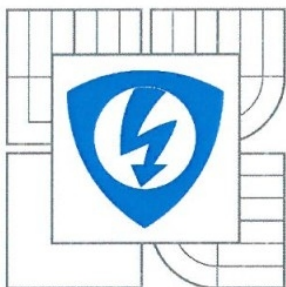


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

HIERARCHICKÉ KÓDOVÁNÍ POHYBLIVÝCH OBRAZŮ

HIERARCHICAL CODING OF MOVING PICTURES

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

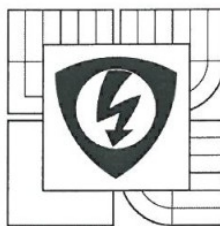
AUTOR PRÁCE
AUTHOR

MILAN ŠPAČEK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. JIŘÍ SCHIMMEL, Ph.D.

BRNO 2011



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student: Milan Špaček

Ročník: 3

ID: 119338

Akademický rok: 2010/11

NÁZEV TÉMATU:

Hierarchické kódování pohyblivých obrazů

POKYNY PRO VYPRACOVÁNÍ:

Prostudujte metody škálování u ztrátové komprese videosignálu podle standardu MPEG-2 a MPEG-4. V prostředí Matlab vytvořte výukovou aplikaci s vlastním grafickým uživatelským rozhraním pro prostorové škálování, kvalitativní škálování typu SNR scalable a Fine granular, které podle zadaných parametrů provedou dané škálování obrazu a zobrazí výstupy jednotlivých vrstev po provedení škálování a po rekonstrukci obrazu. Vstupem programu bude obrázek uložený jako bitová mapa ve formátu RGB.

DOPORUČENÁ LITERATURA:

[1] Iain E. G. Richardson, H.264 and MPEG-4 Video Compression. West Sussex; Willey; 2003; 306s; ISBN 0-470-84837-5.

[2] L. Hanzo, P. Cherriman, J. Streit, Video Compression and Communications, 2nd ed. John Wiley & Sons, Ltd, 2007. ISBN 978-0-470-51849-6

[3] ČSN EN ISO/IEC 11172: Informační technologie – kódování pohyblivých obrazů včetně doprovodného zvuku pro číslicový záznam do rychlostí 1,5 Mbit/s

Termín zadání: 7.2.2011

Termín odevzdání: 2.6.2011

Vedoucí práce: Ing. Jiří Schimmel, Ph.D.

Konzultanti bakalářské práce:

prof. Ing. Kamil Vrba, CSc.
předseda oborové rady



UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Abstrakt

Bakalářská práce se zabývá hierarchickým kódováním – škálováním – pohyblivých obrazů u v současné době velmi využívaných kompresních algoritmů MPEG-2 a MPEG-4. Škálování je proces, při němž je kódovaný datový tok rozdělen do odstupňovaných vrstev, přizpůsobených především možnostem přenosových médií a schopnostem koncových zařízení. Obsahem základní vrstvy je signál v omezené kvalitě. Obsahem jedné či více rozšiřujících vrstev je pak doplňující obrazová informace, rozšiřující signál na plnohodnotný tvar. Dekodér v závislosti na svých schopnostech a vnějších okolnostech může dekodovat pouze relevantní vrstvy z přijatého bitového toku.

Práce se v teoretické části zaměřuje na podrobný popis kódování a dekodování videosignálu dle standardů MPEG při použití tří typů škálování. Konkrétně se jedná o prostorové, kvalitativní SNR a Fine granular škálování.

V praktické části práce byla vytvořena výuková aplikace v prostředí Matlab, s vlastním grafickým uživatelským rozhraním, jenž v sobě implementuje funkce provádějící prostorové, kvalitativní SNR a Fine granular škálování. Na základě uživatelsky zadaných vstupních parametrů provádí škálování obrazu a následně zobrazí výstupy jednotlivých vrstev po provedení škálování a po rekonstrukci obrazu. Vstupem programu, na němž je hierarchické kódování prováděno, je obrázek v podobě bitové mapy libovolných rozměrů, ve formátu RGB.

Klíčová slova

Hierarchické kódování, škálování, kompresní algoritmus, predikce, vektor pohybu, diskretní kosinová transformace, DCT, kvantování, MPEG-2, MPEG-4, kodér, dekodér, SNR, Fine Granular, prostorové škálování.

Abstract

Bachelor's thesis deals with hierarchical coding – scaling – of moving pictures in currently widely used compression algorithms MPEG-2 and MPEG-4. Scaling is the process whereby the encoded data stream is divided into graduated layers, adapted especially to the possibilities of transmission media and capabilities of terminal devices. The content of the base layer is a signal with reduced quality. The content of one or more extension layers is a supplementary picture information, completing the signal to a full quality form. The decoder, depending on its abilities and external circumstances, can decode only the relevant layers from the received bitstream.

The thesis focuses in the theoretical part on the detailed description of encoding and decoding video signal according to MPEG standards using three types of scalability. In the concrete, the spatial, qualitative SNR and Fine granular scalability are mentioned.

In the practical part of thesis was created an educational application in Matlab environment with its own graphical user interface, which implements the functions performing spatial, qualitative SNR and Fine granular scalability. Based on the user-specified input parameters, it performs scaling the image and then displays the outputs of individual layers after scaling and after the image reconstruction. The input of the programme, in which a hierarchical coding is done, is the image in bitmap format with any size in RGB format.

Keywords

Hierarchical coding, scaling, compression algorithm, prediction, motion vector, discrete cosine transform, DCT, quantisation, MPEG-2, MPEG-4, encoder, decoder, SNR, Fine Granular, spatial scalability.

ŠPAČEK, M. *Hierarchické kódování pohyblivých obrazů*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011. 62 s. Vedoucí bakalářské práce Ing. Jiří Schimmel, Ph.D..

Prohlášení

Prohlašuji, že svoji bakalářskou práci na téma Hierarchické kódování pohyblivých obrazů jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne 27. května 2011

.....
podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Jiřímu Schimmelovi, Ph.D., za účinnou metodickou a odbornou pomoc a další cenné rady při zpracování této bakalářské práce.

V Brně dne 27. května 2011

.....
podpis autora

OBSAH

Úvod.....	8
1 Kódování videosignálu.....	9
1.1 Úvod do zpracování videosignálu.....	9
1.2 Videoformáty a datový tok.....	10
1.3 Ztrátové a bezztrátové kódování.....	12
2 Kompresní standard MPEG-2.....	16
2.1 Profily a úrovně.....	16
2.2 Predikce a vektory pohybu.....	18
2.3 Kódování a regulace rychlosti.....	18
2.4 Schéma a popis kodéru a dekodéru soustavy MPEG-2.....	20
2.4.1 Princip kodéru.....	20
2.4.2 Princip dekodéru.....	22
3 Kompresní standard MPEG-4.....	23
3.1 Vlastnosti a obrazové profily.....	23
3.2 Snímkové a objektově orientované kódování.....	25
3.3 Schéma a popis kodéru a dekodéru soustavy MPEG-4.....	27
3.3.1 Princip kodéru.....	27
3.3.2 Princip dekodéru.....	28
4 Hierarchické kódování dle standardů MPEG.....	29
4.1 Prostorové škálování (Spatial scalability).....	30
4.1.1 Schéma a funkce kodéru.....	30
4.1.2 Schéma a funkce dekodéru.....	31
4.2 Kvalitativní škálování (SNR scalability).....	32
4.2.1 Schéma a funkce kodéru.....	32
4.2.2 Schéma a funkce dekodéru.....	33
4.3 Fine granular škálování (FGS).....	33
4.3.1 Schéma a funkce kodéru.....	34
4.3.2 Schéma a funkce dekodéru.....	34
4.4 Další typy škálování.....	35
4.4.1 Časové škálování (Temporal scalability).....	35
4.4.2 Komplexní škálování (Complexity scalability).....	35
4.4.3 Objektové škálování (Object-based scalability).....	36
4.4.4 Rozdělování dat (Data partitioning).....	36
4.4.5 Škálování kombinující více druhů (Hybrid scalability).....	36
5 Realizace škálování v prostředí Matlab.....	38
5.1 Uživatelské rozhraní aplikace.....	38
5.2 Realizace funkce prostorového škálování.....	42
5.3 Realizace funkce kvalitativního škálování.....	44
5.4 Realizace funkce škálování Fine granular.....	47
5.5 Realizace funkce pro výpis hladin ve FGS.....	49
6 Stručný popis obsluhy výukové aplikace.....	56
Závěr.....	58
Použitá literatura.....	60
Seznam zkratk.....	61
Seznam příloh.....	62

ÚVOD

V několika posledních letech došlo k dramatickému rozvoji využívání informačních a komunikačních technologií. Ať už se jedná o stále rostoucí počty prodaných zařízení, či zvyšující se dostupnost datového spojení pro mobilní přístroje. Přenos obrazové informace již dávno není doménou pouze televizního vysílání, ale dostal se do mnoha oblastí lidské činnosti. Účelem jeho šíření není jen poskytnutí multimediální zábavy, či předání seriózní informace, ale i obchodní či osobní prezentace jednotlivců a společností. I v oblasti telekomunikací je patrná snaha rozšířit hlasovou komunikaci o obrazový obsah a dodat tak nabízené službě atraktivitu. Obchodní jednání v podnikatelské a firemní sféře, či komunikace mezi blízkými osobami, tak nabývá další, uživatelsky mnohem přitažlivější rozměr, což je i pro prodejce těchto služeb zajímavým faktorem.

Ruku v ruce s vývojem na poli multimediálního obsahu jde i vývoj hardware sloužícího pro vysílání a příjem takového obsahu. Schopnosti koncových zařízení neustále rostou a nabízejí stále vyšší a vyšší kvalitu a věrnost obrazu, to vše k radosti výrobců i spotřebitelů. Ač by se zdálo že vývoj v této oblasti není ničím omezen a je jen otázkou vývoje poskytnout uživatelům těchto služeb vše, co si lze vůbec představit, existuje přesto faktor, který má na rozvoj podstatný vliv. Tím je ekonomické hledisko. Nutí nás posuzovat a stále zvyšovat efektivitu ve využívání dostupných zdrojů, a to proto, aby finanční náročnost na pořízení a provoz zařízení nebyla pro rozvoj v oblasti multimédií kontraproduktivní.

V tomto okamžiku se dostává ke slovu kódování snižující nadbytečnost (redundanci) a zbytečnost (irelevanci) ve vyjádření obrazové informace. Pomocí různých technik tak lze použitím ztrátové i bezztrátové komprese redukovat informace lidským zrakovým systémem (HVS) nepostřehnutelné, zvýšit efektivitu v bitovém vyjádření signálu a provádět další činnosti, s cílem zachování dobrého subjektivního dojmu z obrazu, při podstatně efektivnějším využití přenosových prostředků.

Z důvodu jednotnosti mezi výrobcí a též jednoznačné definice způsobů kódování a dekódování, vzniklo postupem času několik standardů. Mezi nejvýznamnější z nich patří bezpochyby standardy MPEG-2 a MPEG-4, jimiž se zabývá tato práce. Tyto standardy jednoznačně definují bitový tok, který má být kóděrem produkován a následně v dekodéru zpracován. Velkým pomocníkem je jim v tom specifikace profilů a úrovní, jenž dokáže pomocí odstupňování definovat, které konkrétní funkce a s jakými omezeními budou v zařízení či aplikaci implementovány. Vzhledem k širokému spektru různorodých přijímačů, a tedy i dekodérů, by však v praxi bylo obtížné pro každou specifickou konfiguraci vytvářet kóděrem samostatný datový tok. V tu chvíli přichází na pomoc hierarchické kódování, umožňující kóděru vytvořit jediný multiplexovaný datový tok, odstupňovaný do vrstev, přizpůsobený odlišným typům cílových zařízení. Dekodér pak nemusí disponovat vysokým výpočetním výkonem, ani nemusí implementovat veškeré funkce standardu, neboť si může zvolit pro něj optimální vrstvu, která odpovídá jeho reálným schopnostem.

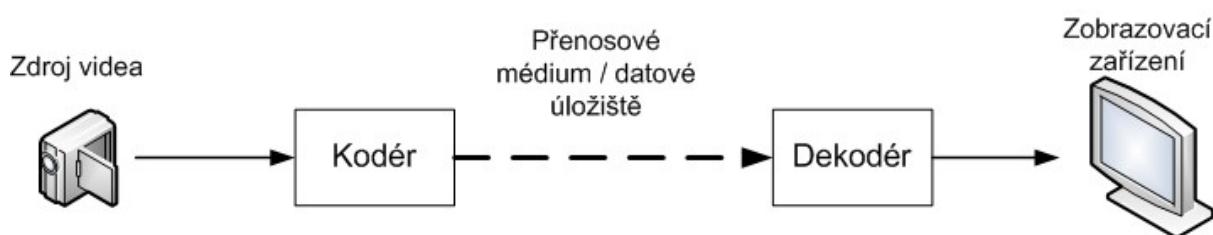
Cílem této bakalářské práce je seznámit čtenáře s metodami škálování u ztrátové komprese videosignálu podle standardů MPEG-2 a MPEG-4. Dále také v prostředí Matlab vytvořit výukovou aplikaci s vlastním grafickým uživatelským rozhraním, která dle zadaných parametrů provádí prostorové, kvalitativní (SNR) a Fine granular škálování a zobrazí výstupy jednotlivých vrstev po provedení škálování a po rekonstrukci obrazu. Vstupem programu bude obrázek v podobě bitové mapy formátu RGB.

V první kapitole se zabývám definicí základních pojmů kódování a zpracování videosignálu. V druhé a třetí kapitole jsou popsány vlastnosti a specifiky kompresních standardů MPEG-2 a MPEG-4. Čtvrtá kapitola je věnována popisu způsobů hierarchického kódování dle standardů MPEG. Pátá a šestá kapitola popisuje realizaci výukového programu, společně s popisem jeho uživatelského rozhraní, funkčnosti a výstupů z něj získaných.

1 KÓDOVÁNÍ VIDEOSIGNÁLU

Předtím, než můžeme sledovat obraz na displejích a obrazovkách televizí, počítačů, či přenosných zařízení, je třeba aby videosignál urazil dlouhou cestu a byl zpracován a upraven mnoha obvody. Tyto procesy jsou zpravidla standardizovány mnoha normami, doporučeními i nařízeními, v tomto dokumentu budou zmíněny zejména [3], [4], [5]. Nermalou roli v nich hraje i zpětná kompatibilita se staršími systémy. Obecné blokové schéma, popsané např. v [1], znázorňující použití kodéru a dekodéru v praxi, uvádí Obr. 1.1.

Co si však představit pod pojmem kódování? Norma [3] uvádí, že kódování je: „proces, který čte tok vstupních obrazů nebo audio vzorků a vytváří platný kódovaný bitový tok“. Jak je tedy z této obecné definice zřejmé, problematika kódování je velmi obsáhlá a vyžaduje detailnější prozkoumání. Podrobněji se proto budu jednotlivým fázím kódování videosignálu věnovat v následujících kapitolách.



Obr. 1.1: Obecné blokové schéma využití kódování [1]

1.1 ÚVOD DO ZPRACOVÁNÍ VIDEOSIGNÁLU

Samotnému kódování videosignálu, jenž je ústředním tématem této práce však předchází několik nezbytných kroků, díky nimž získáme videosignál v takové číslicové podobě, jenž je použitelná pro další zpracování. Uvažuji tedy činnost, počínající samotným snímáním zdroje, při němž je zaznamenávána přirozená obrazová scéna, jenž je prostorově a časově spojitá, do číslicové podoby reprezentované vzorky v prostorovém (obvykle tvořeném body v obdélníkové mřížce) a časovém vyjádření (jako posloupnost jednotlivých obrazových snímků). Každý takový vzorek si lze představit jako číselnou informaci, popisující jas a barvu konkrétního obrazového elementu, pixelu.

Běžně je k pořízení takového záznamu použito kamery, která pro zachycení barevného obrazu využívá jediného CCD snímače (Charge-Coupled Device), jemuž je předřazen filtr propouštějící střídavě jednotlivé RGB složky. Lepší kvality záznamu se však dosáhne při použití 3 CCD snímačů za současného rozkladu světla pomocí filtrů, či dichroických hranolů na samostatné RGB složky. Ty po rozdělení na trojici obrazů jednotlivých barvonosných složek dopadají na tři k tomu určené pole snímacích prvků CCD, v nichž dochází ke vzniku náboje, odpovídajícího dopadajícímu světlu. Následně je obrazová informace, reprezentovaná nábojem těchto snímačů, uložena do paměti, odkud dochází k postupnému řádkovému čtení. Existují ovšem i čtyřdílné televizní kamery, které kromě signálů R, G, B, zpracovávají odděleně i jasový signál, čímž se ještě více podobají lidskému oku [14].

Lidské oko je pro vidění vybaveno dvěma druhy buněk, pracujících jako obrazové snímače. Jedná se o tyčinky (citlivé převážně na jas) a čípky (citlivé na jas i barvu). Tyčinek je v oku podstatně více než čípků (asi 120 milionů tyčinek proti 6 milionům čípků) a současně jsou na světlo asi 200 000 krát citlivější než čípky. To je například i důvodem, proč za šera sice vidíme, ale pouze černobíle. Těchto znalosti se využívá i při záznamu obrazu, kdy jsou redukovány chrominanční složky bez výraznějšího dopadu na sledovaný obraz. Vlastnosti a

specifika lidského zrakového systému HVS (Human Visual System) je nutné zohlednit při následném zpracování a přizpůsobování obrazového signálu proto, aby výsledný dojem působil co nejvíce věrohodně. Experimentálně byly dle [9] zjištěny tyto charakteristiky zdravého lidského oka: prostorová rozlišovací schopnost – 0,1 mm ze vzdálenosti 25 cm; rozlišení šedi (odděleně) – 40 úrovní; rozlišení šedi (porovnáním vedle sebe) – 100 úrovní.

Výstupní spojitý analogový signál snímacího zařízení je třeba převést do číslicové podoby, vhodné k dalšímu zpracování. To bývá realizováno pomocí pulzně kódové modulace PCM, jenž se skládá ze tří základních kroků:

- Vzorkování – vzniká diskretní impulsový signál s proměnnou amplitudou. Vzorkování probíhá s opakovací frekvencí splňující Nyquistovu podmínku.
- Kvantování – původní amplitudě se přiřadí velikost úrovně nejbližší kvantizační hladiny. Rozdíl mezi původní amplitudou a kvantizační hladinou je příčinou zanesení kvantizačního šumu do signálu, projevující se drobným zrněním.
- Kódování – přiřazuje jednotlivým nakvantovaným úrovním signálu binární číslo, odpovídající jejich hodnotám. U běžně používaných barevných modelů se používá rozlišovací schopnost 8 bitů na pixel.

Po provedení výše uvedených operací je získán barevný model, jenž matematicky reprezentuje určitou množinu barev. Jestliže byla zpracována pouze jasová složka, pak je určen černobílý obraz. Při zpracování všech tří barev je získán barevný model RGB, z něž je pak možné vycházet při převodu do jiného z barevných modelů. Model RGB se s výhodou používá u bezeztrátového záznamu nepohyblivých obrazů, pro zpracování videa však není vhodný.

Odlišné barevné modely se dle [6] využívají z několika důvodů. Prvním z nich je požadavek slučitelného přenosu se systémy černobílého televizního vysílání, využívajícího pouze jasovou složku. Druhým důvodem je požadavek na hospodárné využívání přenosového pásma, kdy by z důvodu vysoké citlivosti lidského oka na jas bylo zapotřebí mít všechny tři složky v plném rozlišení, čímž by se šlo proti myšlence redukce lidským okem nepostřehnutelné informace v signálu. Mezi základní barevné modely používané pro záznam obrazu patří:

- RGB – aditivní barevný model složený ze všech tří barevných složek v plném prostorovém rozlišení. Při smíchání všech tří se získá bílé světlo.
- YCbCr – Barevný model složený z jasové složky a dvou chrominančních. Chybějící složka Cg je získána výpočtem ze známých složek. Chrominanční složky jsou zpravidla přenášeny s nižším rozlišením než jasová složka Y. Hodnoty složek jsou vyjádřitelné v 8 bitovém tvaru.
- YUV – model používaný v televizním vysílání, konvertující hodnoty YCbCr na tvar, v němž jasová složka je v intervalu $\langle 0;1 \rangle$ a barevné v intervalu $\langle -0,5;0,5 \rangle$.

1.2 VIDEOFORMÁTY A DATOVÝ TOK

Jestliže je již znám způsob vyjádření obrazové informace pomocí barevného modelu, je třeba se dále zamýšlet nad otázkou prostorového a časového formátu a jejich typického využití. 2D prostorový obraz je určen horizontálním a vertikálním rozlišením, udávajícím počet bodů v řádcích a sloupcích jednotlivých snímků. Některé z nejčastěji používaných formátů jsou uvedeny v Tab. 1.1. Jedná se zejména o sadu formátů CIF (definovanou CCITT pro standard H.261), formáty SIF (používané v MPEG) a formáty určené pro televizní

vysílání. Rovněž je zde uvedena informace, zda je formát určen pro neprokládané p (progressive) či prokládané i (interlaced) řádkování, více viz [2].

Časový sled jednotlivých obrazů je určen snímkovým kmitočtem, udávajícím počet zobrazených snímků za sekundu. Pro televizní vysílání PAL (Phase-Alternation Line) a SECAM (Séquentiel Couleur avec Mémoire) to činí 25 snímků za sekundu. Pro formát NTSC (National Television Systems Committee) 29,97 snímku za sekundu. Z důvodu zajištění plynulosti obrazu z pohledu diváka a zároveň pro úsporné využití dostupné šířky pásma ve slučitelném režimu se televizní obraz většinou vysílá v prokládaném módu o dvojnásobném snímkovém kmitočtu [6].

Tab. 1.1: Přehled základních obrazových formátů [2],[8]

Standard	Rozlišení		Řádkování	Datový tok při 25 snímcích/s [byty]	Použití
	vertikální	horizontální			
Sub-QCIF	128	96	p	921 600	Přenosné osobní videopřehrávače, videotelefony,
QCIF	176	144	p	1 900 800	videokonference přes veřejnou telefonní síť
CIF	352	288	p	7 603 200	Páskové videorekordéry
4CIF	704	576	p	30 412 800	CCITT H.261
16CIF	1408	1152	p	121 651 200	
QSIF	180	120	p	1 620 000	MPEG
SIF	360	240	p	6 480 000	MPEG
CCIR 601 PAL	720	576	i,p	31 104 000	Standard i Enhanced Definition TV (SDTV, EDTV)
HDTV 1280	1280	720	p	69 120 000	High Definition TV (HDTV)
HDTV 1440	1440	1080	i	116 640 000	
HDTV (full HD)	1920	1080	i	155 520 000	

Z výše uvedeného vyplývá, že nekomprimovaný datový tok je určen součinem počtu bitů na jeden obrazový bod, počtu obrazových prvků ve snímku a snímkovou frekvencí, viz rovnice (1.1). Potřebnou přenosovou rychlost nekomprimovaného videa (například v modelu RGB) při rozlišení 8 bitů na pixel tak lze podle [8] určit pomocí vztahu:

$$R = p \cdot N_{op} \cdot f_s \quad [\text{bit/s}] \quad (1.1)$$

, kde:

- R – přenosová rychlost [b/s]
- p – počet bitů na jeden obrazový bod
- N_{op} – počet obrazových prvků ve snímku
- f_s – snímková frekvence [snímků/s]

Pro příklad, při nekomprimovaném vysílání HD videa v modelu RGB v kvalitě 720p (rozlišení obrazu 1280x720 pixelů) bude celková velikost R rovna:

$$R = p \cdot N_{op} \cdot f_s = 8 \cdot 3 \cdot 1280 \cdot 720 \cdot 25 = 552,96 \text{ Mbit/s} = 69,12 \text{ MB/s}$$

Jak je z výše uvedeného výpočtu i z hodnot uvedených v Tab. 1.1 zřejmé, potřebná přenosová rychlost by při barevném rozlišení 8 bitů na pixel byla v nekomprimovaném módu pro přenos neúměrně vysoká a vyžadovala by velkou šířku pásma, což by bylo neefektivní a neekonomické, a proto přichází ke slovu úpravy snižující redundanci a irelevanci obrazové informace, kterým je věnována následující kapitola.

1.3 ZTRÁTOVÉ A BEZEZTRÁTOVÉ KÓDOVÁNÍ

Z důvodu potřeby snížení datového toku, a to ať už z důvodů ekonomických, či technických, jsme u téměř všech typů videosignálu postaveni před problémem, jakým způsobem zakódovat zdrojová data, abychom dosáhli co nejvyššího zhuštění pro účely přenosu či záznamu. Při tom je však nutné současně zvažovat dopad takové činnosti na výslednou kvalitu obrazu a na subjektivní dojem pozorovatele. Takovéto kódování nazýváme „komprese“ a dle [3] je definováno jako: „redukce počtu bitů užitých k reprezentaci datové položky“.

