

VIDEO VQ-KODÉR – IMPLEMENTACE V MATLABU

Pavel Hanzlík, Petr Páta

Katedra Radioelektroniky, ČVUT v Praze, Fakulta Elektrotechnická, Technická 2, 166 27, Praha 6, Česká Republika

Abstrakt. *Ačkoli jsme již dnes snadno schopni uchovávat velké objemy dat, myšlenka komprese dat je stále velmi aktuální. Zejména pokud hovoříme o datech typu video, kde narážíme na potřebu rychle zpracovávat velké objemy dat na vstupní straně komunikačního řetězce. Redukce redundance a irrelevance byla již velmi dobře vyřešena návrháři systémů MPEG, ale v dnešní multimediální společnosti s poptávkou po integraci velkého počtu komunikačních služeb do jediného komplexního systému také hledáme způsoby, jak urychlit přenos dat. Jedním z možných řešení je použití multidimenzionální vektorové kvantizace. Vektorová kvantizace již našla své použití v aplikacích používajících kompresi zvuku a řeči, při konstrukci psychoakustických modelů a v dalších specializovaných oborech. Rádi bychom se zaměřili na její použití ve videu – MPEG 4 standard – nebo v kompresi vědeckých obrazů (obrazů s vysokým rozlišením).*

1. Úvod

Tento článek představuje implementaci a strukturu videokodéru (soustava kodér - dekodér) s použitím vektorové kvantizace (VQ) v programu Matlab. Také se zaměřuje na silné a slabé stránky VQ a popisuje experiment, který využívá výhodných vlastností VQ v kompresi dat typu video. Vektorová kvantizace je již platnou součástí aplikací s kompresí zvuku a řeči, a také některé její základní charakteristiky hovoří pro její výhodné použití ve videu. Proto pravděpodobně nejdůležitější částí návrhu video kodéru je důkladná studie umístění VQ do procesu komprese dat a optimalizace jejího použití pro konkrétní typ přenosu. Návrh smysluplné kompletní koncepce kódovacího řetězce s VQ by měl vést tomu, že chyba mezi daty vstupujícími do kodéru a vystupujícími z dekodéru je buď minimální nebo nepostřehnutelná pro pozorovatele.

Vektorová kvantizace je výhodná metoda pro kompresi dat, pokud hovoříme o texturách, nebo statických obrázcích. Její použití pro vědecké obrázky (obrazy s vysokým rozlišením) nebo ve videu není ještě plně prozkoumanou oblastí. Standard MPEG 4 je otevřeně definován (bez striktních limitací) a struktura videokodeku s použitím VQ by se mohla stát jeho důležitou součástí.

2. Koncepce použití vektorové kvantizace ve videosekvenci tvořené „I“ snímky

Pravděpodobně nejjednodušší přístup k řešení této problematiky můžeme realizovat použitím VQ na obrazový datový tok tvořený „I“ snímky. Některé výhody této metody komprese z oblasti statických snímků mohou být použity i pro videosekvenci. Kódová kniha je tvořena speciálně navrženou množinou subbloků jednotlivých snímků sekvence. Tento přístup je zkoumán i v následujícím experimentu, ale je zde také několik dalších řešení, která mohou sloužit jako účel pro další experimenty v budoucnu.

Použití vektorové kvantizace nám např. umožňuje zaměnit Diskrétní Kosinovou Transformaci na začátku standardního kódovacího řetězce MPEG za transformaci Karhunen-Loève. Potíží při použití různých bází KLT pro jednotlivé realizace bloků se můžeme vyhnout s pomocí předdefinované množiny vektorů charakterizujících báze, kódované vektorovou kvantizací.

Dalším příkladem může být použití VQ s proměnnou délkou slova (dimenzí) ihned za výpočtem DCT ve standardním kódovacím řetězci MPEG. Vektory s proměnnou délkou slova jsou zde vytvořeny stylem čtení „zig-zag“ nenulových koeficientů ve 2D-DCT matici.

