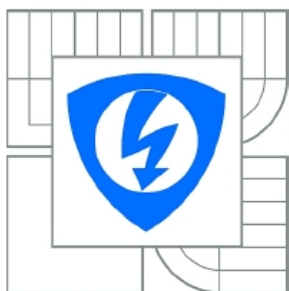


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNologiÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

POROVNÁNÍ MOŽNOSTÍ KOMPRES MULTIMEDIÁLNÍCH SIGNÁLŮ

COMPARISON OF MULTIMEDIA SIGNAL COMPRESSION POSSIBILITIES

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

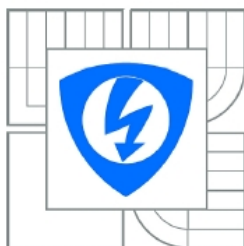
Bc. MILAN ŠPAČEK

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. VLADISLAV ŠKORPIL, CSc.

BRNO 2013



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student: Bc. Milan Špaček

ID: 119338

Ročník: 2

Akademický rok: 2012/2013

NÁZEV TÉMATU:

Porovnání možností komprese multimediálních signálů

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s možnostmi komprese multimediálních signálů se zaměřením na videosignál a na užívané kodeky. Zaměřte se zejména na techniky popsané ve standardech MPEG. V prostředí MATLAB vytvořte funkce simulující činnost funkčních bloků kodéru a dekodéru. Vytvořené funkce implementujte do grafického rozhraní. Vstupem budou obrázky ve formátu RGB, výstupem pak informace z jednotlivých funkčních bloků v kodéru i dekodéru. Při návrhu kodéru uvažujte především bloky úvodního zpracování vstupních snímků (podvzorkování), dále pak DCT, kvantování a kompenzaci pohybu (určení vektorů pohybu, rozdílových snímků). Zhodnoťte vliv jednotlivých technik a algoritmů na výstupní signál.

DOPORUČENÁ LITERATURA:

- [1] PUŽMANOVÁ, R. Moderní komunikační sítě A-Z. Computer Press, Brno 2007.
- [2] KUROSE, J., KEITH, R. Computer Networking. Addison-Wesley, New York 2010
- [3] ŠKORPIL, V. Přístupové a transportní sítě. FEKT, Brno 2012.

Termín zadání: 11.2.2013

Termín odevzdání: 29.5.2013

Vedoucí práce: doc. Ing. Vladislav Škorpil, CSc.

Konzultanti diplomové práce:

prof. Ing. Kamil Vrba, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové 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.

Výzkum popsáný v této diplomové práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Abstrakt

Diplomová práce se zabývá porovnáním možností komprese multimediálních signálů se zaměřením na videosignál a užívané kodeky. Detailně popisuje především kódování a dekódování videosignálu dle standardů MPEG.

V teoretické části jsou uvedeny charakteristické vlastnosti obrazového signálu a zdůvodnění potřeby použití komprese pro jeho záznam a přenos. Dále jsou vysvětleny metody s jejichž pomocí je odstraněna nadbytečnost (redundance) a zbytečnost (irrelevance) z kódovaného obrazového signálu. Rovněž jsou diskutovány způsoby měření kvality obrazového signálu. Samostatná kapitola je věnována charakteristickým rysům v současnosti používaných a perspektivních kodeků.

V praktické části práce byly vytvořeny funkce v prostředí Matlab, implementované do grafického uživatelského rozhraní, jenž simulují činnost funkčních bloků kodéru a dekodéru. Na základě uživatelsky zadanych vstupních parametrů provádí kódování a dekódování obrazu složeného ze snímků ve formátu RGB a následně zobrazí grafické i numerické výstupy z jednotlivých funkčních bloků. Implementovány jsou algoritmy pro úvodní zpracování vstupní sekvence snímků včetně podvzorkování, dále též DCT, kvantování, kompenzace pohybu a k nim inverzní operace.

Samostatné kapitoly jsou věnovány popisu realizace kodeku v prostředí Matlab a výstupům z jednotlivých kroků zpracování. Dále jsou též diskutovány srovnání kompresních algoritmů a vliv změny parametrů na výstupní podobu signálu. V závěru práce jsou shrnuty zjištěné poznatky.

Klíčová slova

Kódování, kodek, kompresní algoritmus, predikce, vektor pohybu, diskrétní kosinová transformace, DCT, kvantování, MPEG-2, MPEG-4, kodér, dekodér.

Abstract

Thesis deals with multimedia signal comparison of compression options focused on video and advanced codecs. Specifically it describes the encoding and decoding of video recordings according to the MPEG standard.

The theoretical part of the thesis describes characteristic properties of the video signal and justification for the need to use recording and transmission compression. There are also described methods for elimination of encoded video signal redundancy and irrelevance. Further on are discussed ways of measuring the video signal quality. A separate chapter is focused on the characteristics of currently used and promising codecs.

In the practical part of the thesis were created functions in Matlab environment. These functions were implemented into graphic user interface that simulates the activity of functional blocks of the encoder and decoder. Based on user-specified input parameters it performs encoding and decoding of any given picture, composed of images in RGB format, and displays the outputs of individual functional blocks. There are implemented algorithms for the initial processing of the input sequence including sub-sampling, as well as DCT, quantization, motion compensation and their inverse operations.

Separate chapters are dedicated to the realisation of codec description in the Matlab environment and to the individual processing steps output. Further on are mentioned compress algorithm comparisons and the impact of parameter change onto the final signal. The findings are summarized in conclusion.

Keywords

Coding, codec, compression algorithm, prediction, motion vector, discrete cosine transform, DCT, quantisation, MPEG-2, MPEG-4, encoder, decoder.

Prohlášení

Prohlašuji, že svou diplomovou práci na téma „Porovnání možností komprese multimediálních signálů“ jsem vypracoval samostatně pod vedením vedoucího diplomové 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 diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následku porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne 29. května 2013

.....

podpis autora

Poděkování

Děkuji vedoucímu práce doc. Ing. Vladislavu Škorpilovi, CSc. za velmi užitečnou metodickou pomoc a cenné rady při zpracování diplomové práce.

V Brně dne 29. května 2013

.....

podpis autora

ŠPAČEK, M. *Porovnání možností komprese multimediálních signálů*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2013. 89 s. Vedoucí diplomové práce doc. Ing. Vladislav Škorpil, CSc..

OBSAH

Úvod.....	8
1 Multimediální signály.....	9
1.1 Charakteristické vlastnosti videa.....	9
1.2 HVS a barevné modely pro obrazový signál	10
1.2.1 Model RGB	11
1.2.2 Model YUV	12
1.2.3 Model YCbCr	12
2 Důvody a principy odstranění redundance a irelevance.....	13
2.1 Metody ztrátového a bezztrátového kódování.....	14
2.1.1 Podvzorkování obrazu a jeho složek	14
2.1.2 Transformační kódování	15
2.1.3 Diskrétní kosinová transformace (DCT).....	16
2.1.4 Diskrétní waveletová transformace (DWT)	17
2.1.5 Kvantování	18
2.1.6 Snímání frekvenčních koeficientů	19
2.1.7 Entropické kódování	20
2.1.8 Prediktivní kódování.....	21
2.1.8.1 Kompenzace pohybu.....	21
2.1.8.2 Vyhledávání vektorů pohybu.....	22
2.1.9 Úplné prohledávací algoritmy	23
2.1.9.1 Algoritmus Full Search	23
2.1.10 Rychlé prohledávací algoritmy.....	24
2.1.10.1 Algoritmus Three Step Search.....	24
2.1.10.2 Algoritmus Nearest Neighbours Search.....	25
2.1.11 Intra predikce.....	26
2.2 Způsoby hodnocení kvality obrazového signálu.....	26
3 Kompresní standardy a kodeky.....	28
3.1 MPEG-1	29
3.2 MPEG-2	30
3.2.1 Profily a úrovně	31
3.2.2 Funkce kodéru	32
3.2.3 Funkce dekodéru.....	33
3.3 MPEG-4	34
3.3.1 Profily standardu MPEG-4.....	35
3.3.2 Kódování obrazu.....	36
3.3.3 Intra predikce.....	38
3.3.4 Inter predikce.....	40
3.3.5 Rekonstrukční (deblocking) filtr	41
3.3.6 Transformace a kvantování	42
3.3.6.1 Kódování 4x4 bloků reziduí	43
3.3.6.2 Kvantování	44
3.3.6.3 Kódování DC koeficientů jasových složek v Intra módu	45
3.3.7 Entropické kódování	46
3.4 H.261	47
3.5 H.263	47
3.6 H.264	47

4	Realizace vybraného kodeku v Matlab	48
4.1	Popis funkce kodéru	48
4.2	Popis funkce dekodéru.....	49
4.3	Popis realizace aplikace a jejích parametrů	50
4.3.1	Vstupní podvzorkování.....	51
4.3.2	Kódování snímku typu I metodou Intra	53
4.3.3	DCT, Celočíselná DCT	54
4.3.4	Hadamardova transformace DC koeficientů	56
4.3.5	Kvantování	57
4.3.6	Určení vektorů pohybu metodou Full Search a vytvoření pohybově kompenzovaného snímku	59
4.3.7	Určení vektorů pohybu metodou Three Step Search a vytvoření pohybově kompenzovaného snímku	60
4.3.8	Určení vektorů pohybu metodou Nearest Neighbours Search a vytvoření pohybově kompenzovaného snímku	62
4.3.9	Subpixelová přesnost určování vektorů pohybu.....	63
5	Zhodnocení výstupů realizovaných funkcí.....	65
5.1	Očekávané výstupy z algoritmů kodeku.....	65
5.2	Vstupní parametry pro vyhodnocení	67
5.3	Vstupní podvzorkování a proměnlivost obrazu	67
5.4	Vliv rychlosti snímání.....	70
5.5	Intra predikce	72
5.6	Inter predikce	73
5.7	Rozsah hledání shody bloku	74
5.8	Přesnost vektorů pohybu.....	75
5.9	Vliv kvantování	77
5.10	Transformace.....	81
	Závěr	83
	Použitá literatura.....	85
	Seznam zkratk.....	87
	Seznam příloh	89

ÚVOD

Ať se v současné době zaměříme na takřka jakoukoliv oblast lidské činnosti, zpravidla vždy narazíme na využití multimediálních signálů. Nemusí se však nutně jednat pouze o pasivní „konzumaci“ již vytvořeného obsahu pro zábavní účely, ale běžně též o multimediální komunikaci mezi dvěma, nebo více účastníky s určitou mírou interaktivity.

Za jednoho z prvních a současně nejvýznamnějších představitelů v oblasti přenosu multimediální informace lze bezpochyby považovat televizní vysílání, které přineslo obraz i zvuk do většiny domácností. Naštěstí však nezůstalo pouze u něj a tak se od pouhého příjmu informací a zábavy vyvinulo nejen v oblasti telekomunikací mnoho služeb, jenž nabízí přenos multimédií mezi uživateli. Kupříkladu to mohou být ve firemní sféře videokonference mezi vzdálenými účastníky, firemní prezentace, podpora prodeje a další. V akademické a vědecké sféře pak sledování přednášek, aktivní účast na diskuzích a konferencích, či možnost využívání datových multimediálních archivů a podobně.

Mezi oblíbenou cílovou skupinu výrobců patří domácnosti a zejména mládež. Ta poskytuje velký potenciál pro poptávku a velký trh na němž se mohou uplatnit výrobky spotřební elektroniky. Oblíbené je vše, co je multimediální, interaktivní, zábavné. Může to být komunikace mezi rodinou a přáteli, hraní her, sledování sdíleného obsahu, i jeho vlastní tvorba. Možnosti jsou takřka nevyčerpatelné.

Aby však výrobek nebo služba byla komerčně úspěšná, musí splňovat několik důležitých požadavků. Minimálně by měla být cenově dostupná a měla by nabídnout dostatečně vysokou kvalitu a množství služeb. V oblasti multimediálního obsahu však při snaze o maximální věrnost obrazu a zvuku narazíme na překážku, kterou je množství informací, které je nutné k dosažení požadované kvality přenést, či zaznamenat. Realistický zvuk a obraz s vysokým rozlišením vyžadují pro nekomprimovaný přenos datový tok v řádu desítek MB/s, což je i v dnešní době nelehké zajistit, a to jak při záznamu, tak především při přenosu [1]. Potřeba požadované šířky pásma by tak byla neúměrná a finančně nákladná, vzhledem k faktu, že snahou každého výrobce je co největší rozšíření produktu. Bylo tak nutné nalézt ekonomicky přijatelné řešení, které by nabídlo uživatelsky atraktivní službu s dostupnými náklady na pořízení a následný provoz.

Řešením dané situace je využití kódování, jenž dokáže snížit nadbytečnost (redundanci) a zbytečnost (irelevanci) ve vyjádření obrazové informace. Díky různým metodám, popsáným též v této práci, lze za pomoci ztrátové i bezztrátové komprese redukovat objem informací, které jsou lidským zrakovým systémem (HVS) nepostřehnutelné, a tím zvýšit efektivitu využití přenosových prostředků, při zachování dobrého subjektivního dojmu z obrazu. K zajištění výše uvedeného bylo vytvořeno několik skupin odborníků, kteří se danou problematikou zabývali. Výsledkem jejich práce pak byl vznik standardů, které definují funkci kodérů a dekodérů a rovněž i bitový tok, který je kodéry produkován. Mezi nejvýznamnější patří bezpochyby standardy skupiny MPEG, jimiž se zabývá i tato práce.

Úkolem této diplomové práce je seznámit čtenáře s možnostmi komprese videosignálu a s perspektivními kodeky, se zaměřením na standardy MPEG. Dále pak v prostředí Matlab vytvořit funkce, implementované do grafického rozhraní, které dle zadaných parametrů simulují jednotlivé funkční bloky kodéru a dekodéru. V neposlední řadě též zhodnotit vliv jednotlivých technik a algoritmů na výstupní signál.

V prvních třech kapitolách se práce zabývá charakteristickými vlastnostmi videosignálu a důvody a principy odstranění redundance a irelevance. Rovněž jsou popsány kompresní standardy a kodeky a způsoby posuzování kvality obrazového signálu. Čtvrtá kapitola je věnována popisu realizace vybraného kodeku v prostředí Matlab, včetně získaných výstupů z jednotlivých funkčních bloků. Zhodnocení výstupů realizovaných funkcí, včetně vlivu parametrů na výstupní signál, je diskutováno v páté kapitole.

1 MULTIMEDIÁLNÍ SIGNÁLY

Při zvažování obsahu pojmu „multimédia“ lze vycházet z mnoha různých definic, které se snaží uchopit větší či menší šíři záběru, jenž je tímto pojmem dotčena. Například dle doslovného výkladu uvedeného v [6], jenž uvádí, že pod slovem multimédia se skrývá „systém obsahující textové, obrazové, zvukové a datové informace, který se obvykle používá ve spolupráci s počítačem“ lze jen těžko odhadnout šíři všech možných praktických realizací, v nichž se lze s multimediálními signály v současné době setkat.

Multimediální signály lze považovat za zásadní část multimediálního systému, který může být tvořen časově spojitými médii, pod nimiž si lze představit především audio a video, ale i médii diskrétního typu, například text, grafika, obrázky, data. Multimediální systém je tak funkčně především charakterizován sjednocením tvorby, zpracování, ukládání, reprezentace a přenosu několika na čas závislých a nezávislých toků médií [11]. Média závislá na čase poskytují informaci, jenž je souvislá v čase a která musí být předána uživateli v daných časových bodech. To představuje potřebu zajištění spojitého datového toku s obměnou dat v pravidelných intervalech [11]. Při současném přenosu několika typů signálů (datových toků) je rovněž klíčovou otázkou vzájemná časová synchronizace při výsledné reprezentaci, věrně odpovídající původní předloze.

S postupem času je patrný růst poptávky po multimédiích s „přidanou hodnotou“, jenž kromě již tradičního audio a video toku obsahují další typy dat, díky nimž je kupříkladu zajištěn popis přenášené informace, či je poskytnutá určitá míra interaktivity, s jejichž pomocí jsou vyřízeny případné uživatelské požadavky v reálném čase [2].

V této práci je pojednáno o multimediálních signálech s výhradním zaměřením na videosignál a na užívané kodeky. Specifické vlastnosti videa i možnosti jeho zpracování jsou obsahem dalších kapitol.

1.1 CHARAKTERISTICKÉ VLASTNOSTI VIDEO

Při popisu videosignálu je vhodné uvážit, jakými parametry jej lze popsat. Hlavními charakteristikami obecně jsou [4]:

- prostorové rozlišení (šířka x výška snímku [pixel])
- barevná hloubka (počet bitů na obrazový bod [bit/pixel])
- obrazová frekvence (počet snímků za sekundu [fps], [snímky/s])

Lze tedy říci, že videosignál je trojrozměrný (3D), neboť k rovině prostorové (2D) přidává i časovou osu. Samotný 2D prostorový obraz je vyjádřen horizontálním a vertikálním rozlišením, jímž je jednoznačně stanoven počet bodů v řádcích a sloupcích jednotlivých obrazových snímků. Každý standard dle svého zaměření definuje několik obrazových formátů, s nimiž pracuje. Mezi nejznámější patří zejména formát CIF (definovaný CCITT pro standard H.261), formát SIF (používaný v MPEG) a další formáty určené pro televizní vysílání. Nejčastěji používané formáty jsou shrnuty v Tab. 1.1. K nim je také uvedena informace, zda je formát primárně určen pro neprokládané *p* (progressive) či prokládané *i* (interlaced) řádkování. Při neprokládaném snímání obrazu jsou obrazové řádky snímány v pořadí tak, jak následují po sobě, což vyžaduje velkou šířku pásma pro zajištění plynulosti obrazu. Pro úspornější využití šířky pásma se v analogových systémech zavedlo prokládané snímání obrazu, díky kterému je možné vysílat s nižším snímkovým kmitočtem, avšak za cenu zhoršení reprodukce horizontálních hran, detailů a rychlých pohybů scény, více viz [5] a [12].

Tab. 1.1: Přehled základních obrazových formátů [5],[11]

Formát	Rozlišení		Řádkování	Datový tok pro 25 snímků/s [byty/s]	Použití, standard
	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	CCITT H.261 - videorekordéry, videokonference
4CIF	704	576	p	30 412 800	CCITT H.261 - SDTV, DVD-video
16CIF	1408	1152	p	121 651 200	CCITT H.261
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
HDTV (full HD)	1920	1080	i	155 520 000	HDTV

V tabulce je rovněž uveden potřebný datový tok nekomprimovaného signálu při běžném počtu snímků pro televizní vysílání ve formátu PAL (Phase-Alternation Line). Ten je určen součinem počtu obrazových bodů ve snímku, počtem bitů na jeden obrazový bod a snímkovou frekvencí. Jak z uvedených hodnot vyplývá, přenos nekomprimovaného videosignálu by byl nejen technicky velmi obtížný, ale i ekonomicky značně neefektivní. Pro přiblížení si lze uvést výpočet přenosové rychlosti nekomprimovaného videa v modelu RGB, a to při rozlišení 8 bitů na pixel [11]. Tento výpočet lze provést pomocí vztahu (1.1):

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

, kde: R – přenosová rychlost [bit/s]

p – počet bitů na jeden obrazový bod [bit/pixel]

N_{op} – počet obrazových prvků ve snímku [pixel]

f_s – snímková frekvence [snímků/s]

Bude-li uvažováno nekomprimované vysílání HDTV videa v modelu RGB při rozlišení obrazu 1280x720 pixelů, pak celková přenosová rychlost R bude rovna:

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

Takovýto datový tok, jehož hodnoty pro jednotlivá rozlišení jsou uvedena i v Tab. 1.1, je však pro přenos pomocí konvenčních přenosových médií nevhodný, neboť by vyžadoval velkou šířku pásma, což by bylo nejen neefektivní, ale i technicky obtížné. Z tohoto důvodu vzniklo v průběhu času několik standardů a nemalý počet kodeků, jenž si kladou za cíl snížit přenosovou rychlost za pomoci ztrátového a bezztrátového kódování. Těmito úpravám, pomocí nichž lze snížit redundanci a irelevanci obrazové informace, jsou věnovány následující kapitoly.

1.2 HVS A BAREVNÉ MODELY PRO OBRAZOVÝ SIGNÁL

Dříve, než budou popsány jednotlivé principy kódování obrazu, je vhodné se zamyslet, jakým způsobem vlastně popsat viditelný obraz, aby byl vhodný k následnému číslcovému zpracování [12]. Běžně je k pořízení pohyblivého obrazového záznamu využito kamery, jenž pomocí CCD snímače (Charge-Coupled Device) zachycuje náboj, který odpovídá

dopadajícímu světlu. S pomocí předřazených filtrů, jenž propouští pouze požadované RGB složky, či dichroických hranolů, jsou tak získány oddělené složky RGB. Tento náboj představuje obrazovou informaci, která je uložena do paměti, odkud je vyčítána po řádcích a dále zpracovávána. Kamery pro snímání obrazu však mohou být i čtyřdílné, přičemž kromě RGB signálu zpracovávají odděleně i informaci o jas, čímž se funkčně blíží lidskému oku [14].

Lidský zrakový systém HVS (Human Visual System) má určitá specifika, jenž je vhodné zohlednit při kódování i reprodukci videosignálu, neboť za běžných podmínek není nutné uchovávat informaci, kterou není pozorovatel schopen rozlišit.

Světlo v lidském oku dopadá na sítnici, jenž je pokryta dvěma druhy receptorů, pracujících jako obrazové snímače. Prvním z nich jsou tyčinky, které jsou citlivé především na jas, a jsou nezbytné pro vidění při nízkých hladinách osvětlení. To je dáno nejen tím, že tyčinek je v oku podstatně více než čípků (asi 120 milionů tyčinek proti 6 milionům čípků), ale i proto, že jsou na světlo mnohem citlivější než čípky. To se prakticky projevuje při vidění za šera, kdy sice vidíme, ale pouze černobíle [10].

Druhým receptorem jsou čípky, citlivé na jas i barvu, avšak na každou z určitých vlnových délek odlišně. Příčinou tohoto jevu je jejich nerovnoměrné zastoupení, jenž je dle [12] rozloženo následovně:

- zelené: 64%
- červené: 32%
- modré: 2%

Kvůli tomuto rozdělení má oko různou citlivost na jednotlivé barvy. Tohoto poznatku bylo využito při sestavování dále uvedených barevných modelů.

1.2.1 Model RGB

Jedná se o takzvaný aditivní barevný model složený ze všech tří barevných složek v plném prostorovém rozlišení. Je založen na principu vyzařování světla, tj. představa zcela zatemněné místnosti a reflektorů R, G, B, jejichž mícháním se získává světlejší výsledek. Při smíchání všech tří složek se získá bílé světlo. Dále je též znám rozšířený model RGBA, který doplňuje model o α -kanál, jenž definuje míru průhlednosti dané hodnoty RGB s případnou barvou na pozadí [12].



Obr. 1.1: Míchání barev v aditivním modelu RGB [12]

1.2.2 Model YUV

Jde o model využívaný především v televizní technice. Podobně jako model YCbCr má oddělenou jasovou složku Y (luminance) od barevných informací U , V (chrominance). Důvodem vzniku byl požadavek slučitelného přenosu se systémy černobílého televizního vysílání, které využívají pouze jasovou složku [10]. Jasová složka je u toho modelu v intervalu $\langle 0;1 \rangle$ a barevné v intervalu $\langle -0,5;0,5 \rangle$.

Konverze signálu vyjádřeného v RGB formátu do formátu YUV je definována následujícím vztahem [5]:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{pmatrix} 0,299 & 0,587 & 0,114 \\ -0,146 & -0,288 & 0,434 \\ 0,617 & -0,517 & -0,100 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad (1.2)$$

1.2.3 Model YCbCr

Tento model je zásadní při zpracování digitálního obrazu a videa. Je u něj oddělena jasová složka od chrominančních. Pro vyjádření všech tří barev postačuje znalost dvou složek. Chybějící složka Cg se získává výpočtem z obou známých. Chrominanční složky jsou díky znalosti HVS přenášeny zpravidla s nižším rozlišením než jasová složka Y . Hodnoty složek jsou vyjádřitelné v 8 bitovém tvaru [10].

Podobně jako u modelu YUV lze definovat složky ve formátu YCbCr, u nějž platí, že jasová složka se určuje stejně jako pro YUV v rovnici (1.2), pro ostatní složky pak platí [4]:

$$\begin{aligned} Cb &= B - Y \\ Cr &= R - Y \\ Cg &= G - Y \end{aligned} \quad (1.3)$$

Nebo též může být využit vztah pro konverzi z formátu RGB [13]:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{pmatrix} 0,299 & 0,587 & 0,114 \\ -0,169 & -0,331 & 0,500 \\ 0,500 & -0,419 & -0,081 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad (1.4)$$

2 DŮVODY A PRINCIPY ODSTRANĚNÍ REDUNDANCE A IRELEVANCE

Jak již bylo uvedeno v kapitole 1.1 a především ukázáno v Tab. 1.1, představuje nekomprimovaný datový tok takový objem dat, že by i při dnešních možnostech přenosových sítí a kapacitách paměťových médií bylo uchovávání a přenos takového množství dat nejen neefektivní, ale i neekonomické a především technicky velmi náročné [12]. Proto šly snahy o efektivnější vyjádření přenášených informací vždy ruku v ruce s rozvojem přenosu dat. Příkladem toho může být Morseova abeceda, používaná již od roku 1844, jenž pro nejčastěji užívané znaky využívá nejkratších kódových slov. Například písmeno „E“ je vyjádřeno tečkou, písmeno „T“ čárkou, písmeno „I“ dvěma tečkami, a podobně. Tak byla již v té době využita myšlenka bezztrátové komprimace dat použitím kódování s proměnnou délkou slova [18].

Více než kdy dřív, se v současnosti jeví jako nutnost každý obrazový signál určitými metodami komprimovat. Ač jsou k dispozici stále výkonnější zařízení, schopná komprese dle nových a účinnějších standardů, tak i přesto neustále každoročně dramaticky narůstá množství dat přenášených sítí Internet, jenž je z větší části tvořeno právě multimediálním obsahem [2]. Dle předpovědi se celkové množství dat přenášených po Internetu v porovnání s dneškem do roku 2016 zvýší na troj až čtyřnásobek. V oblasti mobilních dat však tento nárůst bude ještě významnější [17].

Před kompresí videesignálu je ale nutné zvážit, jakých cílů a za jakých podmínek má být dosaženo [19]:

- Zda je možné použít ztrátovou kompresi, či je požadováno aplikovat bezztrátovou
- Jaká přenosová rychlost sítě je dostupná a jaká je minimální garantovaná
- Jaká je minimální požadovaná kvalita obrazu
- Zda bude možné přistupovat do animace v průběhu, zda bude použit posuv zpět a střih
- Jaká může být složitost reálné realizace, s ohledem na výpočetní výkon použitého systému a velikost paměti

Z dalšího úhlu pohledu tak může komprese z videesignálu odstranit nadbytečnost neboli redundanci, tj. dokáže stejnou informaci vyjádřit efektivněji s pomocí menšího počtu bitů. Tato komprese je bezztrátová, z čehož vyplývá, že rekonstruovaná data se přesně shodují s daty, jenž vstoupily do kodéru. Nevýhodou je však nízká účinnost, jenž se například u formátu JPEG pohybuje dle [4] v poměru 1:3 až 1:4.

V případě ztrátové komprese je redukována informace, jenž je irelevantní, zbytečná, neboť lidský zrakový systém (HVS) ji zpravidla nedokáže postřehnout. Této znalosti bylo dosaženo rozsáhlým zkoumáním HVS, čímž byl vyhodnocen vliv odstranění subjektivní redundance na výsledný vizuální dojem pozorovatele [13].

Dle [12] lze kompresní algoritmy tedy rozdělit na tři základní typy:

- bezztrátová komprese (lossless) – jenž je nejčastěji založena na obecných principech datové komprese, např.: Huffmanovo kódování (použití symbolů různé délky), algoritmy RLE (kódování délkou sledu, vhodné pro opakující se hodnoty sousedních vzorků), komprese LZW (Lempel-Ziv-Welch) použitá ve formátu GIF a donedávna chráněná patenty (princip spočívá v nahrazení vzorků vstupních dat binárními kódy postupně rostoucí délky). Dále též Kvadrantový strom (vhodný pro

obrazy s velkými jednobarevnými plochami), či použití barevné palety (vhodné pro obrazy obsahující omezený počet barev, jenž je mnohem menší než počet věrných barev. Do obrazu se vloží barevná paleta a obraz je zakódován pouze indexy do této tabulky)

- ztrátová komprese (lossy) – která dosahuje vyššího kompresního poměru za cenu pozměnění barevné hodnoty pixelů a zhoršení výsledného obrazu. Existuje též několik metod, např.: transformační kódování (mapování z jedné domény – prostor, čas – do jiné – DCT, DWT), kvantování, redukce barevných složek (podvzorkování ve vertikálním či horizontálním směru), predikční kódování, hierarchické kódování
- zvláštní případy komprese – kupříkladu fraktální komprese, která patří mezi nesymetrické metody; hledá se opakující se vzor v obraze

2.1 METODY ZTRÁTOVÉHO A BEZEZTRÁTOVÉHO KÓDOVÁNÍ

Při realizaci ztrátového a bezetrátového kódování byly u jednotlivých standardů použity některé, nebo případně všechny z níže uvedených metod. Každý standard, či jeho vývojová verze, však specifickým způsobem pozměnila tyto obecné metody. Postupem času totiž docházelo k vývoji stále dokonalejších a úspornějších algoritmů, a to i v návaznosti na růst výkonu použitých kódovacích a dekódovacích systémů. Dále tedy budou uvedeny obecné metody kódování a k nim zmíněny různé varianty řešení.

2.1.1 Podvzorkování obrazu a jeho složek

Používá se zpravidla u barvosložek. Dochází k redukci počtu pixelů na řádek a sloupec, při dekódování následně dochází k interpolaci. Tomu odpovídá omezení spektra videosignálu a ztráta detailů [12]. Této metody se užívá na základě poznatku o lidském zrakovém systému HVS, jenž uvádí, že lidské oko je více citlivé na jas, než na barvy. Při podvzorkování bývá zpravidla zachována původní velikost jasové matice Y, ale matice chrominancních složek jsou podvzorkovány v určitém poměru. U videosignálu to bývá nejčastěji vzorkování 4:2:0, jenž zachovává jasovou matici, ale obě barvosložkové matice podvzorkovává na poloviční horizontální i vertikální rozlišení [4]. Tato i další používané metody jsou uvedeny v Tab. 2.1 [10], [12].

Tab. 2.1: Způsoby podvzorkování barvosložek [10], [12]

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

Jak je patrné, je tato metoda ztrátová, neboť dochází k zásadní redukci informace a tím i potřebného datového toku. Dle použitého typu podvzorkování tak může datový tok pro barvosložky klesnout na polovinu, nebo až čtvrtinu původního datového objemu. To si

lze prakticky ukázat v následujícím příkladu, na němž je patrné výrazné snížení potřebného datového toku. Při zvažovaném vzorkování 4:2:0, rozlišení obrazu 720x576 pixelů a reprezentaci pomocí 8 bitů tak datový tok bude následující [11]:

Jasová složka:

$$R = p \cdot N_{op} \cdot f_s = 8 \cdot 720 \cdot 576 \cdot 25 = 82,944 \text{ Mbit/s} = 10,368 \text{ MB/s} \quad (2.1)$$

Chrominanční složka:

$$R = p \cdot N_{op} \cdot f_s = 8 \cdot 360 \cdot 288 \cdot 25 = 20,736 \text{ Mbit/s} = 2,592 \text{ MB/s} \quad (2.2)$$

Potřebná přenosová rychlost videosignálu při uvedeném vzorkování 4:2:0 bude tedy celkem rovna:

$$R = 82,944 + 2 \cdot 20,736 = 124,416 \text{ Mbit/s} = 15,552 \text{ MB/s} \quad (2.3)$$

Jak je z uvedeného zřejmé, snížení datového toku pro chrominanční složky je zásadní, lze však předpokládat, že subjektivní dojem z videa bude i přesto dobrý. Stále jsou však uvedené hodnoty příliš vysoké a je tedy nezbytné použít dalších kompresních metod.