Komprese jako taková může být buď bezeztrátová, tj. taková jenž odstraňuje nadbytečnost (redundanci) a dokáže tutéž informaci vyjádřit efektivněji pomocí menšího počtu bitů. Z toho plyne, že data rekonstruovaná dekodérem jsou přesnou kopií dat vstupujících do kodéru. Ačkoliv by nám takový stav vyhovoval, největší slabinou této komprese je její nízká účinnost. Podle [1] se například u formátu JPEG pohybuje v poměru 1:3 až 1:4, což by u videosignálu nebylo dostatečné. V takovém případě je třeba použít ztrátové komprese, která dokáže z kódovaného signálu odstranit irelevantní, lidským zrakem nepostřehnutelné informace. Ztrátová komprese je založena na principu odstranění subjektivní redundance, která může být odebrána bez podstatného vlivu na výsledný vizuální dojem pozorovatele. V případě příliš vysoké komprese však může být snížení kvality znatelné a může se jevit jako rušivý element subjektivního dojmu. Hledáním způsobů, jak při co nejnižším bitovém toku, za přispění kompresních algoritmů i znalosti HVS, poskytnout pozorovateli dobrý výsledný dojem, je úkolem pracovních skupin expertů vyvíjejících kompresní standardy [12], [15].

Z povahy videosignálu lze u kompresních kódovacích metod uvažovat zejména o odstranění redundance prostorové a časové. V časové oblasti se setkáváme především s velkou podobností mezi snímky po sobě časově následujícími a to tím více, čím vyšší je počet snímků za časový interval. U prostorového vyhodnocení pak lze pozorovat výskyt stejných objektů (bloků) ve skupině snímků, avšak prostorově posunutých. Všechny výše uvedené skutečnosti se využívá při kódování videosignálu a to především pomocí následujících operací a technik detailně popsanych v [6]:

- Podvzorkování barvonosných složek – na základě znalostí o lidském zrakovém systému HVS, z něž vyplývá že lidské oko je mnohem více citlivé na jas, než na barvy, se využívá metod vzorkování, při němž je zachována původní velikost jasové Y matice, avšak jsou podvzorkovány matice chrominančních složek. Několik nejčastěji používaných způsobů je uvedeno v Tab. 1.2.

Tab. 1.2: Způsoby podvzorkování barvonosných složek [6]

Typ vzorkování	Vzorkovací kmitočet složek			Popis
	Y	Cb	Cr	
4:4:4	13,5 MHz	13,5 MHz	13,5 MHz	Všechny tři matice mají plné vertikální i horizontální rozlišení
4:2:2	13,5 MHz	6,75 MHz	6,75 MHz	Cb, Cr matice mají plné vertikální rozlišení, ale jen poloviční horizontální rozlišení
4:2:0	13,5 MHz	6,75 MHz	6,75 MHz	Cb, Cr matice mají poloviční vertikální i horizontální rozlišení
4:1:1	13,5 MHz	3,375 MHz	3,375 MHz	Cb, Cr matice mají plné vertikální rozlišení, ale jen čtvrtinové horizontální rozlišení

V případě použití chrominančního podvzorkování dochází k výraznému snížení potřebného datového toku, v závislosti na množství vynechaných chrominančních vzorků. Například pro vzorkování 4:2:2 (poloviční horizontální rozlišení C_b a C_r) se u rozlišení obrazu 1280x720 pixelů jedná o následující hodnoty:

Jasová složka:

$$R = p \cdot N_{op} \cdot f_s = 8 \cdot 1280 \cdot 720 \cdot 25 = 184,32 \text{ Mbit/s} = 23,04 \text{ MB/s}$$

Chrominanční složky:

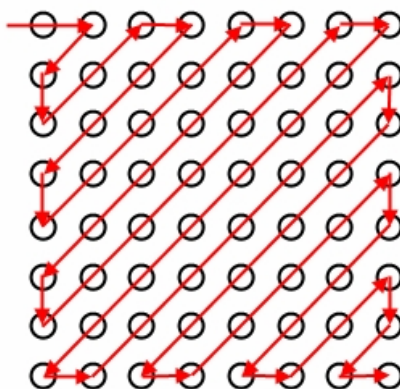
$$R = 2 \cdot p \cdot N_{op} \cdot f_s = 2 \cdot 8 \cdot 640 \cdot 720 \cdot 25 = 184,32 \text{ Mbit/s} = 23,04 \text{ MB/s}$$

Přenosová rychlost videa při vzorkování 4:2:2 v modelu YC_bC_r bude tedy činit:

$$R = 184,32 + 184,32 = 368,64 \text{ Mbit/s} = 46,08 \text{ MB/s}$$

Došlo tak k nemalému poklesu přenosové rychlosti, aniž by výrazněji utrpěl dojem pozorovatele z videa.

- Kvantování – činnost při které je určitému rozsahu hodnot přidělena jedna hodnota, reprezentovaná kvantizační hladinou. Jedná se o operaci, při níž dochází ke ztrátě informace v závislosti na kvantizačním kroku. Účelem je snížení potřebného počtu bitů pro interpretaci informace. Dalším základním pojmem této operace je „kvantizační šum“ vyjadřující rozdíl mezi původní a nakvantovanou hodnotou. Jeho velikost může dosahovat poloviny kvantizačního kroku. Čím větší je vzdálenost kvantizačních hladin, tím účinnější je komprese, avšak současně roste i kvantizační šum. Kvantování se používá ve spojitosti s diskretní kosinovou transformací (DCT) k redukci nepodstatných (vysokofrekvenčních) a k nule se blížících hodnot. V tomto případě mohou být všechny DCT koeficienty kvantovány odlišně, za použití pro daný účel optimalizované kvantovací tabulky. Zpravidla jsou koeficienty nižších frekvencí kvantovány s vyšší přesností, než koeficienty vyšších frekvencí. I zde se vychází z poznatků o HVS.
- Diskretní kosinová transformace (DCT) – operace, při níž je blok 2D prostorových vzorků transformován na matici frekvenčních koeficientů [4]. Současně dochází k přeuspořádání hodnot v matici, kdy koeficient s největší energií se nachází na souřadnicích [0;0], tedy v levém horním rohu. Hodnota v tomto bodě je označována jako *DC* koeficient, neboť jeho kmitočet v horizontálním i vertikálním směru je nulový a jeho hodnota představuje průměrnou hodnotu všech vzorků v bloku. Dále jsou hodnoty *AC* koeficientů v pořadí od nejnižších kmitočtů zapsány způsobem cik-cak a to směrem z levého horního rohu do pravého dolního rohu, viz Obr. 1.2.



Obr. 1.2 Princip čtení koeficientů metodou cik-cak [1], [6]

Diskrétní kosinová transformace je obdobou diskrétní Fourierovy transformace (DFT) a její definice je pro 2-D matici o rozměrech $N \times N$ následující [4]:

$$F(u, v) = \frac{2}{N} C(u) C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos \frac{(2x+1)u\pi}{2N} \cos \frac{(2y+1)v\pi}{2N} \quad (1.2)$$

kde:

$u, v, x, y = 0, 1, 2, \dots, N-1$

x, y jsou prostorové souřadnice ve zdrojové matici

u, v jsou souřadnice v transformované matici

$$C(u), C(v) = \begin{cases} \frac{1}{\sqrt{2}} & \text{pro } u, v = 0 \\ 1 & \text{ostatní} \end{cases} \quad (1.3)$$

Inverzní diskrétní kosinová transformace (IDCT) je definována jako:

$$f(x, y) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u) C(v) F(u, v) \cos \frac{(2x+1)u\pi}{2N} \cos \frac{(2y+1)v\pi}{2N} \quad (1.4)$$

Vstupní informace zdrojové matice pro DCT a výstup z IDCT je reprezentován 9 bity. Koeficienty jsou reprezentovány 12 bity. Dynamický rozsah DCT koeficientů je v rozsahu $[-2048; +2047]$. [4]

- Prediktivní kódování – vylepšuje účinnost komprese na základě předpokladu, že hodnota DC koeficientů sousedících bloků bývá velmi podobná a rovněž sousedící vektory pohybu jsou stejné či podobné. Následně tak dochází pouze ke kódování rozdílu mezi hodnotou předpokládanou a hodnotou skutečně nastalou. Kodér i dekodér totiž dokážou předpovídat – predikovat hodnotu následujícího vzorku na základě předchozího. Rozdíl mezi těmito hodnotami se nazývá chyba predikce a v dekodéru je použita pro korekci. Výhodou je, že chyba predikce bývá často nulová, či blíží se k nule, čímž dochází k mnohem efektivnější kompresi, či dokonce k vynulování rozdílových koeficientů v průběhu kvantování.
- Kompenzace pohybu – využívá se ke zlepšení procesu předpovědi hodnoty bloku pixelů ve snímku z hodnot předchozího snímku za použití vektorů pohybu. Dle [3] je definována jako: „Užití vektorů pohybu k zlepšení účinnosti předpovědi hodnot pixelů; předpověď používá vektorů pohybu k určení posunu do minulých a/nebo budoucích referenčních obrazů obsahujících dříve dekódované hodnoty pixelů, které se užívají k vytvoření předpovědi chybového signálu“. Vektory pohybu tedy určují posun bloku mezi souřadnicemi v aktuálním a předchozím či následujícím referenčním snímku. Výhoda spočívá ve faktu, že v sekvenci rychle po sobě následujících snímků zůstává mnoho objektů – bloků – v klidu na stejné pozici, zatímco menšina ostatních se pohybuje pouze v malém rozsahu. Díky tomu je pomocí DCT zakódován pouze rozdíl obou bloků (predikovaného a skutečného), které jsou takřka shodné a díky tomu je i většina kmitočtových koeficientů rovna nule. Vyhledávání vektorů pohybu bývá lineární, či logaritmické, v prostorově

omezené oblasti – z důvodu výpočetní náročnosti. Jestliže však nedojde k nalezení shodného bloku, musí se zakódovat celý. Pro zajištění lepší predikce, a tím současně menší difference, se využívá subpixelové kompenzace pohybu, kdy lze pomocí interpolace určit shodu až s půlpixelovou, či čtvrtpixelovou přesností.

- Kódování s proměnnou délkou slova – jedná se bezztrátové entropické kódování VLC (Variable-length Coding) sloužící ke snížení redundance. Entropie nám udává nejmenší možný počet bitů pro vyjádření hodnoty jednoho vzorku v závislosti na tom, s jakou pravděpodobností se tato hodnota vyskytuje. Délka slova pro vyjádření vzorku je pak proměnná a je pro často se vyskytující hodnoty krátká, zatímco pro zřídka se vyskytující hodnoty dlouhá. Typickými představiteli kódování tohoto typu jsou Huffmanovo a aritmetické kódování. Pro dosažení optimální účinnosti Huffmanova kódování by bylo třeba, aby pravděpodobnost výskytu koeficientů bloků (a s ní i tabulka Huffmanova kódu) byla určována průběžně během kódování, což je však v praxi nerealizovatelné ze dvou důvodů. Zaprvé by to bylo velmi výpočetně náročné, což by vnášelo neakceptovatelné zpoždění do kódovacího procesu. Zadruhé je zřejmé, že dekodér musí používat pro dekódování stejnou tabulku kódů jako kodér. To by vyžadovalo, aby kodér při každé změně zaslal současně s videosignálem i tabulku Huffmanova kódu, čímž by došlo k poklesu účinnosti komprese, a to zejména u kratších video sekvencí. Z toho důvodu současné kódovací standardy mají definovány tabulky kódových slov, založené na v praxi ověřené pravděpodobnosti výskytu hodnoty vzorků v obecném videosignálu, více viz [6].

2 KOMPRESNÍ STANDARD MPEG-2

Kompresní standard MPEG-2, viz [4], je určen ke kódování zvuku a pohyblivých obrazů. Byl vytvořen pracovní skupinou expertů, seskupených organizacemi ISO a IEC, založenou v roce 1988 pod názvem MPEG (Moving Picture Experts Group). Rodina standardů MPEG se vyznačuje ztrátovým kódováním a asymetričností. Asymetrie spočívá ve vyšší složitosti kódování, než dekódování. To je způsobeno výkonově náročnými technikami provádějícími ztrátovou a bezztrátovou kompresi datového toku.

Prvním uvedeným kompresním standardem byl MPEG-1 označený jako ISO/IEC IS 11172, zveřejněný v roce 1993 [3]. Jeho primárním zaměřením bylo nahrazení VHS, tedy ukládání audia a neprokládaného videa s bitovou rychlostí do 1,5 Mbit/s, cíleného především na disky CD-ROM a videotelefony. Ačkoliv jeho specifikace přestala po čase dostačovat, je tato soustava nadále v oblibě a to i díky široce známému zvukovému kompresnímu formátu MPEG-1 Audio Layer III (MP3). Mezi hlavní rysy standardu MPEG-1 patří: v oblasti audia omezení na dva kanály (stereo); chybějící podpora prokládaného videa, pouze jeden standardní profil CPB (Constrained Parameters Bitstream), problém s kódováním vyšších rozlišení, podpora pouze 4:2:0 barevného modelu.

V roce 1995 byl zveřejněn standard MPEG-2, označený jako ISO/IEC CD 13818 [4]. Tato soustava byla navržena pro distribuci televizního signálu různými prostředími, v různé kvalitě a s různými stupni ochrany proti chybám v přenosu a rušení. Standard je tvořený 11 kapitolami. Soustava MPEG-2 poskytuje velkou variabilitu ve schopnosti přizpůsobení bitového toku kóděrem pro široké spektrum koncových zařízení. Hlavními rozdíly, v nichž se MPEG-2 odlišuje od MPEG-1 je především zvýšení možné bitové rychlosti datového toku při účinnějších komprimačních metodách. Dále zavedl možnost prokládaného řádkování a s tím související úpravy v základních postupech při predikčním kódování, transformaci a kvantování a při entropickém kódování [6].

Velkým přínosem standardu MPEG-2 byla též možnost složení výstupního bitového toku z více částí. V jednom společném toku tak může být přenášen videosignál pro standardní přijímače s nízkou rozlišovací schopností i datový tok pro přijímače s vysokým rozlišením. Záleží pak na dekodéru přijímače jaký druh signálu zvolí a zpracuje. Tato variabilita souvisí s principy škálování – kódování videosignálu do několika odstupňovaných hladin. Standardem MPEG-2 podporované funkce i případná omezení jsou popsána pomocí profilů a úrovní, o nichž pojednává následující kapitola.

2.1 PROFILY A ÚROVNĚ

Ve standardu MPEG-2 je podporováno široké spektrum aplikací od jednoduchých mobilních zařízení až po přijímače HD. Pro mnohá použití by bylo neekonomické i technicky obtížné podporovat celou šířku standardu. K umožnění podpory pouze části z nich definuje standard MPEG-2 jednotlivé profily a úrovně.

Profily specifikují sadu vlastností, jako je kompresní algoritmus, typ podvzorkování chrominančních složek a další algoritmy a funkce. Těchto profilů je pět a v Tab. 2.2 jsou rovněž uvedeny jejich základní rysy, čerpané z [6]. Je patrné, že odlišnost je výrazná zejména v oblasti podpory hierarchického kódování videosignálu. Jsou podporovány zejména dva druhy škálování, a to prostorové a kvalitativní. Další odlišnost je v otázce použití predikovaného B snímku a ve způsobu podvzorkování chrominančních složek.

Úrovně definují omezení kvantitativních parametrů pro daný profil. Jsou určena především maxima pro bitový tok, velikost snímku a snímkový kmitočet uvedená v Tab. 2.1. Více viz [16].

Tab. 2.1: Přehled úrovní a jejich omezení ve standardu MPEG-2 [4], [6]

Název úrovně	Snímkový kmitočet (Hz)	Max. horizontální rozlišení	Max. vertikální rozlišení	Maximum jasových vzorků za sekundu (rozlišení x snímkový kmitočet)	Max. bitový tok v hlavním profilu (Mbit/s)
Nízká	23,976; 24; 25; 29,97; 30	352	288	3 041 280	4
Hlavní	23,976; 24; 25; 29,97; 30	720	576	10 368 000 (14 475 600 / 11 059 200)	15
Vysoká 1440	23,976; 24; 25; 29,97; 30; 50; 59,94; 60	1440	1152	47 001 600 (62 668 800)	60
Vysoká	23,976; 24; 25; 29,97; 30; 50; 59,94; 60	1920	1152	62 668 800 (83 558 400)	80

V praxi se obvykle oba parametry uvádějí současně, zapsané zkratkami ve tvaru „profil@úroveň“, čímž definují maximální možný profil a úroveň MPEG-2 datového toku, kterou může dané zařízení zpracovat. Přehled profilů a úrovní se často uvádí ve společné přehledné tabulce zobrazené rovněž i v [6], viz Tab. 2.2, z níž jsou i lépe patrné možné kombinace. Je třeba uvést, že tato tabulka současně znázorňuje i zpětnou kompatibilitu – slučitelnost, kdy dekodér pro určitý profil a úroveň musí umět dekódovat nižší profily a úrovně a rovněž tak i standard MPEG-1.

Tab. 2.2: Přehled profilů a úrovní standardu MPEG-2 [6]

Profil Úroveň	Jednoduchý	Hlavní	Odstupňovaný	Prostorově odstupňovaný	Vysoký
	Bez predikce B, formát 4:2:0	Bez odstupňování, formát 4:2:0	Jako hlavní profil + odstupňování SNR	Jako SNR profil + prostorové odstupňování	Jako prostorový + formát 4:2:2
Nízká		352 x 288 bodů, 4Mbity/s	352 x 288 bodů, 4(3)Mbity/s		
Hlavní	720 x 576 bodů, 15Mbitů/s	720 x 576 bodů, 15Mbitů/s	720 x 576 bodů, 15(10)Mbitů/s		720 x 576 (352 x 288) bodů, 20/15/4Mbit/s
Vysoká 1440		1440 x 1152 bodů, 60Mbitů/s		1440 x 1152 (720 x 576) bodů, 60/40/15Mbit/s	1440 x 1152 (720 x 576) bodů, 80/60/20Mbit/s
Vysoká		1920 x 1152 bodů, 80Mbitů/s			1920 x 1152 (960 x 576) bodů, 100/80/25Mbit/s

2.2 PREDIKCE A VEKTORY POHYBU

Velmi silným nástrojem standardu MPEG-2, snižující výrazně potřebu datového toku je predikce za použití vektorů pohybu. Predikci je možné použít nejen pro neprokládané řádkování, ale i při prokládaném řádkování, ač tento režim predikci značně komplikuje. Zejména se to týká rozšíření počtu a způsobů volby referenčních makrobloků a určení vektorů pohybu. Základní principy predikce a vektorů pohybu byly popsány v kapitole 1.3 – Ztrátové a bezztrátové kódování, proto zde zmíním pouze odlišnosti pro určování predikce v celosnímkovém a půlsnímkovém módu. Je třeba uvést, že mezi oběma módy přepíná kódér pouze po celých snímcích, nikoliv v průběhu. Detailně je tato technika popsána v [6].

U celosnímkového módu se s dvojicí půlsnímků náležejících témuž typu snímku pracuje jako s jedním snímkem I,P, nebo B. Makroblok vzorků je tedy předpovídan podobným způsobem jako u MPEG-1. Pro predikci typu P je použit pouze jeden vektor, pro predikci typu B dva vektory. Tento postup zvyšuje účinnost komprese, avšak je vhodný pouze pro statické obrazy, neboť makrobloky jsou složeny z navzájem časově posunutých sudých a lichých řádků. U rychlých scén s vodorovným posuvem tak vzniká chyba, kdy z důvodu vodorovného pohybu dochází k roztřepení svislých hran. Proto je vhodnější použít predikci na základě půlsnímku. Toho lze dosáhnout rozdělením makrobloku do dvou částí o rozměrech 16 x 8 obrazových bodů. Poté je předpověď určena pro každou z obou částí rozděleného makrobloku pomocí samostatného vektoru, predikce se tedy vztahuje na operaci mezi vzorky půlsnímků. Určení pohybu je pak přesnější, ale nevýhodou je, že je zapotřebí dvojnásobného počtu vektorů. Informace o tom, zda byl referenční makroblok pro právě kódovaný makroblok současného půlsnímku zvolen z půlsnímku stejného nebo opačného druhu je uvedena v jednobitové informaci v záhlaví makrobloku (tzn. například předpověď lichého půlsnímku z předchozího lichého či sudého půlsnímku). Tato volba může být odlišná pro každý jednotlivý makroblok.

Vzhledem k tomu, že při půlsnímkové predikci dochází k časově náročné operaci zdvojnásobení počtu vektorů pohybu, což není optimální pro účinnost kódování, používá se v celosnímkovém i půlsnímkovém módu místo obousměrné predikce, predikce zvaná Dual Prime. Tento typ predikce je vhodný především pro obrazy s rovnoměrným pohybem v jejich obsahu. Při této predikci dochází ke zprůměrování hodnoty dvou predikcí pro současný makroblok v půlsnímku, tj. predikce z předchozího snímku stejného druhu a z půlsnímku opačného druhu. Predikce pro půlsnímek stejného druhu je dána vektorem pohybu. Pro predikci z druhého půlsnímku se však nepřenáší znovu vektor, ale pouze diferenční vektor a koeficient pro prodloužení či zkrácení původního vektoru. Dva zakódované půlsnímky příslušející jednomu snímku tedy vyžadují přenos dvou hlavních vektorů pro tentýž druh půlsnímků a dvou diferenčních vektorů pro opačný druh půlsnímků.

2.3 KÓDOVÁNÍ A REGULACE RYCHLOSTI

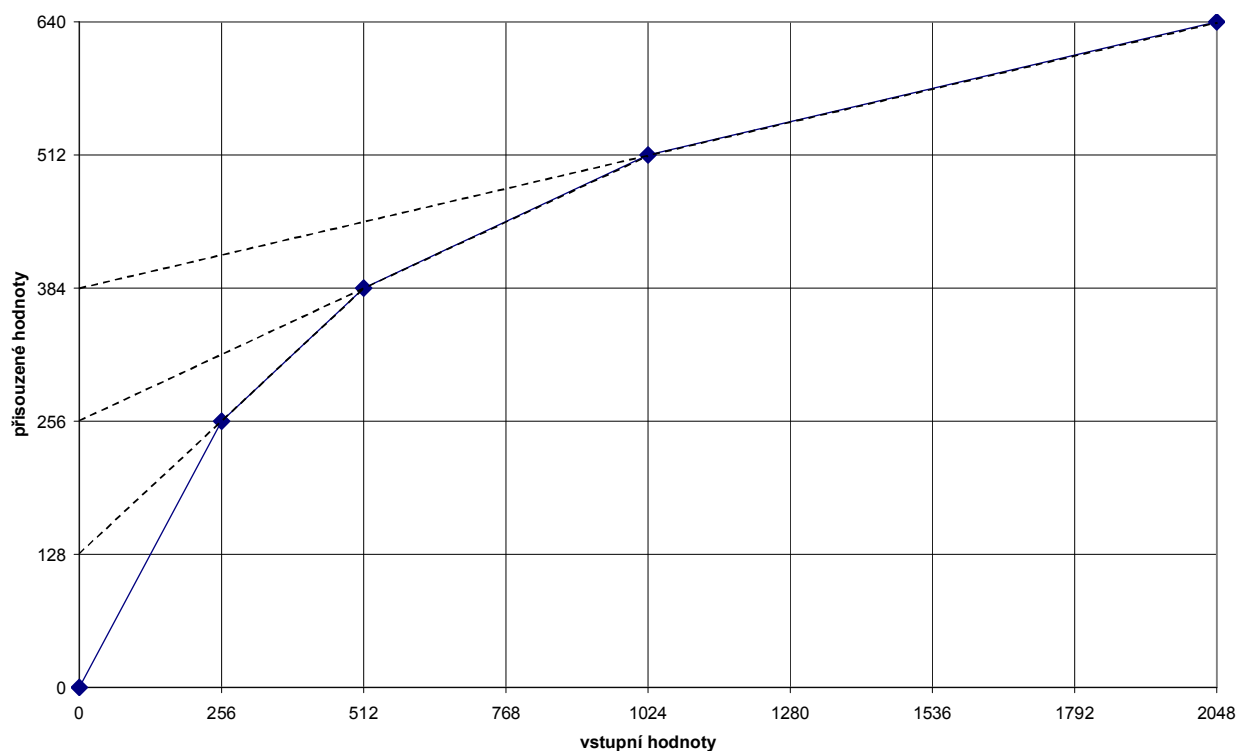
Obecné principy transformačního a entropického kódování byly již popsány v kapitole 1.3 – Ztrátové a bezztrátové kódování. Pro standard MPEG-2 však dochází k několika změnám, především s ohledem na účinnější metody kódování, zaručující vyšší účinnost komprese, při zachování stejné kvality videosignálu. Tak jak již bylo popsáno v předchozí kapitole 2.2, ovlivňuje prokládané řádkování nejen predikční kódování, ale i diskretní kosinovou transformaci a kvantování [6].