Také můžeme studovat použití fraktálové VQ s variabilní dimenzí na snímky „I, B, P“ ve výsledném toku dat MPEG. Dalším přístupem může být řešení s použitím VQ v časové dimenzi. Například pokud použijeme zásobník (buffer) pro několik snímků a zkonstruujeme vektory napříč časovou oblastí. Vektory jsou konstruovány buď z hodnot na přesných pozicích x, y v obrazové matici, nebo sledováním predikce pohybu různých videoobjektů (MPEG 4, 7).

Pokud se zaměříme na analogii s kompresí statických obrazů, musíme zmínit iterativní algoritmy minimalizující chybu komprese. Právě LBG nebo FCM (Fuzzy Clustering Method) algoritmy generují kódovou knihu několika opakujícími se iterativními kroky (viz Obr. 1,2,3 a 4).

Pokud známe statistické rozložení vektorů v Euklidově prostoru (m -dimenzionální vektor) a počáteční počet všech vektorů, můžeme zkonstruovat LBG algoritmus opakovaným přepisováním množiny referenčních vektorů podle středu aktuální Voronoiho buňky. VQ je relace Q z k -dimenzionálního Eukleidovského prostoru R^k do konečné množiny Y z R^k .

Platí:

$$Q: R^k \rightarrow Y,$$

kde

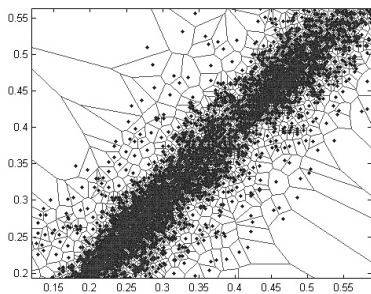
$$Y = (\hat{x}_i; i = 1, 2, \dots, N)$$

je množina předdefinovaných vektorů a N je počet vektorů v této množině.

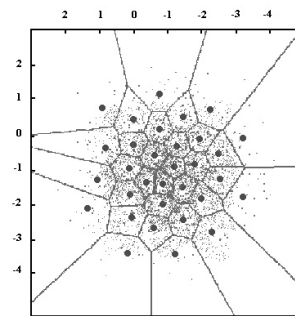
Definujeme měření zkreslení

$$D(x, q(x)) = \frac{1}{N} \sum_{i=0}^{N-1} (x_i - \hat{x}_i)^2,$$

keré měří chybu reprezentace vektorů x jeho modelem x' . Tato chyba může být iterativně minimalizována přepisováním vektorů v takzvané „*trénovací množině*“.



Obr 1. První konstrukce Voronoiho diagramu



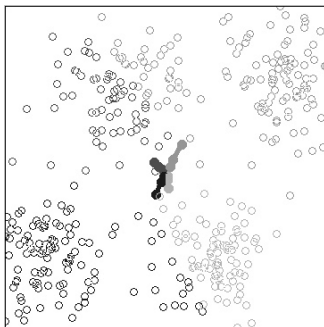
Obr 2. Voronoiho diagram s centroidy

Centroid (část trénovací množiny) je generován z n Voronoiho buněk korespondujících s rozmístěním vektorů v Euklidově prostoru. Algoritmus má dva základní kroky, které po několika iteracích konvergují podle kritéria minimální (kvadratické) chyby.

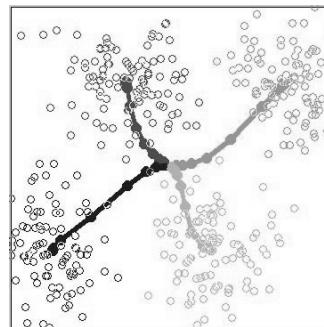
Každá iterace má dvě části – zapsání kódové knihy a její vyladění podle charakteru obrazu. Vyladění je realizováno iterativně přepisováním aktuální trénovací množiny její novou verzí. Pokud dospějeme při výpočtech z různých rozvržení Voronoiho diagramů k lepším výsledkům v MSE, nová trénovací množina se stane referenční. Tento proces konverguje a minimalizuje kvantizační chybu.