2.1.2 Transformační kódování

Cílem transformačního kódování je převod obrazu z jedné oblasti (zpravidla prostorové nebo časové) do jiné oblasti, čímž lze získat vyjádření ve tvaru, jenž je výhodnější pro následné zpracování, zejména z důvodu umožnění lepší komprese, odstranění redundance pomocí entropického kódování, apod. [12],[19].

Pro dvourozměrný obraz lze uvažovat především následující typy transformací [19]:

Diskrétní Fourierova transformace (DFT)

Diskrétní kosinová transformace (DCT)

Diskrétní sinová transformace

Diskrétní Hadamardova transformace

Transformace Karhunen-Loeve (KLT)

U transformací je nezbytným požadavkem existence inverzní transformace, to znamená, aby transformovaný a zpracovaný obraz bylo možné opět převést do původní prostorové (nebo časové) oblasti. Při transformacích se obraz považuje za signál a jako takový je aproximován součtem tzv. váhových bázeových funkcí. U jednotlivých bázeových funkcí se tyto váhy nazývají koeficienty transformace. Bázeové funkce jsou zpravidla předem dány a nezávisí na zpracovávaném obrazu. Avšak transformace Karhunen-Loeve (KLT) je představitelem obrazově (signálově) závislé transformace. Pro stanovení vlastní transformační matice je nejprve třeba určit vlastní čísla a vlastní vektory kovarianční matice obrazové matice. Tato vlastní transformační matice se musí přenést i do dekodéru [19].

Významným pojmem v transformačním kódování je separabilnost transformace. Ta říká, že 2D transformace je separabilní, pokud je možné ji zapsat jako složení dvou po sobě jdoucích transformací, z nichž každá je pouze funkcí jedné proměnné [12]:

$$T(x, y) = F(u, G(v)), \text{ resp. } T(x, y) = F(G(u), v) \quad (2.4)$$

Toto vyjádření tedy odpovídá transformaci signálu obrazové matice nejprve po řádcích a následně po sloupcích, respektive obráceně [12]. Separabilnost lze využít u transformace aplikované nejprve na řádky a poté na sloupce obrazové matice, což umožňuje využití jednorozměrné DCT (1D-DCT) a tím i možnost rychlého paralelního zpracování. Zpracování se provádí na blocích určité velikosti (např. 8x8 pixelů), jenž vzniknou rozdělením původní matice [19].

Mezi nejznámější a zároveň nejpoužívanější patří diskretní kosinová transformace (DCT), mezi jejíž zásadní výhody patří transformace reálných čísel na reálná čísla a koncentrace energie obsažené v signálu do menšího počtu koeficientů (například v porovnání s DFT). DCT se využívá v současnosti v řadě standardů, namátkou JPEG, MPEG, H.26x, atd. DCT nepředstavuje přímo kompresi dat, neboť nedochází ke zmenšení množství informací (matice signálu i koeficientů jsou stejného rozměru), ale slouží spíše k přípravě signálu pro další zpracování [19]. Po DCT tak následuje ztrátová kvantizace a entropické kódování, viz Obr. 2.1, o nichž bude pojednáno dále.



Obr. 2.1: Transformační kodér založený na DCT [19]

2.1.3 Diskretní kosinová transformace (DCT)

Představuje operaci, během které je blok 2D prostorových vzorků transformován na matici frekvenčních koeficientů [11]. Při tom současně dochází k přeuspořádání hodnot v matici, směrem od nízkých kmitočtů k vyšším. Koeficient s největší energií se nachází v levém horním rohu na souřadnicích [0;0]. Hodnota koeficientu v tomto bodě se zpravidla značí jako *DC* koeficient, neboť jeho kmitočet v horizontálním i vertikálním směru je nulový a jeho hodnota představuje průměrnou hodnotu všech vzorků v bloku. Ostatní koeficienty v bloku jsou pak analogicky značeny jako *AC* koeficienty. Vlastní diskretní kosinová transformace je bezztrátová, neboť dochází pouze k odlišnému vyjádření obrazové informace. Ke ztrátové kompresi dochází až v následujícím bloku kvantizéru.

Diskretní kosinová transformace je obdobou diskretní Fourierovy transformace (DFT) a pro 2D matici o rozměrech $N \times N$ je definována následujícím způsobem [11]:

$$F(u, v) = \frac{2}{N} C(u) C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos \frac{(2x+1)u\pi}{2N} \cos \frac{(2y+1)v\pi}{2N} \quad (2.5)$$

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

$$C(u), C(v) = \begin{cases} \frac{1}{\sqrt{2}} & \text{pro } u, v = 0 \\ 1 & \text{ostatní} \end{cases} \quad (2.6)$$

Inverzní diskretní kosinová transformace (IDCT) je definována vztahem:

$$f(x, y) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u)C(v)F(u, v) \cos \frac{(2x+1)u\pi}{2N} \cos \frac{(2y+1)v\pi}{2N} \quad (2.7)$$

Vstupní informace zdrojové matice pro DCT i výstup z IDCT je vyjádřen 9 bity. Koeficienty jsou reprezentovány 12 bity [11]. Dynamický rozsah DCT koeficientů se proto pohybuje v rozsahu [-2048; +2047].

2.1.4 Diskretní waveletová transformace (DWT)

Pracuje na podobném principu jako fourierovské transformace, tj. princip rozkladu signálu do součtu základních prvků, avšak nejsou zde použity harmonické signály sin, cos, nýbrž wavelety – vlnky. K základním pojmům DWT patří dilatace (smrštění, roztažení, změna měřítka) a translace (posunutí). Každý waveletový koeficient signálu zastupuje jednu konkrétní variantu smrštění a posunutí základního waveletu a informuje, jaké množství je ho v signálu [12].

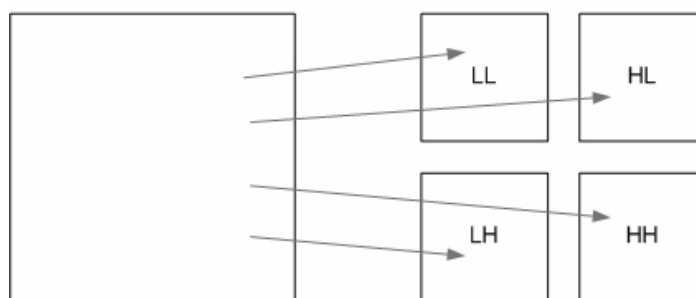
Wavelety jsou skupiny funkcí, které lze získat z funkce ψ tzv. mateřského waveletu pomocí posuvu a rozšíření. Dle [19] je popsána následujícím způsobem: nechť a je konstanta určující rozšíření a b konstanta určující posuv, pak pro daný mateřský wavelet vznikne skupina funkcí definovaná vztahem [19]:

$$\psi_{m,n} = a^{-m/2} (a^{-m}t - nb) \quad (2.8)$$

DWT rozdělí signál do pásem, na něž může být následně uplatněno další zpracování, například vektorová kvantizace. Výhodou této transformace je, že pomocí ní lze lépe zachytit jemné detaily a rovněž také to, že se zvyšujícím se počtem zpracovávaných vzorků, narůstá počet operací lineárně. V neposlední řadě se DWT s výhodou používá při kódování pro velmi nízké přenosové rychlosti. DWT během jednoho kroku dekompozice obrazu tedy prakticky probíhá následovně [12]:

1. filtrace obrázku po řádcích
2. decimace výsledku filtrace v řádcích
3. filtrace výsledku bodu 2 po sloupcích
4. decimace výsledku filtrace ve sloupcích

Uvedené body se provádí zvlášť pro waveletový filtr typu horní propust (H) a dolní propust (L), což tvoří celkem čtyři kombinace (LL, HL, LH, HH), viz Obr. 2.2. Takto jsou získány čtyři množiny koeficientů, které mají v porovnání s původním počtem pixelů pouze čtvrtinový počet pixelů. Získané koeficienty lze sestavit do čtverce a dále pokračovat rekurzivně pro požadovaný počet rekurzí (tím je určena tzv. hloubka dekompozice) Jednou z nejznámějších aplikací waveletové transformace je její použití v kompresním algoritmu JPEG2000 [12].



Obr. 2.2: Schéma jednoho kroku waveletové dekompozice obrazu [12]

2.1.5 Kvantování

Představuje proces, při kterém je určitému rozsahu hodnot přidělena jedna hodnota, reprezentovaná kvantizační hladinou, za účelem snížení potřebného počtu bitů pro interpretaci informace [10]. Jedná se o operaci, při níž dochází k nevratné ztrátě informace v závislosti na kvantizačním koeficientu. Důležitým pojmem spojeným s realizací kvantování je „kvantizační šum“, jehož hodnota vyjadřuje rozdíl mezi původní a nakvantovanou hodnotou [6]. Jeho velikost může nabývat hodnot rovným až polovině kvantizačního kroku. Změnou hodnot kvantizačních koeficientů v kodéru lze účinně regulovat výstupní bitový tok, neboť platí, že čím větší je vzdálenost kvantizačních hladin, tím účinnější je komprese, avšak za cenu nárůstu kvantizačního šumu. Kvantování se používá velmi často po provedení diskrétní kosinové transformace k odstranění nepodstatných vysokofrekvenčních hodnot, jenž se blíží k nule a nejsou pro výsledný dojem z hlediska znalostí o HVS příliš podstatné. Díky tomu pak lze cik-cak čtením získat dlouhé posloupnosti nul, jenž jsou ideální pro kódování RLE (Run-length Encoding) [19].

Při kvantování v kodérech standardů JPEG, MPEG a H.26x jsou jednotlivé prvky kvantizačních matic odlišné a i matice se mohou lišit pro Intra a Inter kódované snímky, pro jasovou a barvonosnou složku, apod. Hodnoty matic jsou sice definovány, lze je však optimalizovat pro požadovaný účel. Pak se liší i informační ztráta jednotlivých prvků matice DCT koeficientů, neboť zpravidla jsou koeficienty nižších frekvencí kvantovány s vyšší přesností, než koeficienty vyšších frekvencí [11]. Viz příklad kvantizačních tabulek [10] na Obr. 2.3.

Operaci kvantování lze matematicky vyjádřit následujícím vztahem [19]:

$$yq_{kl} = \text{round}\left(\frac{y_{kl}}{q_{kl}}\right) \quad (2.9)$$

, kde: yq_{kl} - jsou hodnoty koeficientů DCT po kvantování (zaokrouhlené na celé číslo)
 y_{kl} - jsou koeficienty DCT před kvantováním
 q_{kl} - jsou prvky kvantizační matice, určující kvantizační krok

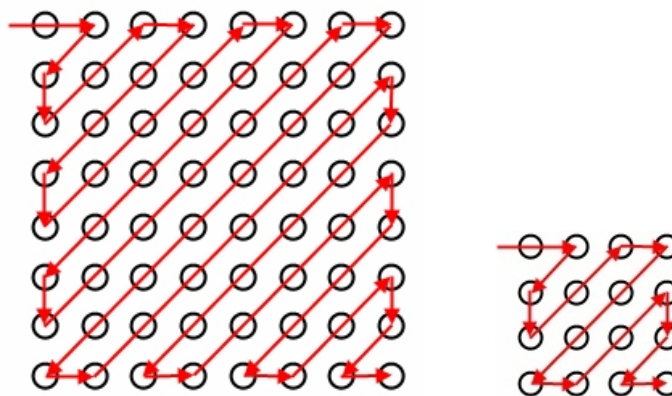
Intra (snímky I)	Non-intra (snímky P, B)
8 16 19 22 26 27 29 34	16 16 16 16 16 16 16 16
16 16 22 24 27 29 34 37	16 16 16 16 16 16 16 16
19 22 26 27 29 34 34 38	16 16 16 16 16 16 16 16
22 22 26 27 29 34 37 40	16 16 16 16 16 16 16 16
22 26 27 29 32 35 40 48	16 16 16 16 16 16 16 16
26 27 29 32 35 40 48 58	16 16 16 16 16 16 16 16
26 27 29 34 38 46 56 69	16 16 16 16 16 16 16 16
27 29 35 38 46 56 69 83	16 16 16 16 16 16 16 16

Obr. 2.3: Standardizované hodnoty kvantizačních tabulek u jasového signálu pro Intra a non-intra kódované snímky v soustavě MPEG-2 [10]

2.1.6 Snímání frekvenčních koeficientů

Jedná se o krok, jenž následuje po provedení kvantování. Hodnoty AC koeficientů jsou v pořadí od nejnižších kmitočtů snímány způsobem cik-cak a to směrem z levého horního rohu do pravého dolního rohu [4], viz Obr. 2.4 pro celosnímkový mód pro blok koeficientů 8×8 a 4×4 . V pulsnímkovém módu je způsob čtení koeficientů odlišný.

Účelem je získat nejprve nejdůležitější nízkofrekvenční koeficienty, jenž se nachází v levém horním rohu a jenž směrem do pravého dolního rohu zvyšují svůj kmitočet, avšak jejich význam postupně klesá. Vlivem kvantování je také mnoho koeficientů vyšších frekvencí v získané posloupnosti rovno nule a díky tomu je tak tento způsob čtení ideální pro kódování RLE (Run-length Encoding) [4], [10].



Obr. 2.4: Princip čtení koeficientů metodou cik-cak pro celosnímkový mód [10]

Při kódování může být rozlišeno, zda se jedná o DC koeficient (stejnoseměrnou složku) na pozici $[0;0]$, nebo o AC koeficienty střídavých složek. DC koeficient je pro výsledný obraz zásadní a proto může být kódován samostatně prediktivním kódováním. Kódovaná hodnota je pak rozdílem ($DIFF$) mezi hodnotou DC koeficientu současného bloku (y_{00}) a DC koeficientu předchozího bloku ($PRED$) [11]. To lze zapsat pomocí vztahu:

$$DIFF = y_{00} - PRED \quad (2.10)$$

Hodnota $DIFF$ se následně kóduje pomocí Huffmanových tabulek, které jsou individuálně stanovené, a to jak pro jasové, tak i pro barvosné složky [11].

2.1.7 Entropické kódování

Tento typ kódování se používá z důvodu snižování nadbytečnosti a dle [6] je definován jako „způsob bezztrátového kódování signálu s proměnnou délkou kódu využívající snížení redundance aplikované v systému MPEG za kvantizačním stupněm“. Dle [20] je motivováno následující úvahou: Pokud by se přizpůsobení zdroje na přenosový kanál realizovalo rovnoměrným kódováním, uskutečňovalo by se toto kódování s malou účinností, tj. entropie na jeden bit značky by byla malá. Při použití rovnoměrného kódování by se tak na přenos požadované informace bylo zapotřebí většího datového toku, nebo případně více času. Přitom však některé zprávy mají menší četnost než jiné zprávy. Jestliže by se ale použilo nerovnoměrného kódování, pak znakům s četnějším výskytem bude přiřazena značka s menším počtem bitů a tím by se množství dat i doba přenosu zkrátila oproti rovnoměrnému kódování. V konečném důsledku by se zkrátila tzv. střední délka značky a průměrná entropie připadající na jeden bit značky by se zvýšila - tj. snížila by se nadbytečnost přizpůsobeného bitového toku [20].

Protože značky jsou u nerovnoměrného kódování různě dlouhé, musí být jejich struktura taková, aby žádná značka nebyla začátkem jiné značky. Mezi nejznámější patří Huffmanovy kódy. O typech kódování snižujících nadbytečnost bude pojednáno dále [20].

- Run-length encoding (RLE) – lze přeložit jako kódování délky běhu. Využívá se znalosti, že po provedení kvantování je mnoho koeficientů rovno nule a není nutné je uvádět jako posloupnost hodnot, ale lze to provést efektivněji, zápisem v kompaktním tvaru uspořádaných dvojic. Prakticky se jedná o zápis sekvence symbolů do dvojic ve tvaru: (počet opakování, symbol). Čím delší jsou sekvence opakujeících se znaků, tím účinnější je komprese [4]. Pro názornost je možné uvést příklad [4]:

Vstupní řetězec: 16, 0, 0, -3, 5, 6, 0, 0, 0, 0, -7, ...

Výstupní hodnoty: (0,16),(2,-3),(0,5),(0,6),(4,-7), ...

- Kódování s proměnnou délkou slova – představuje bezztrátové entropické kódování VLC (Variable-length Coding) snižující redundanci. Délka slova jímž jsou vyjádřeny vzorky je proměnná a je pro často se vyskytující hodnoty krátká, kdežto pro zřídka se vyskytující hodnoty dlouhá. Toto je určeno entropií, jenž udává nejmenší možný počet bitů pro vyjádření hodnoty jednoho vzorku v závislosti na pravděpodobnosti jeho výskytu. Nejznámějšími zástupci tohoto typu kódování jsou Huffmanovo a aritmetické kódování. [10]

Současné standardy používající kódování VLC mají definovány tabulky kódových slov, jenž byly stanoveny z obvyklé pravděpodobnosti výskytu hodnoty vzorků ve videosignálu. Bylo by sice pro dosažení optimální účinnosti Huffmanova kódování vhodnější, aby se na základě pravděpodobnosti výskytu koeficientů bloků průběžně určovala tabulka Huffmanova kódu, avšak to je v praxi jen obtížně realizovatelné. Jednak by uvedený proces byl velmi výpočetně náročný, což by způsobovalo zpoždění během kódování. Rovněž by bylo nutné při každé změně zaslat současně s videosignálem i tabulku Huffmanova kódu, neboť dekodér musí vždy používat pro dekódování stejnou tabulku kódů jako kodér. Tím by ale došlo ke znatelnému poklesu účinnosti komprese, především pak u kratších video sekvencí [10].

- Slovníkové kompresní metody – princip spočívá v nahrazení vzorků vstupních dat binárními kódy postupně rostoucí délky. Nejznámější metodou je bezpochyby LZW – Lempel-Ziv-Welch, jenž se používá pro formát GIF. LZW komprese byla donedávna chráněna patenty. Původní princip byl navržen v roce 1977 jako LZ77 (Abraham Lempel, Jacob Ziv), další verze následovala o rok později – LZ78. V roce 1983 ji rozšířil Terry Welch. Postup je zjednodušeně následující [12]:
 - vynulování řetězce w
 - dokud vstupní posloupnost obsahuje další znaky, pak načte znak ze vstupní posloupnosti do proměnné K a v závislosti na tom, zda již slovník obsahuje řetězec wK , přidá k řetězci další znak ($w = wK$) a opakuje pro další znak, nebo v opačném případě přidá řetězec wK do slovníku. Následně zapíše na výstup pozici řetězce w ve slovníku a nahradí řetězec pouze znakem $w = K$ a cyklus opakuje pro další znak.

Kompresce LZW vyžaduje počáteční inicializaci slovníku. Slovník není nutné zapisovat do výstupního souboru, avšak se zvětšujícím se slovníkem se zpomaluje vyhledávání, proto bývá slovník kapacitně omezen a při překročení je znovu inicializován [12].

2.1.8 Prediktivní kódování

Provádí kódování rozdílů mezi snímkem a výstupem z predikčního algoritmu, který vytváří vzorky obrazového signálu na základě předchozích vzorků ve stejném nebo předcházejícím snímku [12]. Účinnost komprese může být zlepšena na základě předpokladu, že hodnota DC koeficientů sousedících bloků bývá velmi podobná a také sousední vektory pohybu jsou stejné nebo velmi podobné. Tak může být kódován pouze rozdíl mezi hodnotou předpokládanou (předchozí) a hodnotou skutečnou. Využívá se totiž faktu, že kodér i dekodér umí předpovídat (predikovat) na základě předchozího vzorku hodnotu vzorku následujícího. Pro přenos mezi kodérem a dekodérem tak postačuje přenos rozdílů mezi těmito hodnotami vzorku, jenž se nazývá chyba predikce a v dekodéru se používá pro korekci. Efektivnost komprese je umocněna tím, že chyba predikce se často blíží k nule, nebo je dokonce nulová, a to i díky vynulování rozdílových koeficientů během kvantování [4].

2.1.8.1 Kompenzace pohybu

Zjednodušeně lze říci, že při kompenzaci pohybu dochází ke kódování rozdílů mezi aktuálním snímkem posunutým předcházejícím snímkem za použití vektorů pohybu [12], [4]. Vektory pohybu jsou tedy použity ke zlepšení účinnosti předpovědi hodnot pixelů. Předpověď používá vektorů pohybu k určení posunu do minulých a/nebo budoucích referenčních snímků obsahujících dříve dekódované hodnoty pixelů, které se užívají k vytvoření předpovědi chybového signálu [7]. Vektory pohybu mohou být určeny jednou z mnoha známých metod. Volba metody spočívá v nalezení kompromisu mezi přesností hledání nejlepší shody a výpočetní náročností, reprezentovanou dobou hledání. Je zřejmé, že čím větší je shoda, tím menší hodnoty reziduí je nutné zpracovávat [4]. Pro tuto operaci se využívá poznatku, že ve videosekvenci rychle po sobě následujících snímků zůstává mnoho objektů (respektive bloků) nehybně na stejném místě, avšak pouze některé se pohybují, a to zpravidla v malém rozsahu. Vyhledávání může být ovšem též blok, který je otočený, komprimovaný, nebo expandovaný. Vyhledává se shoda podle jasové složky, chrominanční se posouvají dle nalezeného vektoru jasové složky. Za nejvíce odpovídající blok je považován blok s nejmenší reziduální energií. Tato shoda bloků $N \times N$ pixelů je vyjádřena integrací rozdílů jasových složek cílového a vztazného bloku, dle vztahu [12]:

$$\varepsilon(k, l) = \frac{1}{N^2} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} [Y_T(i + m, j + n) - Y_R(i + k + m, j + l + n)]^2 \quad (2.11)$$

, kde i, j jsou souřadnice obrazového prvku v levém horním rohu bloku a vektor (k, l) je vektorem pohybu.

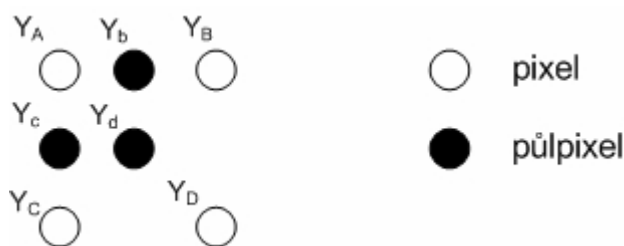
Blok je považován za nalezený, pokud je nalezená nejmenší reziduální energie menší než stanovený práh. V kladném případě je zakódován vektor pohybu (k, l) . K provedení DCT dochází u diferenčního bloku, jenž je určen jako rozdíl vztažného a cílového bloku. Vzhledem k tomu, že rozdílové koeficienty jsou velmi malé, pak po provedení kvantování může také mnoho koeficientů vyšších kmitočtů mít nulové hodnoty. Ve spojení s následným zpracováním signálu (VLC kódování) tak může dojít k zásadní redukci bitového toku v kodéru. Pokud by ovšem při hledání nebyl nalezen blok s reziduální energií nižší než je dovolený práh, pak by byl úplně kódován celý blok. K vyhodnocení shody porovnávaných bloků se využívají metody objektivního posuzování kvality [12].

2.1.8.2 Vyhledávání vektorů pohybu

Vyhledávání bývá lineární, logaritmické, nebo lineární/logaritmické pouze v určitých směrech, a to v dané, prostorově omezené, oblasti – z důvodu výpočetní náročnosti. Logaritmické vyhledávání zkracuje v každém kroku souřadnice testovaných bodů na polovinu ve srovnání s předcházejícím blokem a algoritmus končí při rozdílu souřadnic 1. U standardů bývá rozsah v němž jsou pohybové vektory hledány přímo definován. Kupříkladu dle [7] pro MPEG-1 existuje široký rozsah pro hledání shody, a to při umožnění až půlpixelové přesnosti, avšak bez možnosti pohybových vektorů směřujících za hranice obrazu.

Pro dosažení lepší predikce s kompenzací pohybu a tím současně menší difference, mohou být použity i další techniky, jenž zpřesňují odhad a snižují potřebu datového toku [9]. Ve standardu MPEG-4 se jedná například o tyto:

- Subpixelová kompenzace pohybu – s pomocí interpolace lze určit shodu pohybových vektorů s půlpixelovou, nebo čtvrtpixelovou přesností. Nejprve ovšem musí být nalezena shoda v celočíselném rastru a až poté je v okolí nalezeného bodu hledána nejlepší shoda v subpixelovém rastru [12].



Obr. 2.5: Vzorky na půlpixelových pozicích [12]

Určení hodnot vzorků na půlpixelových pozicích [12]:

$$Y_b = (Y_A + Y_B + 1)/2 \quad (2.12)$$

$$Y_c = (Y_A + Y_C + 1)/2 \quad (2.13)$$

$$Y_d = (Y_A + Y_B + Y_C + Y_D + 2)/4 \quad (2.14)$$

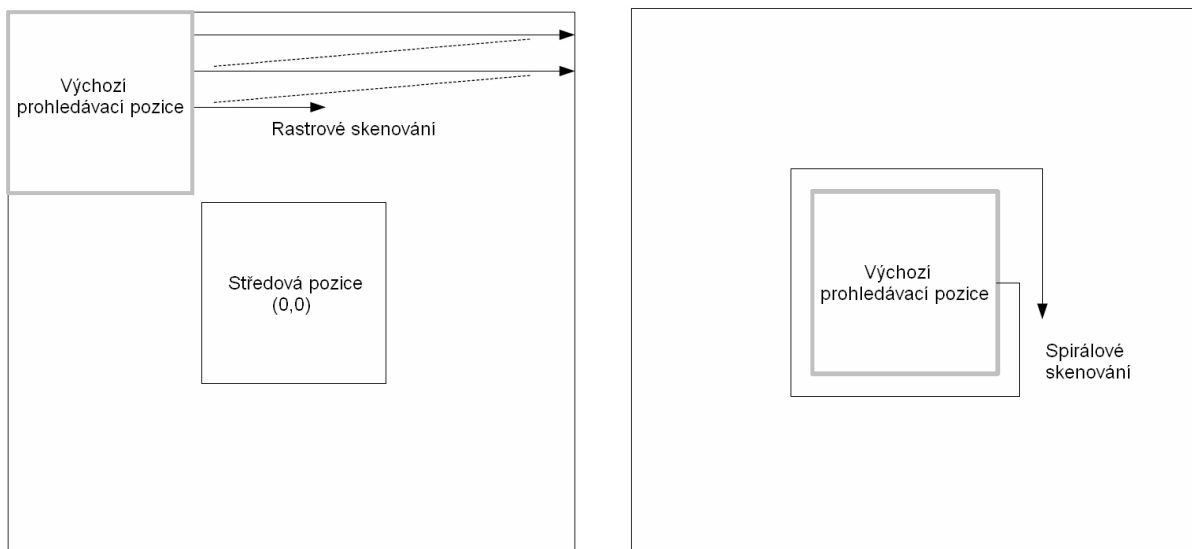
- Neomezené vektory pohybu – referenční makroblok může být použit i v případě, kdy se určitou částí vyskytuje za hranicí snímku. Chybějící hodnoty, jenž leží za hranicí, jsou extrapolovány. Vektory pohybu tak mohou směřovat i do oblasti ležící mimo snímek, což je výhodné zejména při kompenzaci pohybu objektu směrem ven a dovnitř snímku [12].
- Čtyři pohybové vektory na makroblok – běžně se kompenzace za použití vektorů pohybu provádí pro jasový blok o velikosti 16x16 pixelů pro jasovou složku a 8x8 pro barvonosné složky. Použitím čtyř pohybových vektorů bude velikost pohybově kompenzovaného bloku činit 8x8 pro jasovou a 4x4 pro barvonosné složky, což sice zvýší množství použitých vektorů na čtyřnásobek, avšak zpřesní a zefektivní kompenzaci pohybu [12].
- Globální kompenzace pohybu – využívá časté situace, kdy makrobloky tvořící jeden objekt v obraze konají stejný pohyb. Může to být například rotace, pohyb do stran, přiblížení, oddálení. Kodér tak může popsat pohyb celé skupiny bloků tvořících objekt malým množstvím parametrů (postačují i pouze čtyři globální vektory pohybu pro každý VOP). Pro jednotlivé pixely jsou pak při dekódování určeny pohybové vektory pomocí interpolace [13].

2.1.9 Úplné prohledávací algoritmy

K určení optimálního vektoru pohybu existuje mnoho metod, jenž se snaží nalézt co nejrychleji a co nejpresněji největší shodu [4]. Z důvodu výpočetní náročnosti je tedy vždy uvažováno o hledání co nejlepší shody v rámci určité prostorově omezené oblasti. Prohledávání probíhá pouze pro jasovou složku, neboť barvonosné složky jsou zpravidla na vstupu nezanedbatelně podvzorkovány. Vektory pohybu určené z jasové složky se však uplatňují jak u jasové, tak i chrominančních složek. K vyhodnocení míry shodnosti porovnávaných bloků se využívají obecné metody objektivního posuzování kvality, o nichž bude pojednáno dále. Prohledávací algoritmy je možné dle způsobu vyhledávání v prohledávané oblasti rozdělit na algoritmy s úplným hledáním (Full Search) a na algoritmy s rychlým hledáním (Fast Search) [4].

2.1.9.1 Algoritmus Full Search

Metoda s úplným prohledáváním, jenž porovnává referenční blok s každým možným v prohledávané oblasti. Výhodou této metody je, že je zaručeno nalezení bloku s největší mírou shody. Naopak nevýhodou je, že tato metoda je velmi výpočetně náročná. Pokud je rozsah vzorků $\pm S$ směrem od nulové středové pozice, pak pro určení všech hodnot musí být proveden výpočet v $(2S + 1)^2$ lokalitách [13]. Při prohledávání může být postupováno různými cestami. Například rastrovým skenováním směrem z levého horního rohu doprava, po jednotlivých řádcích. Pokud by však byla k dispozici možnost předčasného přerušení výpočtu, pak by se jako vhodnější postup jevil směr od středu k obvodu, v postupně zvětšujících se spirálách dle Obr. 2.6, a to proto, že bylo zjištěno, že u typické videosekvence je většina vektorů pohybu koncentrována poblíž středové pozice, kde bývá nalezena většina minim [4].



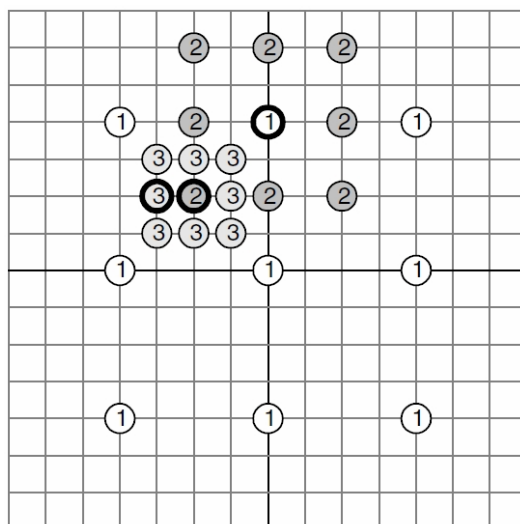
Obr. 2.6: Prohledávací okno pro rastrový a spirálový způsob skenování [4]

2.1.10 Rychlé prohledávací algoritmy

Ačkoliv je u metody s úplným prohledáváním dána možnost předčasného ukončení hledání po nalezení dostatečně odpovídající shody, přesto nelze takovýto postup považovat za optimální, neboť i při více krocích nemusí být získán dobrý výsledek. Zejména u zařízení s výkonovým nebo energetickým omezením je třeba použít sofistikovanější algoritmus, jenž v několika krocích dokáže nalézt dostatečně shodný blok v rámci celé prohledávané oblasti [4]. Dále jsou uvedeny dvě velmi oblíbené metody, které ilustrují jakým způsobem může být při prohledávání postupováno.