Jestliže dochází v obraze k pohybu, je výhodnější zpracovávat bloky s oddělenými sudými a lichými řádky, protože stejně jako při predikci by i zde při celosnímkovém zpracování (ze dvou časově posunutých půlsnímků) nepochybně docházelo k roztřepení

svislých rovných hran. Tím by se také zvyšoval počet transformačních koeficientů, směrem k vyšším frekvencím a rovněž tak by narostla potřeba datového toku. Při statických scénách s minimem pohybu je pak samozřejmě výhodnější a úspornější použít celosnímkový režim zpracování, při němž frekvenční spektrum koeficientů obsahuje jejich menší počet.

Kvantizační tabulky i tabulky pro kódování s proměnnou délkou slova jsou uzpůsobeny pro vyšší počet frekvenčních koeficientů ve svislém směru. Odlišný tak je i způsob čtení koeficientů pro signál s prokládaným řádkováním, než pro signál s neprokládaným řádkováním. U prokládaného řádkování se uplatňuje větší strmost snímání, čímž je současně ovlivněna i délka běhu, z čehož také plyne potřeba modifikace tabulek Huffmanova kódování.

Další odlišností je kvantování. U standardu MPEG-2 je nelineární, dalo by se však říci, po částech lineární. Je rozděleno do čtyř úseků, z nichž v každém z nich má jinou strmost, viz Obr. 2.1, dle [6]. V prvním úseku je vstupním hodnotám koeficientů do velikosti 256 přiřazena tatáž nezměněná hodnota. Při vyšších vstupních hodnotách postupně dochází ke kvantování s menší strmostí, až do posledního úseku, kdy je pro maximální vstupní hodnotu koeficientu 2048 je přiřazena výstupní hodnota 640. Pomocí tohoto nelineárního kvantování lze dosáhnout snížení počtu bitů, v tomto případě z dvanácti (-2048 až 2047) na deset bitů. Se změnou kvantizačních tabulek souvisí i možnost velmi efektivní regulace datového toku vystupujícího z kvantizéru. Zeslabení toku tak lze realizovat v poměru $1 : 2^{n/16}$, pro hodnoty $n = \langle 0;175 \rangle$.



Obr. 2.1: Nelineární kvantování frekvenčních koeficientů v soustavě MPEG-2 [6]

Jak lze tedy konstatovat, pomocí kvantovacích koeficientů (změnou kvantizačního kroku při kvantování koeficientů po DCT) a rovněž vhodnou volbou Huffmanových koeficientů je možné velmi výrazně ovlivnit výslednou bitovou rychlost datového toku.

bloku. Ostatní pozice bloku jsou obsazeny střídavými koeficienty, a to v řazení od nejnižších k nejvyšším frekvencím.

Dalším krokem je kvantování, určené podle kvantizačních tabulek, které mají rozměry shodné s bloky vzorků, tedy 8×8 . Po nakvantování (podělení hodnot vzorků) a zaokrouhlení dostaneme matici po kvantizaci. Nepodstatné složky jsou nakvantováním poděleny a zaokrouhleny často na nulu. Zůstává pouze DC složka a menší množství koeficientů nízkofrekvenčních AC složek. Tato nakvantovaná matice postupuje dále do kodéru a čte se cik-cak, z důvodu vzrůstajícího trendu frekvence. Hodnoty se uspořádají za sebe a proběhne kódování s proměnnou délkou slova. Každému údaji se přiřadí odpovídající Huffmanův kód. Huffmanovo kódování je bezeztrátové, probíhá podle přesně daných tabulek, dochází pouze k redukci bitů. Takto vzniklá bitová posloupnost je výstupem kodéru.

Zároveň s tímto proběhne zpracování ve zpětné větvi a obrázek se dekoduje. Referenční snímek slouží k odhadu dalšího snímku. V tomto případě typu P. Ten jde oběma cestami. Existuje mnoho metod jak směr odhadnout – určit rozdílový snímek. Prediktor pohybu se snaží algoritmem co nejlépe a co nejrychleji odhadnout pohyb objektu. Vezme každý blok a hledá ho v dané vyhledávací oblasti v okolí. Pokud bod najde, bloky se mezi sebou odečtou a získá se difference. Cílem algoritmu je co nejlepší odhad, aby chybová matice byla co nejmenší. Odhadnutý snímek přichází do sčítačky, je odečten od skutečného a kóduje se pouze rozdíl. Rozdíl nastává zejména u hran objektů v obraze.

Současné s odhadem pohybu jsou určeny i vektory pohybu, určující kam má být blok posunut. Tyto vektory jsou zakódovány společně s chybou predikce P či B snímku.

Předtím, než kodér začne pracovat, je nastaven režim – pořadí posloupnosti snímku I,P,B. Předem je tedy známo jaká posloupnost se bude kódovat. V případě náhlé výrazné změny však může dojít k narušení této posloupnosti, rozhodovač určí, že difference je příliš velká a snímek se zakóduje jako I (použije se jiná kvantizační tabulka pro intra/inter kódování) a pokračuje se dál dle určené posloupnosti. Tato změna typu kódování však snižuje účinnost komprese.

V kodéru se rovněž nachází obvod řízení kodéru, jenž je nutný kvůli nastavení a optimalizaci bitového toku. Při požadavku konstantní bitové rychlosti mění řízení kodéru kvantizační tabulky pro vyvážený bitový tok. To vše s ohledem na časovou proměnnost scény a optimální vytížení bufferu. Při větším nároku na vyšší bitový tok, u rychlých změn scény, se mění tabulky, kvantizace se stává hrubší a kvalita se zhoršuje. U variabilního bitového toku (například DVD), kde není řízení bitového toku tak striktní a může docházet k výraznějším výchyldkám, dosahuje maximální bitový tok až 9,8 Mbit/s, průměrný pak kolem 4-5 Mbit/s.

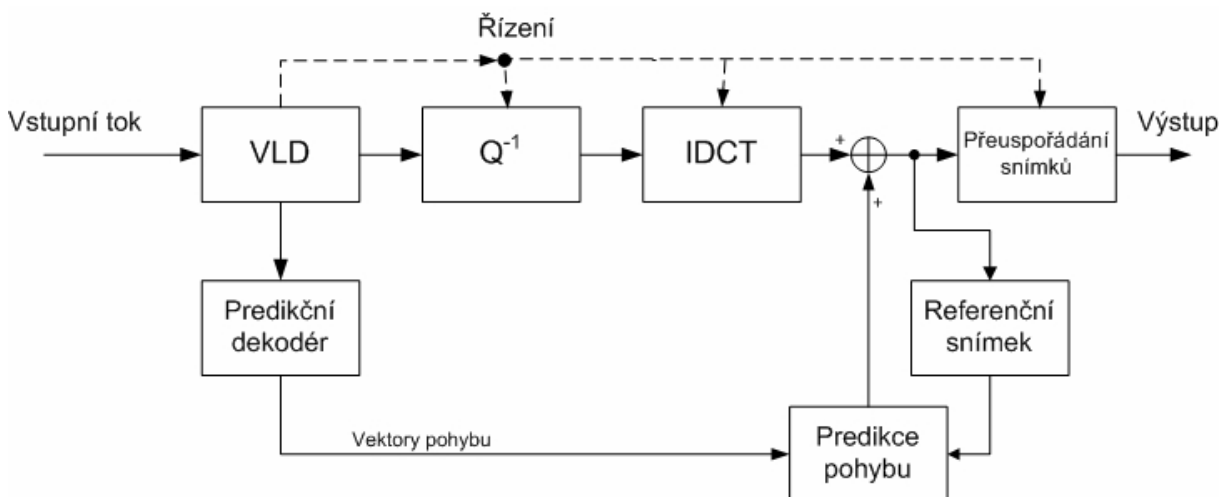
V kodéru se kóduje blok po bloku, každý zvlášť, z toho plyne jeho velký nedostatek, a to ten, že jsou viditelné hranice jednotlivých bloků. Pokud je tedy komprese vysoká (proběhne hrubé kvantování), vznikají blokové artefakty, tj. obraz je kostkovaný. K odstranění tohoto jevu mohou být na výstupu dekodéru použity postprocessingové filtry rozmazávající hranice bloků. Velikost bloku 8×8 obrazových bodů je určena z hlediska výpočetní náročnosti transformace DCT a to zejména kvůli nevýkonným zařízením (telefony, bezdrátové komunikátory, apod.).

Z předchozího popisu vyplývá, že ke ztrátě informace z videosignálu dochází již při vstupním podvzorkování (4:2:2, 4:2:0) a při kvantování, a to v důsledku zaokrouhlení. Kvantizace v podstatě jako jediná řídí kvalitu snímku videa. Kupříkladu u komprese JPEG může faktor nabývat hodnoty 1 – 100, čímž je dána velikost dělení. Pro hodnotu 100 je kvalita nejlepší, kvantovací tabulka je vyplněna samými jedničkami, tedy probíhá beze ztráty. Při faktoru 99 už však dochází ke ztrátě [8].

2.4.2 Princip dekodéru

V dekodéru, znázorněném na Obr. 2.3, detailněji popsáném v [6], dostáváme na vstupu matici, tj. snímek rozdělený po blocích. Nejdříve zpracován – jako první přichází by měl být snímek typu I. Proběhne inverzní kvantování (násobení koeficientů), poté následuje inverzní kosinová transformace IDCT. Tím se uskutečnila rekonstrukce snímku, který se optimálně jen velmi málo liší od kódovaného.

Jako další přichází je predikovaný snímek. Ze vstupu dekodéru se do predikčního dekodéru odeberou vektory pohybu. První snímek typu I byl v předchozím kroku dekódován a zapsán jako referenční, bez přispění vektorů. Jakmile přijde druhý snímek – typu P, ten má jen chybu predikce a vektory pohybu. Vektory se odeberou z toku a v prediktoru pohybu se použijí na bloky referenčního snímku a tím se snímek přeuspořádá. Takto se získá pozměněný, přeuspořádaný snímek, který se sečte s chybou predikce a je získán druhý, úplný snímek typu P, použitelný jako referenční. V predikovaných snímcích jsou tedy přenášeny jen chyby odhadu a vektory pohybu. Za výstupem dekodéru může být užito postprocessingu pro odstranění blokových artefaktů.



Obr. 2.3: Blokové schéma dekodéru standardu MPEG-2 [6]

3 KOMPRESNÍ STANDARD MPEG-4

V roce 1999, tedy 4 roky po uveřejnění standardu MPEG-2 [4], byla zveřejněna specifikace standardu ISO/IEC 14496, známého pod názvem MPEG-4 [5]. Jeho cílovým určením měly být původně především systémy s nízkou přenosovou rychlostí, avšak rychle se rozšířil snad do všech oblastí multimediálních aplikací, až po video s vysokým rozlišením. Svůj podíl má na tom nejenom účinnost komprese, ale i použití dvou jader pro kódování [10].

Kódovací jádra standardu MPEG-4 jsou dvě a každé z nich je zaměřeno na jiný objem toku videosignálu:

- Jádro VLBV (Very Low Bitrate Video) uskutečňuje kódování s nízkou přenosovou rychlostí (5 – 64kb/s), se snímkovým kmitočtem do 15 snímků/s, maximálně v prostorovém rozlišení formátu CIF. Je vhodné pro pomalé datové linky.
- Jádro HBV (High Bitrate Video) poskytuje možnost kódování ve vysokém prostorovém rozlišení při snímkové frekvenci odpovídající televizním normám. Přenosová rychlost bývá v rozmezí 64kb/s – 4Mb/s.

Dalším novým prvkem tohoto standardu je definování typu vstupního videa. To lze rozdělit na přirozené (natural video), syntetické (synthetic video) a hybridní (hybrid video – kombinujícího přirozený a syntetický obsah). Odlišnost typů se projevuje při kódování, kdy přirozené video je kódováno pomocí predikce a kompenzace pohybu, za použití DCT. Pro syntetické video se využívá modelování objektů a přístupu k nim. Při kódování může být videoscéna rozdělena na více samostatných objektů [2].

Obrazový vstup kodéru a výstup dekodéru, dle standardu MPEG-4, může být videosekvence ve formátu 4:2:0, 4:2:2, či 4:4:4 prokládaného či neprokládaného řádkování.

3.1 VLASTNOSTI A OBRAZOVÉ PROFILY

Standard MPEG-4 poskytuje velký počet obrazových, zvukových, grafických a scénu popisujících profilů, uvedených například v [10]. Princip pojetí je obdobný jako u standardu MPEG-2, kdy pomocí zápisu kombinace „*profil@úroveň*“ sdělujeme informaci o tom, jaká podoblast (metody a parametry) standardu je (či bude) implementována v kodéru a dekodéru a rovněž lze na základě takového označení ověřit, zda zařízení tento standard splňuje. Obrazové profily jsou určeny pro kódování přirozeného, syntetického a hybridního obrazového obsahu. Mnoho z těchto profilů podporuje určitý typ škálování, některé z nich i více druhů současně. Následující přehled, převzatý z [10], popisuje základní rysy těchto obrazových profilů. Pro přehlednost jsou profily rozděleny do skupin podle typu kódovaného obsahu.

Pro kódování přirozeného obrazového obsahu je k dispozici tato pětice profilů:

1. **Simple** – výkonné, chybám odolné kódování obdélníkových snímků, vhodné k aplikování v mobilních a bezdrátových sítích.
2. **Simple Scalable** – přidává do profilu Simple podporu pro kódování s použitím časového a prostorového škálování. Vhodné pro poskytování služeb s více úrovněmi kvality, např. z důvodu limitovaného bitového toku, či omezených zdrojů dekodéru.
3. **Core** – doplňuje profil Simple o možnost kódování časově škálovaných objektů libovolného tvaru. Použitelné pro aplikace poskytující relativně jednoduchou obsahovou interaktivitu.
4. **Main** – do profilu Core přidává širokou podporu pro kódování prokládaných, částečně průhledných a jinými vlastnostmi disponujících typů objektů. Použití: vysílání s interaktivním a zábavným obsahem, DVD.

5. **N-Bit** – profil Core doplňuje o možnost kódování objektů s rozlišením vzorků v rozsahu od 4 do 12 bitů. Vhodné pro použití v dohledových systémech.

Následující profily jsou určeny pro kódování syntetického a hybridního obsahu:

6. **Simple Facial Animation** – poskytuje základní prostředky pro animaci modelu lidského obličeje, například pro použití k audiovizuální prezentaci pro osoby s poruchou sluchu.
7. **Scalable Texture** – umožňuje prostorově škálované kódování nehybných objektů (textur), využitelného například pro aplikace požadující více úrovní škálovatelnosti – textury v počítačových hrách, či digitální fotoaparáty s vysokým rozlišením.
8. **Basic Animated 2-D Texture** – poskytuje prostorové a kvalitativní škálování a na drátové síti založené animace pro nepohyblivé obrazové objekty (textury) a rovněž jednoduché animace lidské tváře
9. **Hybrid** – kombinuje schopnost dekódování časově škálovaných přirozených objektů s libovolným tvarem (podobně jako Core) se schopností dekódovat více syntetických či hybridních objektů, včetně lidské tváře, či animací nepohyblivých objektů. Vhodné pro rozličné, obsahově bohaté multimediální aplikace.

Verze číslo 2 standardu MPEG-4 Visual přidává další profily pro přirozené video:

10. **Advanced Real-Time Simple** – poskytuje vyšší odolnost vůči chybám při kódování obdélníkových snímků, při použití zpětného kanálu a vylepšené stálosti časového škálování s nízkým zpožděním.
11. **Core Scalable** – přidává podporu kódování časově a prostorově škálovaných objektů s libovolným tvarem do profilu Core. Hlavním znakem tohoto profilu je objektově zaměřené kvalitativní a prostorově/časové škálování pro oblasti, či objekty zájmu.
12. **Advanced Coding Efficiency** – profil vylepšující účinnost kódování pro obdélníkové snímky i objekty s libovolným tvarem.

Profily pro syntetický a hybridní obsah byly ve verzi 2 doplněny o následující:

13. **Advanced Scaleable Texture** – podporuje dekódování textur libovolného tvaru a nepohyblivých objektů včetně škálovatelného kódování.
14. **Advanced Core** – kombinuje schopnost dekódovat objekty různého tvaru (stejně jako v Core) se schopností dekódovat škálované, nepohyblivé objekty libovolného tvaru (jako v Advanced Scaleable Texture).
15. **Simple Face and Body Animation** – je nadstavbou Simple Face Animation profilu, jenž doplňuje o animaci lidského těla.

V pozdějších verzích byly přidány ještě následující profily:

16. **Advanced Simple** – podobný profilu Simple, neboť podporuje pouze obdélníkové snímky, má však několik vylepšení: účinnější kódování u B snímků, čtvrtpixelovou kompenzaci pohybu, upravené kvantizační tabulky, globální kompenzaci pohybu.
17. **Fine Granularity Scalability** – umožňuje přerušení bitového toku rozšiřující vrstvy na libovolné bitové pozici, takže kvalita přijatého snímku se může snadno přizpůsobit přenosovým a dekódovacím podmínkám. Profil FGS může být použit s profilem Simple nebo Advanced Simple na pozici základní vrstvy.
18. **Simple Studio** – profil s vysokou kvalitou pro použití ve střihačích studiích. Má pouze I rámce, avšak s podporou libovolných tvarů objektů. Bitový tok může dosáhnout až 2 Gbit/s.
19. **Core Studio** – přidává P rámce do profilu Simple Studio, čímž jej činí efektivnějším, avšak za cenu složitější implementace.

3.2 SNÍMKOVÉ A OBJEKTIVĚ ORIENTOVANÉ KÓDOVÁNÍ

Ve standardu MPEG-2 dochází ke kódování snímků složených z bloků, respektive makrobloků. Standard MPEG-4 tuto možnost doplňuje o objektivě orientované kódování [2]. Základními pojmy objektivě orientovaného kódování jsou: objektivá rovina videa VOP (Video Object Plane) a video objekt VO (Video Object). Definice podle [1] uvádí: „Video objekt VO je plocha video scény, která může zabírat oblast libovolného tvaru a může existovat po libovolně dlouhou dobu. Instance video objektu ve specifickém časovém okamžiku se nazývá objektivá rovina videa VOP“.

Z jiného úhlu pohledu lze tedy říci, že snímek je rozdělen do několika objektivých rovin videa VOP. Objektivou rovinu tvoří určitý ucelený fyzický celek daného snímku – entita – představovaná například osobou, zvířetem, či předmětem. Sekvence objektivých rovin náležejících jedné entitě pak představuje video objekt VO. Každý video objekt VO je charakterizován informací o tvaru, textuře a pohybu. Scéna je pak zpravidla složena z několika takovýchto obrazových objektů. Tento pohled od základu mění přístup k uspořádání dat určených pro kódování, známý například z MPEG-1 a MPEG-2, kde VOP byl v podstatě tvořen úplným obrazovým snímkem a sekvence těchto snímků tvořila VO.

Objektivé kódování má tedy potenciál pro to, být mnohem pružnější, než předchozí metody založené na obdélníkových snímcích. Lze tak například provést kódování jednotlivých VO v různé obrazové kvalitě, s odlišným časovým rozlišením, v závislosti na důležitosti daného objektu pro finální scénu. Pro kódování textur může být využita DCT, DWT a jejich modifikace. Nabízí se také možnost nepřeberných možností úprav VO při zpracování videosignálu a takřka libovolné manipulace s objekty nejen přirozeného, ale i syntetického videa a jejich kombinací.

Přestože by se zdálo, že objektivě orientované kódování díky svým možnostem bude hlavním používaným způsobem kódování v MPEG-4, opak je pravdou. Nejoblíbenějším dle [1] nadále zůstává kódování celých snímků. To je prováděno pomocí nástrojů seskupených v tzv. „Simple“ profilech. Tyto nástroje jsou podobné předchozím standardům MPEG-1/2 založených na metodách DCT kódování makrobloků s pohybově kompenzovanou predikcí. Simple profil využívá hybridní DPCM/DCT model s doplňujícími nástroji pro zvýšení účinnosti kódování a přenosové efektivity. Z důvodu velké popularity Simple profilu byly vytvořeny i další obdobné profily, založené na kódování obdélníkových VOP snímků, například Advanced Simple, či Advanced Real-Time Simple.

V jednotlivých obrazových profilech jsou použity různé techniky, pomocí nichž lze provádět operace kódování objektů velmi efektivně. Stručným shrnutím podrobných informací, uvedených v [1] a [2], lze jmenovat například tyto:

- I-VOP – intra-kódovaný obdélníkový VOP. Kódování obdélníkových rámců bez použití predikce, za použití DCT a kvantování pro bloky velikosti 8x8.
- P-VOP – inter-kódovaný obdélníkový VOP. Kódování pomocí predikce z předchozího referenčního I-VOP nebo P-VOP. Kompenzace pohybu je založena na práci s makrobloky o velikosti 16x16 pixelů, může být dosaženo subpixelové přesnosti pohybového vektoru. Po kompenzaci pohybu opět dochází k DCT, kvantování a entropickému kódování.
- B-VOP – inter-kódovaný obousměrně predikovaný VOP. Zlepšuje účinnost kompenzace pohybu použitím obousměrné predikce, avšak za cenu zvýšení požadavků na paměťovou kapacitu a zpoždění kódování.
- Režim krátké hlavičky (Short Header mode) – poskytuje kompatibilitu mezi standardy MPEG-4 Visual a ITU-T H.263. I-VOP a P-VOP pak mají stejnou

syntaxi jako I a P snímky základního režimu H.263, díky čemuž jsou dekodovatelné i v H.263 dekodéru.

- Čtyři pohybové vektory na makroblok – zvyšují efektivitu kompenzace pohybu zmenšením oblasti pro jeden vektor pohybu. Výchozím stavem je totiž kompenzace pro jasový blok o velikosti 16x16 a chrominanční o velikosti 8x8. Pomocí této funkce může kodér zvolit menší velikost pohybové kompenzovaného bloku a to o velikosti 8x8 pro jasovou a 4x4 pro chrominanční složku. V tom případě bude zapotřebí určení čtyř pohybových vektorů na makroblok, avšak při dosažení větší efektivity.
- Neomezené vektory pohybu – umožňují využít referenčního makrobloku i v případě, kdy se částečně vyskytuje za hranicí VOP. Chybějící vzorky za hranicí VOP jsou pak extrapolovány. Vektor pohybu tak může být umístěn i za hranice referenčního VOP, což zvyšuje účinnost kompenzace pohybu obzvláště v případech, kdy se objekty pohybují dovnitř a ven z obrazu.
- Čtvrtpixelová přesnost pohybových vektorů – vzorky referenčního VOP jsou interpolovány na půlpixelovou polohu a následně i na čtvrtpixelovou, a to ještě před predikcí a kompenzací pohybu. Tím vzrůstá komplikovanost predikce a kompenzace pohybu i samotné rekonstrukce obrazu, avšak při dosažení vyšší účinnosti kódování i v porovnání s půlpixelovým rozlišením.
- Globální kompenzace pohybu – vychází z myšlenky, že makrobloky v rámci jednoho objektu provádí často stejný pohyb. A to ať už se jedná o lineární pohyb do stran, rotaci, či přiblížení a oddálení (zoom). To umožňuje kodéru popsat pohyb celého objektu pomocí malého množství parametrů. Pro každý VOP pak postačují čtyři globální vektory pohybu, z nichž je ve výsledku pomocí interpolace vypočten pohybový vektor pro každý pixel.
- VOP kódování libovolného tvaru – tvar každého VOP je kódován současně s dalšími parametry, jako jsou vektory pohybu a textura objektu. Informace o tom, které pixely patří VO v daném časovém okamžiku, definuje BAP (Binary Alpha Plane). Tato BAP v podstatě představuje matici o velikosti shodné s VOP, kde jednotlivé prvky matice mohou nabývat pouze dvou hodnot. V případě, že pixel patří objektu, je hodnota elementu rovna 255; pokud objektu nepatří, pak je rovna 0. Tato matice se nazývá binární maska, či bitmapa. Před samotným kódováním je BAP rozdělen na BAB (Binary Alpha Blocks) o velikosti 16x16 pixelů. Každý BAB je pak kódován nezávisle. Jestliže je blok tvořen vzorky se samými nulami, lze ho považovat za průhledný; naopak pokud jejich hodnoty jsou 255, pak je neprůhledný. Hlavními nástroji pro kódování BAB jsou aritmetické kódování založené na obsahu CAE (Context-based Arithmetic Encoding) a kompenzace pohybu.
- Šedoškálové kódování tvaru – namísto toho, aby BAP matice obsahovala vzorky s pouze dvěma hodnotami (255 a 0), může tato matice obsahovat vzorky s hodnotami od 0 do 255, jejichž hodnota vyjadřuje míru průhlednosti daného pixelu. Hodnota 0 pak značí zcela průhledný pixel, naopak hodnota 255 neprůhledný pixel. Hodnoty vzorků však i přesto nadále reprezentují tvar VOP. Z těchto důvodů je tato myšlenka označována spíše jako šedoškálové než binární kódování. Tyto šedoškálové blokové informace o tvaru jsou následně kódovány pomocí DCT, podobně jako při kódování textur.