“K-means“ metoda – metoda fuzzy shlukování

Zde je myšlenka stejná jako v předchozím odstavci, ale postup při vyladění kódové knihy je zjednodušen, není třeba vypočítávat Voronoiho diagramy. Trénovací množina je předdefinována náhodně, kritérium iterativity je realizováno průměrováním střední Eukleidovské vzdálenosti v m -dimenzionálním prostoru mezi jednotlivými nejbližšími sousedy. Je také respektována váha aktuálního čísla iterace. Tato metoda také konverguje, dříve, s dosažením podobných výsledků.



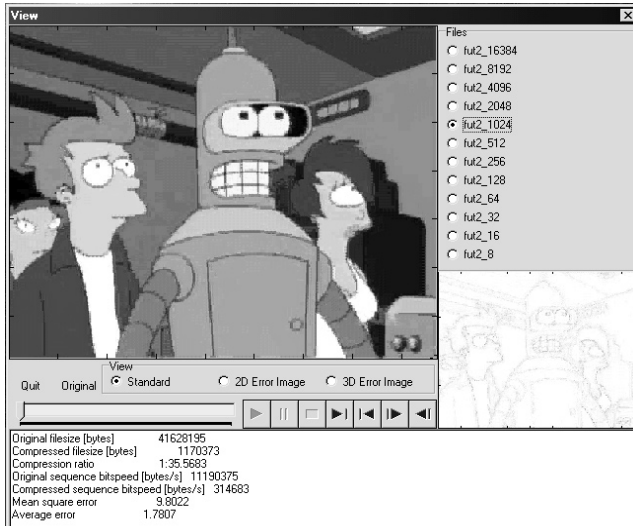
Obr 3. Po prvních 3 iteracích *k-means fcm* metody



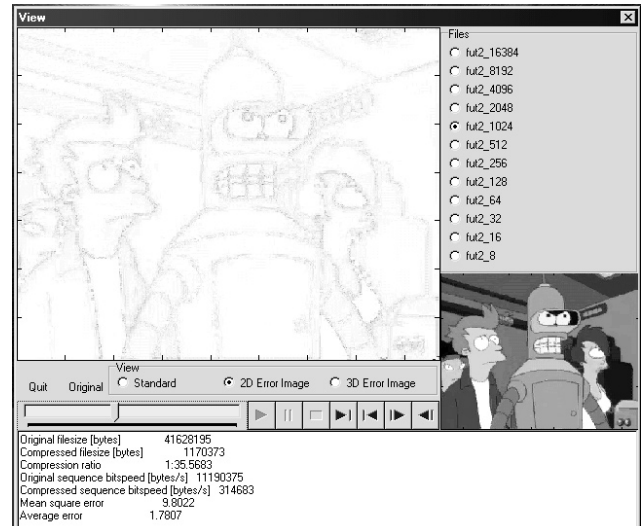
Obr 4. Po poslední (osmé) iteraci *k-means fcm* metody

3. Experimentální část – koncepce v Matlabu

Celková koncepce kodéru je ve svém jádru vektorová kvantizace s adaptivní kódovou knihou. Ta je generována pro vstupní videosekvenci alternativním algoritmem uniformní redukce kódové knihy. Vstupními proměnnými vektorového kvantizéru jsou vektory tvořené ze subbloků velikosti 4x3 obrazové body, snímků vstupní videosekvence. Protože uvažujeme barevnou vstupní videosekvenci, dimenze kvantizovaného vektoru je 36 (tři složky – R,G,B). V experimentu používáme prvních 100 snímků videosekvence (25fps – 8 sec.). Tímto docílíme toho, že motivy obrazu se výrazně nemění v čase a tak může být kódová kniha navržena ze všech vektorů dané sekvence. I když je některý motiv zobrazen pouze po krátkou chvíli, stane se součástí kódové knihy. Toto řešení může samozřejmě být modifikováno s přispěním další analýzy obrazu, např. použití této metody pro jeden celý záběr (od střihu ke střihu).