2.1.10.1 Algoritmus Three Step Search

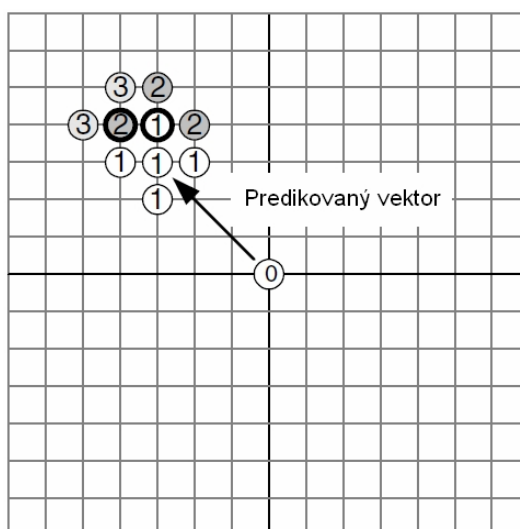
Tato metoda patří do skupiny N-Step-Search algoritmů, u kterých prohledávání probíhá v N krocích. Nejpoužívanější z nich je právě uvedená tříkroková varianta [4]. Obecně platí, že při prohledávání v N krocích se prohledává čtvercová oblast v rozsahu $\pm(2^N - 1)$ vzorků. Při hledání se vychází od pozice (0,0), se zadaným krokem $\pm 2^{N-1}$. Z osmi pozic okolo počáteční pozice a současně i z počáteční pozice je určen blok s největší shodou, jenž je nastaven jako výchozí. Velikost kroku se sníží na polovinu a opět je proveden výpočet a porovnání devíti hodnot. Blok s největší shodou je opět nastaven jako výchozí. Takto se pokračuje tak dlouho, dokud se krok nerovná jedné, viz Obr. 2.7. Dle nejlepší nalezené shody je určen pohybový vektor. Počet hledání je roven $(8N+1)$, což je podstatně méně v porovnání s metodou Full Search $(2^{N+1} - 1)^2$, viz [13].



Obr. 2.7: Prohledávání metoda Three Step Search [4]

2.1.10.2 Algoritmus Nearest Neighbours Search

Rychlý vyhledávací algoritmus, navržený pro kodeky H.263 a MPEG-4, jenž má nízkou výpočetní náročnost, avšak přesností vyhledávání se blíží k algoritmu Full Search. Této vlastnosti se dosahuje na základě poznatku, že sousední bloky mají podobné vektory pohybu [13]. Při provádění prohledávání je nejprve nastavena počáteční pozice (0,0) a z ní je pomocí předpovězeného pohybového vektoru vytýčen bod, v němž probíhá výpočet pro určenou pozici a čtyři sousední bloky ve vzdálenosti 1, jenž tvoří geometrický tvar symbolu „+“. Pokud je nejlepší shoda nalezena ve výchozí pozici, pak vyhledávání končí. Je-li však v některém ze sousedních bloků, pak je tento blok nastaven jako nová výchozí pozice a opět proběhne nové hledání v okolních blocích [4].

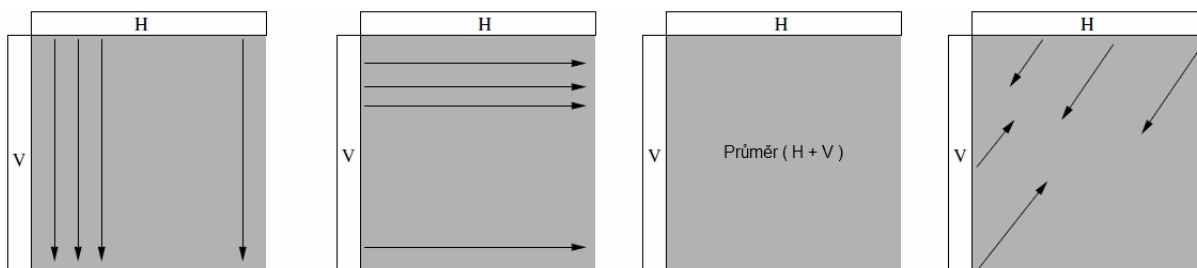


Obr. 2.8: Prohledávání metodou Nearest Neighbours Search [4]

2.1.11 Intra predikce

Na rozdíl od Inter kódování mezi jednotlivými snímky, se pro kódování v rámci jednoho snímku využívá zjištění, že nízkofrekvenční transformované koeficienty sousedních bloků vykazují vysokou míru korelace. Proto lze pro predikovaný blok využít již dekódovaných hodnot sousedních bloků téhož proužku (jako referenčních), které jsou dostupné po rekonstrukci v dekodéru [4]. Pro zvýšení efektivity predikce mohou být koeficienty aktuálního bloku predikovány podle míry shody, a to porovnáním DC koeficientu horního bloku C a levého již dekódovaného bloku A [13]. Směr predikce je pak určen následujícím vztahem: „Jestliže $|DC_A - DC_B| < |DC_B - DC_C|$, pak predikuj z horního bloku C , jinak predikuj z levého bloku A “ [13]. Toho je využito i pro AC koeficienty (MPEG-4 Visual).

Predikce pixelů na principu DPCM mohou být rozděleny dle různých kritérií [3]. Například podle linearity nebo nelinearity predikce z předchozího pixelu, podle přizpůsobivosti příchozím datům (fixní nebo adaptivní prediktory). Dále je též možné dělit podle dimenze na jedno-dimenzionální (k predikci si využívá předešlý pixel na stejném řádku), dvou-dimenzionální (využívá se pixelů ze současného i z předchozích řádků) a mezisnímkové (predikce využívá jedné nebo dvou dimenzí, ale též odlišné snímky) [13]. Ilustraci Intra kódování pro standard MPEG-4 s blokem o velikosti 16x16 podává Obr. 2.9:



Obr. 2.9 :Typy Intra predikce bloku 16x16 pixelů (vertikální, horizontální, DC, rovinná) [5]

2.2 ZPŮSOBY HODNOCENÍ KVALITY OBRAZOVÉHO SIGNÁLU

Při provádění komprese, v průběhu kódování, bývá použita ztrátová a bezztrátová komprese. Jelikož v průběhu ztrátové komprese dochází k nevratné ztrátě informací, je pro určení velikosti degradace obrazu nutné stanovit určitá kritéria, jenž dovolují popsat rozdíl stavů před kompresí a po provedení rekonstrukce. Pro hodnocení kvality existují dva základní principy: subjektivní a objektivní [13].

Systém pro zjišťování subjektivního kvalitativního dojmu pracuje na principu přepínání mezi zdrojovou (originální) a hodnocenou (komprimovanou) obrazovou sekvencí [4]. Při subjektivním hodnocení se vychází z pozorování určité množiny snímků reprezentativní skupinou osob. Tato skupina subjektivně posuzuje kvalitu obrazu a stanovuje hodnocení dle předem dané stupnice. Může to být například slovní hodnocení s použitím středního názorového skóre. Jednotlivé stupně hodnocení jsou dle stupnice MOS (Mean Option Score) následující [13]:

Tab. 2.2: Stupně hodnocení dle stupnice MOS [13]

	Kvalita (quality)	Poškození (impairment)
1	Bad	velmi obtěžující (very annoying)
2	Poor	obtěžující (annoying)
3	Fair	lehce obtěžující (slightly annoying)
4	Good	postřehnutelné, ale neobtěžující (perceptible, but not annoying)
5	Excellent	nepostřehnutelné (imperceptible)

Rovněž může být použito hodnocení známkami (-2, -1, 0, 1, 2) s obdobným významem [13].

Objektivní hodnocení vyjadřuje rozdíly mezi originálním a rekonstruovaným obrazem pomocí matematicky definovaných postupů, jenž jsou výhodné zejména z důvodu možnosti jejich rychlého a jednoznačného stanovení za použití výpočetní techniky. To se uplatňuje mimo jiné i ve výše uvedeném stanovení vektorů pohybu – nejnižší hodnota chyby představuje největší míru shody [4], [19].

Nejpoužívanější metriky jsou:

- **MAE (Mean Absolute Error)** - střední absolutní chyba = Absolutní hodnota součtu rozdílů všech bodů originálního a rekonstruovaného obrazu ve výsledku poděleného rozměrem obrazu [19]

$$MAE = \frac{1}{MN} \cdot \sum_{i=1}^M \sum_{j=1}^N |x(i, j) - \tilde{x}(i, j)| \quad (2.15)$$

- **SAE (Sum of Absolute Errors)** - součet absolutních chyb [19]

$$SAE = \sum_{i=1}^M \sum_{j=1}^N |x(i, j) - \tilde{x}(i, j)| \quad (2.16)$$

- **MSE (Mean Square Error)** - střední kvadratická chyba = součet rozdílů všech bodů originálního a rekonstruovaného obrazu umocněného na druhou a ve výsledku poděleného rozměrem obrazu [19]

$$MSE = \frac{1}{MN} \cdot \sum_{i=1}^M \sum_{j=1}^N |x(i, j) - \tilde{x}(i, j)|^2 \quad (2.17)$$

- **NMSE (Normalized Mean Square Error)** – normalizovaná střední kvadratická chyba [19]

$$NMSE = \frac{1}{L^2} \cdot MSE \quad (2.18)$$

- **PSNR (Peak Signal to Noise Ratio)** - špičkový poměr signálu k šumu [5], [19]

$$PSNR_{dB} = 10 \cdot \log_{10} \frac{(2^n - 1)^2}{MSE} = 10 \cdot \log_{10} \frac{1}{NMSE} \quad (2.19)$$

, kde: n – počet bitů na obrazový vzorek
 M, N – rozměry snímku
 $x, \tilde{x}$ – originální a rekonstruovaný snímek
 L – dynamický rozsah hodnoty jasu obrazového bodu
 i, j – souřadnice obrazového bodu ve snímku

Z uvedených vztahů vyplývá, že čím je rekonstruovaný obraz kvalitnější, tím je větší PSNR a menší MSE. V případě méně kvalitního obrazu je PSNR menší a MSE větší.

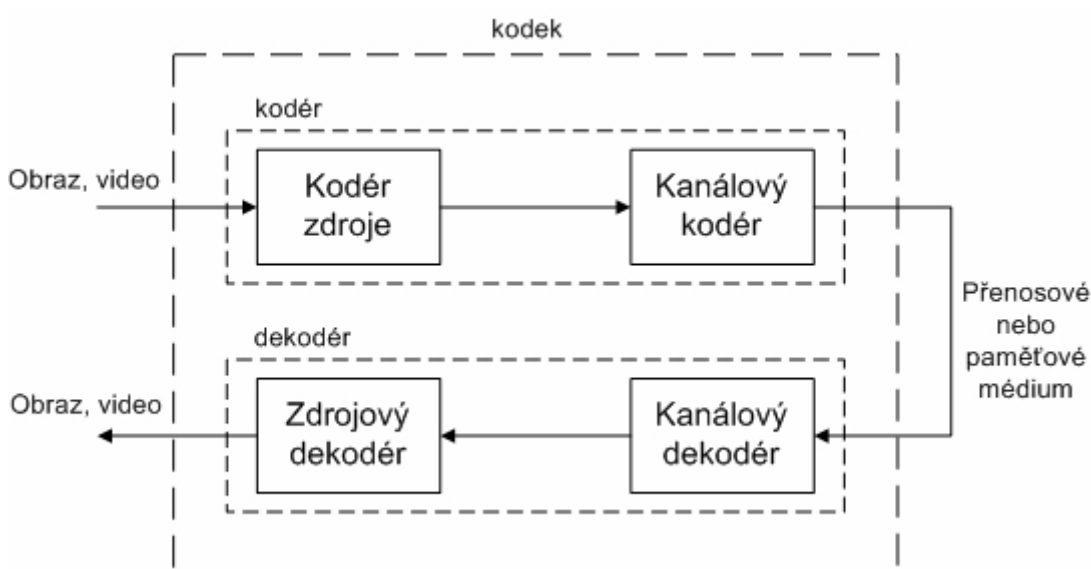
3 KOMPRESNÍ STANDARDY A KODEKY

Slovem „kodek“, převzatým z anglického CODEC (coder – decoder) lze označit počítačový program nebo zařízení sloužící k transformaci datového proudu, nebo signálu [10]. V běžných multimediálních signálech se lze setkat s obrazovými, zvukovými a doplňujícími daty, jenž jsou zapouzdřeny do společného celku, tzv. kontejneru. Vnější poznávacím znakem použitého kontejneru je jeho označení. Mezi nejčastěji užívané patří AVI, WMV, OGG, MKV, MP4 a další [12]. Dále je též každý blok v kontejneru identifikován FourCC značkou o délce 4 byty, jenž informuje o formátu datového proudu (např. typ použitého kodeku). Pro praktické účely této práce je za zpracovávaná data uvažován výhradně videosignál, jenž je složen ze sekvence po sobě jdoucích obrazových snímků.

Obrazový kodek kóduje zdrojový obraz nebo videosekvenci do komprimované podoby a dekóduje ji zpět do podoby, jenž je identická nebo podobná zdrojové sekvenci [4]. Pokud je dekódovaná výstupní sekvence shodná se vstupní, pak se jedná o bezztrátové kódování. Pokud se však obě sekvence liší, je tento proces ztrátový. Kodeky se využívají pro kódování signálu z důvodu přenosu, zaznamenání, nebo šifrování a analogicky též k dekódování, pro obnovení do podoby vhodné pro zobrazení nebo manipulaci s těmito daty. Schéma obecného kompresního systému uvádí Obr. 3.1.

Volba kodeku patří k zásadním rozhodovacím momentům při návrhu video kódování pro konkrétní platformu. A to především z důvodu hledání optimálního vyvážení požadovaného kompresního a výpočetního výkonu, vhodnosti pro cílovou platformu, množství podporovaných funkcí, energetických požadavků a dalších faktorů.

Důležitým faktorem jsou rovněž licenční podmínky, spojené ve většině případů s nutností platby poplatků za vlastnictví oprávnění k užití kodeku [4]. Komerční implementace standardů MPEG-4 Visual, či H.264/MPEG-4 AVC spadá do působnosti množství patentů jenž znesnadňují jejich nasazení. Kupříkladu použití MPEG-4 Visual je ve výsledku vázáno na úhradu autorských poplatků sdružení přibližně třiceti organizací, jenž jsou držiteli patentů. Podobná situace panuje též u H.264/MPEG-4 AVC, kde však u základních funkcí a omezených nasazení není vyžadován poplatek, což má pomoci masovějšímu a snadnějšímu rozšíření. Více informací o licenčních podmínkách různých systémů poskytuje [21].

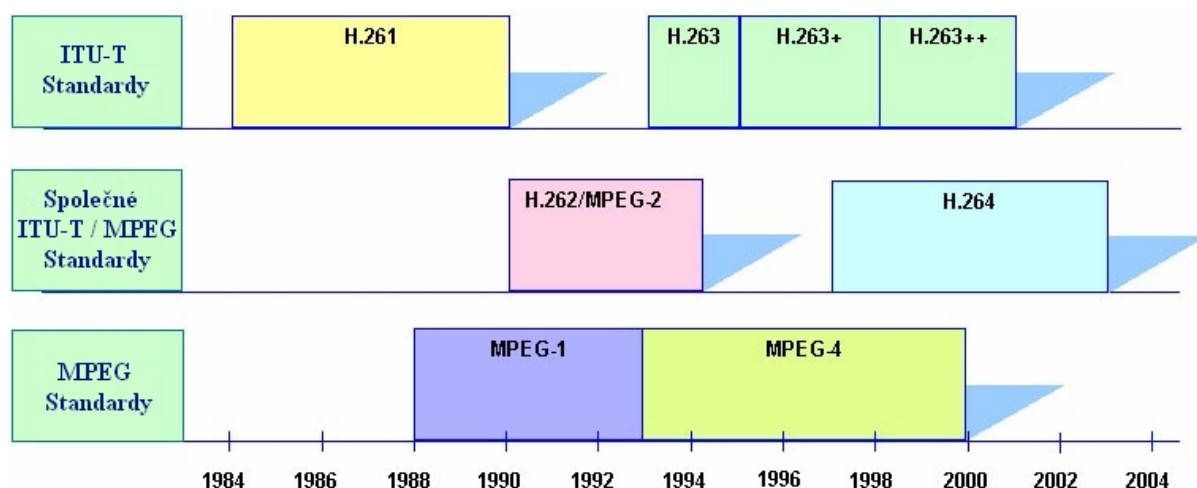


Obr. 3.1: Schéma obecného kompresního systému [19]

Ke komprimaci obrazového signálu bylo s postupem času navrženo a implementováno mnoho algoritmů, jenž svými vlastnostmi odráží potřeby a možnosti realizace kódování v době svého vzniku. K nejznámějším a nejrozšířenějším patří bezpochyby kodeky z rodiny standardů MPEG (Motion Picture Experts Group) a kodeky skupiny H.26x vytvořené organizací ITU-T. Ač byly doporučení ITU-T vydávány nezávisle na standardech MPEG-x organizace ISO/IEC, přesto došlo v minulosti k jejich významné spolupráci.

Kompresní standardy MPEG jsou vytvářeny skupinou expertů, jenž byla sestavena organizacemi ISO a IEC již v roce 1988. Standardy MPEG jsou specifické ztrátovou kompresí, použitím diskrétní kosinové transformace a asymetričností komprese. To značí, že komprese datového toku je časově i výkonově mnohem náročnější než jeho dekomprese [19].

Odlišný vývoj standardů H.26x a MPEG-x má původ již v počátcích, kdy doporučení ITU-T mířily především na video konference, video telefonii a obecně jakékoliv aplikace video komunikace v reálném čase. Naproti tomu standardy MPEG cílily do oblasti ukládání videa (CD, DVD), plošného vysílání video obsahu (televizní vysílání) a na video streaming (video přes Internet, DSL, bezdrátové sítě) [27]. Obě organizace se společně podílely v minulosti již na vzniku standardů MPEG-2 a H.262. V průběhu času vývoj došel do takové fáze, kdy použité techniky byly natolik podobné, že samostatný vývoj ztratil opodstatnění, což vyústilo do spolupráce obou komisí ITU-T VCEG a ISO/IEC JTC1 na návrhu nového standardu H.26L. To dalo v nedávné době možnost vzniknout standardu H.264 / MPEG-4 part 10 AVC. Časovou osu vývoje doporučení a standardů znázorňuje Obr. 3.2 [27].



Obr. 3.2: Vývoj doporučení ITU-T a standardů MPEG [27], [26]

3.1 MPEG-1

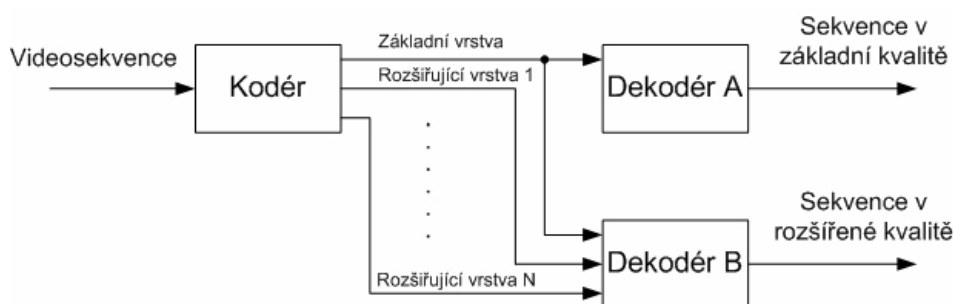
Kompresní standard MPEG-1, označený též jako ISO/IEC IS 11172, byl uveřejněn v roce 1993 a jeho účelem bylo nahrazení VHS [7]. Je určen k ukládání audia a neprokládaného videa s bitovou rychlostí do 1,5 Mbit/s na disky CD-ROM a pro použití ve videotelefonech. Standard MPEG-1 se vyznačuje: v oblasti audia omezení na dva kanály (stereo); v oblasti videa chybí podpora prokládaného videa, obsahuje pouze jeden standardní profil CPB (Constrained Parameters Bitstream), kódování snímku I, P, B, doporučené rozlišení obrazu dle formátu SIF, podpora pouze barevného modelu 4:2:0 a problematické kódování vyšších rozlišení [7]. Z výše uvedeného vyplývá, jak omezené byly možnosti využití tohoto standardu. Ač byl postupem času nahrazen pokrokovějšími, přesto se doposud těší velké oblibě zvukový kompresní formát MPEG-1 Audio Layer III (MP3).

3.2 MPEG-2

V krátké době po uvedení standardu MPEG-1 byl v roce 1995 zveřejněn standard MPEG-2, označený jako ISO/IEC 13818, uzpůsobený pro distribuci televizního signálu. Jeho výhodou je, že již ve specifikaci bylo počítáno s možností šíření videosignálu různými prostředními, v rozličné kvalitě a s různými stupni ochrany proti chybám v přenosu a rušení [8].

Velkým přínosem tohoto standardu byla jeho variabilita ve schopnosti přizpůsobení multimediálního signálu pro příjem koncovými zařízeními. Oproti předchozímu standardu došlo především k zavedení účinnějších kompresních metod a s tím spojené možnosti zvýšení bitové rychlosti datového toku.

Zavedena byla rovněž možnost hierarchického kódování – škálování – s jehož pomocí tak lze současně přenášet videosignál ve více částech. V jednom datovém toku tak může být přenášén kvalitativně odstupňovaný signál pro přijímače s nižším rozlišením i doplňkový signál pro přijímače s vysokým rozlišením [12]. V závislosti na požadavcích a schopnostech dekodéru je zpracována základní kvalitativní úroveň, či všechny dostupné, viz Obr. 3.3:



Obr. 3.3: Obecné schéma principu škálování [4]

U prostorového škálování se datový tok rozděluje na vrstvy s odlišným prostorovým rozlišením obrazu. Při dekódování základní vrstvy je získána video sekvence s nízkým rozlišením. Po dekódování všech vrstev je získán signál s maximálním možným dostupným rozlišením, které bylo zpracováno v kodéru. V kodéru může být toto škálování provedeno pomocí podvzorkování kódovaného signálu. Rozšiřující vrstvy pak obsahují pouze rozdílový signál. Využití lze nalézt v realizaci myšlenky slučitelnosti mezi systémy s odlišnými požadavky na rozlišení obrazu [10].

Při kvalitativním odstupňování (SNR) je v základním bitovém toku zakódován signál s nižší jakostí, tj. s vyšším poměrem šumu. Rozšiřující vrstvy pak tento poměr vylepšují směrem k vyššímu poměru SNR a k získání kvalitnějšího výstupního obrazu. Toto rozdělení může být realizováno pomocí hrubého a jemného kvantování koeficientů v kodéru. Využit lze i možnosti odlišné míry zabezpečení signálu jednotlivých vrstev při přenosu ve ztížených podmínkách [10].

Z dalších vylepšení lze jmenovat zavedení možnosti prokládaného řádkování a k tomu navázané změny postupů v predikčním a transformačním kódování. Úpravy se týkaly též kvantování a entropického kódování [10].

3.2.1 Profily a úrovně

Variabilita standardu MPEG-2 je popsána pomocí profilů a úrovní, jenž specifikují podporované funkce a případná omezení. To je výhodné zejména z důvodu možnosti implementace pouze omezené sady funkcí a nikoliv celé šíře standardu.

MPEG-2 má definováno pět profilů, které specifikují sadu funkčních vlastností. Jedná se především o způsob podvzorkování chrominančních složek, kompresní algoritmus, predikci a podporu škálování. Škálování je definováno pro prostorové a kvalitativní odstupňování [10].

Úrovně specifikují kvantitativní parametry, a to zejména jejich omezení v daném profilu. Jedná se konkrétně o velikost snímku, bitový tok a snímkový kmitočet.

Oba parametry se zapisují zpravidla současně, a to ve tvaru „*profil@úroveň*“, což značí nejvyšší možný profil a úroveň standardu MPEG-2, jenž zařízení podporuje. Analogicky jsou tak podporovány i všechny nižší úrovně standardu [10].

Přehled profilů a úrovní standardu MPEG-2 uvádí následující Tab. 3.1. Z ní je patrné, že *Jednoduchý profil* používá pouze snímky typu I a P při vzorkování ve formátu 4:2:0. To je sice jednoduché pro implementaci a rovněž tak i výpočetně nenáročné, avšak kvůli absenci snímků B nelze dosáhnout vyššího kompresního poměru. Oproti tomu *Hlavní profil* obsahuje všechny zásadní prvky kodeku jako jsou snímky typu B, prokládané řádkování a formát vzorkování 4:2:0.

Oba odstupňované profily standardu MPEG-2 umožňují již dříve zmiňované prostorové a kvalitativní škálování. Vysoký profil v sobě obsahuje všechny předchozí funkčnosti a dále též přidává formát vzorkování 4:2:2. Vzhledem k tomu, že tento profil podporuje výše uvedené funkčnosti, a to i pro vysoké rozlišení, je pochopitelné, že složitost kodéru i dekodéru podporující tento profil je velmi vysoká.

Tab. 3.1: Přehled profilů a úrovní standardu MPEG-2 [10]

Profil	Jednoduchý	Hlavní	Odstupňovaný	Prostorově odstupňovaný	Vysoký
Úroveň	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

3.2.2 Funkce kodéru

Účelem kodéru je zpracování vstupního signálu ve formě obrazových snímků a vytvoření kódovaného komprimovaného výstupního datového toku. Kodér je tvořen přímou kódovací větví a zpětnou, inverzní, zajišťující dekódování obrazu pro potřeby následného zpracování v kodéru.

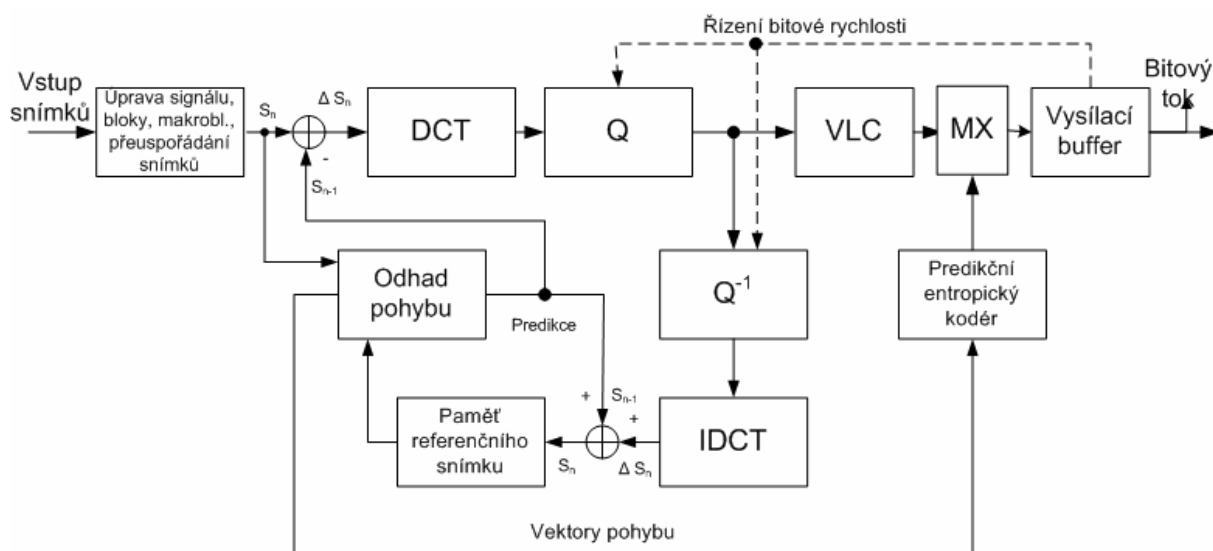
Ve standardu MPEG-2 přichází na vstup kodéru obrazové snímky určitého typu a velikosti, které je třeba nejprve podvzorkovat (např. do formátu 4:2:0 YCbCr) a přeuspořádat. Přeuspořádání se provádí z důvodu predikce, aby při určování jednosměrně a obousměrně predikovaných snímků byly k dispozici potřebné snímky. Pořadí snímků na vstupu je z pořadí I B B P B B P... převedeno do výstupní posloupnosti I P B B P B B... Ač je toto pořadí předem určeno parametrem M , určujícím rozteč snímků P a parametrem N pro rozteč snímků I , je možná jeho změna, a to především v případě podstatné změny scény, kdy použitá predikce nenachází uplatnění.

První snímek v kodéru je typu I a je Intra kódován, bez vazby na jiné snímky. V prvním kroku je provedena DCT na snímku rozděleném do bloků, která převede bloky z prostorové oblasti do oblasti frekvenční. To znamená, že v levém horním rohu se nachází stejnosměrná DC složka s průměrným jasnem celého bloku, od které se směrem do pravého dolního rohu nacházejí postupně složky od nižších k vyšším frekvencím [10].

Dále následuje kvantování, při kterém je každý z 64 vzorků bloku kvantován dle zadané kvantovací tabulky, čímž dojde k redukci vzorků zejména vyšších kmitočtů. Kvantovací tabulky jsou definovány pro Intra i Inter kódování a mohou být pozměněny v závislosti na stavu vysílací paměti, a to tak, aby nedocházelo k jejímu zahlcení, nebo nedostatečnému plnění. Tato činnost je v Obr. 3.4 vyznačena zpětnou vazbou zajišťující řízení toku. Po nakvantování a zaokrouhlení koeficientů, z nichž mnoho je nulových, dochází k cik-cak čtení dle Obr. 2.4, a to směrem od nižších k vyšším koeficientům. Načtená posloupnost se následně kóduje s proměnnou délkou slova VLC, kdy se každému údaji přiřadí odpovídající Huffmanův kód dle zadaných tabulek. Tímto bezztrátovým kódováním dojde k efektivnějšímu vyjádření pomocí redukce bitů. Takto vyjádřený bitový tok přechází do vysílací paměti, jenž zajišťuje nepřerušovaný datový tok s požadovanou bitovou rychlostí, který by měl optimálně využívat přidělenou šířku pásma. Toho je dosahováno pomocí funkce řízení toku s využitím změny váhového koeficientu kvantizační tabulky. Při velkých změnách obrazu totiž může dojít k potřebě podstatného omezení bitového toku, což si vynutí změnu koeficientu a tím i kvantovací tabulky a kvantizace se stane hrubší, čímž může dojít ke vzniku blokových artefaktů, tedy ke stavu kdy jsou patrné hranice jednotlivých bloků [10].

Současně se zpracováním v dopředné větvi dochází ve zpětné větvi k následujícím činnostem. Snímek jenž byl zpracován pomocí DCT a kvantován je ihned inverzně kvantován a je provedena IDCT, čímž je získána rekonstrukce snímku stejně jako v dekodéru. Tento I snímek je uložen v paměti jako referenční a slouží k odhadu vektorů pohybu pro následující příchozí snímek typu P . Pro určení vektorů pohybu existuje mnoho různých metod, pomocí nichž může být určen. Některé z nich byly popsány již v předchozí kapitole. Obecně však platí, že pro každý blok referenčního snímku je v určité oblasti prohledávaného snímku prováděno hledání co nejvíce shodného bloku. Vektor nalezeného bloku je pak následně zaslán na predikční kodér. Po úpravě referenčního snímku nalezenými vektory pohybu je tento odhadnutý snímek odečten od snímku originálního a zakódován je pouze jejich rozdíl, tzv. chyba predikce. Ve VLC kodéru se kóduje tento rozdíl u P a B snímků a k nim jsou rovněž připojeny entropicky kódované vektory pohybu. Pořadí snímků I , P a B je předem známé, avšak při významné změně obrazu (střih, rychlá změna) by již difference byla příliš velká a tak namísto hledání pohybových vektorů je snímek zakódován jako typ I a dále se pokračuje dle nastavené posloupnosti snímků [10].

Jak je z uvedeného popisu patrné, ke ztrátě informace v kodéru dochází již na vstupu, při podvzorkování obrazu do formátu 4:2:2 a 4:2:0, dále též při kvantování, v důsledku zaokrouhlování. Kvantizace je tak rozhodujícím faktorem, jenž řídí kvalitu a bitový tok výsledného videa [11].



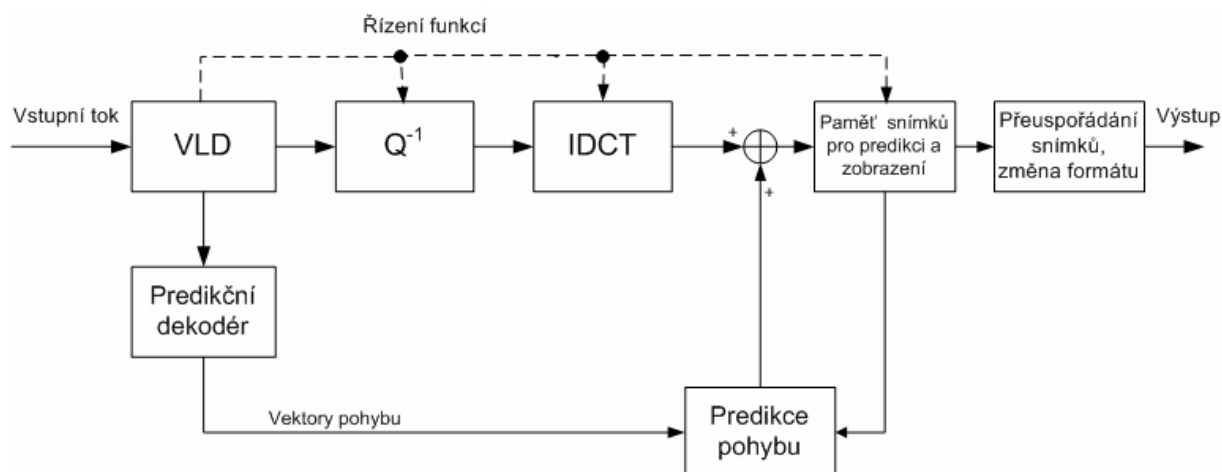
Obr. 3.4: Blokové schéma kodéru standardu MPEG-2 [10]

3.2.3 Funkce dekodéru

Do dekodéru přichází vstupní datový tok tvořený obrazovými snímky v pořadí, v jakém byly vyslány z kodéru, a to: I P B B P B B... Nejprve tedy musí být zpracován Intra kódovaný snímek I, z něž se stane snímek referenční. K tomu však nejdříve musí dojít k dekódování s proměnnou délkou slova VLD, k inverznímu kvantování a k inverzní DCT. Rekonstruovaný snímek je uložen do paměti, kde slouží jako referenční pro následující příchozí snímek.