K dispozici je samozřejmě i řada dalších technik zaměřených na predikci (Newpred – kodér při predikci vybírá ze sady referenčních VOP), kvantování (alternate quantiser – odlišná metoda pro práci s AC koeficienty), či rozlišení (Dynamic Resolution Conversion DRC – kódování VOP se sníženým prostorovým rozlišením) a další. S ohledem na rozsah práce však nebudou podrobněji popisovány.

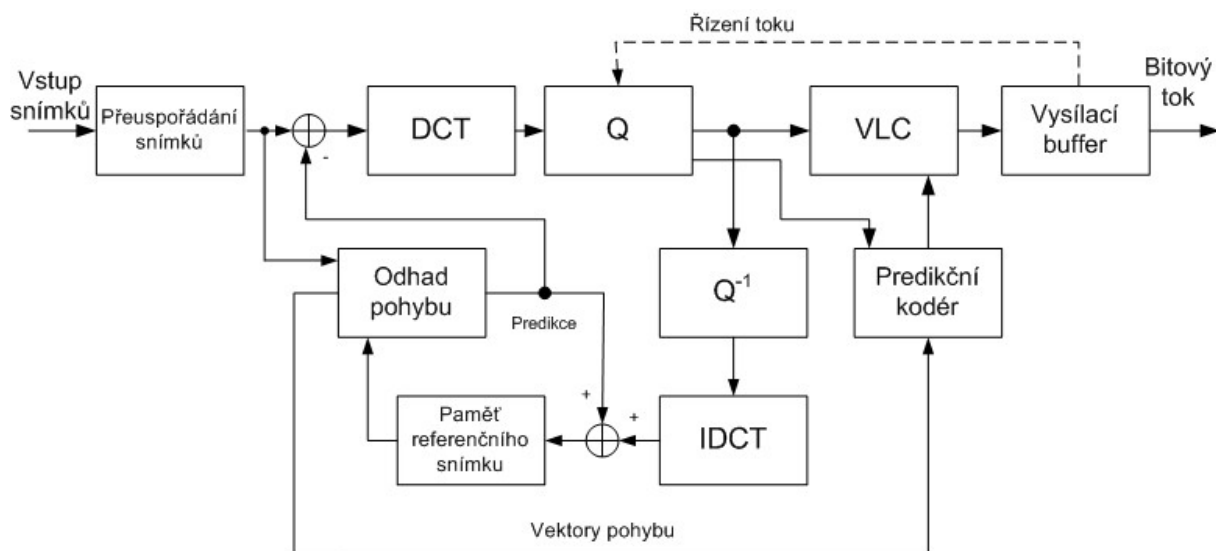
3.3 SCHÉMA A POPIS KODÉRU A DEKODÉRU SOUSTAVY MPEG-4

Podobně jako u předchozích standardů, ani MPEG-4 jednoznačně nedefinuje schéma kodéru a dekodéru, ale určuje zejména strukturu kódovaného datového toku. Ze znalosti principu zpracování videosignálu lze určit blokové schéma kodéru i dekodéru, vyobrazené například v [2]. Jak je z nákrešů Obr. 3.1 a Obr. 3.2 patrné, blokové schéma je u standardu MPEG-4 obdobné schématu MPEG-2. To je způsobeno potřebou implementace podobné sady základních kódovacích a dekódovacích funkcí. Hlavní rozdíly lze však nalézt uvnitř jednotlivých bloků, lišících se v realizaci dle definice konkrétního standardu. Z porovnání obou schémat je také na první pohled zřejmá asymetrie obou systémů. Kodér je mnohem složitější, a to i proto, že kromě samotných obvodů určených pro kódování jsou v něm obsaženy i obvody dekodéru.

3.3.1 Princip kodéru

Kodér, znázorněný na obrázku Obr. 3.1, zpracovává snímky přicházející na vstup a vytváří kódovaný, komprimovaný bitový tok. Kodér je tvořen přímou a zpětnou větví. V přímém směru probíhá kódování datového toku dle [2] takto: na vstup kodéru přichází první snímek typu I, který bude kódován jako Intra, bez vazby na předchozí, či následující snímky. Následuje DCT aplikovaná na bloky luminance nebo chrominance o velikosti 8x8 pixelů. Každý z 64 vzorků je následně kvantován v kvantizéru Q, čímž dojde v závislosti na nastavení kvantizační tabulky k redukci koeficientů vyšších kmitočtů. DC koeficient (určující průměrnou hodnotu jasu bloku) je zpracován odlišně od AC koeficientů. Vzhledem k jeho uvažované podobnosti s předchozím snímkem je kódován diferenční metodou s předchozím snímkem určeným jako referenčním. Následně jsou nenulové hodnoty čteny způsobem cirkak a kódovány s proměnnou délkou – VLC a přesunuty do vysílacího bufferu. Bitová rychlost výstupního toku je ovládána řídicím obvodem pomocí změny váhového koeficientu kvantizační tabulky.

Ve zpětné větvi dochází k inverzní kvantizaci a inverzní DCT, tím se získá rekonstruovaný snímek (stejně jako v dekodéru) a ten je uložen do paměti referenčních snímků. Zde je využit jako referenční k predikci pohybu a k získání vektorů pohybu. Kompenzace pohybu je založena na makroblocích. Pohybové vektory jsou zakódovány a vysílány společně s dotčenými snímky. Predikovaný snímek je pak odečten od skutečně přicházejícího a kódována je pak pouze vzniklá chyba predikce. Tím vznikají Inter-kódované jednosměrně respektive obousměrně predikované P-VOP a B-VOP.

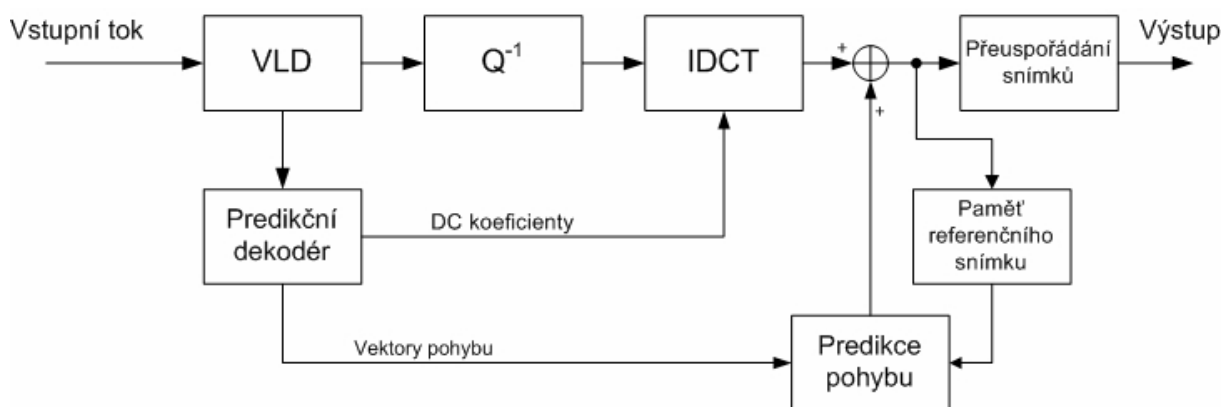


Obr. 3.1: Blokové schéma kodéru standardu MPEG-4 [2]

3.3.2 Princip dekodéru

V dekodéru, popsaném ve [2], znázorněném na Obr. 3.2, je vstupní komprimovaný bitový tok dekódován a rekonstruován do podoby obrazových snímků odpovídajících vstupním snímkům kodéru. Jedná se tedy o činnost inverzní k funkci kodéru. Z důvodu použití komprese nejsou vstupní a výstupní snímky identické.

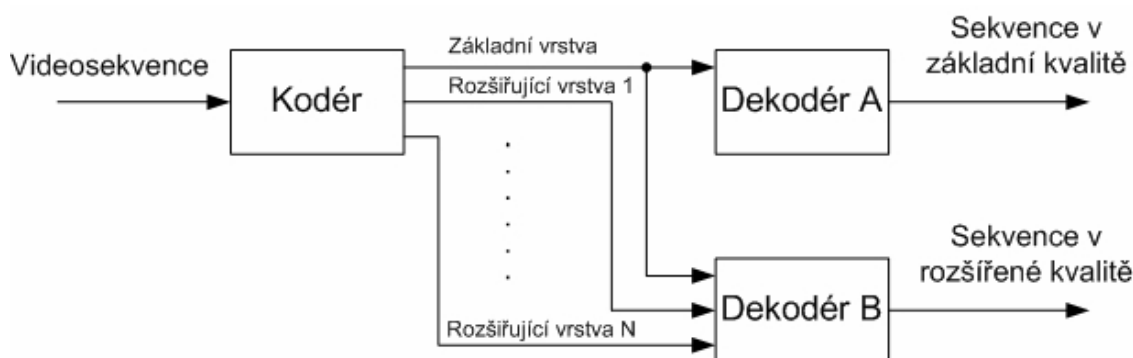
Na vstup dekodéru přichází bitový tok s proměnnou délkou slova, jenž je dekódován VLD obvodem a jsou z něj následně získány makrobloky VOP snímků a vektory pohybu. Makrobloky jsou inverzně kvantovány a zpracovány inverzní DCT. V závislosti na typu snímku získáme buď úplný intra-kódovaný I-VOP snímek, nebo chybu predikce u inter-kódovaných P-VOP a B-VOP. První příchozí snímek typu I-VOP je uložen do snímkového bufferu, kde je prostřednictvím vektorů pohybu následujícího predikovaného VOP upraven do podoby předpovězeného snímku. Ve sčítacím členu dojde k propojení takto upraveného snímku s přijatou chybou predikce, čímž je získán nový úplný snímek, jenž může být opět použit jako referenční pro zpracování dalšího příchozího predikovaného snímku.



Obr. 3.2: Blokové schéma dekodéru standardu MPEG-4 [2]

4 HIERARCHICKÉ KÓDOVÁNÍ DLE STANDARDŮ MPEG

Pod pojmem hierarchické kódování, jinak též nazývané škálovatelné kódování, si lze představit činnost, při níž je v průběhu kódování videosekvence uspořádán do určitého počtu odstupňovaných bitových toků. Tyto bitové toky jsou nazývány „vrstvy“ a jsou složeny ze základní vrstvy a jedné nebo více rozšiřujících vrstev. Obsahem základní vrstvy je bitový tok v základní – výchozí podobě. Rozšiřující vrstvy pak doplňují základní tok o rozšiřující informace, pomocí nichž lze získat datový tok s vyšší informační hodnotou. Dekodér na straně přijímače pak může vzhledem k okolnostem zpracovat pouze některé vrstvy z přijatého bitového toku. Tato myšlenka, uvedená například v [1], je zobrazena na Obr. 4.1, jenž představuje obecný koncept škálování, kdy dekodér A přijímá pouze základní vrstvu datového toku v nižší kvalitě, avšak dekodér B přijímá a zpracovává úplný datový tok.



Obr. 4.1: Obecný koncept škálování [1]

Toto rozdělení na vrstvy má dle [1] a [6] několik výhod. Kupříkladu méně výkonný dekodér přenosného zařízení s malým rozlišením displeje může přijímat pouze základní bitový tok, bez nutnosti dekódování datového toku ve vysokém rozlišení, avšak jiný – výkonnější dekodér může přijímat a dekódovat všechny vrstvy téhož toku a poskytnout tak plnohodnotné video ve vysoké kvalitě. Dále pak může být například z původního datového toku vyseparován a přenášen pouze základní bitový tok přes síťový segment s omezenou kapacitou, bez rizika zahlcení uzlu, či zvýšení ztrátovosti paketů. Rovněž tak může být rozlišena priorita jednotlivých vrstev, kdy základní (na chybovost citlivá) vrstva je přenášena s vyšším zabezpečením i prioritou než rozšiřující vrstvy. V případě výskytu silného rušení tak nedojde k náhlému výpadku příjmu, ale nejdříve vysadí doplňkový bitový tok rozšiřující vrstvy, při zachování přenosu alespoň základní vrstvy. Degradace signálu tak není náhlá, ale pozvolně postupná.

Jak již bylo zmíněno, již ve standardu MPEG-2 bylo škálování použito. Tvoří je zde časové, prostorové a kvalitativní škálování. Ve standardu MPEG-4 došlo k rozšíření především o škálování Fine Granular Scalability (FGS) a také o možnost využití škálování nejen pro obdélníkové snímky, ale i pro video objekty libovolných tvarů.

Vzhledem k předmětu zadání této práce a rovněž tak z důvodu vhodnosti pro praktickou programovou realizaci v prostředí Matlab, provedenou za pomoci dokumentace [13], [19] a [20], budou v následujícím textu podrobněji popsány specifiky tří daných druhů hierarchického kódování. Konkrétně se jedná o prostorové, kvalitativní SNR a Fine granular škálování. Dále budou stručně zmíněny i některé další typy, včetně možností jejich vzájemné kombinace.

4.1 PROSTOROVÉ ŠKÁLOVÁNÍ (SPATIAL SCALABILITY)

Prostorové odstupňování rozděluje datový tok na vrstvy s různým prostorovým rozlišením obrazu. Dekódování základní vrstvy produkuje video sekvenci s nízkým rozlišením. Při zpracování všech vrstev toku je získána sekvence s vyšším, maximálním dostupným rozlišením, které bylo kódérem zpracováno. U videoobjektů jsou podporovány maximálně tři úrovně škálování, pro nepohyblivé obrázky a text to může být až jedenáct úrovní [2].

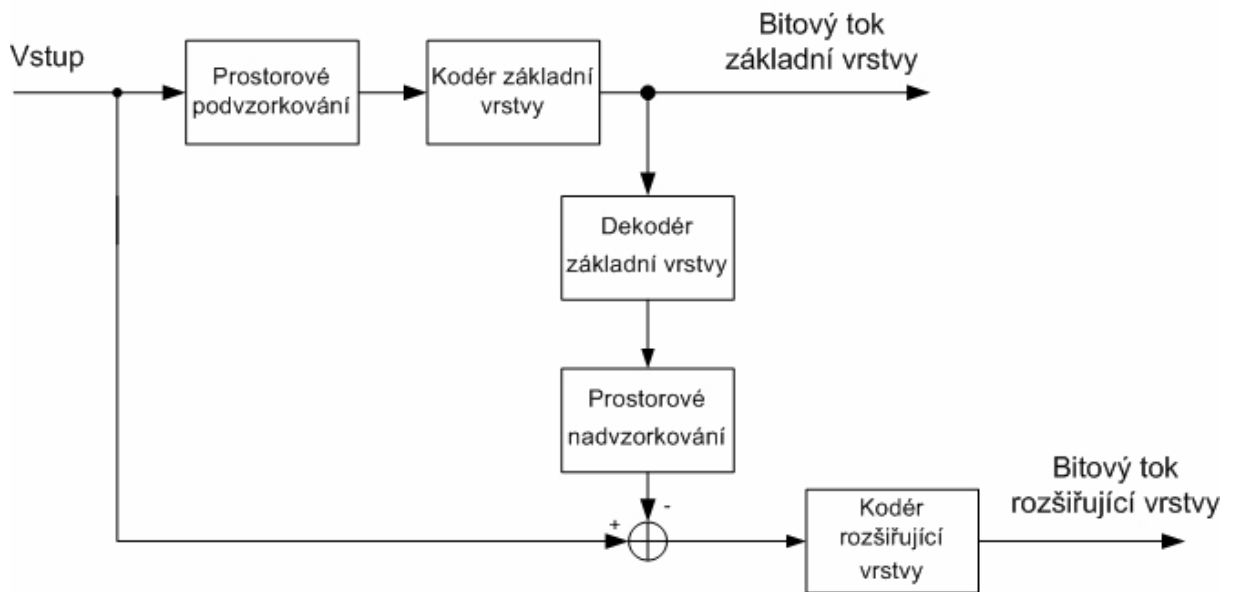
Jednou z hlavních výhod tohoto modelu, uvažovaného již ve standardu MPEG-2, je podle [6] možnost realizace myšlenky slučitelnosti mezi systémy s požadavkem na různé rozlišení obrazu. Pro přiblížení tohoto principu lze uvažovat obraz se standardní rozlišovací schopností pro systémy SDTV v základní vrstvě a doplňkový signál pro zobrazení v systémech vysokého rozlišení HDTV v rozšířené vrstvě. Vytvoření základního datového toku z plnohodnotného vstupního signálu, například typu HDTV, lze provést pomocí podvzorkování. Podvzorkovaný signál odchází v základní vrstvě ven z kódéru, avšak je současně ještě v kódéru převzorkován na originální rozlišení a odečten od nepatrně zpožděného původního signálu, směřujícího do části rozšiřující vrstvy. Zakódován je tak pouze rozdíl mezi originálním signálem a dekodovaným signálem základní vrstvy.

Jiným způsobem využití tohoto odstupňování může být technika PANSCAN, u které se v základní vrstvě přenáší informace o obrazu v poměru stran 4:3 a v rozšiřující vrstvě pouze doplněk pro zobrazení ve formátu 16:9. Současně může být zvolena libovolná pozice výřezu z většího formátu (PAN), nebo ve velkém obrazu může být umístěn jiný menší obraz s libovolnou polohou (SCAN). V záhlaví sekvence i každého snímku je pak přenášena informace o provedeném posuvu.

4.1.1 Schéma a funkce kódéru

Pro realizaci kódování videosekvence do dvou vrstev, detailně popsané v [1] a [6], znázorněné na Obr. 4.2, je třeba, aby v kódéru proběhla následující řada kroků:

1. vstupní snímek, či video objekt je horizontálně a vertikálně podvzorkován.
2. snímek s nižším rozlišením je zakódován do podoby základní vrstvy.
3. současně je tento snímek s redukováným rozlišením dekodován a nadvzorkován do původního rozlišení (vzniká tzv. „prostorově predikovaný“ snímek).
4. Je proveden rozdíl originálního snímku v plném rozlišení a nadvzorkovaného interpolovaného snímku základní vrstvy (tzv. „prostorově predikovaného“).
5. Rozdíl obou snímků je zakódován do podoby rozšiřující vrstvy.

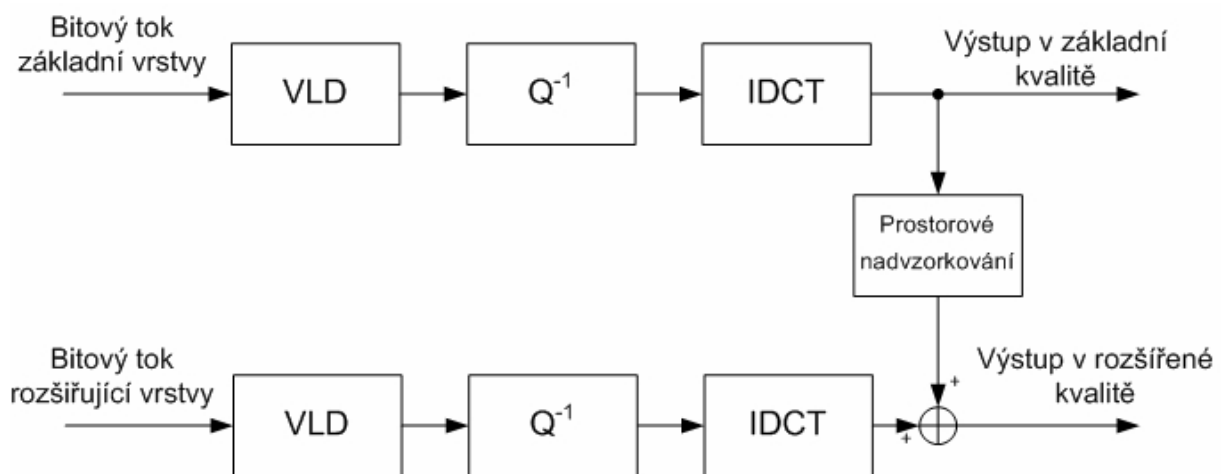


Obr. 4.2: Blokové schéma kodéru pro prostorové škálování [2], [6]

4.1.2 Schéma a funkce dekodéru

Dekódování všech vrstev přijatých v bitovém toku musí být podporováno i na straně dekodéru. Jednovrstvý dekodér dokáže dekódovat pouze základní vrstvu, za současného produkování výstupní video sekvence pouze v redukovaném rozlišení. Pro dekódování obou uvažovaných vrstev je třeba dvouvrstvého dekodéru, jenž dokáže plně zpracovat vstupní datový tok, více viz [6]. V tom případě proběhne v dekodéru, znázorněném na Obr. 4.3 tato série kroků:

1. Je dekódována základní vrstva a interpolována do původního prostorového rozlišení.
2. Dekóduje se rozšiřující vrstva
3. Je sloučena dekódovaná základní vrstva a dekódované rezidua z rozšiřující vrstvy do podoby úplného výstupního snímku.



Obr. 4.3: Blokové schéma dekodéru pro prostorové škálování [6]

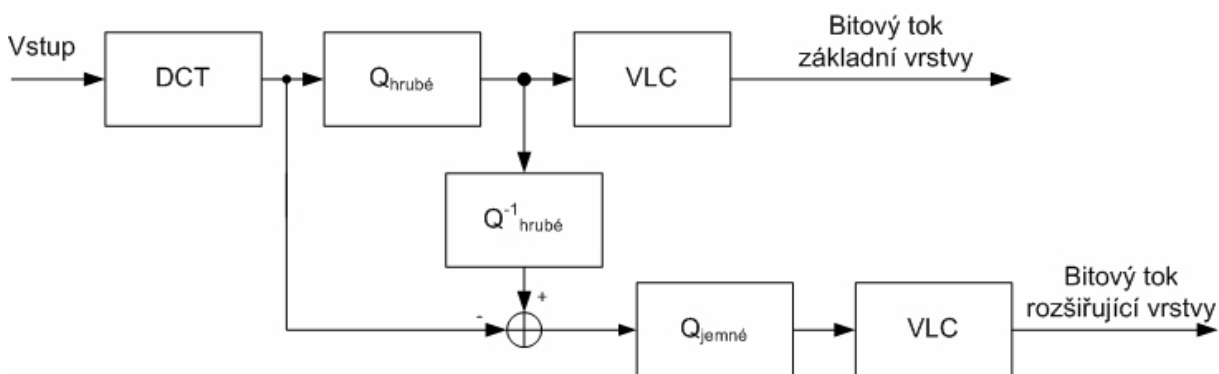
4.2 KVALITATIVNÍ ŠKÁLOVÁNÍ (SNR SCALABILITY)

Kvalitativní odstupňování (Quality scalability) bývá též nazývané „odstupňování podle poměru signál/šum (SNR – Signal to Noise Ratio)“. V základním bitovém toku je zakódován signál s nižší jakostí a tedy s vyšším poměrem šumu, rozšířený tok pak vylepšuje tento poměr a s ním i jakost výsledného obrazu. Tohoto rozdělení na dva jakostně odlišné obrazové signály lze podle [6] využít pro nabídku signálu LDTV nebo SDTV, v závislosti na konstrukci přijímače, nebo z důvodu zajištění schopnosti příjmu signálu ve zhoršených podmínkách. Signál v základní vrstvě je lépe zabezpečený než v rozšiřující vrstvě a tudíž je vyšší pravděpodobnost příjmu i při větším rušení. Za nepříznivých podmínek tak nedochází k náhlému výpadku, ale k postupnému snížení kvality příjmu, neboť i po výpadku rozšiřujícího toku je k dispozici základní, ačkoliv obrazově méně kvalitní bitový tok. Praktická realizace kvalitativního odstupňování je provedena pomocí kvantování koeficientů DCT, kdy koeficienty jsou pro základní bitový tok hrubě nakvantovány kvantizérem. Pro rozšiřující vrstvu je jemně nakvantován rozdíl mezi původními a hrubě nakvantovanými koeficienty. Odlišnost v kvantování je určena rozdílnými kvantovacími tabulkami obou kvantizérů. Přenosová rychlost pro každý z obou toků je obvykle asi 3 Mbit/s.

4.2.1 Schéma a funkce kodéru

Kodér pro kvalitativní škálování, znázorněný na Obr. 4.4, obsahuje dle [2] a [6] část pro realizaci základní vrstvy (která je v podstatě totožná s obecným schématem kodéru) a část rozšiřující vrstvy (snižující kvantizační šum datového toku videosignálu). Posloupnost činností je takováto:

1. Vstupní obrazový signál prochází obvodem DCT, je hrubě nakvantován (dle specifikace kvantizační tabulky) a po provedení VLC odchází jako základní tok.
2. Hrubě nakvantovaný základní tok, s vysokým poměrem kvantizačního šumu, je inverzně kvantován a odečten od původního signálu (před provedením kvantování).
3. Rozdíl obou signálů je v následujícím kvantizéru jemně nakvantován (dle vlastní kvantizační tabulky), zakódován s proměnnou délkou slova v obvodu VLC a vysílán jako přídatný datový tok rozšiřující vrstvy.