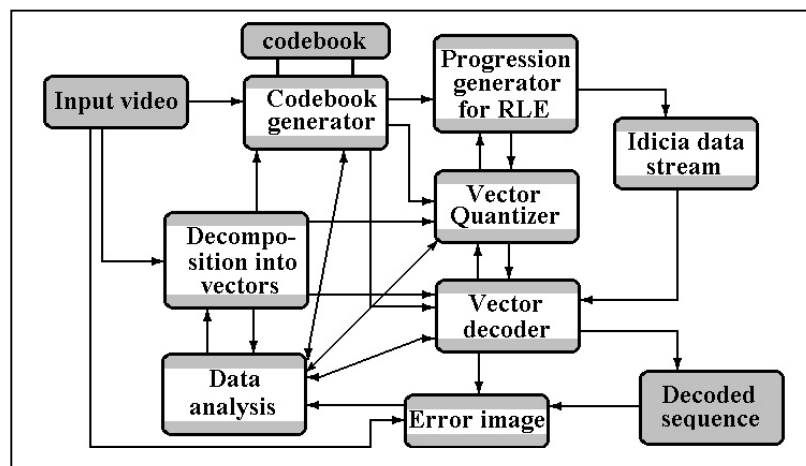
Obr 5.



Obr 6.

Celková struktura návrhu:

- Dekompozice – Vstupní barevná videosekvence je převedena na jednotně velké snímky (výška, šířka) v Matlabu, které tvoří základní „I“ snímky kódované sekvence. Po předchozím zkoumání jsme zvolili velikost porovnatelnou s některými profesionálně navrženými standardy, 435x343 obrazových bodů. Videosekvence je pak tvořena jednotlivými snímky s frekvencí 25 snímků za sekundu.
- Kodér – Vstupní jednotkou do algoritmu VQ je jeden vektor originálního „I“ snímku. Proto také hlavní funkcí kodéru je konverze jednotlivých snímků do vektorové reprezentace obrazového signálu. Vektor má dimenzi 36. Součástí kodéru v experimentu je také kontrolní dekodér, který převádí vektorovou reprezentaci zpět na obrazovou informaci. Tento reverzní algoritmus ověřuje bezztrátovou konverzi v této části kompresního procesu.



Obr 7. Blokový diagram struktury video VQ kodeku

- Generátor kódové knihy – jedna z nejnáročnějších částí programu. Všechny vektory z množiny 100 snímků jsou analyzovány. Uniformní redukci jsou eliminovány ty vektory, které nebudou součástí konečné kódové knihy. Kvalita obrazu nesmí být degradována a velikost kódové knihy musí být maximálně optimalizována uživatelem přidělené datové kapacitě, použité pro kódování resp. přenos.
- Kvantizér – Pokud hovoříme o počtu operací a časové náročnosti, je toto nejnáročnější část programu. Každému vektoru v kódové knize přidělíme ukazatel (indicii). Metodou postupného procházení seznamu vektorů hledáme vektor s minimální odchylkou. Zde můžeme použít různá kritéria od obvyklé střední kvadratické odchylky až po průměrnou chybu nebo chybu minimálního součtu. Podle zvoleného kritéria je vybrán nejlepší vektor, který je zapsán do matice dekódovaného obrazu a jeho ukazatel (indicie) je uložen. Výsledná posloupnost indicií může být komprimována obvyklými algoritmy kódování běhu slova.
- Dekodér – Vektorovou reprezentaci dekódovanou z toku ukazatelů již není obtížné dekódovat algoritmem reverzním vůči tomu, použitému v části kodér.
- Analýza – Tato část je propojena se všemi dříve zmíněnými bloky a umožňuje nám dynamicky měnit metody použité v jednotlivých krocích kódovacího řetězce, za účelem dosažení nejlepších možných výsledků. Data získaná v části kvantizér jsou zde také bezztrátově kódována buď RLE kóděrem nebo Huffmannovým entropickým kóděrem. Je vypočtena výsledná bitová rychlost a objektivní kvalita dekódovaných obrazů.