Dalším zpracovávaným snímkem je snímek typu P, nesoucí chybu predikce, od něž se na vstupu oddělí vektory pohybu, které v prediktoru slouží k pozměnění referenčního snímku do podoby predikovaného snímku. V blocích pro inverzní kvantování a v IDCT je dekódována chyba predikce, která se přičte k predikovanému snímku a je tak získán rekonstruovaný snímek. Takto je získán nový úplný snímek, který může být dále použit jako referenční pro zpracování následujícího příchozího predikovaného snímku typu P nebo B.

Jak je patrné, oproti snímku I, který nese úplnou obrazovou informaci, snímky typu P a B nesou pouze vektory pohybu a chybu predikce. Pro odstranění případných blokových artefaktů z obrazu na výstupu dekodéru může být využito postprocessingu. Jak je z Obr. 3.5 patrné, složitost dekodéru je nižší a rovněž požadavky na výpočetní výkon nejsou tak vysoké jako u kodéru, proto může být dekodér realizován i v relativně levných a jednoduchých zařízeních [10].



Obr. 3.5: Blokové schéma dekodéru standardu MPEG-2 [10]

3.3 MPEG-4

Standard MPEG-4 určený pro kódování obrazových objektů, označený jako ISO/IEC 14496 Part 2 (Visual) byl poprvé zveřejněn v roce 1999 a jeho primárním určením se měly stát zejména systémy s nízkou přenosovou rychlostí [9]. Díky vysoké účinnosti komprese, ale i díky použití dvou jader pro kódování se však brzy rozšířil do nejrůznějších multimediálních aplikací, a to i pro video s vysokým rozlišením.

Obě kódovací jádra standardu MPEG-4 se liší svým určením pro rozdílné objemy toku videosignálu [4]:

- Jádru VLBV (Very Low Bitrate Video) – kódování s nízkou přenosovou rychlostí (5 – 64kb/s), snímkový kmitočet do 15 snímků/s, maximální prostorové rozlišení je formátu CIF. Je vhodné pro pomalé datové linky.
- Jádru HBV (High Bitrate Video) – kódování ve vysokém prostorovém rozlišení při snímkové frekvenci odpovídající televizním normám. Přenosová rychlost v rozmezí 64kb/s – 4Mb/s.

Novými prvky jenž přinesl tento standard je i rozdělení typu videa na přirozené, syntetické a hybridní. Přirozené video se kóduje známým způsobem pomocí predikce a kompenzace pohybu a použitím transformace (DCT). U syntetického videa to je pomocí modelování objektů a přístupu k nim [12].

Zásadní změnu koncepce v chápání videa přinesl posun od kódování snímků pomocí bloků k objektově orientovanému kódování samostatných video objektů VO v objektové rovině videa VOP [9]. V neposlední řadě těž došlo k rozšíření formátu obrazu na vstupu kodéru a výstupu dekodéru na formáty 4:2:0, 4:2:2 a 4:4:4 s možností prokládaného či neprokládaného řádkování.

Dalším vývojovým krokem obrazového standardu MPEG-4 se stala norma ISO/IEC 14496 Part 10 (Advanced Video Coding) publikovaná v roce 2003. Ta je výsledkem spolupráce týmů organizací ISO/IEC a ITU-T. Produktem se stal následník doposud nezávislých standardů řady MPEG-x a H.26x, jenž přebírá metody použité v obou standardech a ty vylepšuje o účinnější algoritmy při kódování signálu.

Tento standard lze tedy označit jako nástupce H.263+ a MPEG-4 profilu Simple s vyšší kompresí a spolehlivostí přenosu, orientovaný na kódování obdélníkových snímků pro videokonference a videotelefonii. Výchozím formátem obrazu je 4:2:0, podporuje až čtvrtpixelovou kompenzaci pohybu v blocích 4x4 pixely [12].

Nově též zavádí rekonstrukční deblocking filtr a Intra predikci makrobloku z již dekodovaných vzorků stejného proužku. Obraz je rozdělen do proužků o velikosti minimálně jeden makroblok a maximálně celý snímek. Proužek I – pouze makrobloky I predikované z dat stejného proužku. Proužek P – makrobloky P predikované z předcházejícího snímku anebo makrobloky I. Proužek B – makrobloky B predikované ze dvou snímků, nebo makrobloky I.

Deblocking filtr vyhlazuje okraje bloků, čímž redukuje zkreslení obrazu blokovými artefakty. Aplikuje se v kodéru na každý makroblok před jeho uložením, nebo v dekodéru před jeho zobrazením [12].

Příkladem společností, jež se zabývají tvorbou kodeků, odvozených ze standardu MPEG-4 jsou: DivX, Envivio, VideoLocus, UBVideo, Hantro, Amphion, Ducas, Sciworx a další [4].

V této kapitole budou dále zmíněny specifika obou norem, avšak s důrazem na novější standard AVC, jež je použit i pro praktickou realizaci této práce.

3.3.1 Profily standardu MPEG-4

Ve standardu MPEG-4 Visual je definován velký počet profilů, jež lze dle jejich zaměření rozdělit do sekcí dle typu kódovaného videa (přirozené, syntetické a hybridní). Princip zápisu je podobně jako ve standardu MPEG-2 realizován zápisem „*profil@úroveň*“, jež uvádí nejvyšší stupeň kódování podporovaný v daném kodéru (dekodéru). Nejčastěji se jedná o kódování přirozeného obrazového obsahu, jež má k dispozici především tyto profily [5]:

- Simple
- Advanced Simple (větší efektivita, podpora prokládaného videa)
- Simple Scalable (podpora časového a prostorového škálování)
- Core (kódování škálovaných objektů libovolného tvaru)
- Main (kódování prokládaných, částečně průhledných a jiných objektů)
- N-bit (kódování objektů s rozlišením vzorků v rozsahu od 4 do 12 bitů)

Standard rovněž obsahuje řadu profilů pro syntetický a hybridní obsah, včetně profilů pro animaci lidského těla a dále též s podporou nového typu škálování Fine Granular SNR Scalability (FGS). Tvůrci se snažili definici profilů připravit na různé použití, takže se následným vývojem celkový počet profilů nakonec vyšplhal až na počet 19, což mohlo být spíše na škodu, než ku prospěchu věci. Nejoblíbenějším se však stal profil Advanced Simple Profile (ASP) jehož vlastností využila i řada kodeků (DivX, XviD, 3ivX, WMV) [4].

Standard MPEG-4 AVC je v této oblasti podstatně skromnější. Má totiž definovány pouze 3 profily, viz Tab. 3.2, konkrétně se jedná o tyto [12]:

- Baseline – podporuje Intra a Inter kódování (proužky I a P) a Context-Adaptive VLC.
- Main – přidává podporu prokládaného videa, Inter kódování pomocí proužků B a Context-Base aritmetické kódování.
- Extended – bez podpory prokládaného videa a aritmetického kódování, avšak vyšší odolnost proti chybám, používá přepínání datových toků. Zavádí proužky SI a SP (switching), které jsou přenášeny paralelně k proužkům profilu Main ve druhém datovém toku. Aplikace tak může přepínat mezi těmito dvěma toky.

Tab. 3.2: Nástroje profilů standardu MPEG-4 AVC

Kódovací nástroje	Dostupnost nástrojů v profilech		
	Baseline	Main	Extended
I a P proužky	X	X	X
CAVLC kódování	X	X	X
Skupiny proužků	X		X
Redundantní proužky	X		X
B proužky		X	X
Vážená predikce		X	X
Prokládané video		X	
CABAC kódování		X	
SP a SI proužky			X
Rozdělení dat			X

3.3.2 Kódování obrazu

Jak již bylo uvedeno, standard MPEG-4 Visual přinesl důraz na kódování objektů a nikoliv jen celých snímků. Kodér tak extrahuje ze scény jednotlivé video objekty (VO) a ty kóduje společně s kompenzací jejich pohybu pomocí kódování tvarů (shape coding, alfa bloky). Scéna je tak prakticky vytvořena z několika video objektů, které jsou představovány určitou oblastí libovolného tvaru, která může existovat libovolně dlouhou dobu. Instance takového objektu v konkrétním časovém okamžiku se nazývá objektová rovina videa (Video Object Plane – VOP) [4]. Použití VO tak dává volnost při definování kvality, časového rozlišení, důležitosti a možnosti manipulace s jednotlivými objekty během kódování.

Standard MPEG-4 Visual dal vzniknout rovněž novým typům objektových obrazových rovin definovaných v souvislosti s použitím VO:

- I-VOP – Intra-kódovaný obdélníkový VOP. Obdélníkové rámce se kódují bez použití predikce, následuje použití DCT a kvantování.
- P-VOP – Inter-kódovaný obdélníkový VOP. Použití predikce z předchozího referenčního I-VOP nebo P-VOP. Kompenzace pohybu pracuje s makrobloky o velikosti 16x16 pixelů, možnost subpixelové přesnosti pohybového vektoru.
- B-VOP – Inter-kódovaný obousměrné predikovaný VOP. Pro zlepšení účinnosti se používá kompenzace pohybu pomocí obousměrné predikce, nevýhodou je zvýšení požadavků na paměťovou kapacitu a zpoždění kódování.

Při kódování VOP libovolného tvaru je současně s dalšími parametry (vektory pohybu, textura) kódována i informace o tom, které pixely náleží danému VO. To je definováno pomocí Binary Alpha Plane (BAP), což je prakticky realizováno maticí o stejné velikosti jako VOP, kde binární prvky vyznačují zda daný pixel náleží VO. Dále též může být využito šedoškálové kódování tvaru, kdy BAP matice neobsahuje pouze dvě hodnoty, ale celý rozsah (0 – 255), jejichž hodnota určuje průhlednost daného pixelu.

Profil Fine Granular Scalability doplnil dosavadní typy časového, prostorového a kvalitativního škálování o typ, jenž umožnil přerušit přenos rozšiřující vrstvy v libovolném

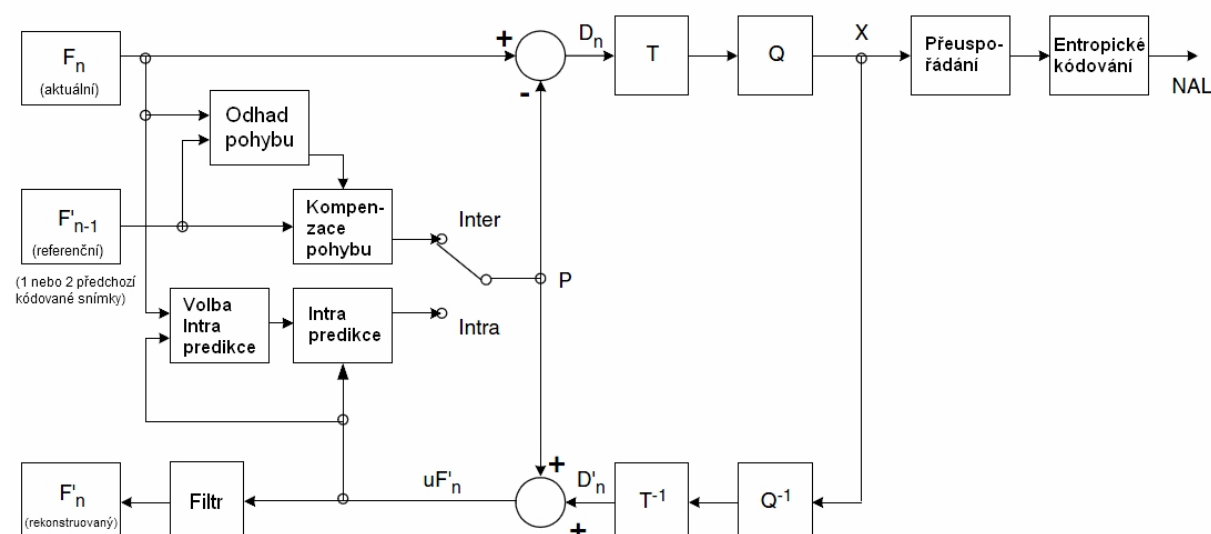
okamžiku, při využití všech do té doby přijatých bitů, což dovoluje dosáhnout nejvyšší možné kvality, jakou jen bitový tok vzhledem k okolnostem dovoluje. Umožňuje tak odstupňování pomocí postupného zvyšování kvality videosekvence po malých krocích. Na první pohled je podobný kvalitativnímu škálování, avšak jeho hlavní výhodou je, že všechny bity přijaté do okamžiku přerušení přenosu jsou pak dekodérem použity pro zvýšení kvality obrazu přeneseného v základní vrstvě.

Standard MPEG-4 AVC se zaměřuje především na efektivní a robustní kódování obdélníkových snímků. Nově též zavedl využití proužků (slices), které jsou tvořeny určitým množstvím makrobloků. Pro predikci kodér i dekodér uchovává jeden nebo dva seznamy referenčních snímků (*list 0*, *list 1*), které již byly kódovány/dekódovány [4]. Rozdělení jednotlivých typů proužků je pak následující:

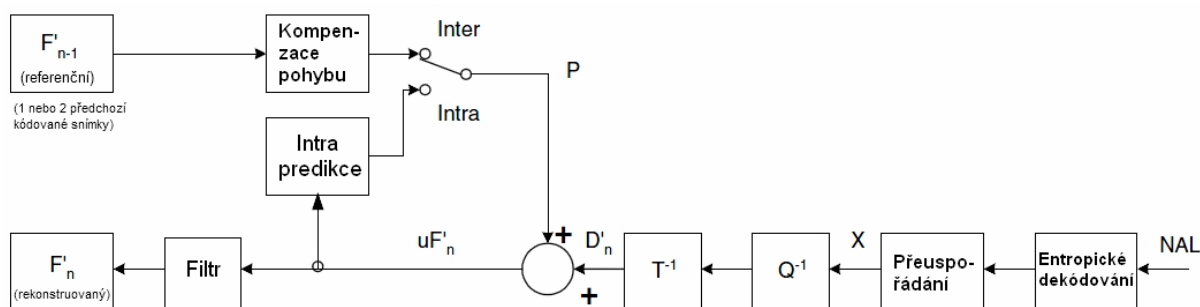
- I (Intra) – obsahuje pouze makrobloky typu I, každý blok je predikován z předchozích kódovaných dat v rámci téhož proužku.
- P (Predicted) – obsahuje makrobloky typu P a/nebo makrobloky typu I. Makrobloky typu P jsou predikovány z jednoho referenčního snímku
- B (Bi-predictive) – obsahuje makrobloky typu B a/nebo makrobloky typu I. Pro predikci lze využít oba seznamy referenčních snímků (*list 0*, *list 1*).
- SI (Switching I) – možnost přepínání mezi datovými toky, obsahuje makrobloky typu SI (zvláštní typ Intra kódovaných makrobloků).
- SP (Switching P) – možnost přepínání mezi datovými toky, obsahuje P a/nebo I makrobloky.

Proužky typu B se nachází jen v profilech Extended a Main, proužky typu SI a SP pak pouze v profilu Extended [12].

Obsahem proužků jsou makrobloky, které jsou v proužcích typu I následně Intra predikovány na základě předchozích kódovaných vzorků v témže proužku. Využívá se oblasti o velikosti 16x16 a 4x4 pro jasové složky a 8x8 pro chrominanci. U proužků typu P, které mohou obsahovat Intra kódované, Inter kódované a vynechané makrobloky, dochází při použití Inter predikce k pohybové kompenzaci s čtvrtpixelovou přesností z předchozích kódovaných snímků. Po predikci pak následuje transformace, kvantování a entropické kódování [4].



Obr. 3.6: Blokové schéma kodéru standardu MPEG-4 AVC [4]



Obr. 3.7: Blokové schéma dekodéru standardu MPEG-4 AVC [4]

3.3.3 Intra predikce

V Intra režimu kódování standardu MPEG-4 AVC jsou predikované bloky vytvořeny na základě dříve kódovaných a rekonstruovaných bloků. Kódovány jsou rezidua, neboli rozdíl mezi předpovězenými bloky a skutečnými bloky. U jasových vzorků může být predikce bloku tvořena pro bloky o velikosti 4x4, nebo pro makrobloky 16x16. Celkem je k dispozici devět možných módů pro předpověď jasového bloku o velikosti 4x4 a čtyři módy pro bloky 16x16. Kodér pak zpravidla volí režim predikce pro každý blok dle míry shody tak, aby byl minimalizován rozdíl mezi predikcí a originálním snímkem [4].

M	A	B	C	D	E	F	G	H
I	a	b	c	d				
J	e	f	g	h				
K	i	j	k	l				
L	m	n	o	p				

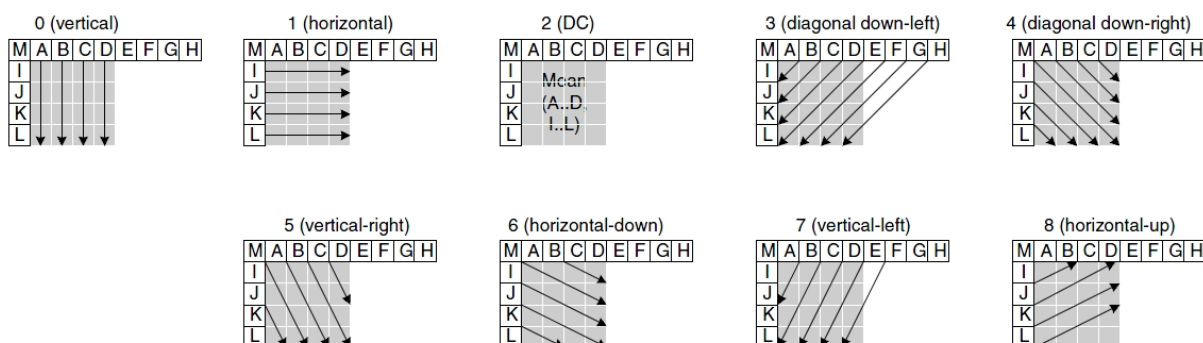
Obr. 3.8: Označení predikovaných vzorků (4x4) [4]

Při Intra kódování bloků o velikosti 4x4 může dojít k situaci, že nejsou k dispozici všechny vzorky A-M. To v módu 2 (DC predikce) způsobí, že predikce je ovlivněna v závislosti na množství dostupných vzorků. V ostatních módech je vyžadována dostupnost všech potřebných vzorků. Jestliže však nebyly dekódovány pouze vzorky E, F, G, H, mohou být jejich hodnoty nahrazeny zkopírováním hodnoty ze vzorku D [4].

Úplný seznam módů pro bloky o velikosti 4x4 je následující:

- Múd 0 (Vertical) – horní vzorky A, B, C, D jsou extrapolovány vertikálně
- Múd 1 (Horizontal) – levé vzorky I, J, K, L jsou extrapolovány horizontálně
- Múd 2 (DC) – všechny vzorky bloku jsou predikovány z průměru vzorků A-D a I-L.
- Múd 3 (Diagonal Down-Left) – vzorky jsou interpolovány pod úhlem 45° mezi dolním levým a horním pravým
- Múd 4 (Diagonal Down-Right) – vzorky jsou extrapolovány pod úhlem 45° dolů doprava

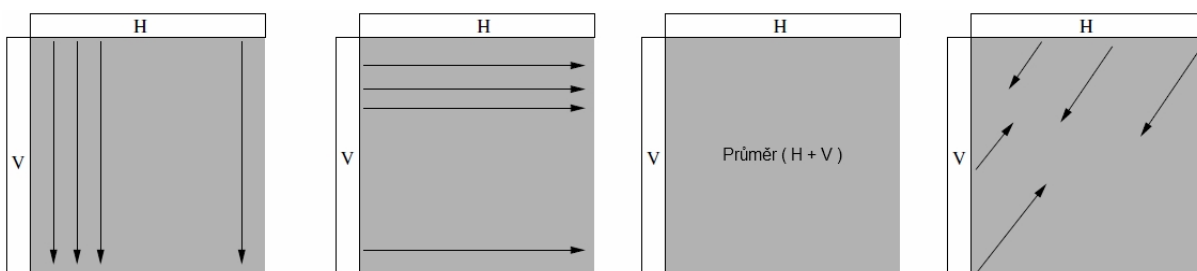
- Mód 5 (Vertical-Right) - vzorky jsou extrapolovány pod úhlem $26,6^\circ$ doleva od vertikálních
- Mód 6 (Horizontal-Down) - vzorky jsou extrapolovány pod úhlem $26,6^\circ$ pod horizontálními
- Mód 7 (Vertical-Left) - vzorky jsou extrapolovány (nebo interpolovány) pod úhlem $26,6^\circ$ napravo od vertikálních
- Mód 8 (Horizontal-Up) - vzorky jsou interpolovány pod úhlem $26,6^\circ$ nad horizontální



Obr. 3.9: Módy jasové predikce bloků 4x4 [4]

U predikce jasových vzorků pro bloky 16x16 se postupuje obdobným způsobem, s tím rozdílem, že predikce jasových složek může být provedena během jedné operace a že celkový počet módů je čtyři, a to [4]:

- Mód 0 (Vertical) – extrapolace z horních vzorků (H)
- Mód 1 (Horizontal) – extrapolace z levých vzorků (V)
- Mód 2 (DC) – všechny vzorky bloku jsou predikovány z průměru horních a levých vzorků ($H + V$).
- Mód 3 (Plane) – vzorky bloku jsou vyplněny pod úhlem 45° mezi horními (H) a levými vzorky (V).



Obr. 3.10: Módy jasové Intra predikce bloků 16x16 (Vertical, Horizontal, DC, Plane) [5]

Existuje ještě jeden kódovací mód, odlišný od výše uvedených. Je označen I_PCM a vyznačuje se přímým přenosem vzorků obrazu bez provedení predikce a transformace. To může být výhodné v určitých případech, jako je neobvyklý obsah obrázku, nebo nízká hodnota kvantovacího parametru, kdy by tradiční metody zpracování obrazové informace (Intra predikce, transformace, kvantování a entropické kódování) nepřinesly dostatečnou účinnost kódování [4].

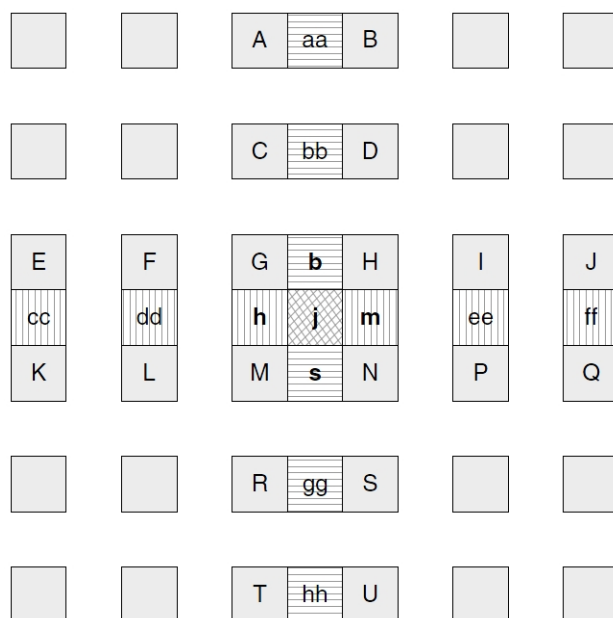
Pro bloky chrominančních složek probíhá predikce v blocích o velikost 8x8 vzorků, a to pomocí čtyřech módů, obdobným způsobem jako u predikce bloků 16x16. Bloky pro predikci jsou: DC (mód 0), horizontal (mód 1), vertical (mód 2) a plane (mód 3). Predikce bloků u obou barvonosných složek probíhá totožným predikčním módem [4].

3.3.4 Inter predikce

Při Inter kódování ve standardu MPEG-4 AVC dochází k predikci z jednoho nebo více předchozích dekódovaných snímků nebo proužků s využitím blokové kompenzace pohybu. Podstatným rozdílem oproti předchozím standardům je podpora bloků o velikosti v rozmezí od 4x4 do 16x16. Toto schéma se nazývá stromově strukturovaná kompenzace pohybu (tree structured motion compensation). Další inovací je též jemné subpixelové určení pohybových vektorů se čtvrtpixelovou přesností pro jasové složky [4].

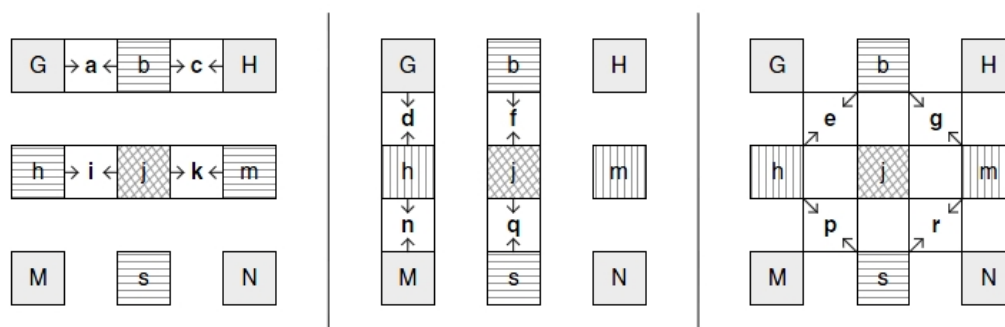
Jasové složky každého makrobloku (16x16 vzorků) mohou být rozděleny pomocí čtyř způsobů, a to na: bloky o velikosti: 16x16, 16x8, 8x16 a 8x8. Při rozdělení na bloky 8x8 může dojít ještě k následnému dělení na sub-bloky o velikostech: 8x8, 8x4, 4x8 a 4x4. Toto dělení makrobloků má za následek velké množství možných kombinací velikostí bloků uvnitř makrobloků. Tato variabilita je výhodná pro kódování, kdy na souvislé homogenní oblasti je použito dělení na větší bloky, kdežto u oblastí s velkou pohybovou aktivitou (více detailů, hrany objektů, apod.) je výhodnější použít více malých bloků. Vzroste sice počet vektorů pohybu, avšak množství reziduí ke kódování zůstane ve vyhovujících mezích [4].

Pro vzniklé definované bloky jsou následně určovány vektory pohybu. Ty mohou být stanovovány z referenčních bloků stejné velikosti, nebo může být využito subpixelové přesnosti. V režimu subpixelové přesnosti dojde nejprve k získání půlpixelových vzorků dle Obr. 3.11 interpolací z hodnot sousedních pixelů pomocí filtru typu FIR (Finite Impulse Response) s vahami (1/32; -5/32; 20/32; 20/32; -5/32; 1/32) [4].



Obr. 3.11: Interpolace jasových vzorků na půlpixelových pozicích [4]

Poté co jsou k dispozici vzorky pro půlpixelovou přesnost odhadu pohybu, dojde k výpočtu vzorků pro čtvrtpixelovou přesnost. To je určeno pomocí lineární interpolace, jako poloviční hodnota součtu dvou sousedních vzorků, což ilustruje Obr. 3.12 .



Obr. 3.12: Interpolace jasových vzorků na čtvrtpixelových pozicích [4]

Při následném kódování vektorů pohybu je současně prováděna předpověď z dříve zakódovaných pohybových vektorů bloků. Problémem není ani odlišná velikost sousedních bloků. S výhodou se tak využije faktu, že sousední vektory pohybu jsou značně korelované. To přináší další úsporu v bitovém toku signálu [4].

3.3.5 Rekonstrukční (deblocking) filtr

Rekonstrukční filtr je aplikován na dekódované makrobloky z důvodu omezení blokového zkreslení. Ve standardu MPEG-4 AVC je povinný, avšak u dřívějších standardů tomu tak zpravidla nebylo, jak uvádí následující Tab. 3.3 [28]:

Tab. 3.3: Použití rekonstrukčního filtru ve standardech [28]

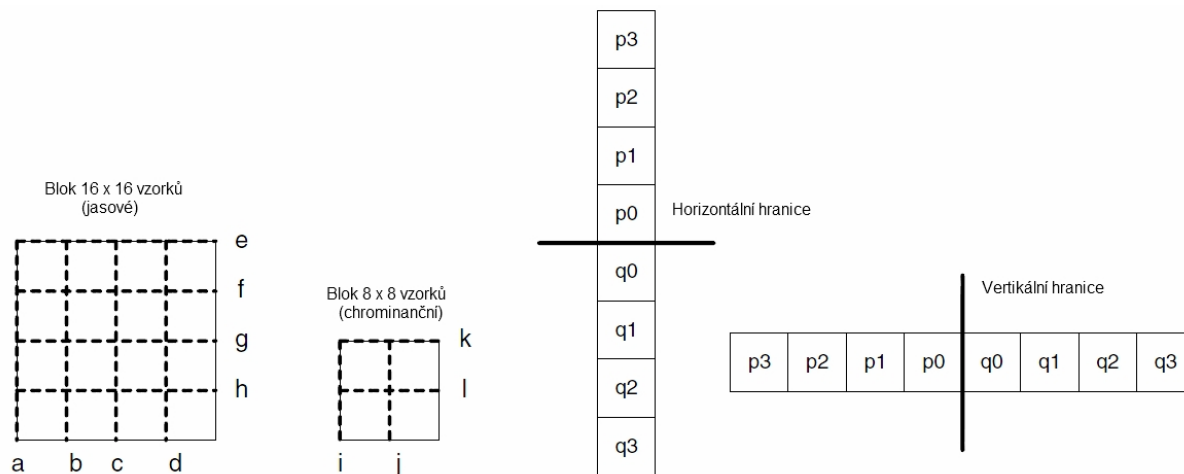
Standard	Rekonstrukční (deblocking) filtr
H.261	Volitelný filtr smyčky
MPEG-1	Bez filtru
MPEG-2	Bez filtru, často použito post-processing zpracování
H.263	Bez filtru
MPEG-4	Volitelný filtr smyčky, post-processing zpracování doporučeno
H.264	Povinný filtr smyčky, post-processing zpracování může být použito

V kodéru standardu MPEG-4 AVC je filtr aplikován na všechny dekódované makrobloky po inverzní transformaci (před rekonstrukcí a uložením makrobloku pro další predikci) a v dekodéru před rekonstrukcí a zobrazením makrobloku. Filtr vyhlazuje hrany bloků, na kterých jsou patrné ostré přechody, což má významný vliv na výsledný vzhled snímku a na subjektivní hodnocení pozorovatele. Objektivní hodnocení, parametr PSNR, však příliš významně ovlivněn není [28].

Filtrovaný obraz je použit pro předpověď pohybové kompenzace budoucích snímků, což může zlepšit výkon komprese, neboť filtrovaný obraz je často více podobný originálu, než rekonstruovaný snímek s blokovými artefakty. Kodér může v průběhu činnosti změnit míru filtrování, nebo dokonce filtr zakázat.