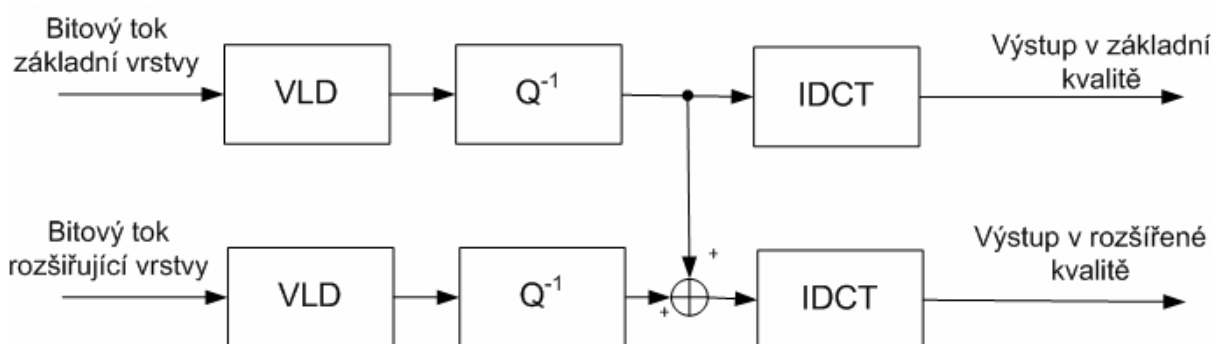


Obr. 4.4: Blokové schéma kodéru pro kvalitativní škálování [6]

4.2.2 Schéma a funkce dekodéru

Pro přichodící datový signál dekodéru, viz Obr. 4.5, tvořený vrstvami základního a rozšiřujícího toku je posloupnost činností, probíhajících po úvodním demultiplexování toků, podle [6] následující:

1. Datový tok základní a rozšiřující vrstvy s proměnnou délkou slova je dekodován ve VLD obvodu.
2. Je provedeno inverzní kvantování, s odlišnými kvantovacími tabulkami pro každý z datových toků obou vrstev.
3. Oba datové toky se sloučí do jednoho a je provedena inverzní DCT.
4. Predikce pohybu probíhá již na sloučeném toku; vektory pohybu jsou přenášeny v základní vrstvě.



Obr. 4.5: Blokové schéma dekodéru pro kvalitativní škálování [6]

4.3 FINE GRANULAR ŠKÁLOVÁNÍ (FGS)

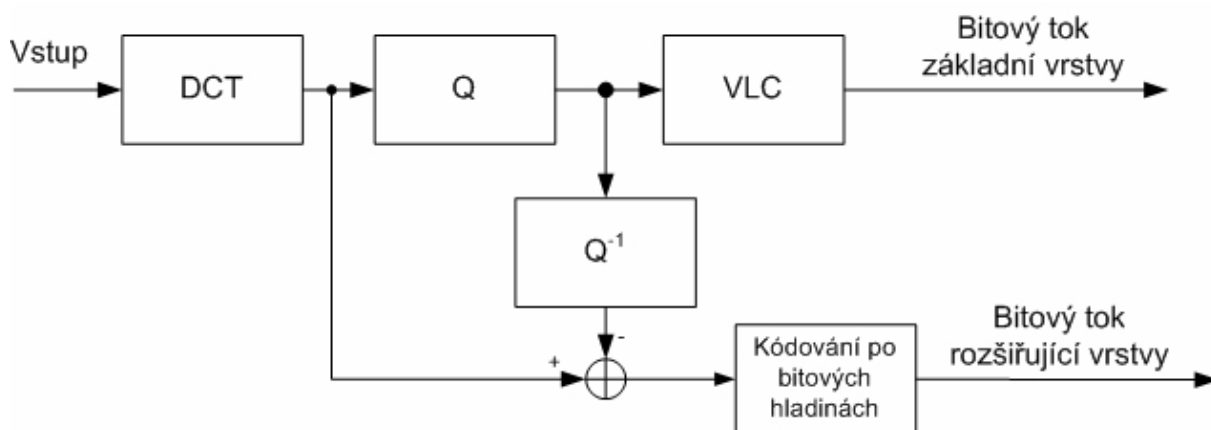
Tento typ hierarchického kódování se poprvé objevil ve standardu MPEG-4, avšak až v pozdější době, jako doplňující profil pod názvem: The Fine Granularity Scalability (FGS), více viz [5] a [10]. Profil FGS umožňuje odstupňování pomocí postupného zvyšování kvality videosekvence po malých krocích. Na první pohled je podobný kvalitativnímu škálování, avšak jeho hlavní výhodou je, že umožňuje přerušení datového toku rozšiřující vrstvy v libovolném okamžiku. Všechny do té chvíle přijaté bity jsou pak použity pro zvýšení kvality obrazu přeneseného v základní vrstvě.

Jeho vhodným využitím je dle [1] streamované video napříč celým spektrem různých druhů síťových zařízení, pro které umožňuje škálování datového toku takovým způsobem, aby bylo dosaženo nejvyšší možné kvality, jakou jen bitový tok vzhledem k okolnostem dovoluje a to vše bez nutnosti vícenásobného kódování videosignálu. Prakticky může být toto přizpůsobení realizováno streamovacím serverem, který na základě požadavku koncového zařízení na zaslání videosekvence o určité bitové rychlosti, provede vysílání základní vrstvy a současně specificky zkrácené podoby rozšiřující vrstvy.

4.3.1 Schéma a funkce kodéru

Na Obr. 4.6 je zobrazeno zjednodušené blokové schéma kodéru, viz [1] a [5]. Z něj je patrné, že hierarchické kódování proběhne takto [1]:

1. Vstupní obrazový tok je transformován pomocí DCT, následně kvantován, zakódován ve VLC a vysílán jako základní vrstva.
2. Kvantovaný datový tok je současně přiváděn do inverzního kvantizéru a následně odečten od DCT koeficientů získaných před kvantováním.
3. Rozdílem kvantovaných a původních koeficientů každého bloku vzniknou matice reziduí, které jsou poté přeuspořádány cik-cak čtením.
4. Postupně jsou pak kódovány koeficienty reziduí vždy po jednotlivých bitových hladinách – v každém kroku tedy bity na stejné úrovni důležitosti, v pořadí od MSB po LSB. Počet potřebných bitových hladin k zakódování je tedy určen nejvyšší nalezenou hodnotou kódovaných reziduí. MSB hladinou je první úroveň, na níž se vyskytuje MSB rezidua s nejvyšší hodnotou.

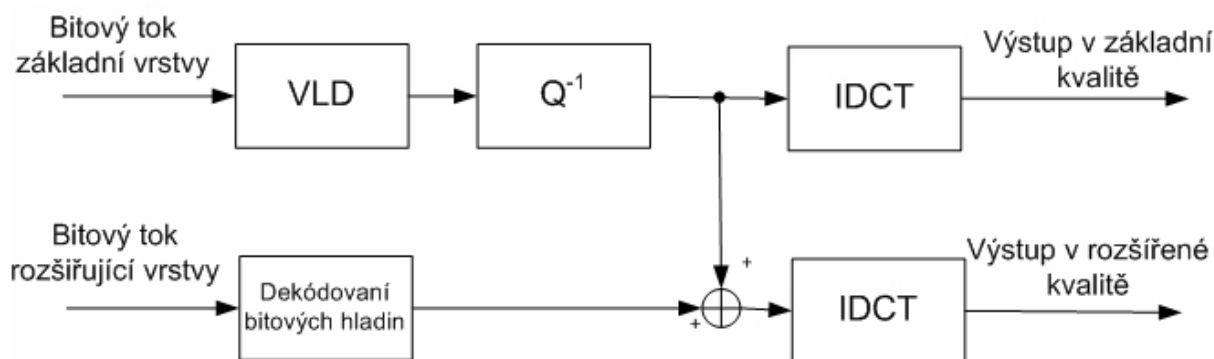


Obr. 4.6: Blokové schéma kodéru pro škálování Fine granular [1]

4.3.2 Schéma a funkce dekodéru

V dekodéru, znázorněném na Obr. 4.7 probíhá dekódování základní vrstvy a úplné, či zkrácené rozšiřující vrstvy následovně [1]:

1. Zakódovaný datový tok základní vrstvy je dekódován VLD, inverzně kvantován a transformován IDCT na obrazový tok v základní kvalitě.
2. Koeficienty reziduí rozšiřující vrstvy jsou obnoveny z přijatých dekódovaných bitových hladin a přičteny ke koeficientům základní vrstvy.
3. Koeficienty obou sloučených vrstev jsou následně inverzně transformovány IDCT obvodem a získána je tak videosekvence v rozšířené kvalitě. Jestliže došlo k přijetí zkrácené rozšiřující vrstvy, pak i přesnost koeficientů bitových hladin je omezená, a to přímo úměrně množství přijatých bitových hladin.



Obr. 4.7: Blokové schéma dekodéru pro škálování Fine granular [1]

4.4 DALŠÍ TYPY ŠKÁLOVÁNÍ

Pro doplnění přehledu v oblasti hierarchického kódování je vhodné uvést i některé další typy škálování, u nichž je odstupňování realizováno na základě více či méně podobných principů jako u již výše zmíněných. Všechny mají ale jednu společnou myšlenku – nalézt kritérium, či dělicí bod, jenž rozdělí datový tok na dvě či více vrstev, odstupňovaných zpravidla dle jejich důležitosti pro výslednou obrazovou informaci.

4.4.1 Časové škálování (Temporal scalability)

Časové odstupňování, dostupné již ve standardu MPEG-2, poskytuje dle [6] rozdělení datového toku na základní vrstvu se sníženou hodnotou snímkového kmitočtu a na rozšiřující vrstvu, doplňující jej na plnohodnotný počet snímku za časový interval. Tímto škálováním však není dotčeno prostorové rozlišení snímků, kdy snímky v základním i rozšiřujícím toku jsou kódovány v plném rozlišení.

Prakticky může být tento režim realizován například tak, že v základní vrstvě jsou vynechávány některé snímky B, neboť dekodér může obraz dekódovat i bez nich. Vynechané snímky jsou pak přenášeny v rozšiřující vrstvě. Plného snímkového kmitočtu pak dosahujeme složením obou druhů toků.

K predikci snímků typu P v horní rozšiřující vrstvě mohou být použity nejen vlastní předchozí snímky této vrstvy, ale i předchozí nebo následující snímky vrstvy základní. Pro predikci snímků typu B v rozšiřující vrstvě jsou použity:

- Předchozí snímek základní a předchozí snímek rozšiřující vrstvy.
- Předchozí snímek rozšiřující a následující snímek základní vrstvy.
- Předchozí a následující snímek základní vrstvy

Snímky typu I v rozšiřující vrstvě jsou kódovány bez jakékoliv predikce.

Z předchozího popisu vyplývá, že v rozšiřující vrstvě není dovolena zpětná predikce, což určitým způsobem zjednodušuje práci dekodéru. Toto škálování je možné použít i pro prokládané řádkování, tím způsobem, kdy se v základní vrstvě přenáší prořídle liché a sudé půlsnímky a v rozšiřující vrstvě jejich doplněk.

4.4.2 Komplexní škálování (Complexity scalability)

Komplexní škálování dle [10] a [18] umožňuje kodérům různé složitosti vytvářet smysluplný bitový tok ze zadaného videosignálu, v závislosti na jejich možnostech. V podstatě tak shrnuje možnost použití různých typů odstupňování, dle možností kodéru i

dekodéru. V dekodéru pak umožňuje z dodaného bitového toku dekódovat signál v závislosti na schopnostech dekodéru a úrovni složitosti. Kvalita rekonstruovaného signálu je pak přímo úměrná komplexnosti použitého dekodéru. To znamená, že méně výkonné dekodéry dekódují pouze část bitového toku, jenž jim byl z kodéru doručen.

4.4.3 Objektové škálování (Object-based scalability)

Objektové škálování, viz [7], dovoluje přidání či odebrání videoobjektů z výsledné scény v závislosti na prioritizaci určitých objektů ve scéně. Při použití této funkce je možné přiřadit objektům zájmu vyšší prostorové, či časové rozlišení, čímž současně dojde ke snížení bitového toku nepodstatných elementů mimo objekt zájmu. Díky prioritizaci přenášených dat se také zlepšuje i odolnost bitového toku vůči chybám.

4.4.4 Rozdělování dat (Data partitioning)

Při tomto druhu škálování dochází po úplném zpracování signálu k rozdělení datového toku na základní a přídatný podle důležitosti jednotlivých částí signálu. Přičemž v obou vrstvách jsou přenášeny všechna záhlaví. Místem, v němž se datový tok dělí je tzv. break point, jenž určuje bod, od kterého jsou data důležitá pro přenos obsažena v základní vrstvě a méně podstatná data naopak obsažena v rozšiřující vrstvě, více v [6].

Prakticky si to lze ilustrovat na I snímku, jehož DC koeficient by byl přenášen v základní vrstvě a zbývající AC koeficienty pak v rozšiřující. Z přijatých DC koeficientů je pak možné sestavit obraz, i když s výrazně horší kvalitou. Bodem rozdělení při použití predikce pak může být místo ihned za vektory pohybu, které jsou přeneseny v základní vrstvě. Kompenzace pohybu predikovaného snímku jsou pak obsahem přídatného toku.

Priorita i zabezpečení základního datového toku je vyšší než u přídatného, je tedy schopen i vyšší odolnosti vůči rušení, díky samoopravitelnosti přijatých chybných bitů. Jak je patrné, tento typ škálování je podobný mnohem známějšímu kvalitativnímu škálování.

4.4.5 Škálování kombinující více druhů (Hybrid scalability)

Hierarchická kódování, provádějící odstupňování dle časového, prostorového, kvalitativního, či jiného hlediska, lze spolu navzájem s výhodou kombinovat, čímž lze docílit vyššího počtu odstupňovaných vrstev, či dosáhnout lepšího využití jejich jednotlivých výhod.

Standard MPEG-2 jako příklad uvádí různé kombinace použití prostorového, časového a kvalitativního škálování. Při použití kombinace dvou různých typů škálování vzniknou tři bitové toky (vrstvy), jenž je třeba dekódovat [4].

Ve standardu MPEG-4 je dle [1] definováno, viz Tab. 4.1, které typy objektů jsou v daném profilu obsaženy. Profily, jak známo, v podstatě určují sadu potřebných kódovacích nástrojů. Prakticky tak lze z Tab. 4.1 například určit, že profil FGS v sobě obsahuje objekty typu Simple, Advanced Simple a FGS. Díky tomu jsou k dispozici nástroje pro realizaci FGS prostorového a FGS časového škálování.

FGS časové škálování spojuje FGS s časovým odstupňováním, kdy rámeček rozšiřující vrstvy je zakódován pomocí dopředné, či obousměrné predikce pouze z rámečků základní vrstvy. DCT koeficienty rámečků rozšiřující vrstvy jsou následně zakódovány do bitových hladin pomocí metody FGS. Podobně je tomu u FGS prostorového škálování, kde je zmíněných technik využito k zakódování každého rámečků jako základní vrstvy a FGS rozšířené vrstvy.

Tab. 4.1: Přehled kombinací profilů a objektů [1]

Profil	Typy objektů																	
	Simple	Advanced Simple	Advanced Real-time Simple	Core	Main	Advanced Coding Efficiency	N-bit	Simple Scalable	Fine Granular Scalability	Core Scalable	Scalable Texture	Advanced Scalable Texture	Simple Studio	Core Studio	Simple Face Animation	Simple Face and Body Animation	Basic Animated Texture	Animated 2D Mesh
Simple	X	X																
Advanced Simple	X																	
Advanced Real-time Simple	X		X															
Core	X			X														
Advanced Core	X			X								X						
Main	X			X	X						X							
Advanced Coding Efficiency	X			X		X												
N-bit	X			X			X											
Simple Scalable	X							X										
Fine Granular Scalability	X	X							X									
Core Scalable	X			X				X		X								
Scalable Texture											X							
Advanced Scalable Texture												X						
Simple Studio													X					
Core Studio													X	X				
Basic Animated Texture											X				X		X	
Simple Face Animation															X			
Simple Face and Body Anim.																X		
Hybrid	X										X				X		X	X

5 REALIZACE ŠKÁLOVÁNÍ V PROSTŘEDÍ MATLAB

Samotná realizace výukové aplikace s grafickým uživatelským rozhraním v prostředí Matlab, tak jak byla specifikována v zadání této práce, byla provedena především za použití literatury [13] a [20]. S její pomocí byly vytvořeny funkce, provádějící prostorové, kvalitativní SNR a Fine granular škálování. Tyto funkce byly následně začleněny do jediné aplikace, doplněny o další funkčnosti a rovněž i o grafické uživatelské rozhraní.

Obsahem této kapitoly je podrobný popis funkcí jednotlivých typů škálování, určených pro zpracování obrazové informace hierarchickým kódováním a rovněž také vývojové diagramy, znázorňující průběh činností po vyvolání dané akce. Vstupem programu může být obrázek s libovolnými rozměry, vzhledem k názornosti je zpracován ve stupních šedi (pouze jasová složka), což je vhodnější pro vyniknutí detailů před a po kódování. Jako vzorový obraz, na němž jsou prezentovány odlišnosti škálování, jsem zvolil snímek rozhledny v Ocmanicích u Náměště nad Oslavou, získaný z vlastního archivu, zobrazený na Obr. 5.1.

Aplikace byla napsána a odladěna v prostředí Matlab verze 7.4.0.287 (R2007a) a samotný kód byl naprogramován metodou Switched Board Programming, tj. především za použití příkazů Switch a Case, které umožňují využití té vlastnosti funkcí, kterou je možnost volat sama sebe s různými parametry [20]. Jednotlivé grafické objekty byly definovány textovým způsobem přímo v programovém kódu, který ačkoliv je pracnější, poskytuje možnost lepší kompatibility mezi jednotlivými verzemi prostředí Matlab, snadnou a přehlednou definici jednotlivých parametrů, kratší a optimálnější zdrojový kód a v neposlední řadě též nezávislost na jinak nezbytném binárním *fig* souboru definujícím grafické uživatelské rozhraní.



Obr. 5.1: Obraz rozhledny použitý pro ověřování funkcí škálování

5.1 UŽIVATELSKÉ ROZHRANÍ APLIKACE

Grafické uživatelské rozhraní je pojato střídmě, za použití pouze nezbytných objektů, nutných pro definování vstupních parametrů a pro vyvolání požadovaných činností. Funkčně propojené objekty jsou vzájemně seskupeny pomocí rámců, které vizuálně ohraničují jednotlivé skupiny. Jedná se především o sekci pro volbu obrazového souboru a tři sekce pro jednotlivé typy škálování. Dále je rovněž aplikace doplněna několika tlačítky, po jejichž

stisku dochází k volání funkce společně se zadanými parametry. Vzhled aplikace je patrný z Obr. 5.2, který představuje hlavní uživatelské rozhraní aplikace.

Hierarchické kódování obrazů

RGB soubor:

☐ **Prostorové** Podvzorkovací krok (horizontální i vertikální):

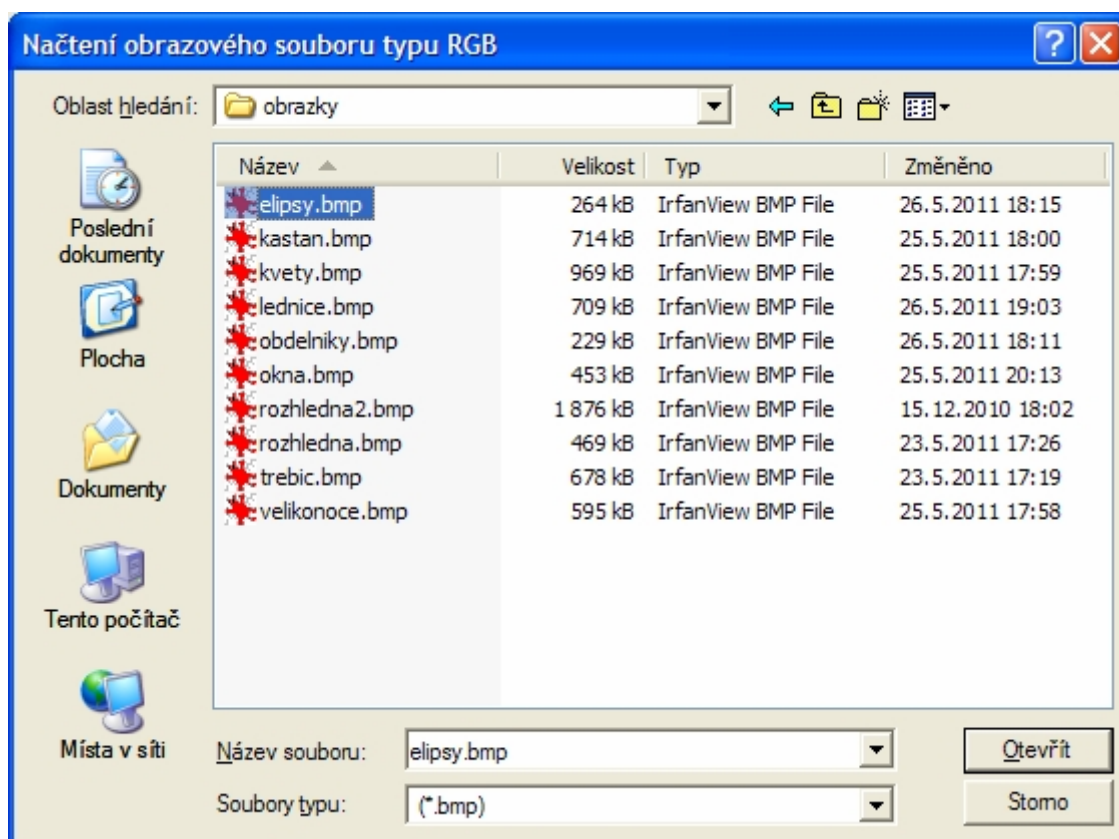
☐ **Kvalitativní** Kvantovací koeficient první vrstvy (max. 2047): = hrubé kvantování
Kvantovací koeficient druhé vrstvy: = jemné kvantování

☐ **Fine granular** Kvantovací koeficient první vrstvy:
Počet zpracovaných hladin druhé vrstvy:

Obr. 5.2: Uživatelské rozhraní aplikace v okamžiku po spuštění

Pro samotnou funkčnost aplikace je nezbytné uživatelské zadání obrazového RGB souboru, tj. obrázku v podobě bitové mapy. Tento soubor může být vložen pomocí klávesové zkratky **CTRL+V**, nebo výběrem z okna průzkumníku po stisku tlačítka „Procházet“. Typ souborů je omezen na bitové mapy s příponou *.bmp, čímž jsou odfiltrovány soubory jiných typů, viz Obr. 5.3. Pro výběr souboru je třeba jej v okně označit a výběr potvrdit stiskem tlačítka „Otevřít“. Vybírat lze soubory z libovolného umístění, které je operačnímu systému dostupné. Po stisku tlačítka „Otevřít“ dojde k přenesení názvu souboru, včetně úplné cesty k němu, do příslušného pole v okně aplikace. Tento výběr lze přerušit stiskem tlačítka „Storno“ a vrátit se tak zpět do hlavního okna aplikace.

V následujících sekcích grafického uživatelského rozhraní, v rámcích pro jednotlivé typy škálování, je třeba zatrhnout zaškrťovací políčko, tzv. checkbox, jenž při vyvolání funkce pro zobrazení metod škálování, informuje program o vyžádaných metodách. Jsou povoleny všechny kombinace zatržení. Pokud není zatrženo žádné políčko, není zobrazena žádná metoda, při zatržení jednoho a více, jsou zobrazena okna se všemi vybranými metodami. Je však třeba brát v úvahu fakt, že při zvolení více metod a při současném zadání velkého obrazového souboru, může zpracování a zobrazení zvolených metod trvat delší dobu, obvykle v řádu jednotek sekund.



Obr. 5.3: Rozhraní pro výběr obrazového souboru

Pro zobrazení výstupu prostorového škálování je nutné mít kromě zatrženého checkboxu rovněž určen podvzorkovací krok, který udává horizontální a vertikální prostorové podvzorkování vstupního obrázku. Jako výchozí hodnota je zadáno „2“, tzn. poloviční vertikální a horizontální rozlišení základní vrstvy. Dostupné volby jsou 1 - 8. Hodnotu podvzorkovacího kroku je možné zadat z klávesnice, nebo myší, klepnutím na šipky posuvníku. Hodnoty mimo daný rozsah jsou aplikací ignorovány.

Kvalitativní škálování je prováděno při současném zatržení výběrového políčka a při zadání dvou parametrů, definujících kvantovací koeficient první a druhé vrstvy, tj. hrubé a jemné kvantování. Oba kvantovací koeficienty jsou zadávány uživatelsky ručně. Hodnota kvantovacího koeficientu první vrstvy má z důvodu použití DCT (12-ti bitové koeficienty) smysl pouze pro hodnoty do 2047. Hodnota kvantovacího koeficientu druhé vrstvy má smysl pouze pro hodnoty menší, než je hodnota prvního koeficientu, a to z toho důvodu, že dochází ke kvantování rozdílu původních a hrubě kvantovaných hodnot, tedy k jemnému kvantování.