Po předchozím zkoumání velikosti základního bloku (1x2, 2x2,..., 8x8, atd.) jsme zvolili velikost bloku 4x3 obrazové body. Čtyři na délku (horizontálně) a tři na výšku (vertikálně). Každý obrazový bod je díky RGB souřadnicím reprezentován třemi čísly z intervalu celých čísel od 0 – 255. Tím dostáváme 8 bitů pro každou souřadnici, 24 bitů celkově. Základní blok 4x3 obrazové body je tedy reprezentován 36 čísly - vektorem.

Uniformní redukce kódové knihy

Hlavní problém v aplikacích s VQ je stále optimalita použité kódové knihy. Navrhujeme kódovou knihu tak, abychom dosáhli minimální odchylky mezi kódovaným a dekódovaným obrazem. Vektorová kvantizace s adaptivní kódovou knihou aproximuje vlastnosti optimální a univerzální kódové knihy a je schopna flexibilně se přizpůsobit momentálním nárokům uživatele. Suboptimalita však v tomto případě nesmí být znatelná pozorovateli.

V první fázi konstrukce kódové knihy vybereme všechny vektory z definované množiny snímků. Tato počáteční kódová kniha je tzv. „kompletní“. Základem „prvně redukované“ kódové knihy jsou pouze reprezentativní vektory jednotlivých podmnožin vysoce korelovaných vektorů. Další redukce může následovat pomocí optimalizujících iterativních algoritmů, které běžně používáme pro VQ (LBG, GLA algoritmy). Bohužel, tyto algoritmy vysoce prodlužují celkový kompresní čas. Tyto algoritmy snadno dosáhnou minimálních chyb komprese, ale i použití robustnějších metod přináší srovnatelné výsledky a to v daleko kratším čase.

Také si uvědomme, že pokud pozorujeme videosekvenci s pulsníkovou frekvencí 25fps, lidské oko je stále schopno integrovat jednotlivé snímky a akceptovat vyšší chybovost, než je viditelné na samostatném statickém obraze.

„Kompletní“ kódová kniha je nejprve redukována eliminací vícenásobně se vyskytujících vektorů. Ve zbylé množině vysoce korelovaných vektorů také vybereme pouze ty nejreprezentativnější. V této fázi je kompresní chyba stále velmi malá a vizuálně nepostřehnutelná po dekódování. Jsme stále schopni dál zmenšovat tuto „prvně redukovanou“ kódovou knihu výběrem uniformně rozmístěných vektorů z celkové množiny vektorů v Euklidovské prostoru. Maximální velikost kódové knihy je limitována pouze kvalitativními požadavky uživatele.

Předchozími empirickými studiemi byla vytvořena aproximační tabulka, která ukazuje minimální velikost „prvně redukované“ kódové knihy, abychom dosáhli uniformní redukci minimální chybu a zároveň nejefektivnější využití velikosti „konečné“ kódové knihy.

Konečná kód.k. - max. vel.	1638	4096	1024	256	64	32	16
Prvně redukov. kód. k. - min. vel.	8000	1800	4000	1500	40	20	10
	0	0			0	0	0

Tab 1. Velikosti „prvně redukované“ a „konečné“ kódové knihy

Kódové knihy jsou vybrány pomocí následujícího kritéria: vektor, který nebyl eliminován v druhé fázi redukci, (přechod od prvně redukované ke konečné kódové knize) je nejreprezentativnější z konkrétní množiny vektorů. Tato konkrétní množina vektorů se nerozkládá dále než 150 [q²] - (q je kvantizační krok skalární uniformní kvantizace RGB složek 0-255) - střední kvadratické Eukleidovské vzdálenosti. Porušení tohoto kritéria by způsobilo viditelné a

nepříjemné jevy v charakteru jednotlivých snímků (bloková struktura). Tento efekt pravděpodobně nemůžeme eliminovat při nižších velikostech kódové knihy, ale pro danou velikost můžeme tento efekt oddálit.