Filtrování je prováděno na hranách bloků o velikosti 4x4 v makrobloku (kromě hran na hranicích proužků), a to následujícím způsobem:

1. Filtrování čtyř vertikálních hranic jasové složky (v pořadí a, b, c, d)
2. Filtrování čtyř horizontálních hranic jasové složky (v pořadí e, f, g, h)
3. Filtrování dvou vertikálních hranic barvonosných složek (i, j)
4. Filtrování dvou horizontálních hranic barvonosných složek (k, l)



Obr. 3.13: Vyznačení svislých a vodorovných hranic mezi vzorky makrobloků [4]

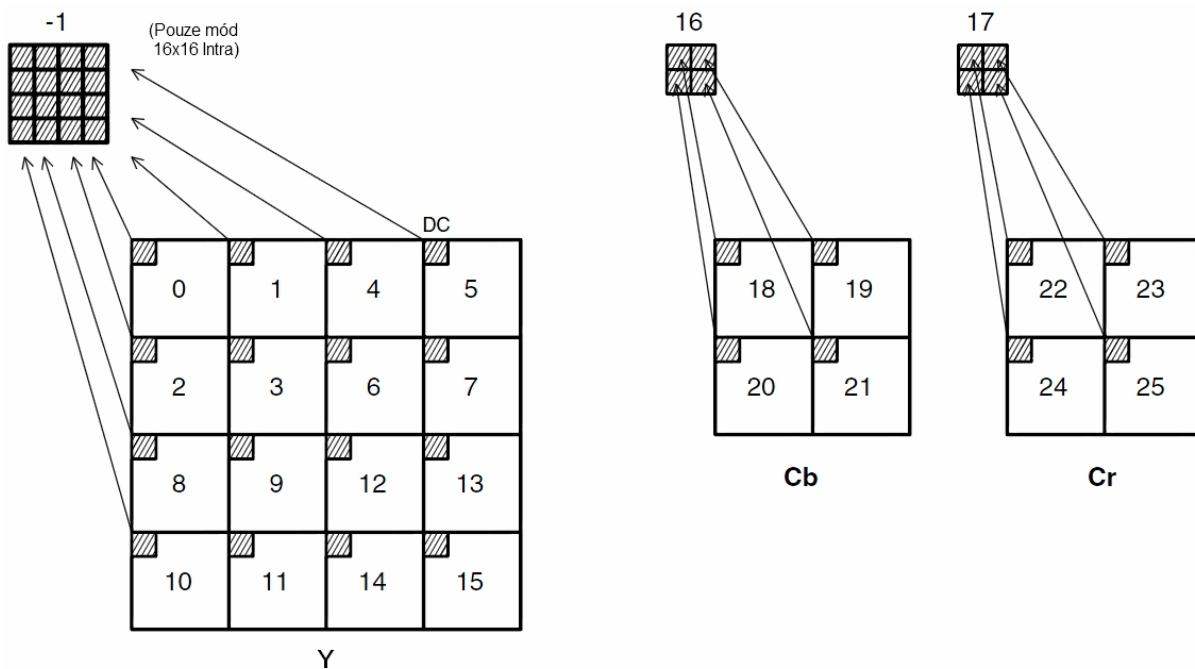
Každá operace filtrování má vliv až na tři vzorky na každé straně hranice. Míra filtrování závisí na aktuálním kvantování, kódovacích módech sousedních bloků a na odlišnosti obrazových vzorků na hranicích bloků [4].

3.3.6 Transformace a kvantování

Standard MPEG-4 AVC používá tři typy transformací v závislosti na typu reziduálních dat, jenž mají být kódována. Jedná se konkrétně o tyto:

- Hadamardova transformace bloků 4x4 jasových *DC* koeficientů makrobloků Intra predikovaných v módu 16x16
- Hadamardova transformace bloků 2x2 chrominančních *DC* koeficientů (v každém makrobloku)
- Transformace založená na DCT pro všechny ostatní 4x4 bloky reziduí

Přenos dat v případě makrobloku kódovaného v Intra módu 16x16 probíhá v pořadí nejprve od *DC* koeficientů každého bloku 4x4 jasové složky. Tento blok 4x4 *DC* koeficientů je označen -1. Dále jsou přeneseny bloky reziduí, avšak bez *DC* koeficientů. Následují bloky 16 a 17, jenž obsahují bloky *DC* koeficientů chrominančních složek o velikosti 2x2. Závěrem dojde k přenosu zbývajících bloků reziduí bez *DC* koeficientů chrominančních složek, viz Obr. 3.14 [4].



Obr. 3.14: Pořadí čtení reziduí v makrobloku [4]

3.3.6.1 Kódování 4x4 bloků reziduí

Standard MPEG-4 AVC provádí transformaci založenou na DCT na 4x4 blocích dat reziduí (označených 0-15 a 18-25 na Obr. 3.14) po provedení Intra predikce, či predikce kompenzace pohybu. Odlišnost transformace založené na DCT od „pravé“ DCT spočívá v celočíselnosti transformace (všechny operace mohou být provedeny pomocí celočíselných výpočtů), dále v implementaci jádra transformace jen za pomoci sčítání a binárního posunu, i v tom, že škálovací násobení (část transformace) je součástí kvantizéru [4].

Inverzní kvantování a operace inverzní transformace mohou být provedeny použitím 16-bitových celočíselných výpočtů, pouze s jednou operací násobení na koeficient, a to bez ztráty přesnosti. Ke zjednodušení implementace transformace založené na DCT se využilo několika úprav, a tak vztah (3.1) poskytuje téměř shodné výsledky jako DCT, avšak s vědomím určitého rozdílu mezi výstupy obou realizací DCT [4].

$$Y = C_f X C_f^T \otimes E_f = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \begin{bmatrix} X \\ \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 1 & -1 \end{bmatrix} \otimes \begin{bmatrix} a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \end{bmatrix} \quad (3.1)$$

, kde CXC^T je jádro 2D transformace. E je matice váhovacích prvků a symbol $\otimes$ značí, že každý prvek CXC^T je násoben váhovacím faktorem na stejné pozici v matici E . Konstanty a a b jsou definovány následovně: $a = \frac{1}{2}$, $b = \sqrt{\frac{2}{5}}$

Inverzní transformace je dána vztahem (3.2), který definuje tuto transformaci jako posloupnost aritmetických operací [4]:

$$Y = C_i^T (Y \otimes E_i) C_i = \begin{bmatrix} 1 & 1 & 1 & \frac{1}{2} \\ 1 & \frac{1}{2} & -1 & -1 \\ 1 & -\frac{1}{2} & -1 & 1 \\ 1 & -1 & 1 & -\frac{1}{2} \end{bmatrix} \left(\begin{bmatrix} X \end{bmatrix} \otimes \begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix} \right) \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{2} & 1 \\ 1 & \frac{1}{2} & -\frac{1}{2} & -1 \\ 1 & -\frac{1}{2} & -1 & 1 \\ \frac{1}{2} & -1 & 1 & -\frac{1}{2} \end{bmatrix} \quad (3.2)$$

3.3.6.2 Kvantování

Pro kvantování v dopředném směru kvantizéru standardu MPEG-4 AVC platí základní vztah:

$$Z_{ij} = \text{round} \left(\frac{Y_{ij}}{Qstep} \right) \quad (3.3)$$

, kde Y_{ij} je koeficient transformace popsané výše. $Qstep$ je velikost kvantovacího kroku a Z_{ij} je kvantovaný koeficient. Operace zaokrouhlení nemusí být nutně k nejbližšímu celému číslu, neboť zaokrouhlení „dolů“ k nejbližšímu menšímu celému číslu může pozorovateli poskytnout lepší kvalitativní vjem [4].

Standard MPEG-4 AVC definuje celkem 52 hodnot kvantovacího kroku $Qstep$, jenž jsou indexovány kvantovacím parametrem QP , viz Tab. 3.4. Hodnota $Qstep$ zdvojnásobuje svou velikost pro každý násobek šesti parametru QP . Široký rozsah spektra kvantovacích kroků dává kodéru možnost pružně a přesně měnit poměr mezi bitovým tokem a kvalitou. Hodnoty parametru QP v daném rozsahu 0-51 se mohou lišit pro jasové a pro chrominanci složky; mohou být ručně definovány [4].

Tab. 3.4: Velikost kvantizačního kroku ve standardu MPEG-4 AVC [4]

QP	0	1	2	3	4	5	6	7	8	9	10	11	12
Qstep	0,625	0,6875	0,8125	0,875	1	1,125	1,25	1,375	1,625	1,75	2	2,25	2,5
QP	13	14	15	16	17	18	19	20	21	22	23	24	25
Qstep	2,75	3,25	3,5	4	4,5	5	5,5	6,5	7	8	9	10	11
QP	26	27	28	29	30	31	32	33	34	35	36	37	38
Qstep	13	14	16	18	20	22	26	28	32	36	40	44	52
QP	39	40	41	42	43	44	45	46	47	48	49	50	51
Qstep	56	64	72	80	88	104	112	128	144	160	176	208	224

Rovněž na operaci kvantování a inverzního kvantování je kladen požadavek na eliminaci výpočetně náročných operací, jako je např. dělení, a proto jsou preferovány a implementovány funkce provádějící ekvivalentní činnost pomocí operací jako je sčítání a binární posun.

Ve standardu MPEG-4 Visual není operace kvantování přímo definovaná standardem, avšak detailně je popsán postup inverzního kvantování. Tento standard uvádí dvě metody pro

tuto operaci. Společná jim je závislost na kvantovacím parametru QP , který může nabývat hodnot od 1 do 31, přičemž vyšší hodnoty tohoto parametru produkují vyšší hodnotu kvantovacího kroku. Výchozí a jednodušší *Metoda 2* inverzního kvantování má pro obnovu DC koeficientů definován následující vztah [4]:

$$DC = DC_Q \cdot dc_scaler \quad (3.4)$$

, kde: DC_Q je kvantovaný koeficient, DC je rekonstruovaný koeficient a dc_scaler je parametr definovaný standardem, viz Tab. 3.5. V režimu krátké hlavičky (short header mode), zajišťujícím kompatibilitu se standardem H.263, je hodnota parametru dc_scaler rovna vždy 8. V ostatních případech je hodnota dc_scaler vypočtena v závislosti na hodnotě QP . Všechny ostatní koeficienty jsou pak vypočteny na základě následujících vztahů [4]:

$$\begin{aligned} |F| &= QP \cdot (2 \cdot |F_Q| + 1) && \text{(jestliže } QP \text{ je lichý a } F_Q \neq 0) \\ |F| &= QP \cdot (2 \cdot |F_Q| + 1) - 1 && \text{(jestliže } QP \text{ je sudý a } F_Q \neq 0) \\ F &= 0 && \text{(jestliže } F_Q = 0) \end{aligned} \quad (3.5)$$

, kde F_Q je kvantovaný koeficient a F je rekonstruovaný koeficient. Znaménko F je stejné jako znaménko F_Q .

Tab. 3.5: Hodnoty parametru dc_scaler v závislosti na rozsahu QP [4]

Typ bloku	$QP \leq 4$	$5 \leq QP \leq 8$	$9 \leq QP \leq 24$	$25 \leq QP$
Jasový	8	$2 \times QP$	$QP + 8$	$(2 \times QP) - 16$
Chrominanční	8	$(QP + 13)/2$	$(QP + 13)/2$	$QP - 6$

Jak je z výše uvedeného zřejmé, jsou i ve standardu MPEG-4 Visual, v *Metodě 2*, všechny koeficienty (s výjimkou Intra DC koeficientu) kvantovány a obnoveny se stejnou hodnotou kvantovacího kroku [4].

V případě využití *Metody 1* inverzního kvantování jsou definovány dvě kvantovací matice, a to pro Intra a Inter kódované bloky. Volba kvantovací metody spočívá v nastavení jednobitové hodnoty návěští *quant_type*, která indikuje použitou *Metodu 1* (*flag* = 1) nebo *Metodu 2* (*flag* = 0) [9].

3.3.6.3 Kódování DC koeficientů jasových složek v Intra módu

Jestliže je makroblok kódován v 16x16 Intra predikčním módu (např. celý jasový makroblok je predikován ze sousedních vzorků), pak každý ze 4x4 bloků reziduí je nejprve transformován pomocí jádrové transformace CXC^T popsané výše. DC koeficienty každého 4x4 bloku jsou poté znovu transformovány použitím 4x4 Hadamardovy transformace ve tvaru [4]:

$$Y_D = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} W_D \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} / 2 \quad (3.6)$$

, kde W_D je blok 4x4 DC koeficientů a Y_D je blok po transformaci. Získané koeficienty Y_D jsou dále kvantovány v závislosti na kvantovacím parametru QP pro získání bloku kvantovaných koeficientů.

V dekodéru je aplikována inverzní Hadamardova transformace dle (3.7), která je dále následována inverzním škálováním [4].

$$W_{QD} = \begin{pmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} Z_D \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \end{pmatrix} \quad (3.7)$$

Rekonstruované DC koeficienty W'_D jsou vloženy na jejich pozice do odpovídajících 4×4 bloků a každý z těchto bloků je inverzně transformován použitím na DCT založené inverzní transformaci $C_i^T W' C_i$.

Tato transformace využívá faktu, že v každém 16×16 Intra kódovaném makrobloku je většina energie koncentrována do DC koeficientů každého 4×4 bloku. Po této doplňující transformaci je energie dále koncentrována do malého počtu významných koeficientů [4].

3.3.7 Entropické kódování

Po provedení transformací a kvantování dochází v kodéru standardu MPEG-4 AVC k přeuspořádání koeficientů z čtvercových bloků o rozměru 4×4 do pole o velikosti 16-ti prvků, a to pomocí cik-cak vyčítání, viz Obr. 2.4. Pokud byl makroblok kódován v Intra módu 16×16 , pak DC koeficienty jednotlivých bloků byly čteny a zpracovány již dříve (a tvoří blok DC koeficientů 4×4), a proto je načteno jen zbylých 15 AC koeficientů, s počátkem od 2. pozice [4].

Před samotným přenosem jsou data všech typů entropicky kódována. Standard AVC podporuje dvě různé metody entropického kódování, a to CAVLC (Context Adaptive Variable Length Coding) a CABAC (Context Adaptive Binary Arithmetic Coding). Z důvodu odlišné koncepce mezi oběma metodami je pak CABAC účinnější než CAVLC. Přičemž samotný CAVLC je lepší než konvenční VLC (Huffmanovo kódování použité v jiných standardech [26]). Základní rysy obou metod jsou následující [4]:

- a) CAVLC – Pokud je kodek v tomto režimu, pak rezidua (zbytková data) jsou kódována pomocí CAVLC, ale ostatní data jsou kódována pomocí jednoduchých Exp-Golomb kódů. Tyto data jsou nejprve vhodně mapovány na Exp-Golomb kódy, v závislosti na typu dat (záhlaví makrobloků, vektorů pohybu, atd.) a následně jsou přenášeny odpovídající kódová slova. Cik-cak skenované kvantované koeficienty bloku reziduí jsou kódovány pomocí CAVLC. Dříve kódované informace sousedních bloků (zpravidla horní a levé bloky) a kódovací status aktuálního bloku určují kontext. Optimalizované VLC tabulky jsou speciálně stanoveny pro každý kontext, k zajištění účinného kódování koeficientů v různých statistických podmínkách.
- b) CABAC – V tomto režimu jsou generovaná data, včetně záhlaví a reziduí, kódována pomocí binárního aritmetického kódování. Vylepšení komprese metody CABAC spočívá v přiřazení čísel s neceločíselnou délkou, v adaptivním pravděpodobnostním odhadu a ve vylepšeném schématu kontextového modelování. Ve standardu AVC je CABAC navržen tím způsobem, že může být implementován s pomocí pouze jednoduchých celočíselných operací a vyhledávacích tabulek, čímž je přizpůsobený pro aplikace s nízkou složitostí.

3.4 H.261

Standard H.261 byl uveden v roce 1989 se zaměřením na videokonference, v podobě doporučení organizace ITU-T (International Telecommunications Union - Telecommunications standardization sector). Byl navržen pro přenos audiovizuálních dat pomocí ISDN linek s rychlostí danou násobky $p \times 64$ kbit/s, kde p je celé číslo v rozsahu od 1 do 30. Tento standard podporuje dvě velikosti snímků: CIF (352x288) a QCIF (176x144), s použitím podvzorkování typu 4:2:0 a byl rovněž prvním, v němž se poprvé objevil koncept makrobloku. Ten byl určen jasovou složkou o rozměrech 16x16 jasových vzorků a 8x8 barvonosných vzorků. Mezi hlavní charakteristické prvky rovněž patří určování vektorů pohybu, DCT bloků 8x8 pixelů a využitím obvodu pro redukci vlivu blokových artefaktů (loop filter). Při kvantování může kodér volit z 32 různých kvantovačů, které se liší velikostí kvantovacího kroku, a to odlišně pro každý makroblok, dle požadované přenosové rychlosti a velikosti DCT koeficientů rozdílového bloku. Rychlost datového toku, a tím i využití vyrovnávací paměti, může být regulováno nejen změnou použitého kvantovače, ale i možností zahazování rámců [12]. Tento standard nepodporuje snímky typu B a rovněž má velmi omezenou velikost pohybových vektorů [19].

3.5 H.263

Tento standard byl vyvinut pro kódování pro velmi nízké přenosové rychlosti, zejména pro videotelefonii s ohledem na možnosti veřejné telefonní sítě. Vychází ze standardu H.261, jenž doplňuje a vylepšuje v efektivitě kódování. H.263 používá pět velikostí snímků, a to: sub-QCIF, QCIF, CIF, 4CIF a 16CIF. K zásadním vylepšením oproti H.261 patří neomezené vektory pohybu (vektor může ukazovat i ven z obrazu), kompenzace pohybu s půlpixelovou přesností (jenž nahrazuje filtr smyčky) a různě velkých bloků a rovněž možnost až čtyř vektorů pohybu na makroblok. Dále také používá snímky PB složené z jednoho P a jednoho B rámce, které jsou kódovány jako jeden. Standard H.263 Annex O rovněž specifikuje možnost škálovatelnosti, jenž je trojího typu:

- časová – základní vrstvu tvoří I a P snímky, rozšiřující vrstvu pak B snímky.
- prostorová – základní vrstva je složena z I a P snímků, rozšiřující vrstva ze snímků EI a EP, jako doplněk do vyššího rozlišení.
- kvalitativní – v základní vrstvě jsou opět I a P snímky, v rozšiřující vrstvě jsou snímky EI a EP. Snímky EI jsou predikovány pouze ze snímků v základní vrstvě, avšak snímky EP jsou predikovány nejen ze základní vrstvy, ale i z předchozích snímků rozšiřující vrstvy. Snímky B se neuplatňují [12].

Standard dále umožňuje využití aritmetického kódování, jenž nabízí účinnější kódování oproti kódování s proměnnou délkou. V průběhu času došlo k definování novější normy H.263+, která doplnila další volitelné kódovací módy a rovněž i typy obrazových formátů [19].

3.6 H.264

Standard H.264 je standard pro video kompresi dle označení ITU-T, který vznikl spoluprací týmu VCEG (ITU-T Video Coding Experts Group) s týmem MPEG (ISO/IEC JTC1 Moving Picture Experts Group). Vzhledem k tomu, že společným dílem obou týmů, označených jako JVT (Joint Video Team) je standard po technické stránce shodný se standardem ISO/IEC 14496-10 – MPEG-4 Part 10, Advanced Video Coding, jenž byl popsán již dříve v kapitole 3.3, nebude zde tedy z tohoto důvodu popisován opakovaně.

4 REALIZACE VYBRANÉHO KODEKU V MATLAB

Pro praktickou realizaci funkce kodéru a dekodéru byly zvoleny funkční bloky ze standardu MPEG-4 AVC (Part 10) a dále pro možnost porovnání i funkce dle MPEG-4 Visual (Part 2). Pro tyto standardy platí, že kodér není jednoznačně definován, avšak je určena přesná podoba kódovaného datového toku i způsob dekódování. Díky znalostem principů ztrátového a bezztrátového kódování tak lze navrhnout blokové schéma kodéru a dekodéru jehož hlavní rysy se shodují v rámci celé rodiny standardů MPEG a rovněž jsou podobné jednotlivým standardům ITU-T H.26x. Ačkoliv se na první pohled zdá být podobnost napříč standardy velmi výrazná (s výjimkou rekonstrukčního filtru), zásadní rozdíly se skrývají uvnitř jednotlivých bloků, které realizují definované funkčnosti jednotlivých profilů konkrétních standardů určitými specifickými technikami a s odlišnými výpočetními nároky.

Pro programovou realizaci byly vybrány zejména bloky, jejichž funkce má zásadní vliv na zpracováváný obrazový signál a u nichž lze změnou parametrů významně ovlivnit dopad na výsledný obrazový signál a ten následně též vyhodnotit.

4.1 POPIS FUNKCE KODÉRU

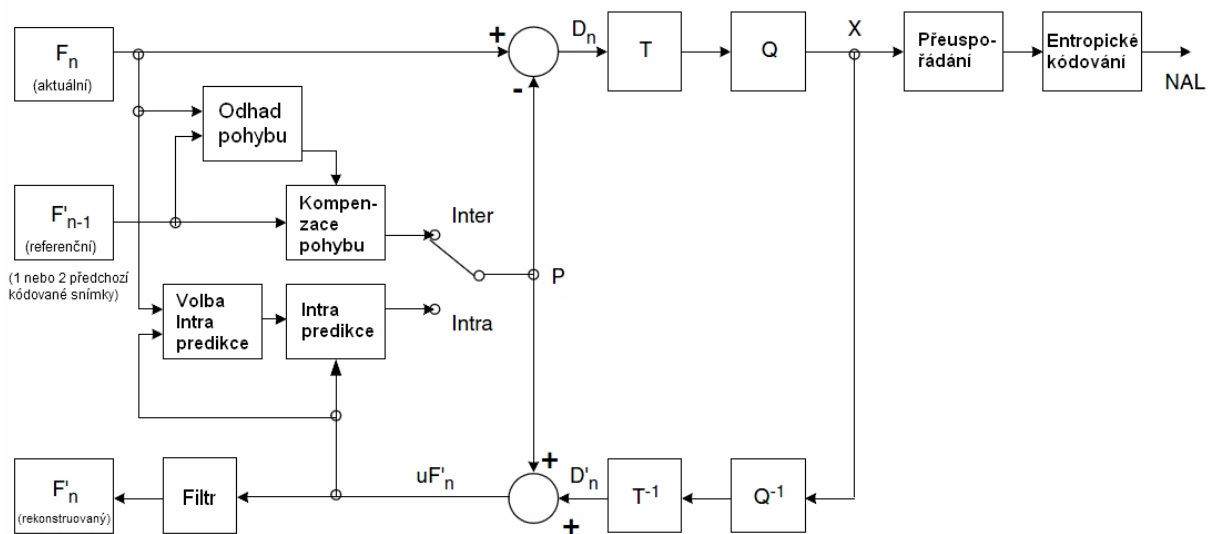
V kodéru standardu MPEG-4 AVC, viz Obr. 4.1, lze pozorovat dvě cesty datového toku. V horní větvi ve směru zleva doprava tzv. dopředný kódovací směr a ve spodní větvi ve směru zprava doleva tzv. rekonstrukční směr. Podobnost rekonstrukční cesty kodéru a dekodéru je tak u obou schémat velmi zřejmá. Před samotným popisem realizovaných funkcí je vhodné uvést zjednodušený průběh kódování obrazového signálu v kodéru [4].

Každý vstupní snímek F_n je zpracováván ve skupinách makrobloků. Každý makroblok je kódován v režimu Intra nebo Inter a pro každý blok v makrobloku je určována predikce na základě rekonstruovaných obrazových vzorků. V režimu Intra je predikce vytvořena ze vzorků v aktuálním proužku, které již byly kódovány, dekódovány a rekonstruovány. Za zmínku stojí, že k predikci se využívají nefiltrované vzorky, ve schématu označené jako uF'_n .

V Inter módu se predikce tvoří pomocí pohybově kompenzované predikce z jednoho nebo dvou referenčních snímků zvolených ze sady seznamu 0 nebo seznamu 1 referenčních snímků. V blokovém schématu kodéru je referenční snímek označen jako F'_{n-1} , ač ve skutečnosti může být predikční reference pro každý makroblok v Inter módu zvolena z dostupných minulých nebo budoucích snímků, které již byly kódovány, rekonstruovány a filtrovány.

Predikce P je odečtena od aktuálního bloku, čímž je vytvořen rozdílový blok diferencí D_n , který je následně transformován pomocí blokové transformace a kvantován pro vytvoření sady kvantovaných transformovaných koeficientů, které jsou následně přeuspořádány a entropicky kódovány. Koeficienty po entropickém kódování, společně s doplňkovými informacemi nutnými pro dekódování každého bloku v rámci makrobloku (režim predikce, kvantovací parametr, informace o vektoru pohybu) tvoří komprimovaný bitový tok, který přechází do vrstvy NAL (Network Abstraction Layer) pro účely přenosu nebo záznamu.

Stejně jako v kodéru probíhá kódování a vysílání každého bloku v makrobloku, tak současně probíhá i operace opačná – dekódování, tj. rekonstrukce, k získání reference pro další predikce. Koeficienty X jsou inverzně kvantovány (Q^{-1}) a inverzně transformovány (T^{-1}), a tak je vytvořen blok diferencí D'_n . Predikovaný blok P je sečten s diferencí D'_n a výsledkem se stává rekonstruovaný blok uF'_n , což značí dekódovanou a nefiltrovanou podobu originálního bloku. Filtr je aplikován k omezení vlivu blokových artefaktů. Rekonstruovaný referenční snímek je vytvořen z posloupnosti bloků F'_n [4].

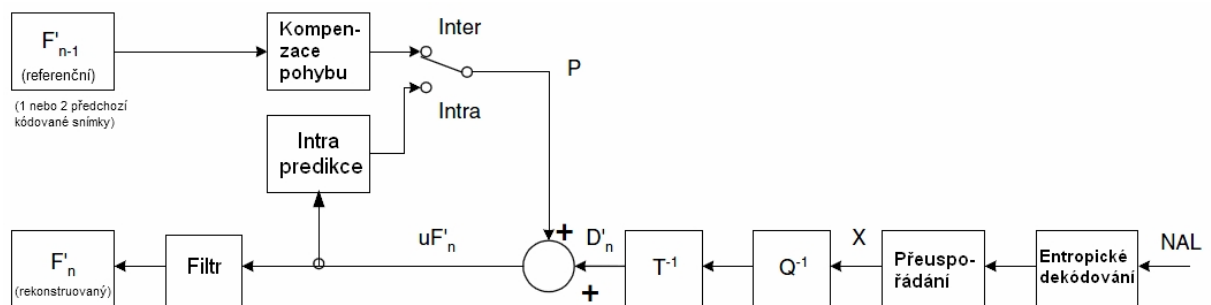


Obr. 4.1: Blokové schéma kodéru standardu MPEG-4 AVC [4]

4.2 POPIS FUNKCE DEKODÉRU

Dekodér, znázorněný na Obr. 4.2, přijímá komprimovaný bitový tok z NAL a provádí entropické dekódování a přeuspořádání. Tímto je získána sada kvantovaných koeficientů X , které jsou následně inverzně kvantovány a inverzně transformovány. Je získán blok diferenci D'_n , shodný s blokem D'_n ve zpětné větvi kodéru.

Použitím informací ze záhlaví, dekódovaných z bitového toku, dekodér vytvoří predikovaný blok P , totožný s originální predikcí sestavenou v kodéru. Predikce P je sečtena s diferencí D'_n , a vznikne tak uF'_n , který je poté filtrován. Výsledkem je dekódovaný blok F'_n , jenž se podobá, či odpovídá, originálnímu kódovanému snímku [4].



Obr. 4.2: Blokové schéma dekodéru standardu MPEG-4 AVC [4]

4.3 POPIS REALIZACE APLIKACE A JEJICH PARAMETRŮ

Pro tvorbu grafického rozhraní a funkcí, jenž v kodéru a dekodéru vykonávají požadované algoritmy bylo použito především literatury [15] a [16] a pro praktickou realizaci dále též zdrojů z [22], [23] a [24]. Zdrojový kód byl napsán a odladěn v prostředí Matlab verze 7.4.0.287 (R2007a) za pomoci volně dostupných obrazových sekvencí z [25]. Jednotlivé funkční části jsou odděleny do samostatných souborů, které funkčně zastřešuje aplikace *kodek.m*, která jednotlivé funkce za běhu volá a zobrazuje grafický i numerický výstup. Popis jednotlivých funkcí následuje.

Vzhled grafického rozhraní aplikace představuje Obr. 4.3. Na něm je patrné, že nezbytnými uživatelskými vstupy jsou obrázky ve formátu RGB, kterým je uživatelsky přiřazena role I, B a P snímku při následném zpracování algoritmem kodeku. Dále je možné volit mezi mnoha možnými kombinacemi zpracování, kdy v každé graficky oddělené sekci lze zvolit s jakým nastavením mají být obrazové snímky zpracovány. Tím lze docílit nastavení kódování nejen dle standardu MPEG-4 AVC (Part 10), ale rovněž pro možnost porovnání i dle MPEG-4 Visual (Part 2), ba dokonce i pro jiné kódovací kombinace.

Realizace kodeku MPEG-4 (AVC & Visual)

Snímek typu I: C:\sekvence\foreman\foreman001.bmp Procházet

Snímek typu B: C:\sekvence\foreman\foreman002.bmp Procházet

Snímek typu P: C:\sekvence\foreman\foreman003.bmp Procházet

Podvzorkování: ☐ 4:1:1 ☒ 4:2:0 ☐ 4:2:2 ☐ 4:4:4

Transformace: ☐ DCT 8x8 ☐ DCT 4x4 ☐ Integer DCT 4x4 ☒ Integer DCT 4x4 + Hadamard

Kvantování: ☐ dle MPEG-4 Visual ☒ dle MPEG-4 AVC ☐ Vlastní --> Definovat

Kvantovací parametr QP: 20 => QStep = 6.5 ; dc_scaler = 28

Intra predikce: ☐ 4x4 H ☐ 4x4 V ☐ 4x4 DC ☐ 16x16 H ☐ 16x16 V ☒ 16x16 DC ☐ I_PCM

Inter predikce (velikost bloku na vektor): ☒ 4x4 ☐ 8x8 ☐ 16x16

Typ prohledávacího algoritmu: ☐ Full Search ☒ Three Step Search ☐ Nearest Neighbour S.

Přesnost vektorů pohybu: ☐ Pixelová ☐ Půlpixelová ☒ Čtvrtpixelová

Zpracovat a zobrazit Info Konec

Obr. 4.3: Vzhled grafického rozhraní realizování aplikace

Pro zpracování a zobrazení výstupů jsou v následujícím textu použity s oblibou používané obrazové sekvence „Football“, „Foreman“, „Garden“ a „Missa“, jejichž rastrová podoba ve formátu RGB je uvedena na Obr. 4.4.



Obr. 4.4: Ukázka snímků použitých obrazových sekvencí [25]

4.3.1 Vstupní podvzorkování

Po ručním udělení pokynu ke zpracování a zobrazení, pomocí stejnojmenného tlačítka, dochází v aplikaci, stejně jako ve skutečném kodéru, nejprve k načtení zadaných snímků (souborů) a k jejich převodu z rastrového formátu RGB do formátu YCbCr. Zpracování vstupních snímků zajišťuje funkce *podvzorkuj.m*, která ze zadaných parametrů, kterými jsou obrazové soubory a způsob podvzorkování, provede požadované podvzorkování, zobrazení jednotlivých složek a vypíše minimální a maximální hodnotu vzorku ve složce. Hlavní aplikaci vrací zpět jasové složky zadaných souborů. Pro zpracování je možnost výběru ze čtyř dostupných typů podvzorkování, pomocí nichž je určen způsob podvzorkování chrominančních složek. Jsou definovány tyto formáty:

4:4:4 – všechny tři matice složek mají plné vertikální i horizontální rozlišení, nedochází ke ztrátě informace podvzorkováním. U barvonosných složek Cb a Cr platí, že čím světlejší daný obrázek zobrazující tuto složku je, tím více je daná složka v obrázku zastoupena.



Obr. 4.5: Formát 4:4:4 – grafický výstup algoritmu, složky Y, Cb, Cr

4:2:2 – matice jasové složky má úplné prostorové rozlišení. Matice chrominančních složek Cb a Cr mají plné vertikální rozlišení, ale jen poloviční horizontální rozlišení. Toho je prakticky dosaženo tím způsobem, že pro výslednou matici je vynechán každý sudý horizontální barvonosný vzorek ze zdrojové matice.



Obr. 4.6: Formát 4:2:2 – grafický výstup algoritmu, složky Y, Cb, Cr

4:2:0 – matice jasové složky má úplné prostorové rozlišení. Matice chrominancních složek Cb a Cr mají poloviční vertikální i horizontální rozlišení. Pro výslednou matici chrominancních vzorků je z původní zdrojové matice vynechán každý sudý horizontální i vertikální barvosný vzorek.