K aktivaci škálování Fine granular dochází při současném zatržení checkboxu a při zadaném kvantovacím koeficientu první vrstvy, společně se zadaným počtem zpracovaných hladin druhé vrstvy. Pro kvantovací koeficient první vrstvy platí stejná pravidla jako u kvalitativního škálování, tj. zadává se ručně a maximální hodnota je 2047 pro toto hrubé kvantování. Počet zpracovaných hladin druhé vrstvy je možné rovněž zadat z klávesnice, nebo klepnutím myší na šipky posuvníku. Výchozí hodnota a současně minimum je jedna hladina, maximum je deset hladin. To odpovídá hodnotě rezidua vyjádřené binárně deseti bity. Hodnoty zadané ručně, nevyhovující danému rozsahu jsou aplikací ignorovány.

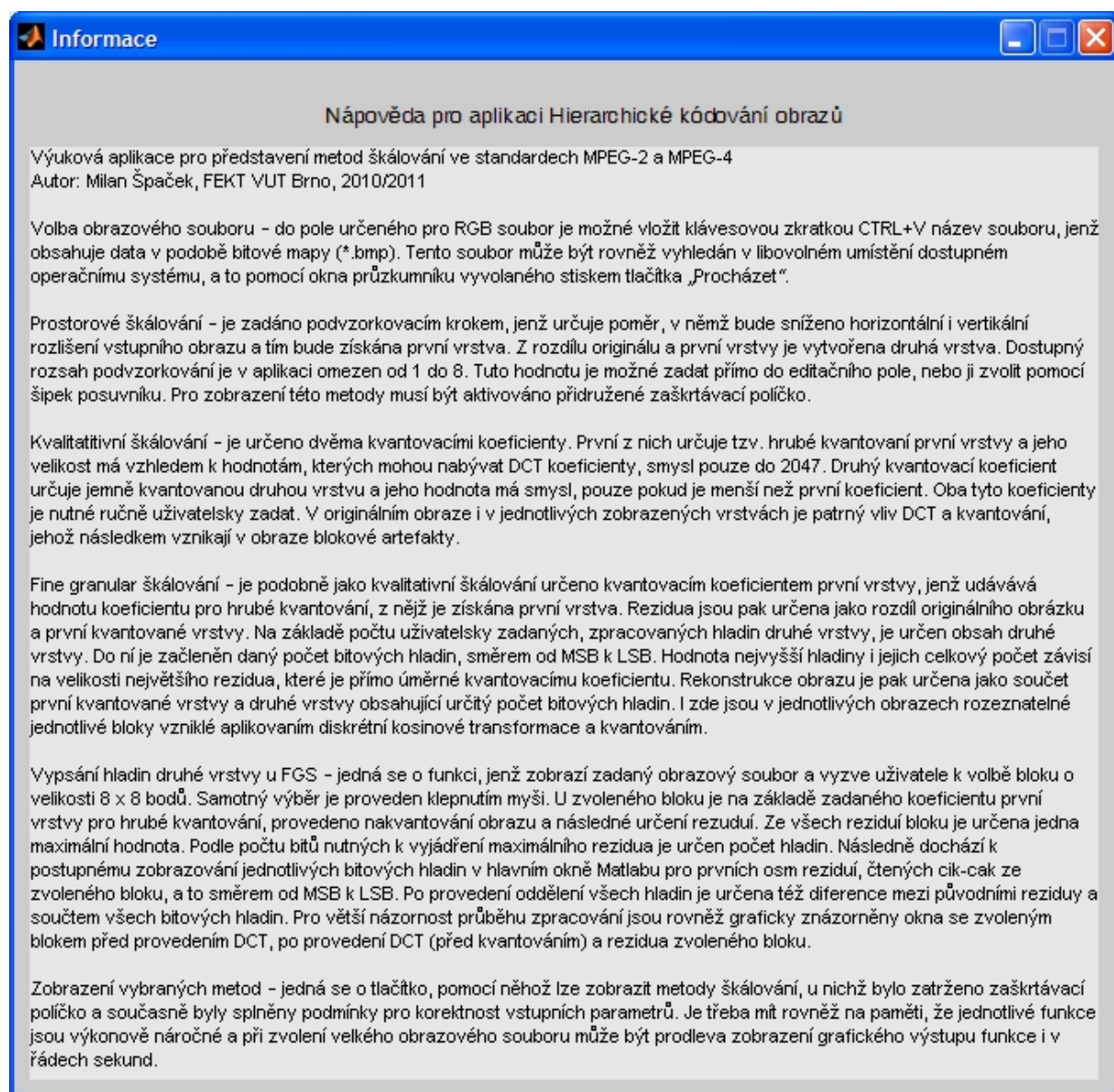
V sekci škálování Fine granular se dále nachází tlačítko „Vypsát hladiny“. Jeho primárním úkolem je do okna programu Matlab vypsát všechny hladiny, které mohou být v zadaném obrázku s určitým kvantovacím koeficientem první vrstvy přeneseny. Z důvodu přehlednosti se prakticky jedná o jeden, uživatelsky zvolený, blok, z nějž je vypsáno pro

každou bitovou hladinu prvních osm bitů jednotlivých reziduí, v pořadí od MSB směrem k LSB. Pro aktivaci této funkce musí být zadán vstupní RGB soubor a kvantovací koeficient první vrstvy. Jiné hodnoty a nastavení nemají na funkci vliv. Pro zvýšení názornosti této aplikace jsem provedl rovněž doplnění o funkčnost pro grafické zobrazení vybraného bloku před provedením DCT, dále pak po provedení DCT (s výrazným DC koeficientem) a rovněž také zobrazení reziduí daného bloku (tj. rozdílu originálu a první kvantované vrstvy).

Ve spodní části aplikace se nachází tři tlačítka. První a nejdůležitější z nich je „Zobrazit vybrané metody“. Po klepnutí na něj je spuštěna funkce, která v případě zadání vstupního obrazového souboru zjistí, která zatrhávací pole jsou zvolena, u těchto typů škálování ověří zadání požadovaných hodnot v polích daných sekcí a jejich korektnost. Posléze je pak pro každý typ škálování zobrazena v novém okně sada šesti obrázků, zobrazujících výstupy jednotlivých vrstev po provedení škálování a po rekonstrukci obrazu.

Následující tlačítko „Info“ poskytuje základní informace o aplikaci, a to zobrazením v novém okně, viz Obr. 5.4. A konečně tlačítko „Konec“ ukončuje aplikaci.

V dalších kapitolách budou detailněji popsány jednotlivé funkce, které mají zásadní vliv na funkčnost celé aplikace.



Obr. 5.4: Nápověda pro aplikaci

5.2 REALIZACE FUNKCE PROSTOROVÉHO ŠKÁLOVÁNÍ

Zdrojový kód, realizující prostorové škálování, využívá principů popsaných v předchozích kapitolách 4.1.1 a 4.1.2 pro kodér i dekodér, včetně jejich blokových schémat z Obr. 4.2 a Obr. 4.3.

Prakticky tedy při prostorovém škálování v programu dochází k horizontálnímu a vertikálnímu prostorovému podvzorkování zadaného obrázku, a to s krokem, jenž je uživatelsky zadán, viz Výpis kódu 5.1. Tím je získána základní vrstva $L1$. Rozšiřující vrstva je následně určena jako rozdíl původního obrázku a nadvzorkované (interpolované) první vrstvy. Druhá, rozšiřující, vrstva $L2$ tak obsahuje pouze jejich rozdíl. V oblasti rekonstrukce signálu pak dochází k součtu obou vrstev, a to vrstvy $L1$ interpolované na původní rozměr a vrstvy $L2$ nesoucí rozdílovou informaci. Detailně je tento princip rozebrán ve vývojovém diagramu na Obr. 5.5. Příklad znázorňující funkčnost kódu při podvzorkování první vrstvy na polovinu horizontálního i vertikálního rozlišení je uveden na Obr. 5.6.

Výpis kódu 5.1: skalovani.m – prostorové škálování zadaného obrazu

```
% vstupní parametry: X - obrázek, step - podvzorkovací krok

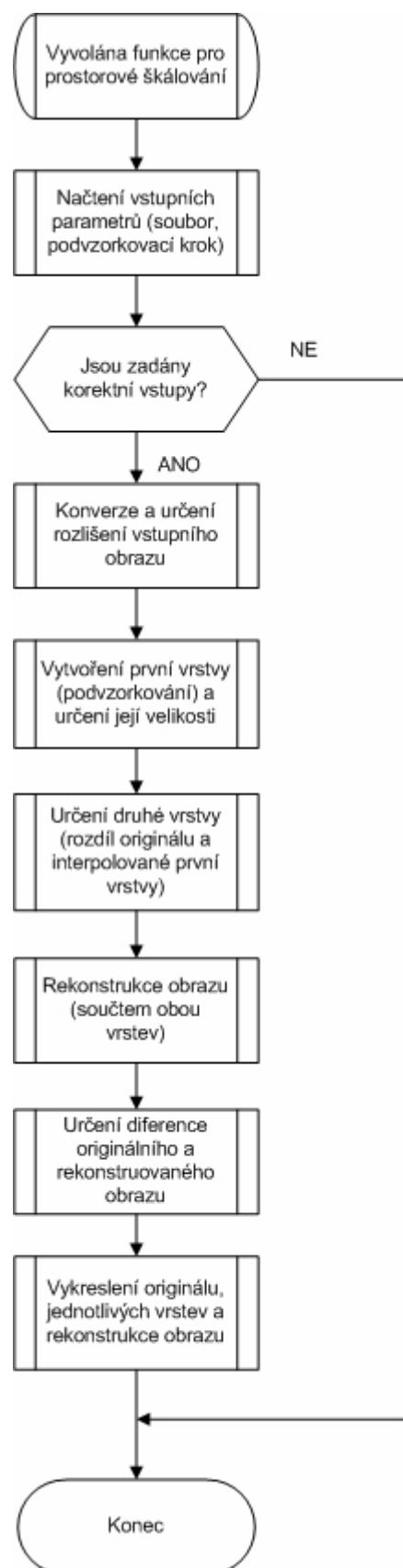
[rows cols] = size(X);
L1i = X(1:step:rows, 1:step:cols);
[rowsi colsi] = size(L1i); % vytvoření první vrstvy a určení velikosti

L1 = ones(size(X));
L1(1:rowsi, 1:colsi) = L1i; % vytvoření obrázku podvzork. vrstvy pro
zobrazení v okně s bílým pozadím

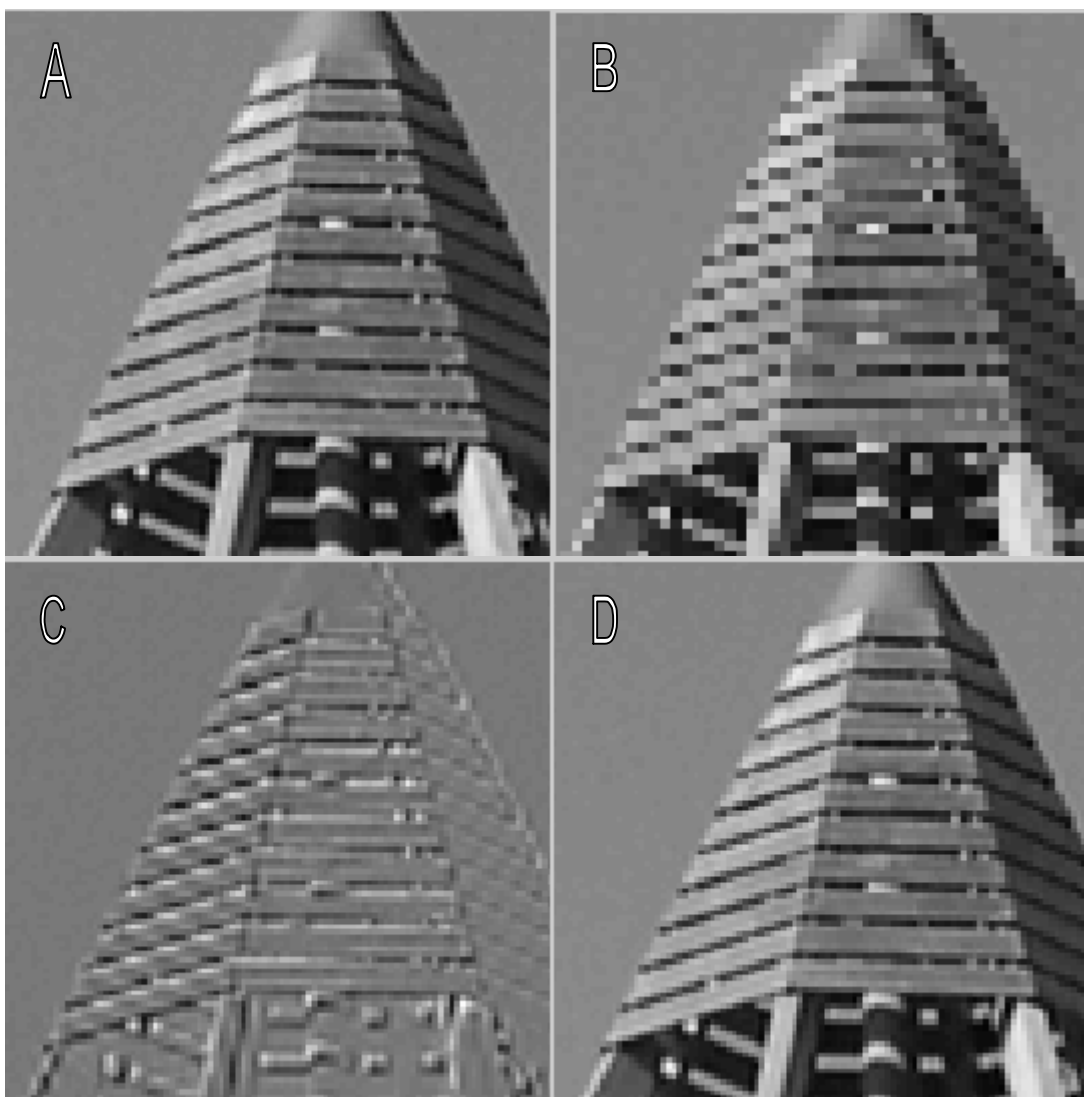
L2 = X - imresize(L1i, [rows cols], 'bicubic');
% určení druhé vrstvy jako rozdíl originálu a interpolované první
vrstvy, možné interpole: nearest, bilinear, bicubic

Out = imresize(L1i, [rows cols], 'bicubic') + L2;
% rekonstrukce obrazu součtem obou vrstev

Dout = X - Out;
% difference originálního a rekonstruovaného obrazu
```



Obr. 5.5: Vývojový diagram funkce prostorového škálování



Obr. 5.6: Detailní výřez pro : a) originální snímek, b) podvzorkovanou první vrstvu, c) rozšiřující vrstvu, d) rekonstruovaný obraz

5.3 REALIZACE FUNKCE KVALITATIVNÍHO ŠKÁLOVÁNÍ

Jak je z dříve uvedeného Obr. 4.4 a Obr. 4.5 patrné, odlišnost mezi oběma vrstvami datového toku je realizována především pomocí různých kvantovacích koeficientů obou kvantizérů sloužících pro hrubé a jemné kvantování. Uživatelsky určenými parametry programu tak jsou kromě vstupního obrázku rovněž kvantizační koeficienty obou kvantizérů.

Při praktické realizaci v aplikaci, při níž bylo postupováno dle vývojového diagramu na Obr. 5.7, z něž je nejdůležitější část uvedena ve Výpis kódu 5.2, nejprve dochází k blokovému zpracování diskrétní kosinové transformace obrázku po maticích o velikosti 8×8 bodů. Následně je pomocí funkce *quant* vstupní bitový tok nakvantován s hrubým krokem, čímž je získán signál základní vrstvy $Xq1$ s velkým kvantizačním šumem. Rozšiřující vrstvu $Dq2$ pak tvoří jemně nakvantovaný rozdíl původního obrázku X a základní vrstvy $Xq1$. Rekonstruovaný obraz $Xq12$ je pak určen jako součet obou vrstev $Xq1$ a $Dq2$. Příklad provedení tohoto typu škálování je zobrazen na Obr. 5.8. Vzhledem k tomu, že jsem při kvantování rozšiřující vrstvy použil pro názornost záměrně velký koeficient, jsou v jednotlivých vrstvách i v rekonstruovaném obrázku znatelné blokové artefakty.



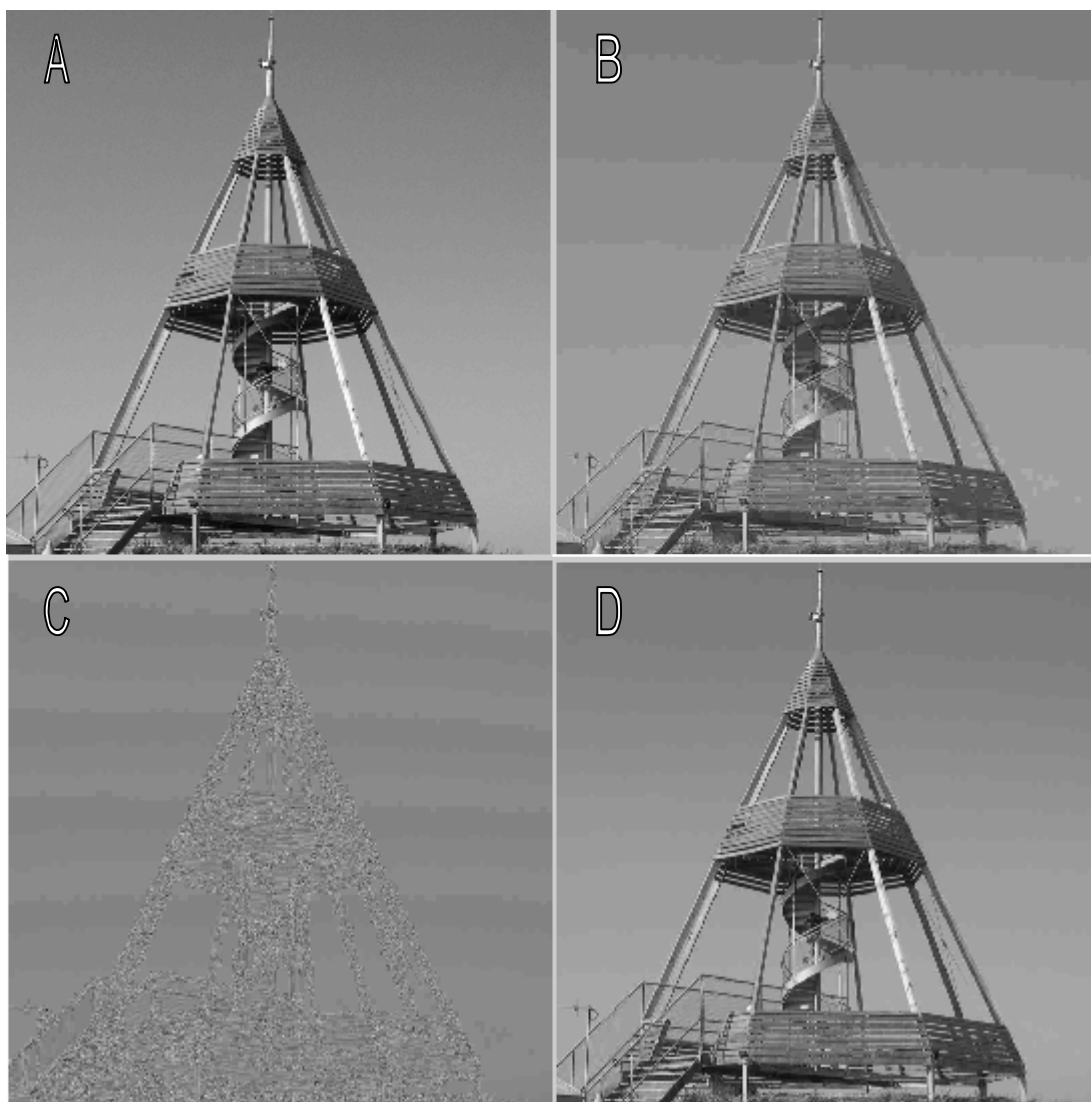
Obr. 5.7: Vývojový diagram funkce kvalitativního škálování

Výpis kódu 5.2: skalovani.m – kvalitativní škálování zadaného obrazu

```
% vstupní parametry: g - obrázek, N1, N2 - kvantizační koeficienty

T=@dct2;
U=@idct2;
X = blkproc(g,[8 8],T); % blokové provedení DCT 8x8

Xq1 = quant(X,N1); % vytvoření první kvantované vrstvy
D = X - Xq1; % určení rozdílu originálu a první vrstvy
Dq2 = quant(D, N2); % vytvoření druhé kvantované vrstvy
Xq12 = Xq1 + Dq2; % rekonstrukce obrazu z první a druhé vrstvy
D12 = X - Xq12; % difference originálního a rekonstruovaného obrazu
```



Obr. 5.8: Celkový pohled na: a) originální snímek, b) hrubě kvantovanou první vrstvu, c) jemněji kvantovanou rozšiřující vrstvu, d) rekonstruovaný obraz

5.4 REALIZACE FUNKCE ŠKÁLOVÁNÍ FINE GRANULAR

Tento typ škálování se svým zpracováním částečně podobá kvalitativnímu škálování SNR. Zejména datový tok základní vrstvy je určen prakticky stejným způsobem, jak je patrné z Obr. 4.6 a Obr. 4.7. Vstupními parametry této funkce jsou: vstupní obrázek, kvantizační koeficient a počet úspěšně zpracovaných hladin rozšiřující vrstvy FGS.

Z Výpis kódu 5.3 a rovněž i z vývojového diagramu na Obr. 5.9 je patrné, že v aplikaci nejdříve dochází k blokovému zpracování obrázku pomocí diskretní kosinové transformace, a to po maticích o velikosti 8x8 bodů. Základní vrstva X_{q1} je stejně jako u kvalitativního škálování získána nakvantováním vstupního obrazu zadaným koeficientem. Z rozdílu originálního obrazu a základní vrstvy je získána matice reziduí D , jenž je určena pro provedení FGS. Z hodnoty největšího vzorku, vyskytujícího se v matici D , je určen počet bitových hladin potřebných pro zakódování reziduí. Kódování FGS probíhá směrem od MSB k LSB pro uživatelem zadaný počet úspěšně dekódovaných hladin. Rozšiřující vrstva D_{q2} je tak ve výsledku určena postupně dekódovanými hladinami, přičemž nedekódované hladiny vnáší do rekonstruovaného obrazu odchylky, a to přímo úměrně počtu nedekódovaných hladin a jejich hodnotě. Výsledný rekonstruovaný signál je tvořen součtem základní vrstvy X_{q1} a rozšiřující vrstvy D_{q2} , kódované FGS pro určitý počet hladin. Příklad takového kódování je uveden na Obr. 5.10. V tomto příkladu jsem záměrně zvolil podobnou hodnotu kvantovacího koeficientu základní vrstvy jako u kvalitativního škálování, avšak pro kódování rozšiřující vrstvy jsem určil, že budou zpracovány pouze tři nejvyšší bitové hladiny.