Poslední fáze – nejvíce ohrožující kvalitu dekodovaného videa – je fáze eliminace nejbližší susedících vektorů z předchozí fáze (tímto myslíme iterativní eliminaci podle kritéria minimální kvadratické Eukleidovské vzdálenosti, aplikované na celou množinu zbývajících vektorů). Obraz může ztratit jemné přechody mezi barvami, detaily obrazu, ale motiv obrazu zůstane zachován. Kvalita obrazu závisí na nárocích uživatele.

Kódování ukazatelů (datová část VQ – indicie)

Jestliže je kódovací předpis (kódová kniha) známa na obou stranách přenosového řetězce, můžeme videosekvenci zakódovat pomocí posloupnosti indicíí – ukazatelů na přesné místo v prostoru kódové knihy.

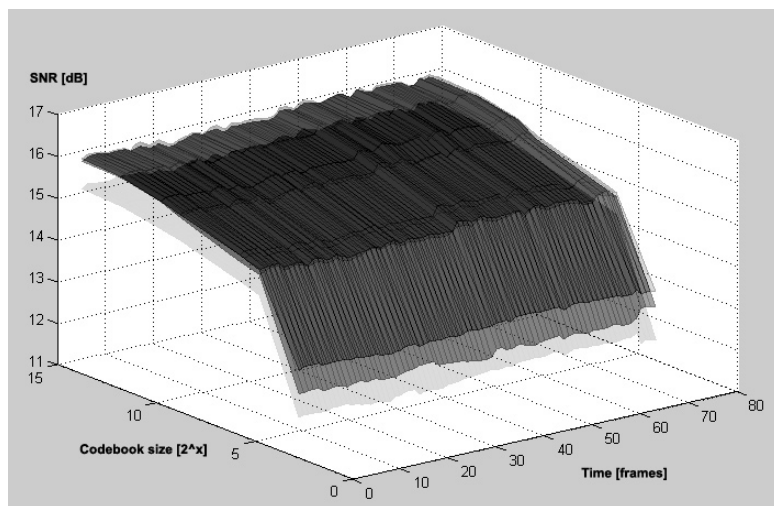
Například, pokud je kódová kniha tvořena 128 vektory, každý reprezentuje subblok 8x8 pixelů, můžeme přenést celý subblok pomocí jediné řadové číslovky ukazující na umístění vektoru v kódové knize.

Toto má za následek, a to také v případě, že je kódová kniha součástí přenosu dat, vysokou kompresi potřebné přenosové kapacity. Hlavním důvodem je vysoká korelace vektorů nejen v rovině obrazů, ale i v čase.

Přenos indicíí může být dále komprimován některou z bezztrátových kompresních metod, jako např. RLE (Run Length Encoding) nebo Huffmanovým entropickým kóděrem. Kódér i dekodér potřebují snímkový zásobník (buffer) na předzpracování videosekvence pro konkrétní typ datového přenosu.

4. Porovnání s ostatními kompresními metodami

Pokud chceme porovnávat výše zmíněnou metodu s některým z ostatních profesionálně navržených kompresních algoritmů, můžeme s výhodou použít Matlab k vytvoření 3D simulace, která by zachycovala chování kodéru v čase. V následujícím grafu (obr 8.) je na ose x zachycena časová dimenze, na ose y je výsledné SNR a na ose z velikosti jednotlivých použitých kódových knih.



Obr 8. R,G,B složky SNR v závislosti na čase a velikosti použité kódové knihy

Správnost návrhu kódové knihy pro celou videosekvenci můžeme snadno ověřit tím, že SNR by mělo být přibližně konstantní - nezávislé na časové dimenzi. Jak můžeme vysledovat ze získaných dat, ačkoliv SNR nedosahuje kvalit kompresních metod statického obrazu, tato metoda optimalizuje opravdu celou videosekvenci a ne pouze některé vybrané snímky. V každé rovině RGB obrazu dosahujeme SNR od 15 do 17dB. Pokud zvyšujeme velikost kódové knihy, také SNR stoupá.

