se potom pouze chyba predikce - rozdíl mezi skutečnou a predikovanou hodnotou
- nejsložitější je kódování tzv. oboustranně **interpolovaných snímků** (**B**), ty mohou být závislé na dvou referenčních snímcích - jednom předcházejícím a jednom následujícím.

Typická sekvence zakódovaných snímků má asi takové složení, tučně jsou vyznačeny referenční snímky (**I** a **P**):

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

Pro snadnou možnost vyhledávání a přímého přístupu se doporučuje, aby mezi dvěma referenčními snímky nebyla prodleva větší než 1/10 s.

Z hlediska návrhu je zajímavá implementace predikovaných a interpolovaných snímků: každý snímek je rozdělen na tzv. **makrobloky** velikosti 16×16 pixelů. Každý makroblok může mít jeden ze čtyř typů:

- **Intra-makroblok** - nezávisí na jiných snímcích (pouze takové makrobloky se smějí vyskytovat ve snímku typu **I**)
- **přímo predikovaný makroblok** - závisí na předchozím referenčním snímku, při predikci se kompenzuje pohyb a kóduje se jen chyba predikce
- **zpětně predikovaný makroblok** - závisí na budoucím referenčním snímku, jinak se neliší od předchozího typu
- **průměrovaný makroblok** - závisí na předchozím i budoucím referenčním snímku, oba vzory se nejprve opraví o pohybové vektory a pak se spočítá predikce jako jejich aritmetický průměr. Kóduje se jen chyba této predikce.

Díky filozofii zpětné predikce je nutné jednotlivé snímky před přenosem trochu zpřeházet, aby dekodér byl schopen dekódovat interpolované snímky. Posloupnost uvedená výše se tak bude přenášet jako

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

6. Literatura

[Adámek89] Adámek J.: Kódování, SNTL, Praha 1989

[Ahmed75] Ahmed N., Rao K.: Orthogonal Transforms for Digital Signal Processing, Springer-Verlag, Berlin 1975

[Bell89] Bell T., Witten I., Cleary J.: Modeling for Text Compression, ACM Computing Surveys, Vol.21, No.4, December 1989

[Cooley65] Cooley, J.-Tukey, J.: An Algorithm for the Machine Computation of Complex Fourier Series, Math. Comput., No.19, 1965, pp. 297-301

[Chen77] Chen W., Smith H., Fralick S.: A Fast Computational Algorithm for the Discrete Cosine Transform, IEEE Transactions on Communications, Vol. COM-25, No.9, September 1977, pp. 1004-1009

[Chen90] Chen D., Bovik A.: Visual Pattern Image Coding, IEEE Transactions on Communications, Vol. COM-38, No.12, December 1990, pp. 2137-2146

[Chen92] Chen D., Bovik A.: Hierarchical Visual Pattern Image Coding, IEEE Transactions on Communications, Vol. COM-40, No.4, April 1992, pp. 671-675

[Foley90] Foley J., van Dam A., Feiner S., Hughes J.: Computer Graphics, Principles and Practice, second edition, Addison-Wesley, Reading 1990

[Frydrych93] Frydrych M.: Komprese obrazové informace, diplomová práce, MFF UK Praha 1993

[GIF90] GIF™ - Graphics Interchange Format™ - A standard defining a mechanism for the storage and transmission of raster-based graphics information, Version 87a, 89a, CompuServe Inc., Columbus, Ohio 1990

[Gonzales87] Gonzales R., Wintz P.: Digital Image Processing, second edition, Addison-Wesley, Reading 1987

[Hamilton92] Hamilton E.: JPEG File Interchange Format, Version 1.02, C-Cube Microsystems, 1992

[Haykin89] Haykin S.: An Introduction to Analog and Digital Communications, Wiley, New York 1989

[Huffman52] Huffman D.: A Method for the Constructin of Minimum Redundancy Codes, Proceedings of the Institute of Radio Engineers 40, 9(September), 1952, pp. 1098-1101

[Jain81] Jain A.: Image Data Compression: A Review, Proceedings of the IEEE, Vol.69, No.3, March 1981, pp. 349-389

[Kučera89] Kučera L., Nešetřil J.: Algebraické metody diskrétní matematiky, SNTL, Praha 1989

[Kunt85] Kunt M., Ikonomopoulos A., Kochler M.: Second-Generation Image-Coding Techniques, Proceedings of the IEEE, Vol.73, No.4, April 1985, pp. 549-574

[LeGall91] Le Gall D.: MPEG: A Video Compression Standard for Multimedia Applications, Communications of the ACM, Vol.34, No.4, April 1991, pp. 47-58

[Limb77] Limb J., Rubinstein C., Thompson J.: Digital Coding of Color Video Signals - A Review, IEEE Transactions on Communications, vol. COM-25, November 1977, pp. 1349-1385

[Liou91] Liou M.: Overview of the px64 kbit/s Video Coding Standard, Communications of the ACM, Vol.34, No.4, April 1991, pp. 60-63

[Loeffler89] Loeffler C., Ligtenberg A., Moschytz G.: Practical Fast 1-D DCT Algorithms with 11 Multiplications, IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing 1989 (ICASSP '89), pp. 988-991

[Netravali80] Netravali A., Limb J.: Picture Coding: A Review, Proceedings of the IEEE, Vol.68, No.3, March 1980, pp. 366-406

[Pritchard77] Pritchard D.: U.S. Color Television Fundamentals - A Review, IEEE Transactions on Consumer Electronics, vol. CE-23, No.4, November 1977, pp. 467-478

[Rao90] Rao K., Yip P.: Discrete Cosine Transform - Algorithms, Advantages, Applications, Academic Press, London 1990

[Skala91] Skala V.: Počítačová grafika II, VŠSE Plzeň, 1991

[Shannon48] Shannon C.: The mathematical theory of communication, Parts I and II, Bell Syst. Tech. J., Vol.27, 1948, pp. 379, 623

[Skarbek93] Skarbek W.: Methods of Digital Image Archivization, Part Three: Compressing Images, Machine Graphics and Vision, Vol.2, No.1, 1993, pp. 53-86

[Special80] Proceedings of the IEEE - Special Issue On Digital Encoding of Graphics, edited by Netravali A., Vol.68, No.7, July 1980, pp. 755-944

[Wallace91] Wallace G.: The JPEG Still Picture Compression Standard, Communications of the ACM, Vol.34, No.4, April 1991, pp. 31-44

[Welch84] Welch T.: A technique for high-performance data compression, IEEE Computer, Vol.17, No.6, June 1984, pp. 8-19

[Ziv77] Ziv J., Lempel A.: A universal algorithm for sequential data compression, IEEE Trans. Inf. Theory, Vol. IT-23, No.3, May 1977, pp. 337-343

[Ziv78] Ziv J., Lempel A.: Compression of individual sequences via variable-rate coding, IEEE Trans. Inf. Theory, Vol. IT-24, No.5, September 1978, pp. 530-536