Výpis kódu 5.3: skalovani.m – škálování Fine granular zadaného obrazu

```
% vstupní parametry: g - obrázek, N - kvant. koeficient, s - počet
zpracovaných hladin

T = @dct2;
U = @idct2;
X = blkproc(g,[8 8],T); % provedení DCT blokově 8x8

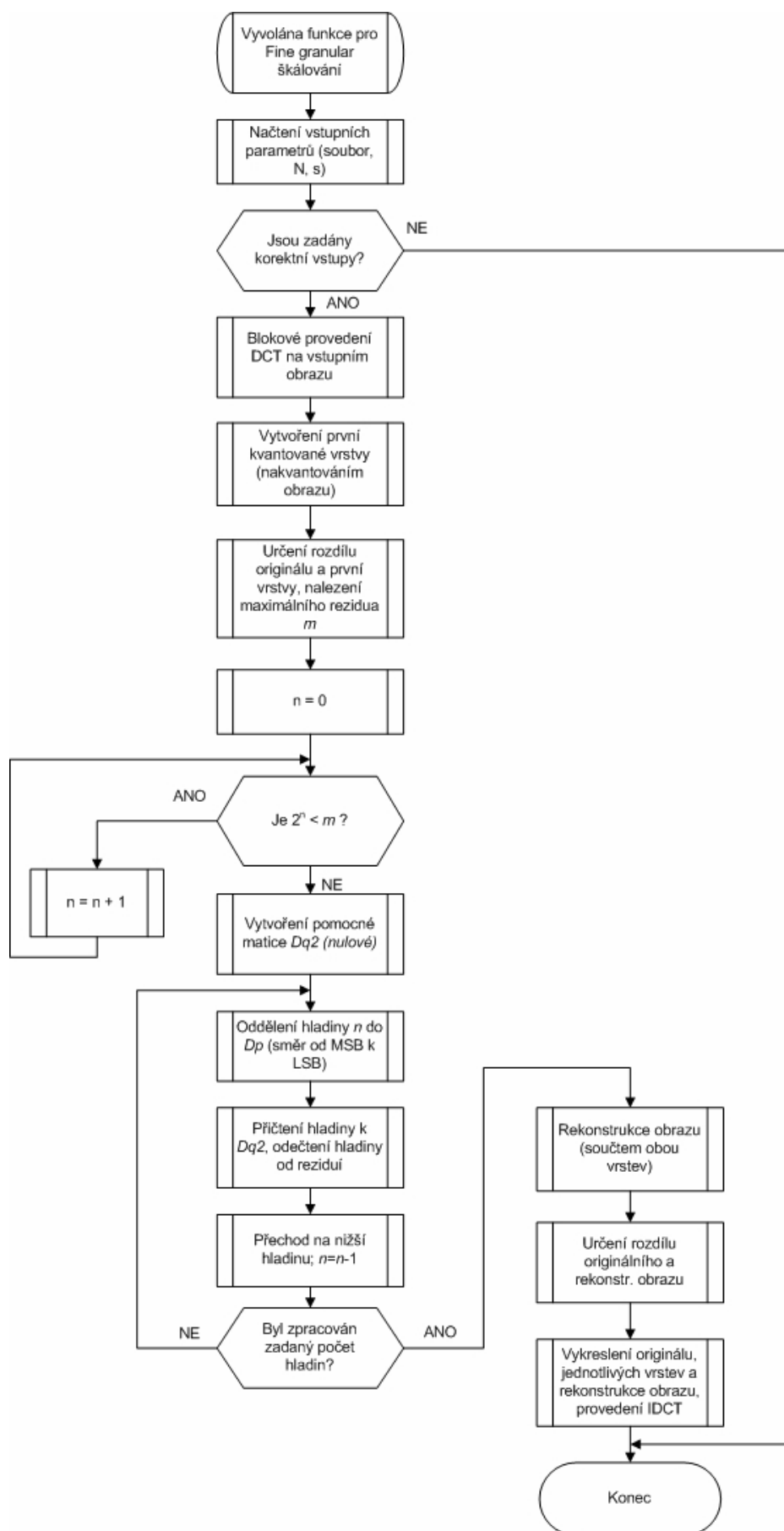
Xq1 = quant(X,N); % vytvoření první kvantované vrstvy
D = X - Xq1; % určení rozdílu originálu a první vrstvy
m = max(max(abs(D))); % určení maximálního rezidua

n = 0; % cyklus pro určení max. počtu hladin, dokud 2^n < m (N/2)
while pow2(1,n) < m
    n = n + 1;
end

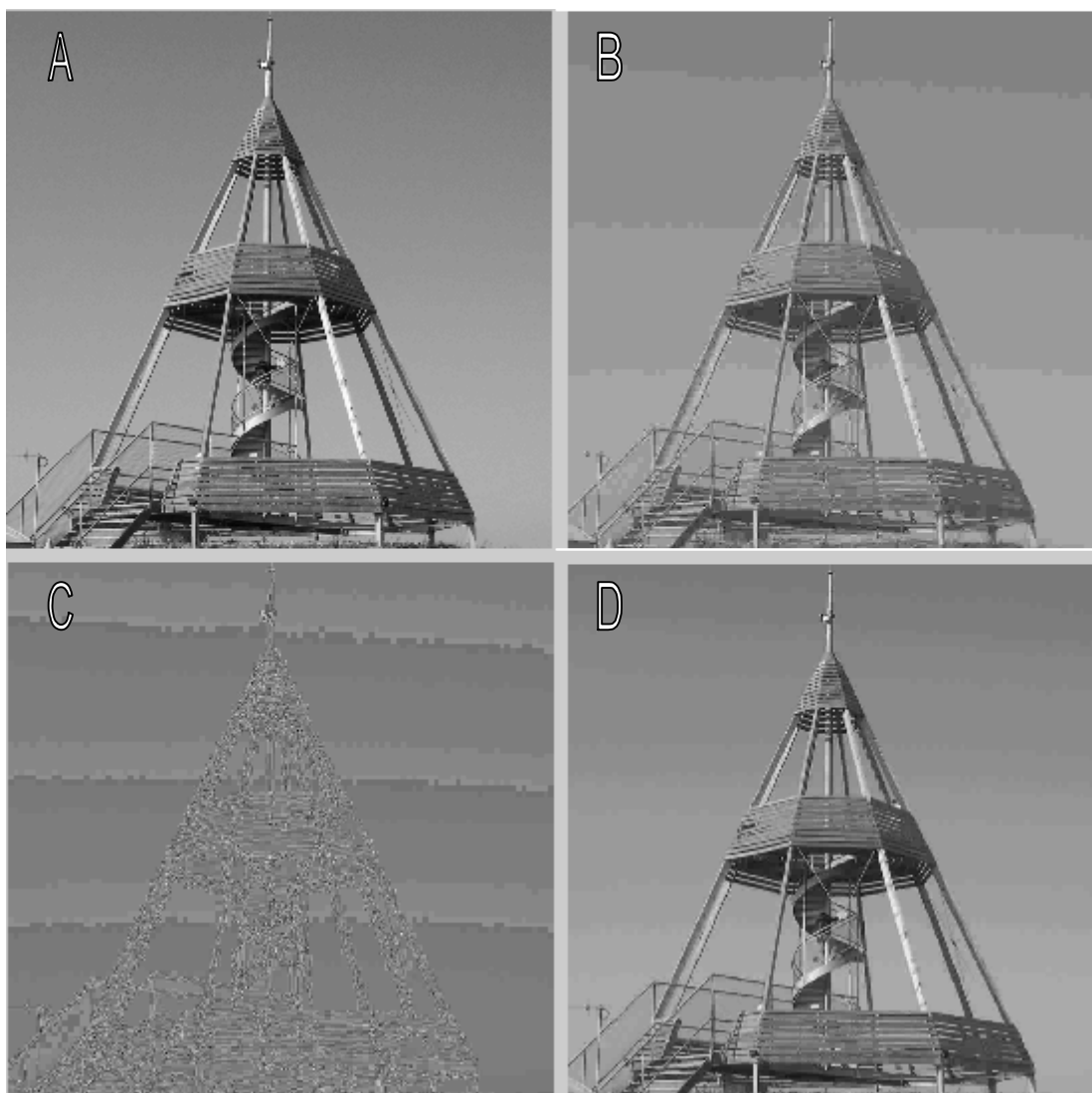
Dq2 = zeros(size(g)); % nulová matice
Dp2 = D; % rozdíl originálu a 1. vrstvy

% cyklus pro oddělení "s" hladin do 2. vrstvy a ponechání zbytku
for c = 1:s
    Dp = pow2(1,n) * fix(Dp2/pow2(1,n)); % separace hladin podle 2^n,
    směrem od MSB
    Dq2 = Dq2 + Dp; % vytvoření 2. vrstvy z "s" hladin
    Dp2 = Dp2 - Dp; % od rozdílu orig. a 1. vrstvy odečte hladinu
    n = n - 1; % přechod na nižší hladinu, od MSB k LSB
end

Xq12 = Xq1 + Dq2; % rekonstrukce obrazu z 1. a 2. vrstvy
D12 = X - Xq12; % rozdíl originálu a rekonstruovaného obrazu
```



Obr. 5.9: Vývojový diagram funkce škálování Fine granular



Obr. 5.10: Celkový pohled na: a) originální snímek, b) hrubě nakvantovanou první vrstvu, c) rozšiřující vrstvu s redukováným počtem hladin, d) rekonstruovaný obraz

5.5 REALIZACE FUNKCE PRO VÝPIS HLADIN VE FGS

Funkce pro výpis hladin patří v této aplikaci zařazením do sekce Fine granular škálování, avšak její rozsah a množství výstupů je natolik velký, že jsem se rozhodl její popis umístit do vlastní kapitoly. Ač je funkce skryta pod poměrně nevelkým tlačítkem „Vypsát hladiny“, samotný zdrojový kód, jehož důležité části jsou uvedeny ve dvou následujících výpisech kódu, je poměrně obsáhlý a zaslouží si podrobnějšího popisu. Vývojový diagram celé této funkce je uveden na Obr. 5.11.

Stiskem tlačítka dochází k aktivaci funkce, viz Výpis kódu 5.4. V ní je nejprve ověřeno, zda jsou zadány vstupní parametry, kterými jsou název souboru a kvantovací koeficient první vrstvy. V případě že nejsou korektně zadány, je funkce ukončena. V případě splnění podmínek je načten zadaný vstupní soubor a je na něm po blocích 8 x 8 bodů provedena diskretní kosinová transformace. Následně je zobrazena původní podoba obrázku, v němž lze klepnutím myši zvolit určitý blok o velikosti 8 x 8 bodů pro další zpracování, viz Obr. 5.12.

Po klepnutí v oblasti obrázku jsou načteny souřadnice udávající polohu a z nich je určen počátek bloku tak, aby byla načtena matice o násobku 8 bodů vertikálně i horizontálně směrem od levého horního rohu, což je nutné pro korektní určení matice po DCT, tak aby byly získány údaje počínaje souřadnicí [0, 0], udávající DC koeficient bloku. Souřadnice, jež jsou násobkem osmi, jsou vypsané do hlavního okna Matlabu (Command Window), viz Obr. 5.16.

Následně jsou zobrazeny dvě okna. V prvním z nich jsou vykresleny hodnoty vzorků vybraného bloku před provedením DCT - Obr. 5.13, ve druhém pak koeficienty po provedení DCT, ještě před kvantováním - Obr. 5.14. Z porovnání obou je zřetelný význam i princip diskrétní kosinové transformace. Dále dochází ke kvantování vybraného bloku zadaným koeficientem první vrstvy. Rovněž jsou určena rezidua bloku jako rozdíl originálního a kvantovaného bloku. Pro názornost je opět blok reziduí vizuálně ztvárněn, viz Obr. 5.15.

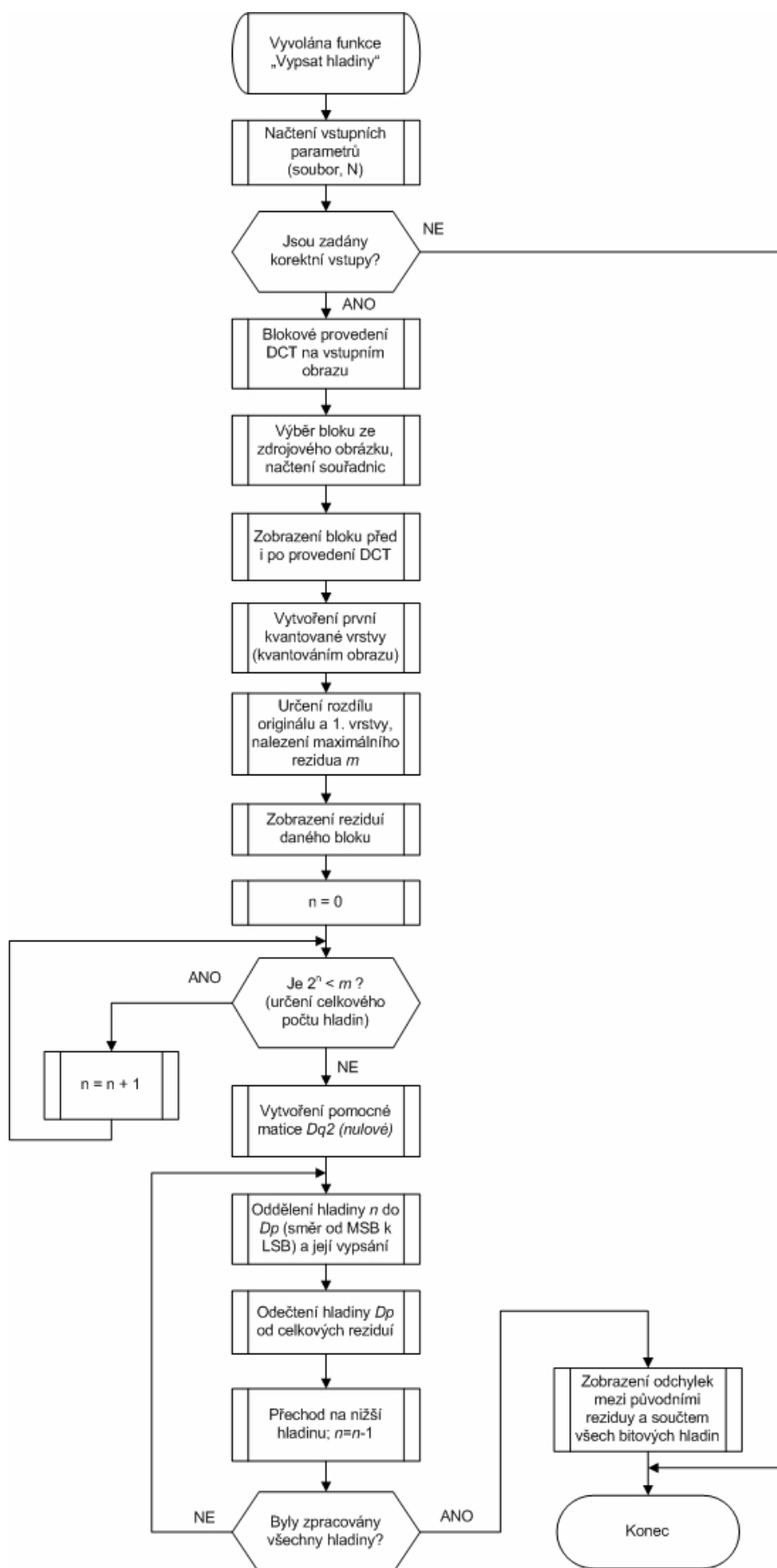
Výpis kódu 5.4: skalovani.m – funkce pro výpis hladin, první část

```
% vstupní parametry: g - obrázek, N - kvantovací koeficient

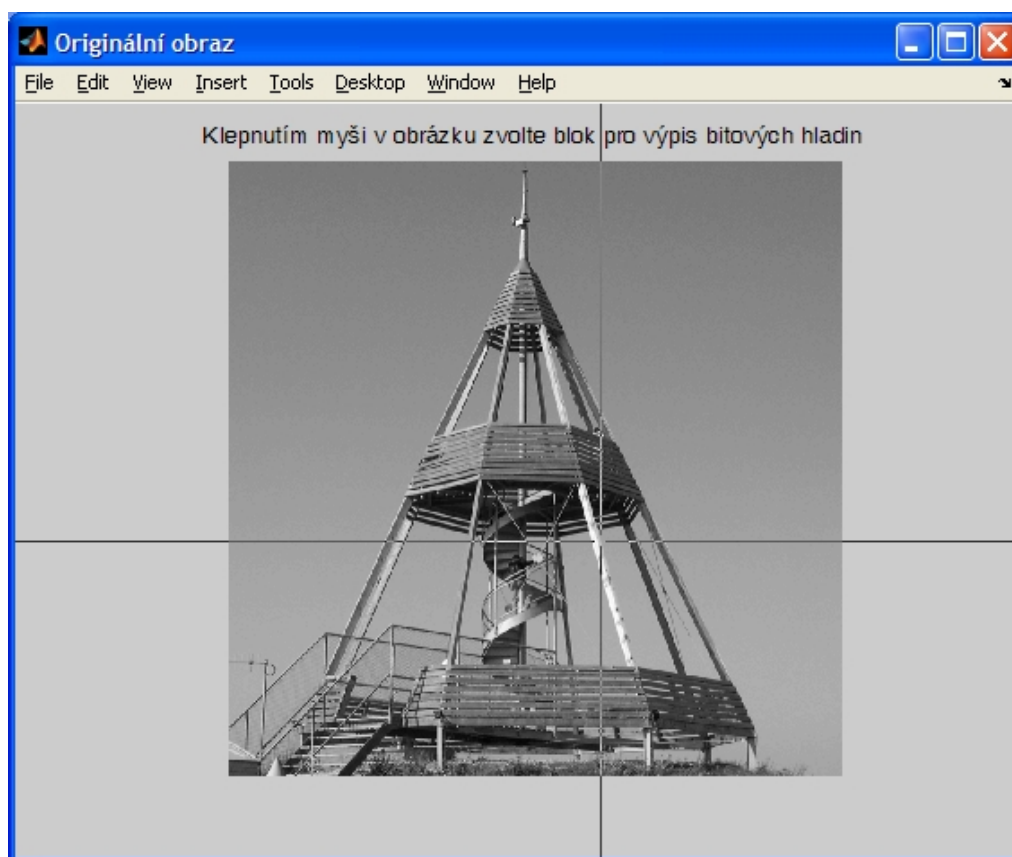
T = @dct2;
U = @idct2;
X = blkproc(g,[8 8],T); % provede DCT blokově
m = abs(max(max(X))); % zjistí maximální DC koef.

Xx = blkproc(X,[8 8],U); % provede IDCT blokově
figure('Name','Originální obraz (po započítání vlivu DCT)','NumberTitle','off','Toolbar','none','Tag','Figure10');
imagesc(Xx);
colormap('gray');
axis off;
axis image;
title('Klepnutím myši v obrázku zvolte blok pro výpis bitových hladin');

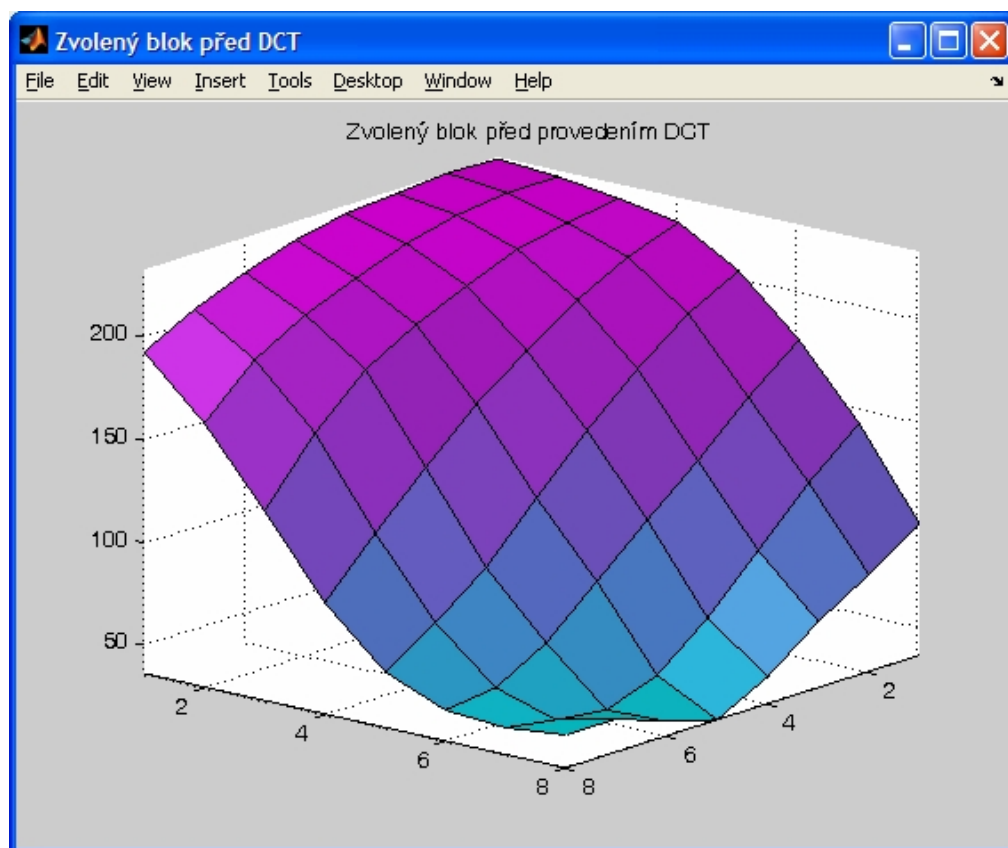
Souřadnice = ginput(1); % získat souřadnice bodu
Xosa=round(Souřadnice(1)); % zaokr. x-ové souřadnice na celé číslo
Yosa=round(Souřadnice(2)); % zaokr. y-ové souřadnice na celé číslo
Xroh=8*(fix(Xosa/8)); % zaoukr. výsledek směrem k nule a vynásobit 8x
Yroh=8*(fix(Yosa/8)); % získá se souřadnice levého horního rohu bloku
disp(' ')
disp(['Zvolený blok má počáteční souřadnice [x,y]: [' num2str(Xroh) ',
' num2str(Yroh) ']']);
Blok_pred_kvantovanim=X((Xroh+1):(Xroh+8), (Yroh+1):(Yroh+8)); %
načtení bloku 8x8 z DCT obrázku
Blok_pred_DCT=Xx((Xroh+1):(Xroh+8), (Yroh+1):(Yroh+8)); % načtení
bloku 8x8 z IDCT obrázku
Xq1 = quant(Blok_pred_kvantovanim,N); % vytvoření první kvantované
vrstvy
D = Blok_pred_kvantovanim - Xq1; % určení rozdílu originálu a první
vrstvy
```



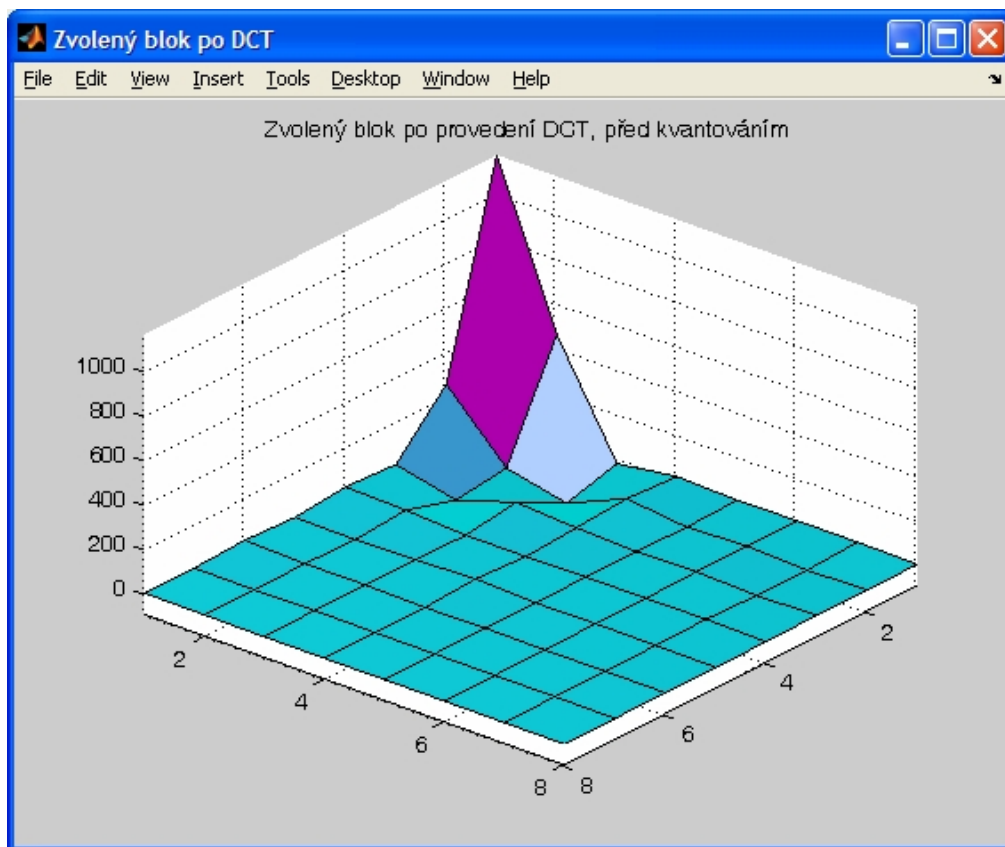
Obr. 5.11: Vývojový diagram pro výpis hladin ve FGS



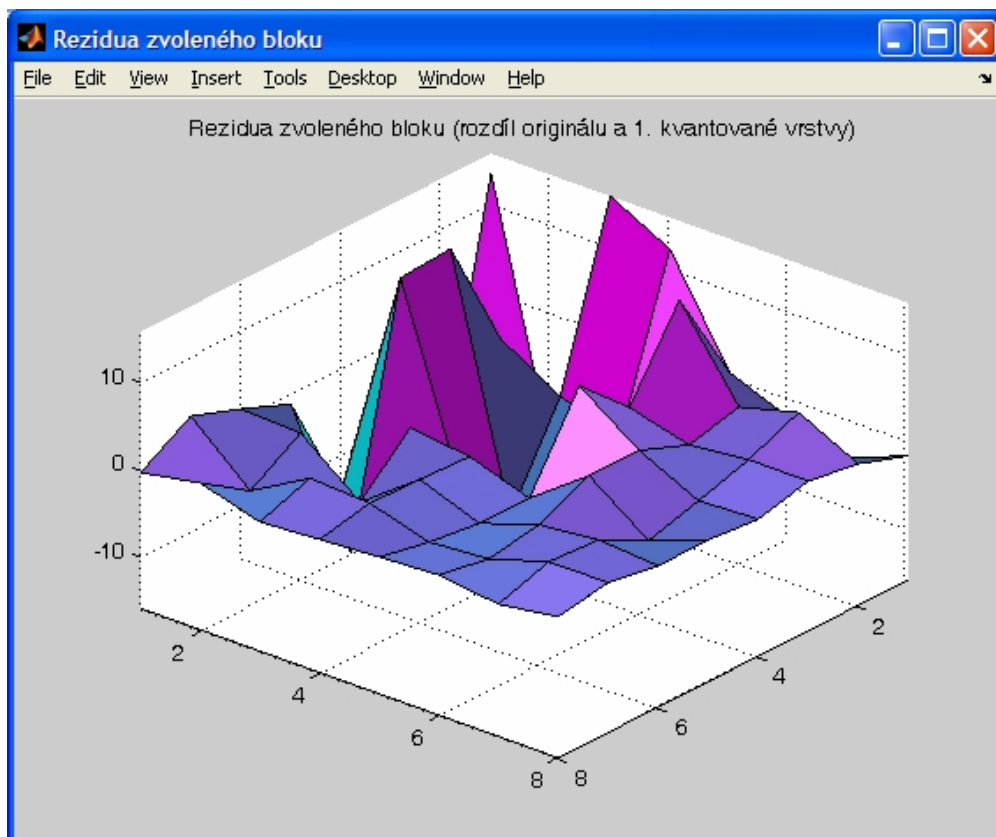
Obr. 5.12: Okno pro výběr bloku pro další zpracování



Obr. 5.13: Okno zobrazující vybraný blok před provedením DCT



Obr. 5.14: Okno zobrazující koeficienty vybraného bloku po provedení DCT



Obr. 5.15: Okno zobrazující rezidua zvoleného bloku, (kvant. koeficient 1. vrstvy = 35)

V další části, uvedené ve Výpis kódu 5.5, je určena maximální hodnota rezidua v bloku, a to z toho důvodu, aby bylo možné určit, kolik bitových hladin bude třeba pro úplné vyjádření reziduí. Počet bitových hladin je zjištěn pomocí cyklu, jenž postupně inkrementuje proměnnou n , vyjadřující počet bitů, až do okamžiku, kdy počet bitů n postačuje k vyjádření hodnoty největšího rezidua v bloku. Informace o celkovém potřebném počtu bitových hladin a současně prvních osm reziduí, čtených cik-cak od počátku, je vypsáno do hlavního okna Matlabu.