Obr. 4.7: Formát 4:2:0 – grafický výstup algoritmu, složky Y, Cb, Cr

4:1:1 – matice jasové složky má opět úplné prostorové rozlišení. Avšak matice chrominancních složek Cb a Cr mají plné vertikální rozlišení, ale jen čtvrtinové horizontální rozlišení. Toho je prakticky dosaženo tím, že do výsledné obrazové matice je zapsán jen každý čtvrtý horizontální barvosný vzorek ze zdrojové matice.



Obr. 4.8: Formát 4:1:1 – grafický výstup algoritmu, složky Y, Cb, Cr

Následně, po provedení vstupního podvzorkování, je u jasových matic jednotlivých snímků, v případě potřeby, provedena úprava velikosti snímku tak, aby jeho strany byly násobkem šestnácti. To je provedeno z toho důvodu, aby nedošlo k ovlivnění následných operací vlivem pouze částečného zaplnění makrobloků vzorky.

4.3.2 Kódování snímku typu I metodou Intra

Pro intra kódovaný snímek typu I se uplatňují operace Intra predikce, DCT (celočíslné - integer - DCT), Hadamardovy transformace, kvantování a jim odpovídající inverzní funkce. Vypracovaná aplikace nabízí možnost výběru režimu Intra predikce z následujících voleb:

- Horizontální predikce s velikostí bloku 4x4 nebo 16x16 (funkce *intraH.m*)
- Vertikální predikce s velikostí bloku 4x4 nebo 16x16 (funkce *intraV.m*)
- DC predikce daná průměrem horních a levých dříve dekódovaných vzorků, predikce je provedená na blocích o velikosti 4x4 nebo 16x16 (funkce *intraDC.m*)
- I_PCM – použitím této volby dochází k přímému přenosu vzorků obrazu bez provedení predikce a transformace. To může být výhodné v případech, jako je neobvyklý obsah obrázku, nebo nízká hodnota kvantovacího parametru. Předěje se tak stavu, kdy by použití celého řetězce zpracování nepřineslo dostatečnou efektivitu a účinnost kódování.

Všechny metody Intra predikce pro bloky 4x4 a 16x16 dostupné ve standardu MPEG-4 AVC představuje Obr. 3.9 a Obr. 3.10 a jsou popsány v kapitole 3.3.3. Princip této metody spočívá v možnosti redukce množství přenášené informace použitím již dekódovaných informací z předchozích bloků v rámci proužku.



Obr. 4.9: Režimy jasové Intra predikce bloků o velikosti 4x4 (vlevo) a 16x16 (vpravo)

4.3.3 DCT, Celočíselná DCT

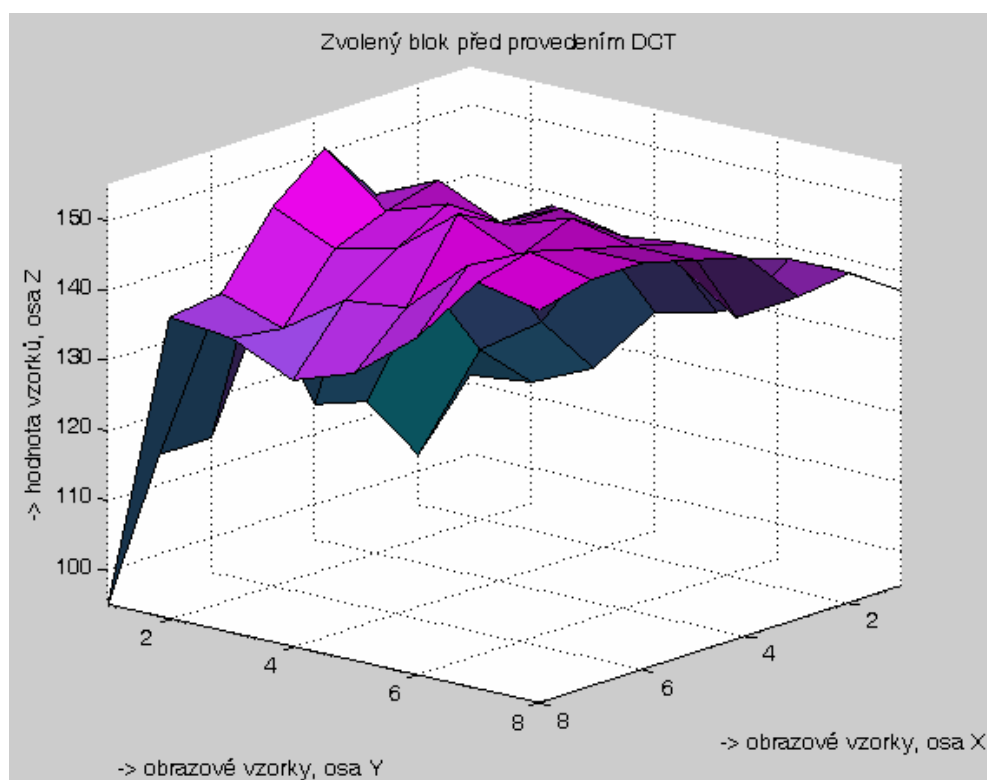
Funkce pro realizaci diskrétní kosinové transformace provádí převod obrazové informace z prostorové do frekvenční oblasti. Tato funkce má zadánu matici vzorků obrazu a velikost strany čtvercového bloku na němž je DCT provedena. Funkce pro inverzní DCT (tj. IDCT) převádí frekvenční koeficienty zpět do prostorové oblasti. Průběh těchto algoritmů na maticích reziduí zobrazují následující obrázky. Při DCT provedené bez Intra predikce by byla u snímků typu I patrná zejména výrazná hodnota *DC* koeficientu, viz Obr. 4.11, jenž odráží hodnoty a prostorové rozložení vzorků ve zvoleném bloku snímku z Obr. 4.10.

Vzhledem k tomu, že u standardů MPEG-4 se využívá v Intra i Inter módu predikce, jsou i hodnoty reziduí, představující chybu predikce, určené pro provedení transformace, v podstatně nižší hodnotové úrovni, viz Obr. 4.12. Taktéž koeficienty po provedení DCT na Obr. 4.13 nedosahují vysokých hodnot. Díky tomu dochází po provedení kvantování k výrazné redukci koeficientů, což podstatně zmenšuje potřebnou bitovou rychlost přenosového kanálu.

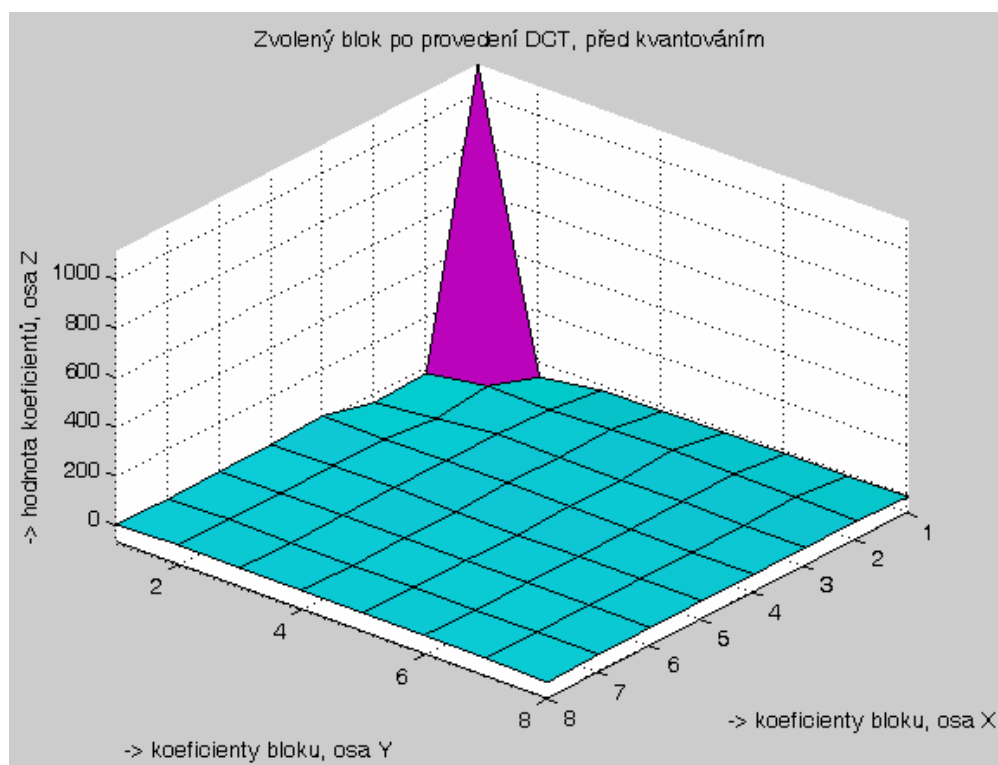
V této práci byly realizovány dvě funkce jenž provádí diskrétní kosinovou transformaci. První funkce *DCT.m* ji provádí tradiční, ovšem výpočetně náročnou operací, s využitím funkce *cosinus*, použitou v MPEG-4 Visual, popsanou v kapitole 2.1.3.

Druhá, novější metoda, provádí transformaci založenou na DCT na 4x4 blocích dat reziduí, avšak pomocí celočíselné transformace. V ní se využívá 2D jádra transformace a matice váhovacích prvků, díky čemuž je průběh transformace podstatně méně náročnější než klasická DCT. Celočíselná DCT je realizována funkcí *IntDCT.m* a obdobně jako *DCT.m* provádí transformaci blokově, avšak vždy po blocích 4x4, jak je uvedeno ve standardu. Detailně je postup celočíselné transformace popsán v kapitole 3.3.6.1.

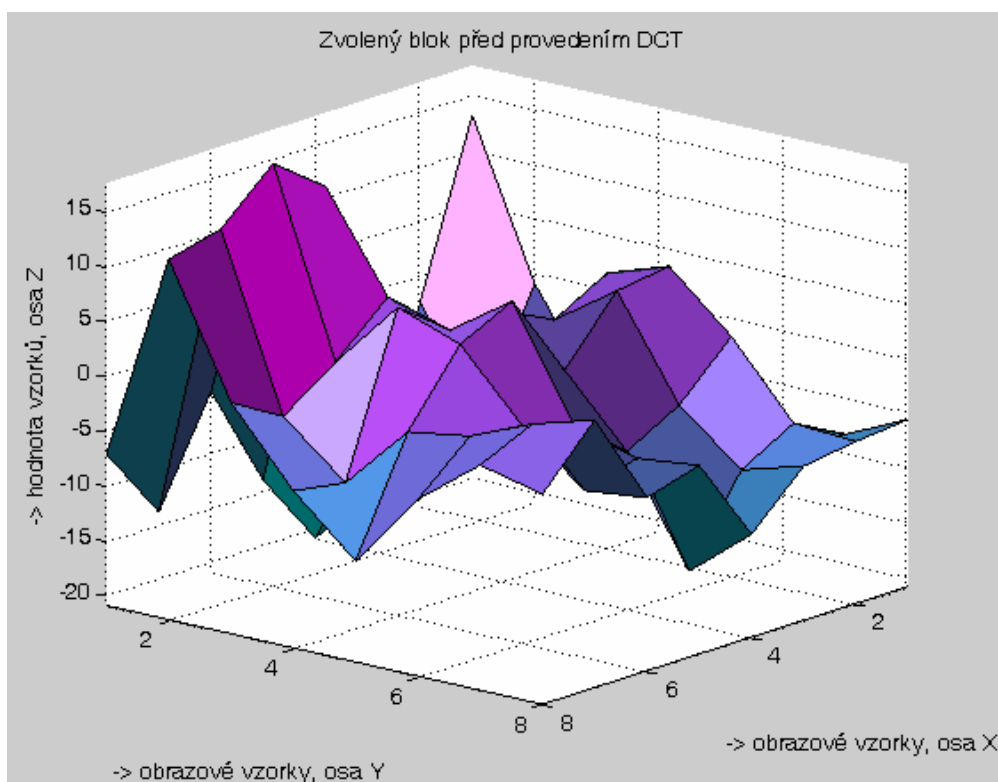
Obě funkce mají i svou inverzní transformaci, která je definována v souboru *iDCT.m* (pro koeficienty získané „klasickou“ transformací DCT), respektive *iIntDCT.m* (pro celočíselnou transformaci).



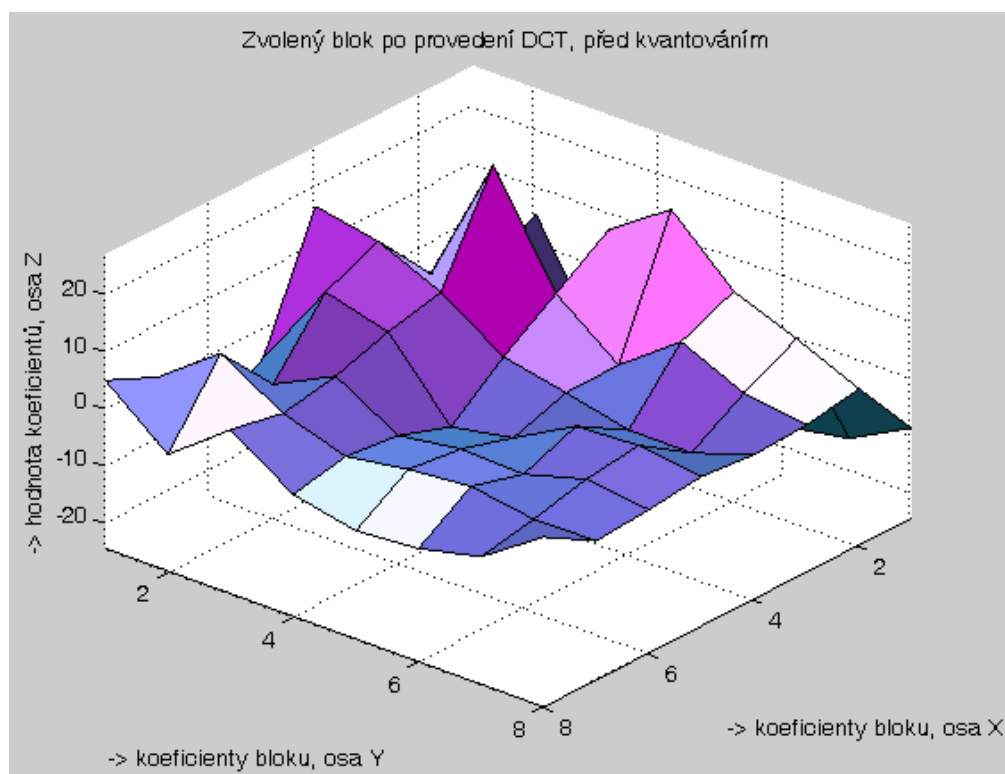
Obr. 4.10: Příklad bloku vzorků snímku typu I, před provedením DCT (bez Intra predikce)



Obr. 4.11: Příklad bloku vzorků snímku typu I, po provedení DCT (bez Intra predikce)



Obr. 4.12: Příklad bloku reziduí (chyby predikce), před provedením DCT



Obr. 4.13: Příklad bloku reziduí (chyby predikce), po provedení DCT

4.3.4 Hadamardova transformace DC koeficientů

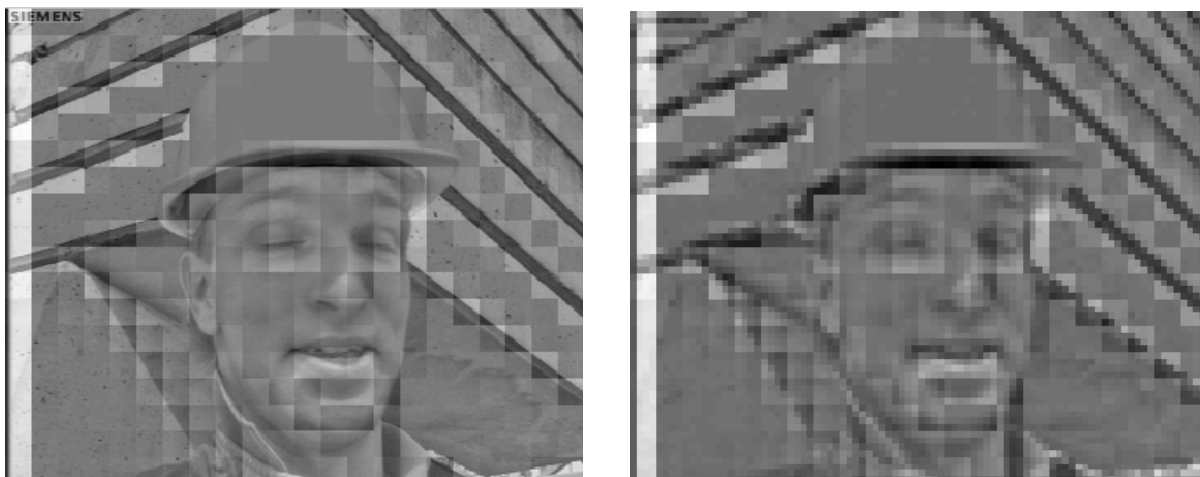
Hadamardova transformace se ve standardu MPEG-4 AVC uplatňuje na blocích 4x4 jasových *DC* koeficientů makrobloků Intra predikovaných v módu 16x16 a dále též na blocích 2x2 chrominančních *DC* koeficientů. V aplikaci slouží pro realizaci algoritmu Hadamardovy transformace funkce *HadamTran.m* a inverzně pak funkce *iHadamTran.m*. K pomocným účelům, zejména pro získání reziduí bez *DC* koeficientů a k zobrazení Hadamardovy matice reziduí, je v aplikaci dále použita funkce *iHadamRozdil.m*.

Pokud je makroblok kódován v 16x16 Intra predikčním módu (např. celý jasový makroblok je predikován ze sousedních vzorků), pak každý ze 4x4 bloků reziduí je nejprve transformován pomocí jádrové transformace CXC^T . *DC* koeficienty každého 4x4 bloku jsou poté znovu transformovány použitím 4x4 Hadamardovy transformace. Získané koeficienty po transformaci jsou dále kvantovány v závislosti na kvantovacím parametru *QP* pro získání bloku kvantovaných koeficientů.

Při dekódování je aplikována inverzní Hadamardova transformace, která je dále následována inverzním škálováním. Rekonstruované *DC* koeficienty jsou následně vloženy na jejich pozice do odpovídajících 4x4 bloků a každý z těchto bloků je inverzně transformován použitím celočíselné DCT inverzní transformace.

Tato transformace využívá faktu, že v každém 16x16 Intra kódovaném makrobloku je většina energie koncentrována do *DC* koeficientů každého 4x4 bloku. Po této doplňující transformaci je energie dále koncentrována do malého počtu významných koeficientů.

Účinnost této transformace je demonstrována na Obr. 4.14, kde je porovnávána původní matice reziduí, získaná odečtením predikce od originálního snímku a vedle ní pak snímek reziduí vytvořený pouze z rekonstrukce *DC* koeficientů které prošly dopřednou a inverzní Hadamardovou transformací. Použita byla Intra predikce 16x16 *DC* a *QP* = 20 (tj. *QStep* = 6,5).



Obr. 4.14: Původní matice reziduí a rekonstruovaná matice pouze s využitím DC koeficientů

4.3.5 Kvantování

Funkce realizující kvantování *kvantuj.m* požaduje na vstupu získat matici hodnot na nichž bude provedeno kvantování (např. reziduí po provedení DCT), dále též hodnotu kvantovací matice a velikost strany bloku, po kterých bude kvantování blokově prováděno. Přednastavené je v aplikaci kvantování dle standardu MPEG-4 Visual a MPEG-4 AVC, popsané v kapitole 3.3.6.2. Dále je rovněž možné ruční zadání kvantovacích matic.

Při kvantování dle standardu MPEG-4 AVC se používá rozsah celkem 52 hodnot kvantovacího kroku $Qstep$, jenž jsou indexovány kvantovacím parametrem QP . Pro výpočet kvantovaných koeficientů je užít základní vztah:

$$Z_{ij} = \text{round}\left(\frac{Y_{ij}}{Qstep}\right) \quad (4.1)$$

, kde Y_{ij} je koeficient po provedení transformace, $Qstep$ je velikost kvantovacího kroku a Z_{ij} je kvantovaný koeficient.

Pro kvantování dle standardu MPEG-4 Visual jsou standardem definovány dvě metody pro tuto operaci. Společná jim je závislost na kvantovacím parametru QP , který může nabývat hodnot od 1 do 31, přičemž vyšší hodnoty tohoto parametru představují vyšší hodnotu kvantovacího kroku. V aplikaci je použita výchozí a jednodušší *Metoda 2* inverzního kvantování, která má pro obnovu DC koeficientů definován následující vztah:

$$DC = DC_Q \cdot dc_scaler \quad (4.2)$$

, kde: DC_Q je kvantovaný koeficient, DC rekonstruovaný koeficient a dc_scaler je parametr definovaný standardem, jehož hodnoty se nachází v Tab. 3.5. Způsob výpočtu ostatních koeficientů je uveden v kapitole 3.3.6.2.

V aplikaci jsou tedy přednastaveny dva způsoby kvantování, a to dle standardu MPEG-4 Visual a MPEG-4 AVC. K nim je přidružena volba kvantovacího parametru, pomocí něž lze nastavovat kvantovací krok dle dříve uvedených algoritmů. Pole „kvantovací parametr“ je určen pro oba standardy, jak je patrné z navázaných textových polí, které v závislosti na změně parametru QP mění pro MPEG-4 AVC hodnotu kvantovacího kroku $QStep$ a pro MPEG-4 Visual hodnotu parametru dc_scaler . Rozsah hodnot je pro MPEG-4 AVC v rozmezí QP od 0 do 51. Pro MPEG-4 Visual je definován v rozsahu od 1 do 31.

Pro umožnění uživatelského zadání vlastních kvantovacích matic je dostupná volba „Vlastní“, pomocí níž lze definovat vlastní kvantovací matice. Výchozí podoba uživatelsky definované kvantovací matice je zvolena podle standardu ITU-T H.264, a to především z toho důvodu, neboť v tomto standardu jsou definovány matice pro Intra i Inter kódování, pro velikost bloků 4x4 a 8x8. Jejich podoba je na Obr. 4.15.

Intra 8x8 (proužky I) <pre> 6 10 13 16 18 23 25 27 10 11 16 18 23 25 27 29 13 16 18 23 25 27 29 31 16 18 23 25 27 29 31 33 18 23 25 27 29 31 33 36 23 25 27 29 31 33 36 38 25 27 29 31 33 36 38 40 27 29 31 33 36 38 40 42 </pre>	Inter 8x8 (proužky P, B) <pre> 9 13 15 17 19 21 22 24 13 13 17 19 21 22 24 25 15 17 19 21 22 24 25 27 17 19 21 22 24 25 27 28 19 21 22 24 25 27 28 30 21 22 24 25 27 28 30 32 22 24 25 27 28 30 32 33 24 25 27 28 30 32 33 35 </pre>
Intra 4x4 (I) <pre> 6 13 20 28 13 20 28 32 20 28 32 37 28 32 37 42 </pre>	Inter 4x4 (P,B) <pre> 10 14 20 24 14 20 24 27 20 24 27 30 24 27 30 34 </pre>

Obr. 4.15: Standardizované hodnoty kvantizačních matic pro Intra a Inter kódování dle standardu ITU-T H.264 [10]

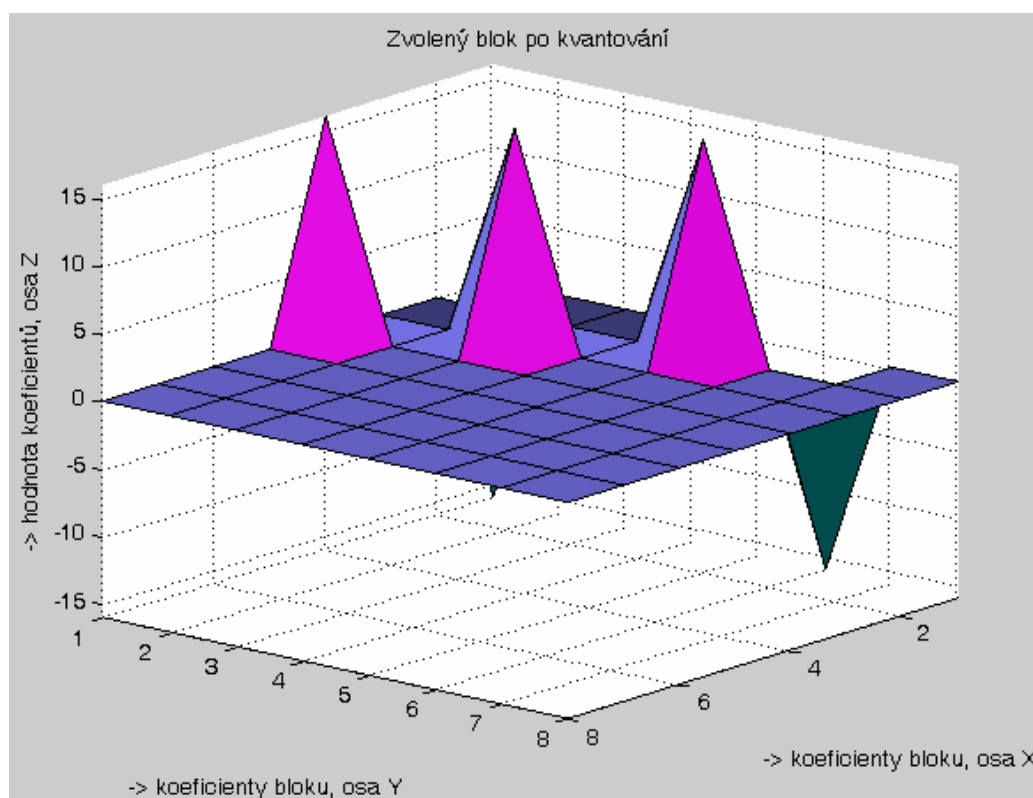
Výše uvedené hodnoty však mohou být uživatelsky snadno pozměněny, respektive přímo zadány. V aplikaci to lze provést stiskem tlačítka „Definovat“, po kterém se objeví nové okno s předvyplněnými kvantovacími maticemi, viz Obr. 4.16. Zadání hodnot kvantovacích matic musí odpovídat formátu zápisu v programu Matlab, tj. hodnoty jsou odděleny mezerou a řádky středníkem.

Obr. 4.16: Okno pro zadání vlastních kvantovacích matic

U kvantovacích matic standardů MPEG a H.26x a je zřetelná preference vzorků nižších kmitočtů, které mají podstatnější význam pro výsledný zrakový dojem, než vzorky vyšších kmitočtů, čemuž odpovídá i jejich tzv. hrubější kvantování.

Podoba kvantovací matice v kodéru se v praxi mění zpravidla v závislosti na stavu vysílací paměti, a to takovým způsobem, aby nedocházelo k jejímu zahlcení, nebo nedostatečnému plnění.

Realizovaná funkce po spuštění vykoná kvantování obdržného rozdílového snímku. A to takovým způsobem, že postupně čte bloky ze zdrojového snímku, kvantuje koeficienty dle kvantovací matice a zapisuje je následně do výsledného snímku. Díky tomu dochází k podstatné redukci koeficientů, v porovnání s nekvantovanou maticí, jak je patrné z náhodně vybraného bloku Obr. 4.13 před kvantováním a Obr. 4.17 po kvantování.



Obr. 4.17: Příklad bloku chyby predikce snímku typu P, po provedení kvantování

4.3.6 Určení vektorů pohybu metodou Full Search a vytvoření pohybově kompenzovaného snímku

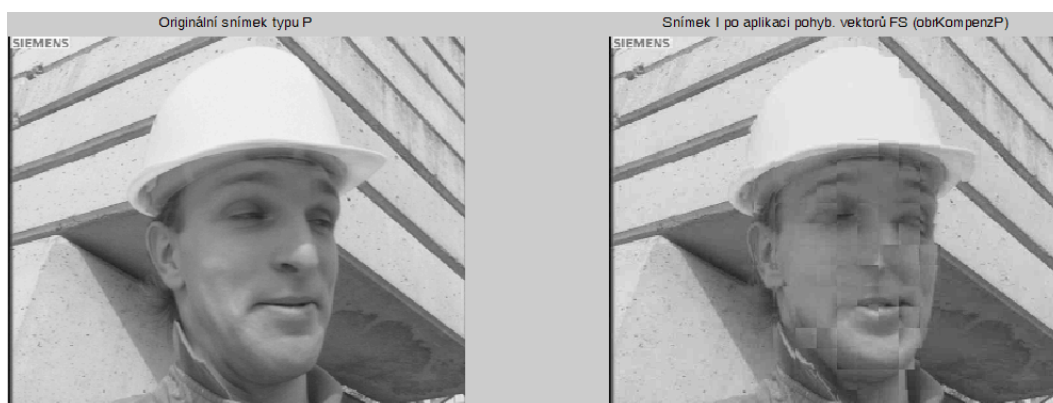
U snímků s predikcí typu P a B se uplatňují vektory pohybu a kompenzace pohybu. U vstupního snímku typu P, na něhož je použit odhad pohybu, se k následnému zpracování DCT a kvantování použije pouze chyba predikce, jenž představuje odlišnost mezi skutečným a predikovaným snímkem. Nalezené vektory pohybu jsou v reálném kodéru odděleně entropicky kódovány a až před samotným vysíláním sloučeny s datovým tokem, jenž je tvořen kódovanou chybou predikce [10].

Funkce hledající vektory pohybu metodou úplného prohledávání *mv_full.m* postupně prochází snímek typu P, v němž jsou hledány vektory pohybu a porovnáním jednotlivých bloků s referenčním snímkem určuje souřadnice vektoru pro osu x a y . Ty zapisuje do matice, jenž tyto hodnoty uchovává pro následnou rekonstrukci k vytvoření pohybově kompenzovaného snímku. Ten se v dekodéru tvoří tím způsobem, že je použit referenční

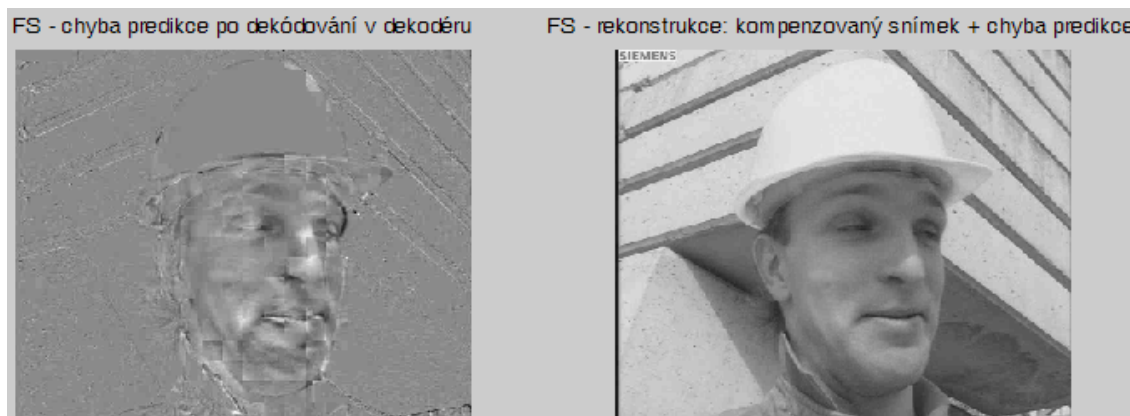
snímek (např. přijatý snímek I) a v něm jsou na jednotlivé bloky aplikovány vektory pohybu. To prakticky znamená, že blok z referenčního snímku je na základě informace o vektoru pohybu zapsán na novou pozici ve výsledném pohybově kompenzovaném snímku.

Tento způsob prohledávání poskytuje vektory pohybu, jenž směřují na pozici bloku s největší mírou shody, avšak za cenu velké výpočetní náročnosti [4].

V ukázkové sekvenci tak hodnota PSNR u pohybově kompenzovaného obrázku s použitím metody Full Search, kdy byl porovnáván originální obrázek typu P a pohybově kompenzovaný obrázek, činila 30,45 dB. Srovnání originálního a výsledného pohybově kompenzovaného obrázku uvádí Obr. 4.18. Vzájemné odchylky jsou patrné v oblastech, u nichž bylo z důvodu rychlého pohybu objektu ve snímku, aplikováno vektorů pohybu a kde tak jsou zřetelné odlišnosti od originálního snímku. Tento rozdíl je patrný na Obr. 4.19, kde je nejprve uvedena chyba predikce, tak jak je interpretována v dekodéru, tedy po DCT, kvantování a IDCT. Následně je pak zobrazen rekonstruovaný snímek, který je tvořený pohybově kompenzovaným snímek společně s chybou predikce. Po součtu obou komponent tak výsledný obraz na straně dekodéru má v porovnání s obrazem na vstupu kodéru PSNR = 40,42 dB (15,1% nenulových reziduí, QP=25, velikost bloku na vektor: 16x16).