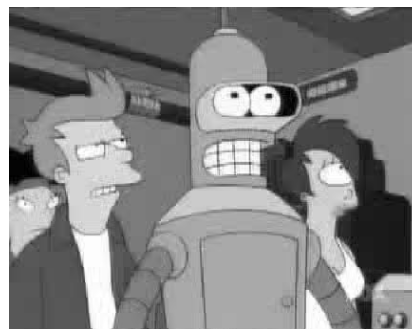
Pro srovnání, pokud použijeme jednoduchou Fourierovu transformaci (velikost bloku 16, bitová hloubka 8) stejného vstupního obrazu, aplikovanou na každou RGB složku zvlášť, rekonstruujeme výsledný obraz se SNR 33-36dB. Toto měření je sice aktuální pouze pro jediný statický obraz z videosekvence a neukazuje posloupnost v časové oblasti, výsledky jsou však výrazně lepší.

Diskrétní Kosinová Transformace se stejnými vlastnostmi (velikost bloku 16, bitová hloubka 8) je na tom o malinko hůře, ale stále dosáhneme SNR 26-29dB. Subjektivní kvalita rekonstruovaného obrazu je však již srovnatelná s VQ. Také jsme studovali Walsh-Hadamardovu a Slantovou transformaci a dospěli jsme k zajímavému

zjištění. Ačkoliv dosáhneme těmito transformacemi SNR 23-26dB, resp. 24-25dB, subjektivní kvalita obrazu se pozorovateli jeví lepší při použití VQ.



Obr 9. Walsh-Hadamardova Transform.: SNR 26dB



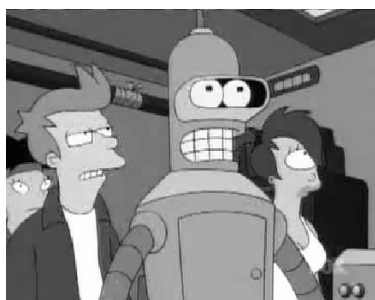
Obr 10. Slantova Transformace: SNR 25dB

Fourierova transformace se zdá být nejlepší použitou metodou, v tomto případě všechny ostatní metody jsou srovnatelné s vektorovou kvantizací kritériem subjektivního vnímání kvality obrazu. Snímky kódované pomocí VQ se jeví v mnoha případech pozorovateli lépe s použitím VQ, ačkoliv naměřené SNR je nižší. DCT, ani Walsh-Hadamardova nebo Slantova Transformace se ale nemůže srovnávat s VQ v případě, pokud porovnávaným parametrem bude kompresní poměr. Pokud výše zmíněnými metodami dosáhneme kompresního poměru mezi 2:1 až 6:1, kompresní poměry při použití VQ 25:1 až 75:1 jsou výrazně lepší.

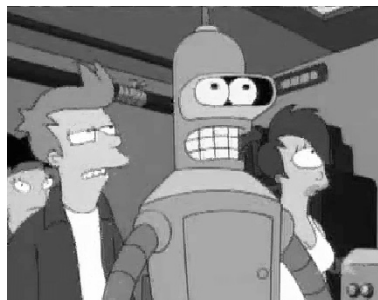
Original video-sequence size	41.628.195 [bytes]
Bit rate of the original video-sequence	11.190.375 [bytes/s]

Code-book size [bits]	2^x	Compressed video-seq. [kB]	Compression ratio		Reduced bit rate [kB/s]	MSE	Avg. Err.
16384	14	2063	1:20,1763	1.895bpp	555,68	6,47%	1,06%
8192	13	1873	1:22,2197	1.080bpp	504,49	7,33%	1,22%
4096	12	1770	1:23,5154	1.021bpp	476,24	7,83%	1,32%
2048	11	1575	1:26,4372	0.9078bpp	423,45	8,72%	1,58%
1024	10	1170	1:35,5683	0.6748bpp	314,68	9,80%	1,78%
512	9	1073	1:38,7885	0.6187bpp	288,56	10,13%	2,07%
256	8	888	1:46,8890	0.5118bpp	238,70	11,14%	2,44%
128	7	774	1:53,7537	0.4465bpp	208,25	11,83%	2,74%
64	6	565	1:73,7070	0.3256bpp	151,87	12,68%	3,03%
32	5	382	1:109,0312	0.2201bpp	102,66	13,02%	3,24%
16	4	279	1:149,1882	0.1609bpp	75,03	16,09%	4,59%
8	3	202	1:205,6601	0.1167bpp	54,43	20,69%	6,98%