Výpis kódu 5.5: skalovani.m – funkce pro výpis hladin, druhá část

```
m = max(max(abs(D))); % určení maximálního rezidua
disp('-----');
disp(['Maximální absolutní hodnota rezidua v bloku: ' num2str(m)]);

% cyklus pro určení celkového počtu hladin, dokud 2^n < m (N/2)
n = 0;
while pow2(1,n) < m
    n = n + 1;
end

disp('-----');
disp(['Celkový počet potřebných bitových hladin: ' num2str(n)]);
disp(' ');

Dq2 = zeros(size(Blok_pred_kvantovanim)); % nulová matice
Dp2 = D; % rozdíl originálu a 1. vrstvy
disp('-----');
disp('Prvních osm reziduí (rozdíl mezi originálem a 1. kvant. vrstvou), čtené cik-cak:');
disp([ Dp2(1,1) Dp2(1,2) Dp2(2,1) Dp2(3,1) Dp2(2,2) Dp2(1,3) Dp2(1,4) Dp2(2,3)]);

% cyklus pro oddělení a zobrazení hladin 2. vrstvy a ponechání zbytku
s=n; % chci zobrazit všechny vrstvy
n=n-1;
for c = 1:s
    Dp = pow2(1,n)* fix(Dp2/pow2(1,n)); % rozdělení hladin 2^n, směrem od MSB
    disp('-----');
    disp(['Prvních osm bitů (vyjádřených hodnotou) bitové hladiny ' num2str(n) ' :']);
    disp([ Dp(1,1) Dp(1,2) Dp(2,1) Dp(3,1) Dp(2,2) Dp(1,3) Dp(1,4) Dp(2,3)]);
    Dq2 = Dq2 + Dp; % vytvoření 2. vrstvy z "s" hladin
    Dp2 = Dp2 - Dp; % od rozdílu orig. a 1. vrstvy odečte hladinu
    n = n - 1; % přechod na nižší hladinu, od MSB k LSB
end
disp('-----');
disp('Ochylky mezi původními reziduy a součtem všech bitových hladin:');
disp([Dp2(1,1) Dp2(1,2) Dp2(2,1) Dp2(3,1) Dp2(2,2) Dp2(1,3) Dp2(1,4) Dp2(2,3)]);
```

Následně dochází k opakovanému průběhu cyklu, jenž odděluje a zobrazuje jednotlivé bitové hladiny druhé vrstvy bloku. Pracuje směrem od MSB k LSB a hodnota bitů v hladině odpovídá jeho pořadí, tj. například v hladině číslo 3 jsou zpracovány rezidua s hodnotou nejméně 2^3 . Logický bit „1“ tak vyjadřuje hodnotu 8. Poslední, nejnižší hladina má číslo 0, tedy bit „1“ vyjadřuje hodnotu 2^0 . Každá takto oddělená hladina je zapsána do proměnné Dp a je s ní dále pracováno. Je tedy především odečtena od původního bloku reziduí, aby tak v matici $Dp2$ nakonec zůstaly pouze zanedbatelné difference s hodnotou menší než 1, které nebylo možné vyjádřit ani v nejnižší, nulté, bitové hladině. Ukázka výstupu ze zpracování náhodně zvoleného bloku je zobrazena na Obr. 5.16, který představuje výřez z hlavního okna Matlabu, v němž jsou tyto údaje vypsány.

```

Zvolený blok má počáteční souřadnice [x,y]: [16, 208]

-----

Maximální absolutní hodnota rezidua v bloku: 16.02

-----

Celkový počet potřebných bitových hladin: 5

-----

Prvních osm reziduí (rozdíl mezi originálem a 1. kvant. vrstvou), čtené cik-cak:
  13.3750  -16.0200  -15.3348   15.6537  -0.1651  -14.2853  -14.3387   12.9989

-----

Prvních osm bitů (vyjádřených hodnotou) bitové hladiny 4 :
    0   -16    0    0    0    0    0    0

-----

Prvních osm bitů (vyjádřených hodnotou) bitové hladiny 3 :
    8    0   -8    8    0   -8   -8    8

-----

Prvních osm bitů (vyjádřených hodnotou) bitové hladiny 2 :
    4    0   -4    4    0   -4   -4    4

-----

Prvních osm bitů (vyjádřených hodnotou) bitové hladiny 1 :
    0    0   -2    2    0   -2   -2    0

-----

Prvních osm bitů (vyjádřených hodnotou) bitové hladiny 0 :
    1    0   -1    1    0    0    0    0

-----

Odchylky mezi původními reziduy a součtem všech bitových hladin:
    0.3750   -0.0200   -0.3348    0.6537   -0.1651   -0.2853   -0.3387    0.9989

```

Obr. 5.16: Textový výstup aplikace ze zpracování náhodně vybraného bloku

6 STRUČNÝ POPIS OBSLUHY VÝUKOVÉ APLIKACE

Tato aplikace představuje uživateli různé druhy hierarchického kódování obrazů, tak jak jsou popsány ve standardech MPEG-2 a MPEG-4. Konkrétně se jedná o prostorové, kvalitativní a Fine granular škálování. Uživatel může u jednotlivých metod volit vstupní parametry a zásadně tak ovlivnit obsah jednotlivých vrstev, které jsou v obrazových tocích přenášeny. Tím dochází prakticky k ovlivnění kvality, bitové rychlosti, a tím i potřebné šířky pásma. Aplikace je napsána pro prostředí Matlab, odladěna byla ve verzi 7.4.0.287 (R2007a) a je obsažena v jediném souboru *skalovani.m*. V něm je definováno jak uživatelské rozhraní, tak i samotné výkonné funkce.

- Spuštění aplikace – v prostředí Matlab je možné zadat v hlavním okně název aplikace *skalovani.m*, případně tento soubor otevřít v editoru a spustit klávesou F5. Tím by mělo dojít k zobrazení grafického uživatelského rozhraní. Uživatelské rozhraní je rozděleno do několika sekcí. První z nich je určena pro zadání vstupního obrazového souboru a následující tři jsou určeny pro výběr a nastavení parametrů jednotlivých typů škálování. V dolní části se nachází tři tlačítka. Levé provede spuštění funkcí realizujících zobrazení vybraných metod škálování, a to v závislosti na tom, které zatrhávací políčko u jednotlivých metod bylo zvoleno. Tlačítkem „Info“ se zobrazí nápověda k použití aplikace. Stiskem tlačítka „Konec“ se program ukončí.

Hierarchické kódování obrazů

RGB soubor: C:\Documents and Settings\Administrator\Plocha\rozhledna.bmp Procházet

☒ **Prostorové** Podvzorkovací krok (horizontální i vertikální): 4

☒ **Kvalitativní** Kvantovací koeficient první vrstvy (max. 2047): 45 = hrubé kvantování
Kvantovací koeficient druhé vrstvy: 10 = jemné kvantování

☒ **Fine granular** Kvantovací koeficient první vrstvy: 35 » Vypsát hladiny
Počet zpracovaných hladin druhé vrstvy: 3

Zobrazit vybrané metody Info Konec

Obr. 6.1: Příklad nastavení parametrů aplikace

- Volba obrazového souboru – do pole určeného pro RGB soubor je možné vložit klávesovou zkratkou **CTRL+V** název souboru, jenž obsahuje data v podobě bitové mapy (*.bmp). Tento soubor může být rovněž vyhledán v libovolném umístění dostupném operačnímu systému, a to pomocí okna průzkumníku vyvolaného stiskem tlačítka „Procházet“. Z okna výběru souboru je možné se vrátit stiskem tlačítka „Storno“.
- Prostorové škálování – je zadáno podvzorkovacím krokem, jenž určuje poměr, v němž bude sníženo horizontální i vertikální rozlišení vstupního obrazu a tím bude získána první vrstva. Z rozdílu originálu a první vrstvy je vytvořena druhá vrstva. Dostupný rozsah podvzorkování je v aplikaci omezen od 1 do 8. Tuto hodnotu je možné zadat přímo do editačního pole, nebo ji zvolit pomocí šipek posuvníku. Pro zobrazení této metody musí být aktivováno přidružené zaškrtačkové políčko.
- Kvalitativní škálování – je určeno dvěma kvantovacími koeficienty. První z nich určuje tzv. hrubé kvantování první vrstvy a jeho velikost má vzhledem k hodnotám, kterých mohou nabývat DCT koeficienty, smysl pouze do 2047. Druhý kvantovací koeficient určuje jemně kvantovanou druhou vrstvu a jeho hodnota má smysl, pouze pokud je menší než první koeficient. Oba tyto koeficienty je nutné ručně uživatelsky zadat. V rekonstruovaném obraze i v jednotlivých zobrazených vrstvách je zpravidla patrný vliv DCT a kvantování, jehož následkem vznikají v obraze blokové artefakty.
- Fine granular škálování – je podobně jako kvalitativní škálování určeno kvantovacím koeficientem první vrstvy, jenž udává hodnotu koeficientu pro hrubé kvantování, z něž je získána první vrstva. Rezidua jsou pak určena jako rozdíl originálního obrázku a první kvantované vrstvy. Na základě počtu uživatelsky zadáných, zpracovaných hladin druhé vrstvy, je určen obsah druhé vrstvy. Do ní je začleněn daný počet bitových hladin, směrem od MSB k LSB. Hodnota nejvyšší hladiny i jejich celkový počet závisí na velikosti největšího rezidia, které je přímo úměrné kvantovacímu koeficientu. Rekonstrukce obrazu je pak určena jako součet první kvantované vrstvy a druhé vrstvy obsahující určitý počet bitových hladin. I zde jsou v jednotlivých obrazech rozeznatelné blokové artefakty vzniklé aplikováním diskrétní kosinové transformace a kvantováním.
- Vypsání hladin druhé vrstvy u FGS – jedná se o funkci, jenž zobrazí zadaný obrazový soubor a vyzve uživatele k volbě bloku o velikosti 8 x 8 bodů. Samotný výběr je proveden klepnutím myši. U zvoleného bloku je na základě zadaného koeficientu první vrstvy pro hrubé kvantování, provedeno nakvantování obrazu a následné určení reziduí. Ze všech reziduí bloku je určena jedna maximální hodnota. Podle počtu bitů nutných k vyjádření maximálního rezidia je určen počet hladin. Následně dochází k postupnému zobrazování jednotlivých bitových hladin v hlavním okně Matlabu pro prvních osm reziduí, čtených cik-cak ze zvoleného bloku, a to směrem od MSB k LSB. Po provedení oddělení všech hladin je určena též difference mezi původními reziduy a součtem všech bitových hladin. Pro větší názornost průběhu zpracování jsou rovněž graficky znázorněny okna se zvoleným blokem před provedením DCT, po provedení DCT (před kvantováním) a rezidia zvoleného bloku.
- Zobrazení vybraných metod – jedná se o tlačítko, pomocí něhož lze zobrazit metody škálování, u nichž bylo zatrženo zaškrtačkové políčko a současně byly splněny podmínky pro korektnost vstupních parametrů. Je třeba mít rovněž na paměti, že jednotlivé funkce jsou výkonově náročné a při zvolení velkého obrazového souboru může být prodleva zobrazení grafického výstupu funkce i v řádech sekund.

ZÁVĚR

V této bakalářské práci bylo mým cílem prostudovat metody škálování u ztrátové komprese videosignálu podle standardů MPEG-2 a MPEG-4. V úvodních kapitolách 1, 2 a 3 jsem se nejprve zabýval problematikou kódování a její realizace ve standardech MPEG-2 a MPEG-4. Je zřejmé, že ačkoliv oba standardy mezi sebou dělí nejen několik let vývoje, ale i odlišná úroveň vyspělosti, obsahují přesto řadu společných technik, jenž byly postupem času vylepšeny k co největší efektivitě komprese při zachování obrazové kvality. Podobnost je patrná i při porovnání blokových schémat kodérů a dekodérů obou standardů. Hlavní rozdíly jsou však realizovány uvnitř funkčních celků, kde došlo k největšímu pokroku. Ze studia problematiky rovněž vyplynulo, jak důležitou roli hraje v obou standardech definice profilů a úrovní. A to nejen z důvodu zajištění kompatibility mezi zařízeními různých výrobců, ale i pro popis a testování schopností jednotlivých zařízení.

Hlavním jádrem této práce jsou však metody hierarchického kódování standardů MPEG-2 a MPEG-4. Výhody použití škálování jsou nesporné, a to především z pohledu možnosti dosáhnout vhodného přizpůsobení videosignálu pro široké pole nejrůznějších koncových zařízení. Díky odstupňování do vrstev tak lze současně kódovat obraz pro nepříliš výkonná přenosná zařízení s malým množstvím paměti a zobrazovačem s nízkým rozlišením, stejně jako pro multimediální zařízení s vysokým rozlišením, podporující nejvyšší úroveň kvality zobrazení. Dekodér přijímače tak zpracovává pouze signál na úrovni, jenž je pro něj relevantní a nezabývá se datovým tokem, pro nějž není hardwarově ani softwarově přizpůsoben. Škálování má však řadu dalších výhod, jímž je věnována především kapitola 4. Je však vhodné poznamenat, že jeho pole působnosti není omezeno jen na pasivní příjem videosignálu, ale lze jej stejně dobře aplikovat u videotelefonů, videokonferencí a obecně snad u všech typů multimédií.

Praktickou částí této práce, popsanou v kapitolách 5 a 6, bylo vytvoření výukové aplikace v prostředí Matlab, s vlastním grafickým uživatelským rozhraním, jenž v sobě implementuje funkce provádějící prostorové, kvalitativní SNR a Fine granular škálování. Na základě uživatelsky zadaných, vstupních parametrů, provádí škálování obrazu a následně zobrazí výstupy jednotlivých vrstev po provedení škálování a po rekonstrukci obrazu. Vstupem programu, na němž je hierarchické kódování prováděno, je obrázek v podobě bitové mapy libovolných rozměrů, ve formátu RGB.

Aplikace byla napsána a odladěna v prostředí Matlab, ve verzi 7.4.0.287 (R2007a) a je obsažena v jediném souboru *skalovani.m*. Tento soubor v sobě obsahuje jak definici grafického uživatelského rozhraní, tak i jednotlivé výkonné funkce. Důvodem, proč jsem zvolil pro definici grafických objektů textový zápis a nikoliv binární soubor *.fig*, ačkoliv se jedná o pracnější způsob, byla moje snaha o snadno modifikovatelnou, přehlednou definici jednotlivých prvků a parametrů, kratší a optimálnější zdrojový kód a co nejmenší závislost na použité verzi prostředí Matlab. Domnívám se, že integrací všech součástí do jediného souboru nikterak neutrpěla přehlednost zdrojového kódu, ač se jedná o přibližně 560 řádků kódu, neboť jednotlivé funkce jsou vzájemně opticky odděleny a zdrojový kód je detailně okomentován. Dobré čitelnosti by měla napomoci i použitá programovací metoda Switched Board Programming, tzn. především za použití příkazů Switch a Case, které umožňují využití té vlastnosti funkcí, kterou je možnost volat sama sebe s různými parametry.

Po vyhotovení této práce, při zpětném hodnocení dosažených cílů, se domnívám, že zadání této bakalářské práce bylo splněno, neboť uvedená výuková aplikace plní všechny požadované funkce a podává očekávané výstupy. Navíc, nad rámec zadání, byla vypracována funkčnost pro výpis hladin u škálování typu Fine granular, jenž nabízí uživateli možnost interaktivně si zvolit blok ze zdrojového obrázku a detailně tak prozkoumat průběh jeho zpracování. Graficky tak lze sledovat hodnoty jednotlivých prostorových bodů v originálním

obrázku, dále pak koeficienty bloku po provedení diskrétní kosinové transformace, při které je blok 2D prostorových vzorků transformován na matici frekvenčních koeficientů (s velmi výrazným DC koeficientem). Rovněž tak i matici reziduí, která je nejen graficky znázorněna, ale navíc je přehledně do hlavního okna prostředí Matlab vypsán postup kódování reziduí do jednotlivých bitových hladin.

Osobně doufám, že případným zájemcům o hierarchické kódování, jímž se do rukou dostane tato práce, bude tato výuková aplikace dobrým pomocníkem při pochopení a praktickém vyzkoušení jednotlivých druhů škálování a že jim rovněž poskytne očekávanou míru variability a interaktivity v uživatelsky příjemném prostředí, s možností snadné volby parametrů kódování, a to vše navíc s okamžitým výstupem v grafické podobě.

Závěrem lze říci, že s rostoucím počtem koncových zařízení, schopných přijmout a zpracovat signál kódovaný dle standardu MPEG-2, či MPEG-4 a rovněž tak i díky dostupnosti stále rychlejších a cenově dostupnějších způsobů připojení – především pro mobilní zařízení, lze očekávat i stále vyšší oblibu využívání různých metod škálování. Zájem lidí o multimediální obsah určený pro zábavu, ale i pro seriózní obrazovou komunikaci je vysoký, a tak i hierarchické kódování bude mít do budoucna v přenosu obrazové informace jistě co říci.

POUŽITÁ LITERATURA

- [1] RICHARDSON, Iain E. G. *H.264 and MPEG-4 Video Compression*. John Wiley & Sons Ltd. 2003. ISBN 0-470-84837-5
- [2] HANZO L., CHERRIMAN P., STREIT J. *Video Compression and Communications, 2nd edition*. John Wiley & Sons Ltd. 2007. ISBN 978-0-470-51849-6
- [3] ČSN EN ISO/IEC 11172: *Informační technologie – Kódování pohyblivých obrazů včetně doprovodného zvuku pro číslicový záznam do rychlostí 1,5 Mbit/s*
- [4] ČSN EN ISO/IEC 13818: *Informační technologie – Obecné kódování pohyblivých obrazů a doprovodné zvukové informace*
- [5] ČSN EN ISO/IEC 14496: *Informační technologie – Kódování audiovizuálních objektů*
- [6] VÍT, V. *Televizní technika – přenosové barevné soustavy*. BEN – technická literatura. 1997. ISBN 80-86056-04-X
- [7] BOVIK, A. C. *Handbook of image and video processing, 2nd edition*. Elsevier Academic Press. 2005. ISBN: 0-12-119792-1
- [8] ČÍKA, P. *Multimediální služby*. Skriptum FEKT VUT, Brno, 2007 .
- [9] WIKIPEDIA : the free encyclopedia. *Kvantování* [online]. [cit. 2010-11-15]. Dostupné z WWW: [http://cs.wikipedia.org/wiki/Kvantov%C3%A1n%C3%AD_\(sign%C3%A1l\)](http://cs.wikipedia.org/wiki/Kvantov%C3%A1n%C3%AD_(sign%C3%A1l))
- [10] KOENEN, R. *Overview of the MPEG-4 Standard* [online]. 2002. [cit. 2010-11-22]. Dostupné z WWW: <http://mpeg.chiariglione.org/standards/mpeg-4/mpeg-4.htm>
- [11] Vcodex Limited [online]. 2010 [cit. 2010-11-20]. Dostupné z WWW: <http://www.vcodex.com>
- [12] MPEG-4 Industry Forum [online]. 2010 [cit. 2010-11-20]. Dostupné z WWW: <http://www.m4if.org>
- [13] ZAPLATÍLEK K., DOŇAR B. *MATLAB pro začátečníky*. BEN – technická literatura. 2003. ISBN: 80-7300-095-4
- [14] WIKIPEDIA : the free encyclopedia. *Charge-coupled device* [online]. [cit. 2010-11-17]. Dostupné z WWW: http://en.wikipedia.org/wiki/Charge-coupled_device
- [15] The MPEG Home Page [online]. 2010 [cit. 2010-11-19]. Dostupné z WWW: <http://mpeg.chiariglione.org>
- [16] WIKIPEDIA : the free encyclopedia. *MPEG-2* [online]. [cit. 2010-11-19]. Dostupné z WWW: <http://en.wikipedia.org/wiki/MPEG-2>
- [17] WIKIPEDIA : the free encyclopedia. *MPEG-4* [online]. [cit. 2010-11-19]. Dostupné z WWW: <http://en.wikipedia.org/wiki/MPEG-4>
- [18] OHM, J. *Introduction to MPEG-4 Video (rectangular)* [online]. 2005. [cit. 2010-11-24]. Dostupné z WWW: [http://mpeg.chiariglione.org/technologies/mpeg-4/mp04-vid\(r\)/index.htm](http://mpeg.chiariglione.org/technologies/mpeg-4/mp04-vid(r)/index.htm)
- [19] MathWorks - MATLAB and Simulink for Technical Computing [online]. 2010 [cit. 2010-11-23]. Dostupné z WWW: <http://www.mathworks.com>
- [20] ZAPLATÍLEK K., DOŇAR B. *MATLAB – tvorba uživatelských aplikací*. BEN – technická literatura. 2004. ISBN: 80-7300-133-0

SEZNAM ZKRATEK

AC	Alternating Current
BAB	Binary Alpha Blocks
BAP	Binary Alpha Plane
B-VOP	Bidirectionally predictive-coded Video Object Plane
CAE	Context-based Arithmetic Encoding
CCD	Charge Coupled Device
CCIR	Consultative Committee on International Radio
CCITT	Telegraph and Telephone Consultative Committee
CD-ROM	Compact Disc Read-only Memory
CIF	Common Interchange Format
DC	Direct Current
DCT	Discrete Cosine Transform
DPCM	Differential Pulse-code Modulation
DVD	Digital Versatile Disc
DWT	Discrete Wavelet Transform
FGS	Fine Granularity Scalability
HBV	High Bitrate Video
HD	High Definition
HVS	Human Visual System
IDCT	Inverse Discrete Cosine Transform
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
ITU	International Telecommunication Union
I-VOP	Intra-coded Video Object Plane
JPEG	Joint Photographic Experts Group
LDTV	Low-definition Television
LSB	Least Significant Bit
MP3	MPEG-1 Audio Layer III
MPEG	Moving Picture Experts Group
MSB	Most Significant Bit
NTSC	National Television Systems Committee
PAL	Phase-Alternation Line
PCM	Pulse-code modulation
P-VOP	Predictive-coded Video Object Plane
QCIF	Quarter Common Interchange Format
QSIF	Quarter Source Input Format
RGB	Red, Green, Blue
SDTV	Standard-definition Television
SECAM	Séquentiel Couleur avec Mémoire
SIF	Source Input Format
SNR	Signal to Noise Ratio
VHS	Video Home System
VLBV	Very Low Bitrate Video
VLC	Variable-length Coding
VLD	Variable-length Decoding
VO	Video Object
VOP	Video Object Plane

SEZNAM PŘÍLOH

A CD-ROM médium

- A.1 Výuková aplikace pro realizaci metod škálování (skalovani.m)
- A.2 Tento dokument ve formátu PDF (BP_Spacek_119338.pdf)
- A.3 Tento dokument ve formátu DOC (BP_Spacek_119338.doc)
- A.4 Adresář s ukázkovými obrazovými soubory (\obrazky)