Obr. 4.18: Originální snímek typu P a jeho pohybově predikovaná podoba metodou FS



Obr. 4.19: Chyba predikce po přijetí v dekodéru a následně rekonstruovaný snímek

4.3.7 Určení vektorů pohybu metodou Three Step Search a vytvoření pohybově kompenzovaného snímku

Tato metoda je realizována funkcí *mv_tss.m*, která podobně jako u předchozí metody očekává na vstupu předání referenčního snímku a prohledávaného snímku, v němž jsou hledány vektory pohybu. Princip hledání vektorů pohybu se však liší, neboť se jedná o rychlý prohledávací algoritmus, patřící do skupiny N-Step-Search algoritmů, kde prohledávání probíhá v N krocích. V této realizaci byla zpracována nejpoužívanější tříkroková varianta.

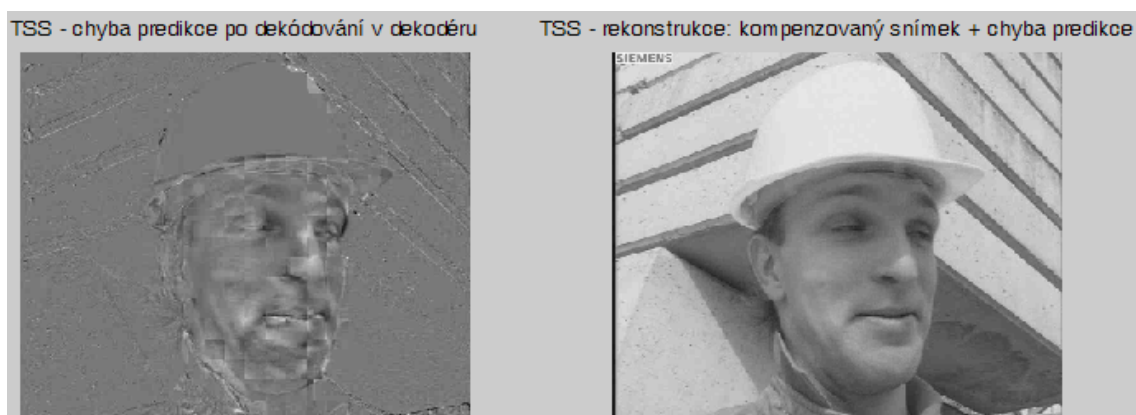
Funkce provádějící hledání vektorů pohybu metodou TSS prohledává snímek od pozice (0,0), se zadaným krokem. Nejprve je hledána shoda v počátku a v osmi pozicích okolo něj. Z těchto devíti pozic je vybrána ta s největší shodou a ta je určena jako nová výchozí pozice. Současně dochází ke zkrácení prohledávacího kroku na polovinu. S tímto krokem je opět prozkoumáno osm okolních pozic včetně počáteční. Opět je určen nejvíce shodný blok, jenž je určen za počáteční a je zkrácen krok na polovinu. Následně je opět provedeno hledání největší shody. Finální souřadnice polohy nejvíce shodného bloku jsou zaznamenány jako vektory pohybu pro osu x a y daného bloku. Matice vektorů pohybu je uchována pro sestavení pohybově predikovaného snímku v dekodéru [4].

Rekonstrukce je provedena s pomocí referenčního snímku, ze kterého jsou postupně čteny bloky, které jsou na základě informace z matice vektorů pohybu umístěny na požadovanou pozici a následně zapsány do výsledného, pohybově kompenzovaného, snímku.

V použité ukázkové sekvenci hodnota PSNR u pohybově kompenzovaného obrázku, viz Obr. 4.20, za použití metody Three Step Search, kdy byl porovnáván originální snímek typu P a pohybově kompenzovaný obrázek, činila 30,23 dB a je tak logicky nižší než u metody úplného prohledávání. Za povšimnutí však bezpochyby stojí to, že hodnota PSNR se u obou metod liší v řádu desetin decibelu, což je vzhledem k podstatně rychlejšímu algoritmu hledání velmi dobrá hodnota. Dobrého výsledku bylo rovněž dosaženo po rekonstrukci snímku v dekodéru, kde byl součtem pohybově kompenzovaného snímku a chyby predikce získán výsledný snímek, jehož hodnota PSNR při porovnání s originálním snímkem na vstupu kodéru činí 40,41 dB (15,5% nenulových reziduí, QP=25, bloky: 16x16). Tato hodnota se téměř shoduje s výsledkem po dekódování metodou Full Search (odlišnost se pohybuje v řádu setin dB) a v praxi by tak byla lidským okem bezpochyby nepostřehnutelná, viz Obr. 4.21.



Obr. 4.20: Originální snímek typu P a jeho pohybově predikovaná podoba metodou TSS



Obr. 4.21: Chyba predikce po přijetí v dekodéru a následně rekonstruovaný snímek

4.3.8 Určení vektorů pohybu metodou Nearest Neighbours Search a vytvoření pohybově kompenzovaného snímku

Metoda Nearest Neighbours Search patří stejně jako předchozí metoda do skupiny rychlých prohledávacích algoritmů. Je realizována funkcí *mv_nns.m* a pro svůj běh vyžaduje znalost referenčního snímku (např. typu I) a prohledávaného snímku (např. typu P), z nichž určuje vektory pohybu. Dále je jí předána informace o počtu kroků, během nichž hledá nejlepší shodu.

Tato funkce při svém běhu využívá předpokladu, že sousedící bloky mají obdobné vektory pohybu. Nejprve je tedy nastavena nulová počáteční pozice ze které se vychází při hledání. Prohledávání probíhá pro zadaný počet kroků a porovnávány jsou vždy okolní pozice tvořící geometrický tvar symbolu „+“. Z nich je nalezena pozice s největší shodou, vyhodnocením metrikou MAE. Další výchozí pozice je nastavena z polohy nejvyšší shody předchozího kroku [4].

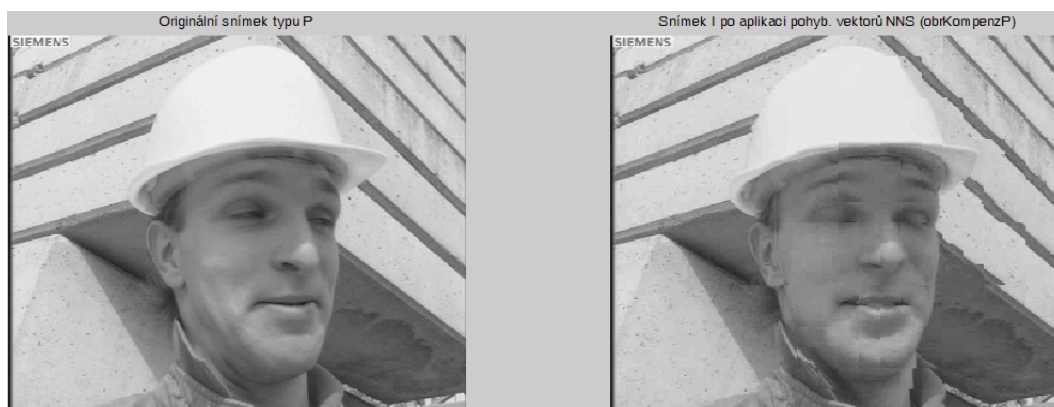
Následující počáteční pozice v dalším bloku je pak následně určena ze znalosti předchozího vektoru pohybu. Pro počáteční prohledávací pozici je tak upravena řádková a sloupcová souřadnice počátku. Souřadnice nejlépe odpovídající pozice, po vyhodnocení metrikou MAE, je po vyhodnocení každého bloku zapsána do matice vektorů pohybu.

Při rekonstrukci pomocí funkce *kompenzace.m* je využit referenční snímek, na jehož jednotlivé, postupně čtené, bloky jsou aplikovány vektory pohybu a tyto bloky jsou zapsány na nově určenou pozici ve výsledném, pohybově kompenzovaném, snímku.

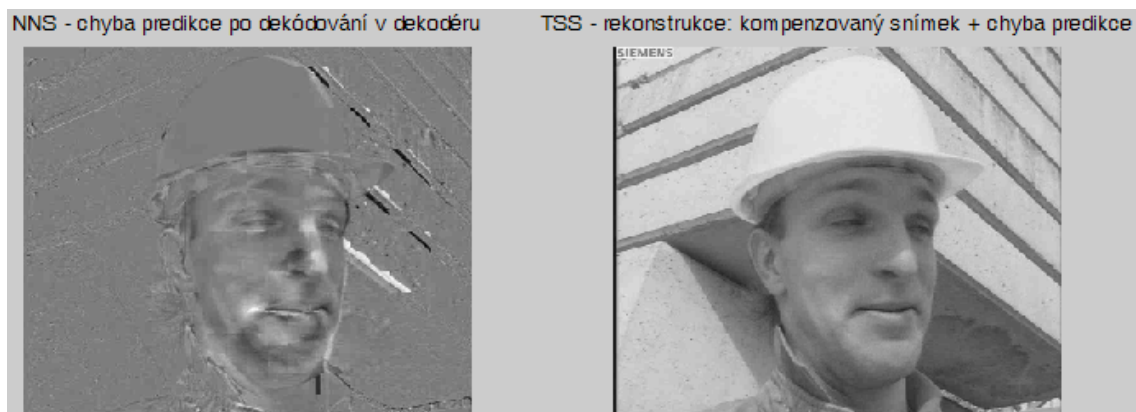
V totožné použité ukázkové sekvenci, jako v předchozích případech, byl za pomoci metody Nearest Neighbours Search vytvořen pohybově kompenzovaný snímek, viz Obr. 4.22, jehož hodnota PSNR, při porovnání s originálním snímkem činila 27,32 dB. Tato hodnota je výrazně horší, než u předchozích dvou metod, což lze přisuzovat nevhodně zvoleným parametřům této metody pro tuto konkrétní sekvenci snímků.

Při pohledu na snímek dekódované chyby predikce na Obr. 4.23 je patrné, že nejen pohyb osoby v popředí, ale i pohyb pozadí (svislé čáry v pravém horním rohu) činily při určování vektorů potíže, což mohlo být způsobeno především nízkou hodnotou počtu prohledávacích kroků. Ten byl nastaven na hodnotu 3, což vedlo zřejmě k nemožnosti algoritmu rychle reagovat na velké pohyby bloků.

Rekonstrukce snímku provedená součtem pohybově kompenzovaného snímku a kvantované chyby predikce poskytla přesto dobrý výsledek, když hodnota PSNR činila 40,28 dB a velmi se tak přiblížila k hodnotám předchozích metod, což je především zásluhou doplňujících reziduí (17,4 % nenulových reziduí, QP=25, velikost bloku na vektor: 16x16).



Obr. 4.22: Originální snímek typu P a jeho pohybově predikovaná podoba metodou NNS



Obr. 4.23: Chyba predikce po přijetí v dekodéru a následně rekonstruovaný snímek

4.3.9 Subpixelová přesnost určování vektorů pohybu

V souladu se standardy MPEG-4 Visual a MPEG-4 AVC poskytuje vypracovaná aplikace možnost určení přesnosti při určování vektorů pohybu. K dispozici je volba pro pixelovou, půlpixelovou a čtvrtpixelovou přesnost vyhledávání, a to v blocích o velikosti 4x4, 8x8 a 16x16.

Pokud je uživatelsky zvolena subpixelová přesnost, pak dochází k volání funkce *subPixelMV.m*, která provádí interpolaci snímku referenčního i prohledávaného, a to z toho důvodu, aby bylo dosaženo požadovaného počtu vzorků na pozicích mezi originálními pixely. V návaznosti na to musí být upraveny i vektory pohybu určené pixelovou přesností, aby i nadále směřovaly do správných míst v obraze.

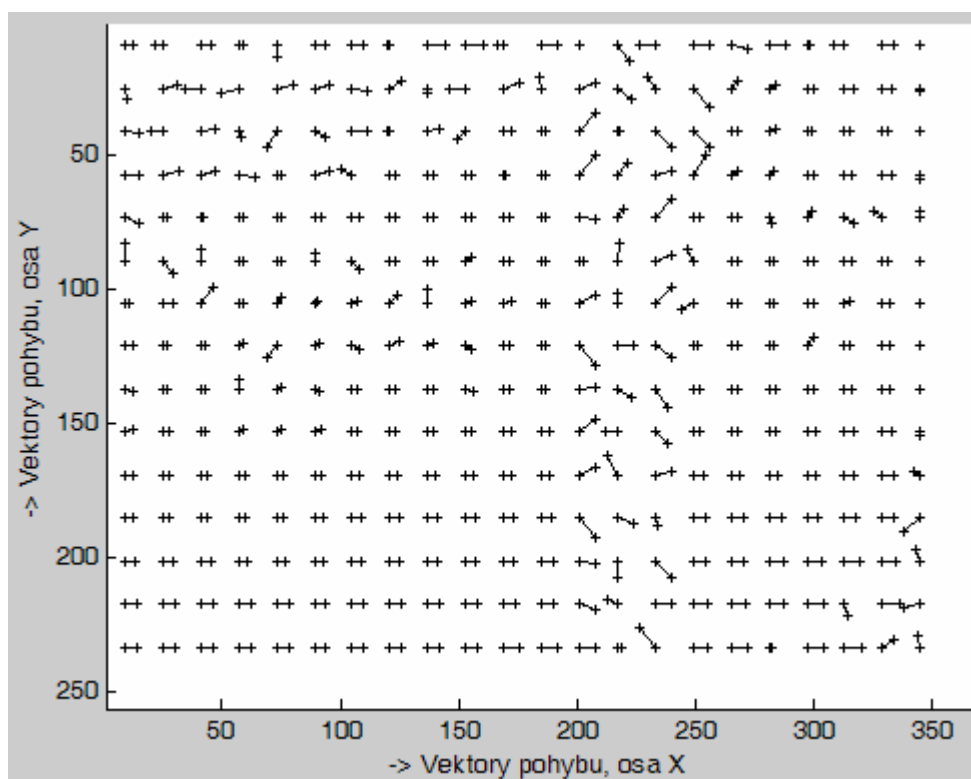
Dále je volána funkce *mv_subfull.m*, které jsou předány interpolované snímky a vektory pohybu. Tato funkce s využitím přijatých informací provede vyhledání přesnější shody bloků na subpixelových pozicích, a to v okolí pozice nalezené již dříve metodou s pixelovou přesností. Hledání probíhá metodou úplného prohledávání (Full Search), což však vzhledem k počtu porovnávaných pozic není z hlediska výpočetní náročnosti problém. Na základě přesnějšího určení vektorů pohybu tak může být lépe sestaven predikovaný snímek, jehož rezidua dosahují menších hodnot.

Při následné kompenzaci snímku s využitím vektorů pohybu funkcí *kompenzace.m* jsou využity určené vektory pohybu, a to ať už s pixelovou, viz Obr. 4.25, nebo subpixelovou přesností, viz Obr. 4.26. Subpixelová přesnost je na obrázcích znázorňujících vektory pohybu jednotlivých bloků patrná z vynásobení počtu vzorků snímku a tím i znásobením rozměru v ose x a y .

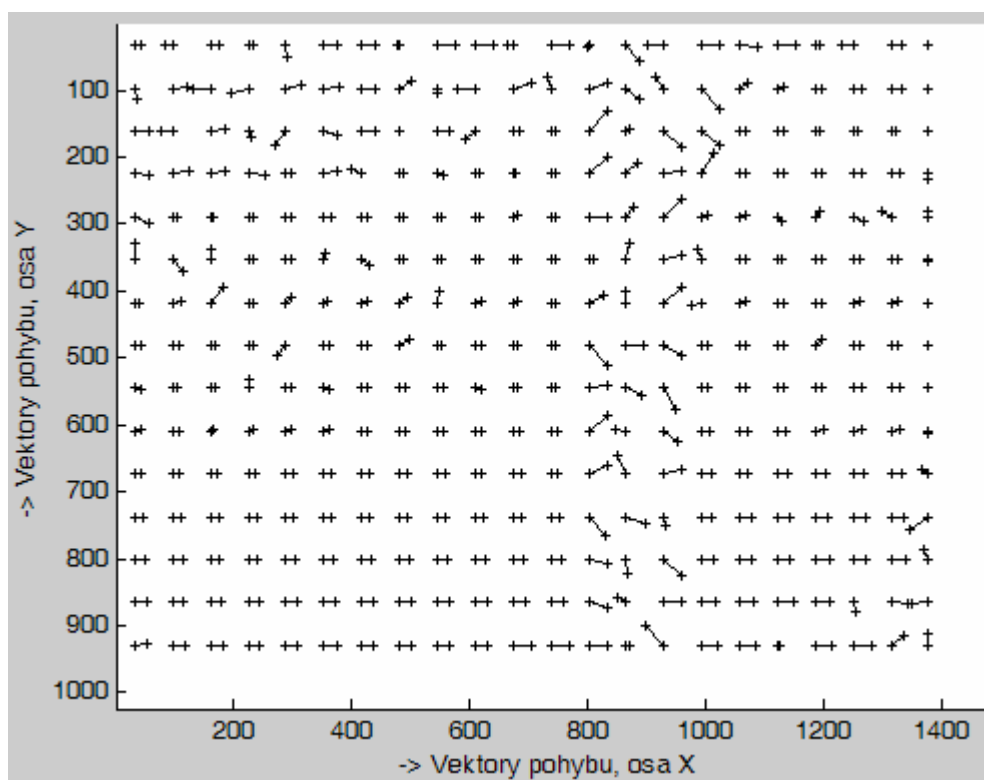


Obr. 4.24: Snímek s horizontálním posunem objektů, strom v popředí rychleji (sekvence Garden)

Pro zobrazení vektorů pohybu je volána funkce *zobrazMV.m*, která pro zadanou velikost bloků (4x4 , 8x8 , 16x16 pixelů) provede vykreslení vektorů, jenž znázorňují směr a velikost pohybu mezi počáteční a cílovou pozici. V kombinaci s pixelovou, půlpixelovou a čtvrtpixelovou přesností tak může být vykresleno mnoho různých variant.



Obr. 4.25: Vektory pohybu pro snímek s pixelovou přesností (sekvence Garden)



Obr. 4.26: Vektory pohybu pro snímek s čtvrtpixelovou přesností (sekvence Garden)

5 ZHODNOCENÍ VÝSTUPŮ REALIZOVANÝCH FUNKCÍ

Vypracované funkce tvoří skupinu algoritmů, jenž představují logický funkční celek, realizující činnost kodéru a dekodéru. Samotné funkce jsou zastřešeny pod soubor *kodek.m*, jenž na základě vstupních parametrů provádí jejich volání a rovněž též zobrazení grafických a numerických výstupů. Obrazová část výstupů byla popsána již v předchozích kapitolách, proto bude následující text zaměřen spíše na numerické vyhodnocení výstupů a též na srovnání jednotlivých možností komprese pomocí realizovaných funkcí.

5.1 OČEKÁVANÉ VÝSTUPY Z ALGORITMŮ KODEKU

Jak již bylo v úvodu práce zmíněno, nekomprimovaný obrazový signál představuje datový tok v řádu desítek MB/s, který je nutné zaznamenat, nebo přenést k cílovému zařízení, což není ani ekonomické ani technicky snadné realizovat [1]. Proto je využití algoritmů pro ztrátovou i bezztrátovou kompresi nezbytné. Vytvořená aplikace provádí simulaci činností probíhajících v kodéru a dekodéru. Souhrnný popis realizovaných funkcí provádějících ztrátovou i bezztrátovou kompresi v simulovaném kodeku následuje dále.

Na vstupu jsou aplikaci uživatelsky zadány snímky sekvence, které mají být programově zpracovány. U nich dochází ke vstupnímu podvzorkování snímků, kdy je v závislosti na zvoleném formátu sníženo vertikální a horizontální rozlišení. U formátu 4:4:4 zůstávají jednotlivé složky bez podvzorkování, tudíž nedochází ke kompresi. To však při zpracování signálu není v praxi příliš obvyklé. Dalším formátem je 4:2:2, kdy jasová složka zůstává beze změny, ale barvonosné složky Cb a Cr jsou horizontálně podvzorkovány na polovinu. To přináší třetinovou úsporu v celkovém množství vzorků oproti původní podobě. Nejčastěji používaným formátem je 4:2:0, ve kterém mají chrominancní složky Cb a Cr poloviční vertikální i horizontální rozlišení, čehož je dosaženo vynecháním sudých vzorků ve vertikálním i horizontálním směru. Jasová složka zůstává i zde v plném rozlišení. Použitím formátu 4:2:0 tak lze již na vstupu dosáhnout snížení počtu vzorků a tím i datového toku snímků na polovinu (úplný jasový a dvě čtvrtinové barvonosné složky snímku proti třem úplným složkám snímku).

Dalším funkčním blokem, jenž je programově zpracován je diskrétní kosinová transformace (DCT), respektive její celočíselná varianta. V ní dochází k transformaci vzorků z prostorové oblasti do podoby frekvenčních koeficientů. Účelem této operace není komprese signálu, ale jeho odlišné vyjádření, jenž je vhodné k následnému ztrátovému kódování. Při DCT totiž nedochází k redukci datového toku, ale naopak k jeho navýšení. Původní osmibitové vzorky jsou totiž po DCT vyjádřeny pomocí 12bitových frekvenčních koeficientů. Vyjádření signálu pomocí koeficientů od nízkých kmitočtů až po vyšší kmitočty však poskytuje relativně snadný způsob jak redukovat pro HVS nepodstatné vysokofrekvenční složky. V kodéru dochází k totožnému zpracování DCT pro všechny snímky. Jedná se vždy o blokové zpracování snímku, kdy na matici 8x8, nebo 4x4 je provedena DCT. Velikost bloku 8x8 byla v počátcích zvolena jako kompromis pro zařízení s nižším výpočetním výkonem, neboť operace DCT je relativně výpočetně náročná.

Mezi Intra kódovanými snímky I a predikovanými snímky P a B panuje na vstupu DCT odlišnost. Snímky typu I přicházejí na vstup s hodnotami reziduí, rovnými rozdílu mezi originálním snímkem a Intra predikovaným, kdežto u snímků P a B to je již jen chyba predikce odhadu pohybu. Chyba predikce se zpravidla vyznačuje nižšími hodnotami vzorků a rovněž také určitou nehomogenitou mezi hodnotami sousedních vzorků. Tyto rysy se rovněž projevují po provedení DCT, kde jsou patrné na struktuře rozložení a též na hodnotách jejich koeficientů. Z důvodu názornosti je výstupem aplikace rovněž grafické zobrazení hodnot

bloku zvolené matice (ve výchozím stavu načtené zprostřed obrázku) před provedením DCT i po provedení DCT. Toto srovnání zobrazení je zajímavé zejména při porovnání mezi snímky typu I (zejména bez použití predikce) a typu P.

Po DCT přichází na řadu kvantování, které je typickým představitelem ztrátové komprese. Myšlenkou je dělení bloků koeficientů zadanými hodnotami kvantizačních tabulek, s následným zaokrouhlením na celá čísla. Kvantizační tabulky se liší pro Intra kódované snímky typu I a pro Inter kódované snímky typu P a B. V realizované kvantovací funkci je kvantování blokově prováděno na přijatém snímku po provedení DCT, přičemž je očekávána definice snímku a kvantovací matice. Kvantovací matice je definovaná dle zadaného typu snímku a uživatelsky zvoleným kvantovacím parametrem. Následně je zobrazen zvolený blok po provedeném kvantování, na němž je zpravidla patrná zejména redukce koeficientů nízkých hodnot. Díky této redukci pak může dojít k zásadnímu omezení datového toku, ale též ke snížení hodnoty parametru PSNR.

Funkce, které hledají vektory pohybu, jsou v této práci definovány tři. Jedna určuje vektory metodou úplného prohledávání (Full Search), druhá pomocí rychlého prohledávacího algoritmu Three Step Search (TSS) a třetí taktéž pomocí rychlého prohledávacího algoritmu Nearest Neighbours Search (NNS). Bylo prakticky zjištěno, že při úplném hledání může být provedeno až 10x více výpočtů než u rychlého prohledávacího algoritmu TSS. Tato výpočetní náročnost úplného prohledávání však při srovnání obou typů metod postrádá opodstatnění, neboť u výrazně rychlejšího prohledávacího algoritmu TSS se odlišnost u hodnoty PSNR pro obě metody pohybovala často v řádu desetin decibelu. Přesnost vyhledání nejvíce shodného bloku je také závislá na velikosti bloku pro nějž je hledána shoda (obvykle makroblok 16 x 16) a též na oblasti ve které je shoda v referenčním snímku hledána. Těmito parametry je ovlivněna nejen míra nalezené shody, ale i obsah snímku s chybou predikce a v neposlední řadě též výpočetní náročnost. Velikost bloku pro nějž jsou hledány vektory pohybu má rovněž vliv na datový tok. Platí totiž, že čím větší blok je, tím méně vektorů pohybu je zapotřebí a tím je i menší datový tok z prediktoru. Negativem však zůstává, že při takto hrubém určování vektorů dochází k nárůstu chyby predikce a tak je nutné kódovat a přenášet větší množství těchto dat. Omezit to lze hrubším kvantováním, ovšem za cenu ztráty informace a zhoršení obrazového výstupu dekodéru.

Vektory pohybu však mohou být určovány z bloků menších než 16 x 16 vzorků, a to například pro bloky 8 x 8, nebo dokonce 4 x 4. Počet vektorů pohybu pak je čtyřnásobný, respektive 16ti násobný, ovšem rezidua chyby predikce se dostávají na výrazně nižší hodnoty, které se následným kvantováním mohou citelně redukovat. Díky tomu tak může dojít ještě k výraznějšímu omezení datového toku. Lze totiž početně určit [10], že u snímku typu P o rozlišení 704 x 576 vzorků je 1584 makrobloků rozměru 16 x 16, přičemž se určují dvě souřadnice x , y , které mohou mít 2 – 14 bitů. To představuje datový tok v rozmezí 6336 - 44352 bit/snímek. Nelze rovněž opomenout, že dané hodnoty pro snímek typu B jsou dvojnásobné, neboť tyto snímky mají dvojnásobný počet vektorů, z důvodu dopředné a zpětné predikce. Jednoduchým vynásobením si lze určit datový tok i pro predikci s použitím menších bloků, a to jako čtyř, nebo šestnáctinásobek uvedeného počtu bitů. Tento datový tok je i přesto relativně velmi nízký, obzvláště je-li uvaženo, že použitím predikce v dekodéru je již získán pohybově kompenzovaný snímek s větší či menší mírou chyby predikce.

K omezení nutnosti přenášet dva vektory na blok byla experimentálně vypracována funkce *vyberB.m*, která simuluje přenos pouze jednoho vektoru pohybu, a to toho, jenž dle porovnání parametru MAE zajišťuje větší shodu u jednoho ze dvou referenčních snímků. Snímek B se tak v podstatě určí jako výsledek z vyhodnocení predikce mezi dvěma snímky a postačuje tak jeden vektor a informace, ze kterého referenčního snímku byl určen. Všechny uvedené předpoklady jsou dále v textu ověřeny a graficky znázorněny.

5.2 VSTUPNÍ PARAMETRY PRO VYHODNOCENÍ

Při testování výstupů algoritmů a pro získání požadovaných tabulek a grafů byly použity již dříve zmíněné obrazové sekvence, jenž patří mezi základní sekvence pro ověřování účinnosti kódování v oblasti komprese obrazu. Jejich základní charakteristika je následující:

- Football (352 x 240 pixelů) – zachycuje rychlé změny v obraze, relativně malé objekty, pozadí se pohybuje
- Foreman (352 x 288 pixelů) – rychlý pohyb mluvčího v popředí, pozadí se pomalu pohybuje
- Garden (352 x 240 pixelů) – souvislý pomalý pohyb celé scény s výrazným prvkem v popředí, který se pohybují rychleji
- Missa (352 x 272 pixelů) – pomalu se pohybující osoba, pozadí je jednolitě, bez výraznějších změn

Při vyhodnocení byly použity zejména funkce a jejich kombinace, jenž odpovídají použití ve standardech MPEG-4 AVC a MPEG-4 Visual. Pro každé vyhodnocení však budou následně uvedeny relevantní podmínky, za kterých byl daný výstup získán.

V průběhu testování je uvažováno bezetrátové prostředí, tudíž výstupní hodnoty nejsou ovlivněny chybou způsobenou ztrátou nebo zkreslením informace během přenosu.

5.3 VSTUPNÍ PODVZORKOVÁNÍ A PROMĚNLIVOST OBRAZU

Prvním krokem v celé posloupnosti zpracování signálu je jeho převod na složkový tvar a následné podvzorkování chrominančních složek. Jedná se o pevně určený algoritmus jenž není závislý na typu zpracovávaného signálu, ale je dán potřebou redukce datového toku. Jasová složka Y není podvzorkována, neboť ze znalosti HVS je lidský zrak nejvíce citlivý právě na změnu jasu. Podvzorkování se tak realizuje pouze u barvonosných složek, a to v horizontálním i vertikálním směru. Nejčastěji používaným typem podvzorkování je režim 4:2:0, při kterém jsou barvonosné složky na výstupu operace pouze s polovičním horizontálním i vertikálním rozlišením. Více k tomuto tématu bylo popsáno v kapitole 2.1.1. Přínos podvzorkování na redukci datového toku uvádí Tab. 5.1, která srovnává jednotlivé typy podvzorkování a množství redukovaných vzorků.

Tab. 5.1: Redukce množství vzorků v závislosti na typu podvzorkování

Typ vzorkování	Množství vzorků ve složkách			Součet složek	Redukce dat
	Y	Cb	Cr	Y+Cb+Cr	
4:4:4	100%	100%	100%	300%	0%
4:2:2	100%	50%	50%	200%	33%
4:2:0	100%	25%	25%	150%	50%
4:1:1	100%	25%	25%	150%	50%

Z tabulky je patrné, že často používaný typ podvzorkování 4:2:0 přináší úsporu 50 % z celkového datového toku, aniž by zřetelně utrpěl subjektivní dojem z takto ztrátově komprimovaného datového toku.

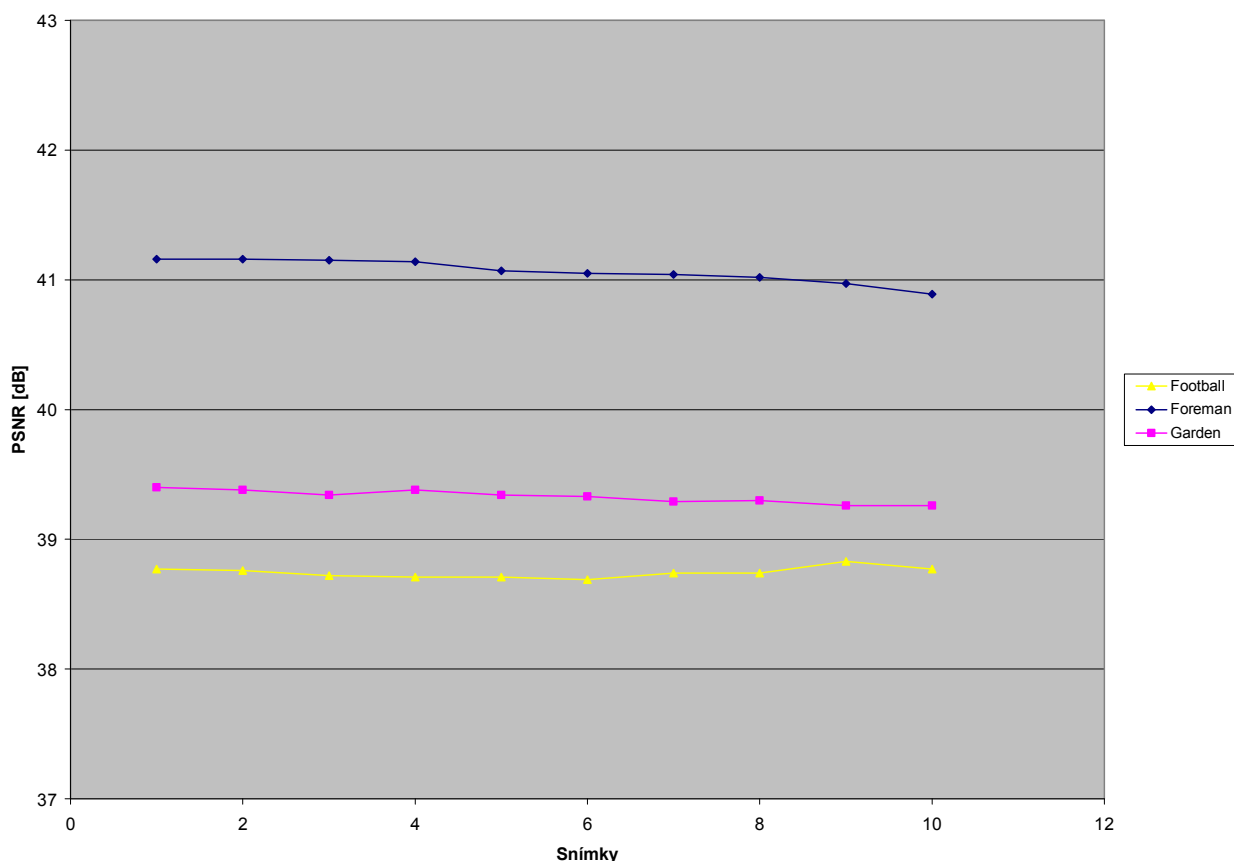
Jak již bylo uvedeno, zásadní vliv na výsledný obrazový dojem má především jasová složka, a proto i na její efektivní zpracování je kladen zvýšený důraz. Na barvonosné složky

jsou často jen aplikovány transformace a operace, které byly výstupem z vyhodnocení jasové složky. Proto se i tato vypracovaná aplikace zaměřuje na jasovou složku a její zpracování.