Tab 2. Kompletní tabulka měřených parametrů videosekvence



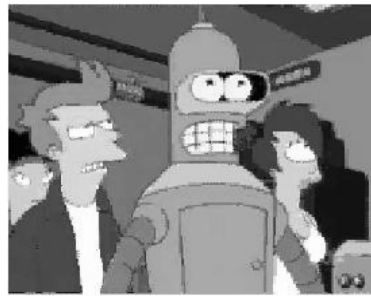
Obr 11. Originál vstupního obrazu



Obr 12. Velikost kód. k.: 16384 elementů (2^{14})



Velikost kód. k.: 4096 elementů (2^{12})



Velikost kód. k.: 256 elementů (2^8)



Velikost kód. k.: 64 elementů (2^6)

5. Závěr

Klasické algoritmy generace kódové knihy pro VQ jsou založeny na základech komprese statických obrazů a textur. Mají silnou zpětnou vazbu k originálnímu obrazu a efektivně tak dosahují minimalizace kvantizační chyby. U dat typu video je ale třeba tento přístup změnit, zejména je pak důležité je, zaměřit se na správný návrh struktury video kodéru. Zde bychom opravdu mohli s výhodou využít některých dobrých vlastností VQ a zakomponovat je do MPEG 4 – otevřeného video standardu – což by vedlo jistě ke zkvalitnění služeb tohoto standardu.

Kompresní časy jsou v tomto případě stále ještě příliš dlouhé pro online zpracování, ale vzhledem ke strmému stoupání možností v oboru informačních technologií, pravděpodobně již brzy nebudeme muset na tento problém hledět a budeme používat rychlejší kódovací přístroje. Redukce datového toku může najít své uplatnění například v aplikacích pro UMTS, nebo jiných vysokorychlostních video aplikacích.

Nové verze Matlabu jsou v tomto případě svým maticově orientovaným zaměřením dobrou pomůckou, zejména s kompletní knihovnou zpracování obrazu. Vizuální a programovací rozhraní umožňuje uživateli dynamicky měnit své požadavky a nastavení jednotlivých algoritmů za účelem získání nejlepších možných výsledků.

6. Poděkování

Tato práce byla vytvořena na katedře radioelektroniky ČVUT – FEL v Praze pod patronací výzkumného projektu "Kvalitativní aspekty kompresních metod obrazu v multimediálních systémech" a byla podporována grantem č. 102/02/0133 Grantové Agentury České Republiky

Partie práce zabývající se návrhem komprese pro vysokorychlostní přenos video dat byly podpořeny grantem v rámci výzkumného záměru doktorského projektu Grantové agentury ČR č. 102/03/H109 „Metody, struktury a komponenty elektronické bezdrátové komunikace“.

7. Reference

- [1] Gersho, A., Gray, R.M. – **VQ and signal compression** C1992
- [2] Abut, H. – **Vector Quantization** C1990
- [3] Klíma M., Bernas M., Hozman J., Dvořák P. - **Zpracování obrazové informace** (•VUT 1996)
- [4] Vít, V. a kolektiv - **Televizní technika** SNTL Praha, 1979
- [5] Košťál, E. - **Obrazová a televizní technika II. - Televize**, •VUT 1998
- [6] **Encyclopedia of computing science and technology** 29 (supplement 14)
- [7] **Encyclopedia of computing science and technology** 33 (supplement 18)

8. Kontakt

Hanzlip@feld.cvut.cz, Pata@feld.cvut.cz

Katedra Radioelektroniky
ČVUT FEL v Praze
Technická 2
Praha 6, 166 27
Česká Republika