Jednotlivým snímkům je po podvzorkování přiřazen konkrétní typ, jakým bude prováděna predikce. Zda bude kódován bez vazby na ostatní snímky, v režimu Intra, nebo bude využívat jednosměrnou nebo obousměrnou predikci v režimu Inter. Snímky jsou nejprve přeuspořádány tak, aby byla možná dopředná i zpětná predikce a následně jsou na ně aplikovány predikce.

Při predikci typu Intra je využita zejména prostorová redundance, tj. předpoklad, že jednotlivé sousedící vzorky jsou shodné nebo velmi podobné. Naproti tomu Inter predikce využívá faktu, že obraz s určitou snímkovou frekvencí se mění jen pomalu a že objekty na jednom snímku lze nalézt na stejné nebo blízké pozici v následujícím snímku.

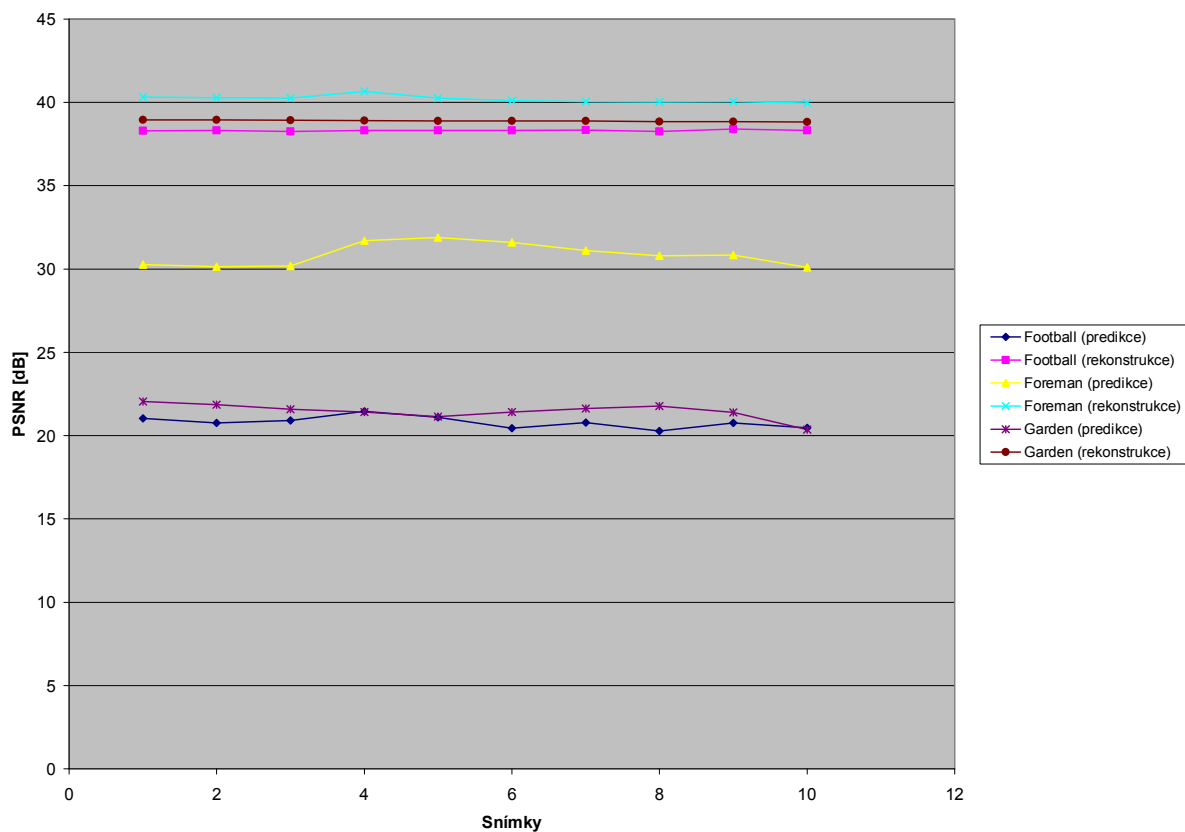
Nedojde-li v sekvenci ke střihu, tj. k okamžité zásadní změně scény, pak i posloupnost snímků zpracovaných konkrétní posloupností algoritmů vykazuje na výstupu značnou korelaci. To zobrazují hodnoty PSNR pro obrázky v dané sekvenci snímků. Na Obr. 5.1 je vidět závislost hodnoty PSNR na pořadí snímku I.



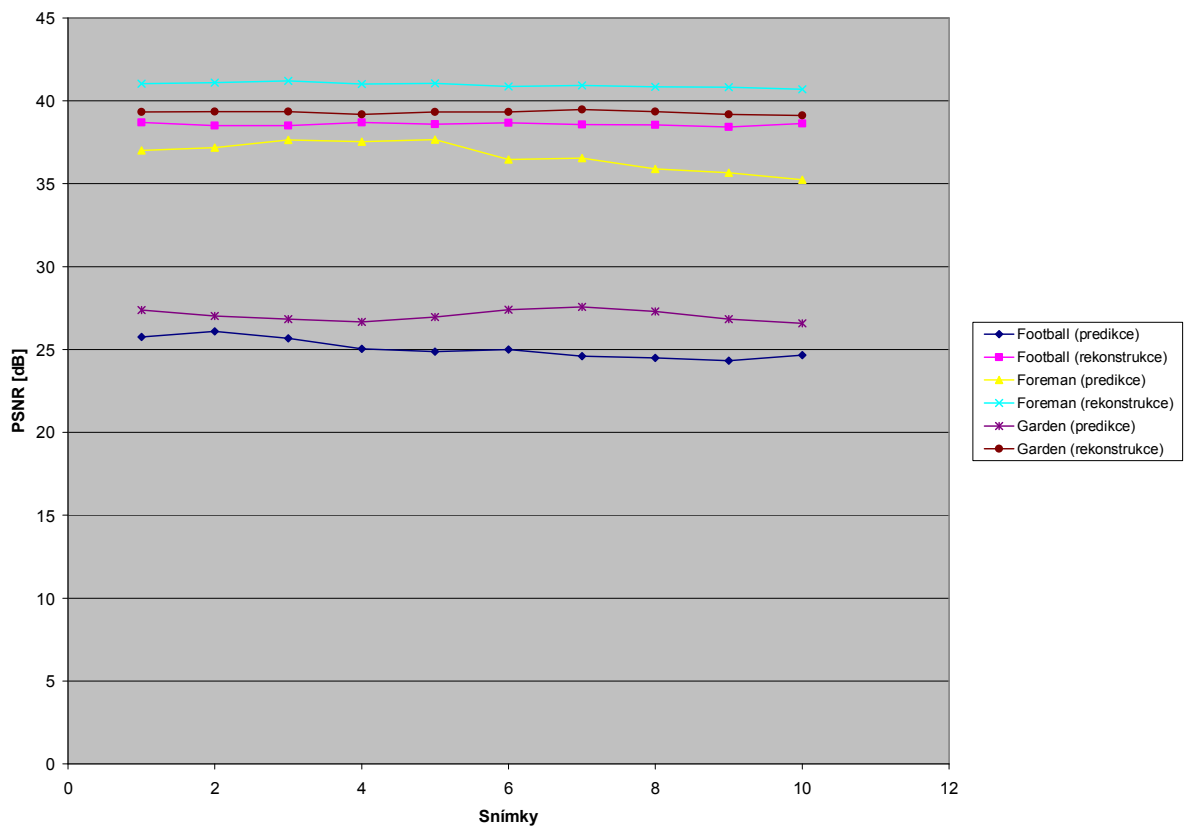
Obr. 5.1: Zobrazení závislosti PSNR na pořadí snímku typu I (QP = 25, TSS, kvantování AVC)

U snímků typu P a B se jedná především o časovou redundanci. Vzhledem k tomu, že sekvence mají vysokou snímkovou frekvenci, tak i hodnoty PSNR pro jednotlivé snímky sekvencí jsou obdobné, a to při použití stejných kódovacích kritérií.

V grafu snímků typu P na Obr. 5.2 je patrná nižší hodnota PSNR, neboť se vychází pouze z jednoho referenčního snímku, na rozdíl od snímků typu B, viz Obr. 5.3, které při rekonstrukci využívají dvou referenčních snímků. Tento odlišný princip rekonstrukce pomáhá u snímků typu B zlepšit hodnotu PSNR přibližně o 5 – 7 dB.



Obr. 5.2: Zobrazení závislosti PSNR na pořadí snímku typu P (QP = 25, TSS, kvantování AVC)



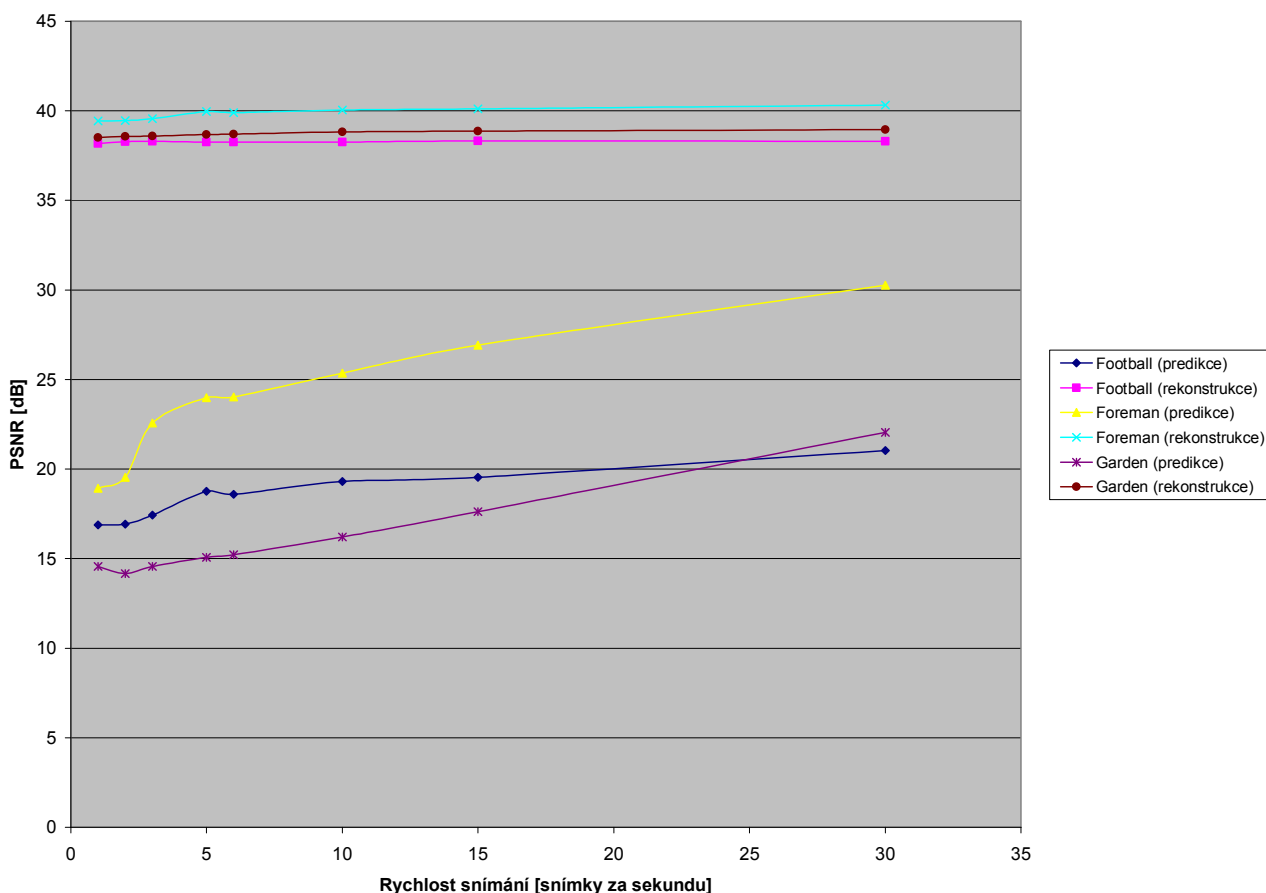
Obr. 5.3: Zobrazení závislosti PSNR na pořadí snímku typu B (QP = 25, TSS, kvantování AVC)

5.4 VLIV RYCHLOSTI SNÍMÁNÍ

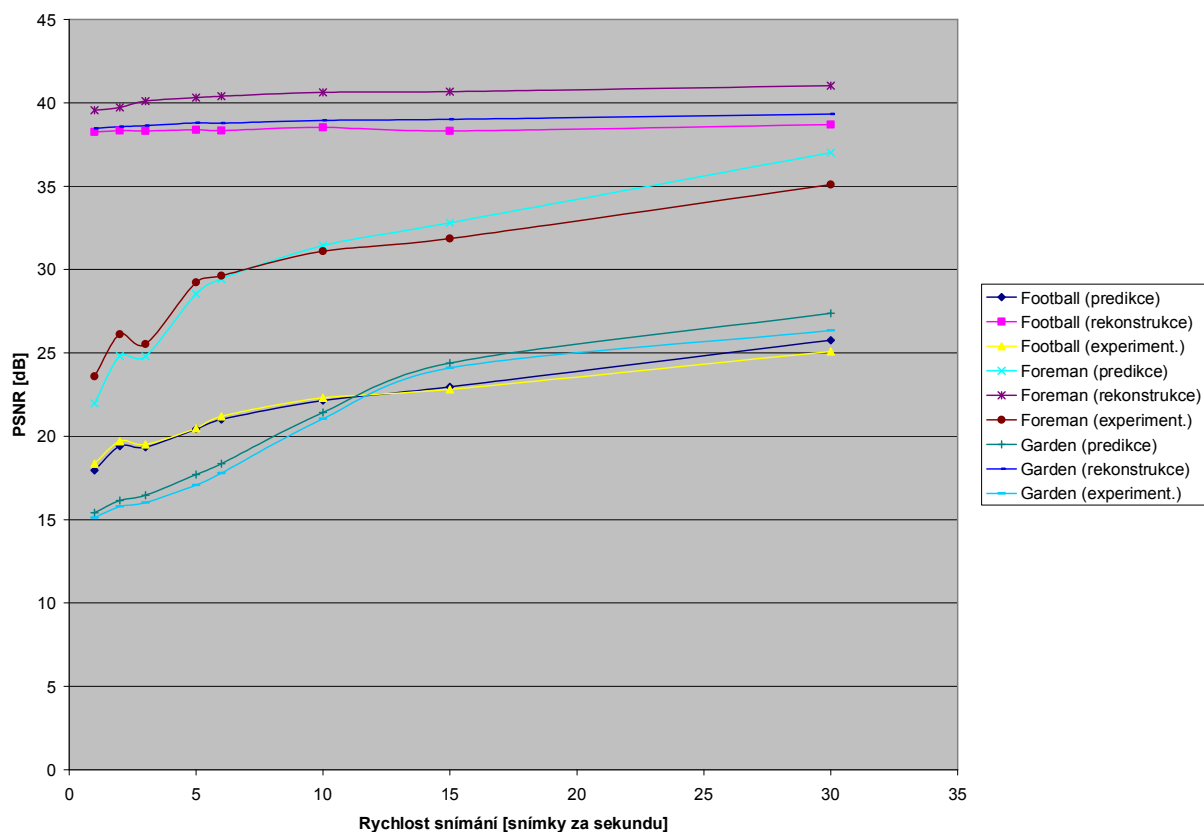
Snímková frekvence hraje významnou roli pro určování vektorů pohybu a pro tzv. časovou redundanci. Platí, že čím je snímání scény častější, tím menší změny pohybu mohou být zachyceny. Se zvoleným prohledávacím parametrem, tj. velikostí oblasti ve které bude vyhledáván nejvíce odpovídající blok, tak může být dosaženo lepších výsledků při určování vektorů pohybu. Na Obr. 5.4 je patrný výrazný vliv vyšší rychlosti snímání na hodnotu PSNR u jednostranně predikovaných snímků typu P. Lepších výsledků je dále dosaženo při obousměrné predikci u snímků typu B, s využitím dvou vektorů pohybu na predikovaný blok, viz Obr. 5.5.

V grafu pro obousměrnou predikci u snímků typu B je rovněž znázorněno experimentální určení snímku typu B pomocí „jednoho vektoru a dvou referenčních snímků“. Bylo zjištěno, že tato experimentální funkce dosahuje lepších výsledků při výraznějších změnách scény (nízká snímková frekvence) a také při zásadních změnách scény (při střihu) oproti klasickému principu predikce.

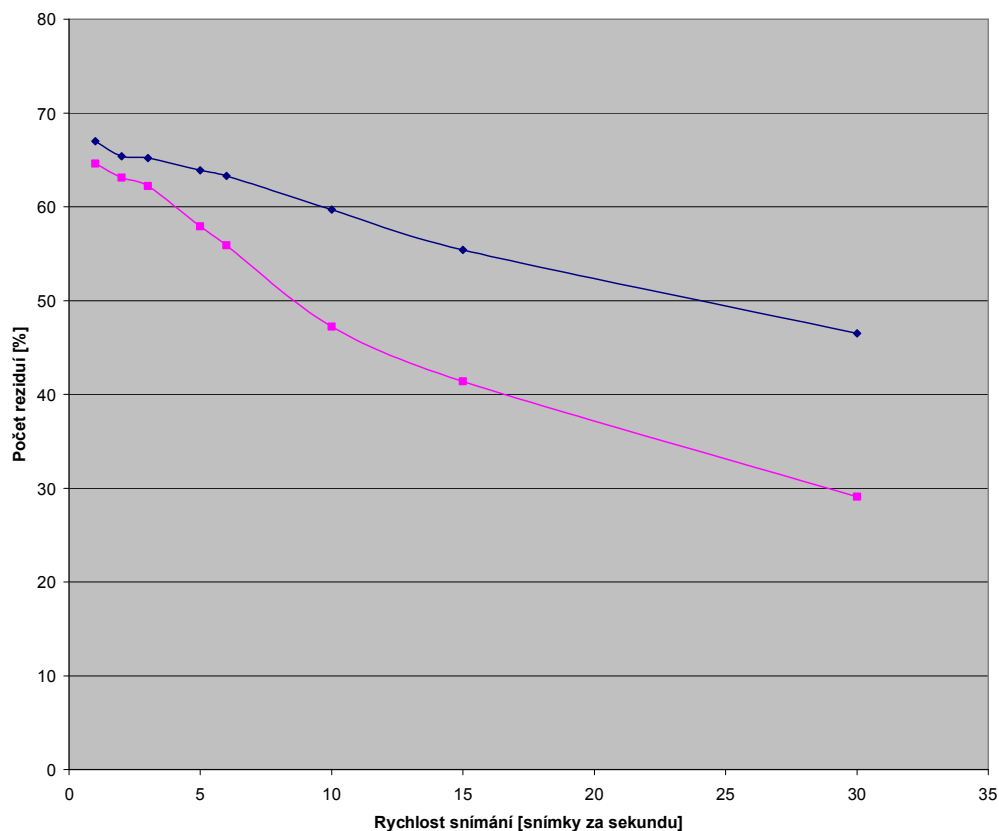
Při zvyšující se frekvenci snímání lze tedy lépe detekovat pohyb objektů (bloků) a toho využít pro určení kompenzace pohybu. Není tak nutné přenášet vysoký počet reziduí (která představují chybu predikce), čímž se ušetří potřebný bitový tok. Pokles počtu reziduí v závislosti na zvyšující se frekvenci snímání, při konstantním kvantovacím parametru QP , představuje obrázek Obr. 5.6. Je třeba mít ovšem také na mysli, že s vyšší snímkovou frekvencí roste také počet snímků, které musí být zpracovány a tím roste i zátěž pro kódovací a dekódovací zařízení a v neposlední řadě roste i datový tok tvořený vektory pohybu i chybou predikce u těchto snímků.



Obr. 5.4: Závislost PSNR na snímkové frekvenci u snímků P ($QP=25$, TSS, kvant. AVC, 16x16)



Obr. 5.5: Závislost PSNR na snímkové frekvenci u snímků B (QP=25, TSS, kvant. AVC, 16x16)



Obr. 5.6: Závislost počtu reziduí na snímkové frekvenci (sekvence Garden, QP=25, TSS, kv. AVC)

5.5 INTRA PREDIKCE

Při provádění Intra predikce dochází k úspoře datového toku využitím již dekodovaných vzorků po jejich rekonstrukci. Intra predikce je u jasové složky tvořena pro bloky o velikosti 4x4 nebo 16x16, vždy pouze v rámci konkrétního proužku. Následně tak jsou kódovány jen rezidua, která nesou informaci o rozdílu predikovaných bloků a originálních bloků. Detailně byly principy této predikce popsány v kapitole 3.3.3.

Pro bloky o velikosti 4x4 je ve standardu definováno celkem 9 režimů predikce, mezi nimiž kodér vybírá ten nejvhodnější dle míry shody. Potřebný počet vzorků pro vytvoření predikovaného bloku činí 25 % – 50 % z celkového počtu šestnácti vzorků, viz Tab. 5.2.

Bloky o velikosti 16x16 jsou tvořeny obdobným způsobem, avšak jsou definovány pouze čtyři módy jasové Intra predikce, které jsou obdobou režimů u bloků 4x4. Celkový počet vzorků v bloku je 256 a tak je zřejmé, že je při využití znalosti pouze horního a levého sloupce vzorků získán lepší poměr úspory přenesených dat. Ten dle použitého režimu predikce činí 6,25 % nebo 12,5 %.

Tab. 5.2: Potřebný počet vzorků pro vytvoření predikovaného bloku dle režimu predikce

Predikce	Režim	Potřebný počet vzorků	Poměr
4x4 16 vzorků na blok	0 Vertical	4	25,00%
	1 Horizontal	4	25,00%
	2 DC	8	50,00%
	3 Diagonal Down-Left	7	43,75%
	4 Diagonal Down-Right	7	43,75%
	5 Vertical-Right	5	31,25%
	6 Horizontal-Down	5	31,25%
	7 Vertical-Left	5	31,25%
	8 Horizontal-Up	4	25,00%
16x16 256 vzorků na blok	0 Vertical	16	6,25%
	1 Horizontal	16	6,25%
	2 DC	32	12,50%
	3 Plane	32	12,50%

Z praktického vyhodnocení různých režimů predikce je patrné, že při použití predikce pro bloky o velikosti 16x16 dojde v porovnání s bloky 4x4 ke zmenšení hodnoty PSNR predikovaného snímku. Nejlépe z ověřovaných režimů v rámci jedné velikosti bloku vychází ve většině případů predikce typu *DC*. Vyhodnocení pro $QP = 51$, tj. získání PSNR pouze pro predikci, s potlačením vlivu reziduí na rekonstrukci, poskytuje Tab. 5.3.

Tab. 5.3: Hodnota PSNR pro Intra kódované snímky s kvantovacím parametrem $QP = 51$

QP = 51 Typ	4 x 4			16 x 16		
	Horizontální	Vertikální	DC	Horizontální	Vertikální	DC
Missa - PSNR [dB]	33,01	33,3	32,83	31,04	31,03	31,63
Missa - rezidua [%]	0	0	0	0	0	0
Football - PSNR [dB]	22,44	22,45	22,42	21,58	21,68	22,31
Football - rezidua [%]	0,8	0,5	0,9	1,1	1	0,9
Garden - PSNR [dB]	20,57	20,02	20,44	19,8	19,64	20,39
Garden - rezidua [%]	0,8	1,3	1	1,8	1,7	1

Pro ilustraci Intra predikce s reálnou hodnotou kvantovacího parametru byla provedena tatáž simulace, avšak s $QP = 20$. Z porovnání hodnot je patrný velký vliv kvantování na výslednou hodnotu PSNR. Díky použití reziduí při rekonstrukci bylo dosaženo výsledného hodnocení scény ve všech případech v rozmezí 43 – 45 dB.

Tab. 5.4: Hodnota PSNR pro Intra kódované snímky s kvantovacím parametrem $QP = 20$

QP = 20	4 x 4			16 x 16		
	Horizontální	Vertikální	DC	Horizontální	Vertikální	DC
Missa - PSNR [dB]	44,43	44,72	44,76	44,36	44,69	44,76
Missa - rezidua [%]	17,3	15	14,9	18,7	17,7	14,9
Football - PSNR [dB]	43,09	43,08	43,1	43,07	43,06	43,1
Football - rezidua [%]	57,3	57,8	57,2	58,6	59,1	57,2
Garden - PSNR [dB]	43,61	43,58	43,62	43,57	43,56	43,62
Garden - rezidua [%]	55,4	56,5	55,7	56,6	57,5	55,7

5.6 INTER PREDIKCE

Při Inter predikci dochází v realizovaném kodéru k určování vektorů pohybu u kódovaného snímku. Uživatelsky je dána možnost volby velikosti bloku pro který budou vektory určovány. Tato velikost bloku je použita na celý kódovaný snímek. Rovněž je možné změnou prohledávacího parametru *prohl_par* v kódu aplikace ovlivnit rozsah, neboli oblast, kde budou vektory hledány.

Oba parametry, jimiž je velikost bloku a prohledávací parametr, mají zásadní vliv na počet nezbytných výpočtů a rovněž tak i na dobu běhu algoritmu. Představu o počtu hledání v rámci snímku poskytuje Tab. 5.5, jenž uvádí dostupné velikosti bloků a zvolený prohledávací parametr, jenž prakticky omezuje prohledávání okolí bloku v rozsahu necelé poloviny velikosti daného bloku. Počet hledání u jednotlivých prohledávacích metod vychází z výpočtů uvedených v kapitole 2.1.9 a 2.1.10.

Tab. 5.5: Počty hledání vektorů pohybu na blok a makroblok (16 x 16)

Počty hledání MV		Pixelová přesnost					
Velikost		Full search		Three step search		NNS, počet kroků = 3	
Blok	prohl_par	Na blok	Na makroblok	Na blok	Na makroblok	Na blok	Na makroblok
4x4	1	9	144	9	144	13	208
8x8	3	49	196	17	68	13	52
16x16	7	225	225	25	25	13	13
Počty hledání MV		Půlpixelová přesnost +9					
Velikost		Full search		Three step search		NNS, počet kroků = 3	
Blok	prohl_par	Na blok	Na makroblok	Na blok	Na makroblok	Na blok	Na makroblok
4x4	1	18	288	18	288	22	352
8x8	3	58	232	26	104	22	88
16x16	7	234	234	34	34	22	22
Počty hledání MV		Čtrtpixelová přesnost +49					
Velikost		Full search		Three step search		NNS, počet kroků = 3	
Blok	prohl_par	Na blok	Na makroblok	Na blok	Na makroblok	Na blok	Na makroblok
4x4	1	58	928	58	928	53	848
8x8	3	98	392	66	264	53	212
16x16	7	274	274	74	74	53	53

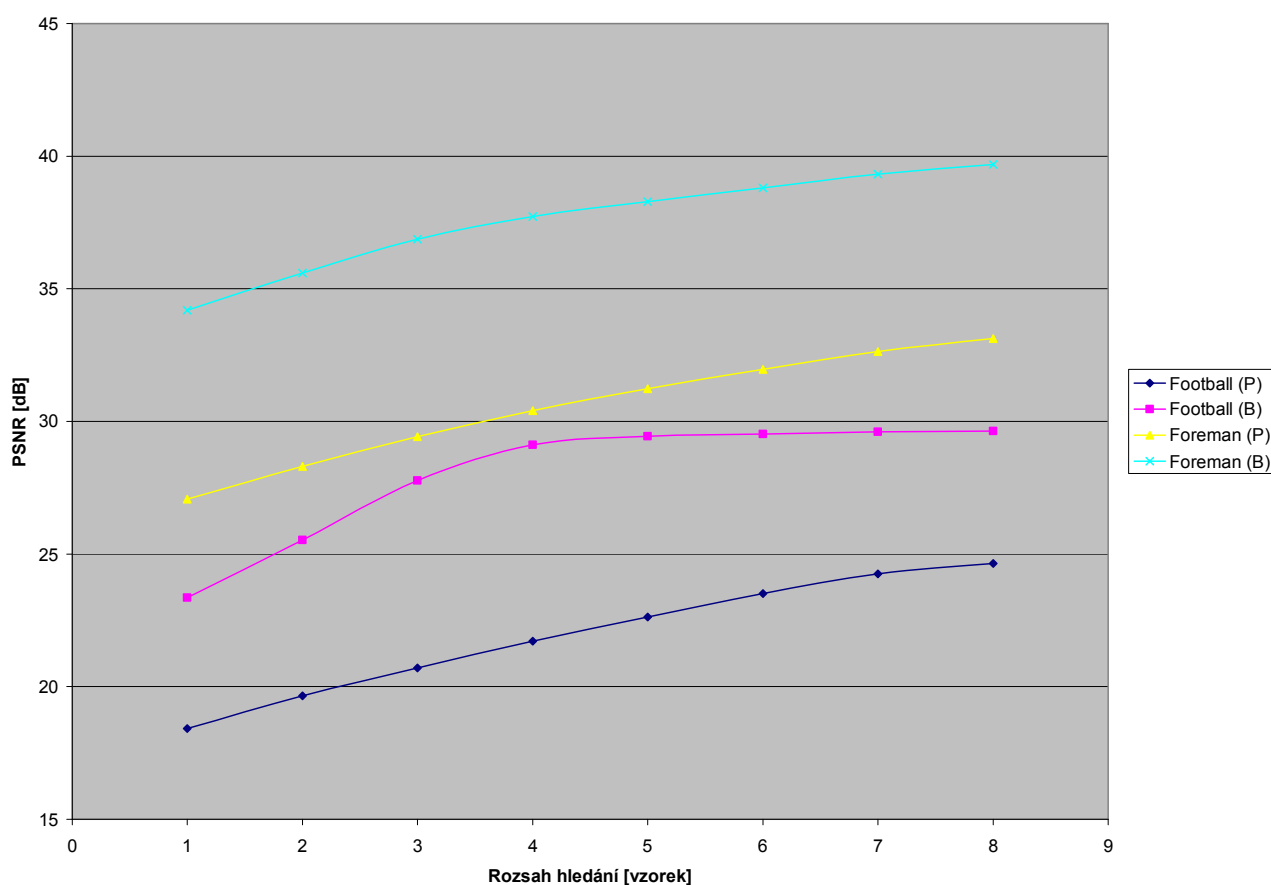
Nárůst počtu hledání vektorů pohybu (MV) pro půlpixelovou přesnost (+9, tj. 3 x 3) a čtvrtpixelovou přesnost (+49, tj. 7 x 7) vychází z potřeby dohledat optimální vektor pohybu se subpixelovou přesností. Nejprve je totiž určen vektor pohybu s pixelovou přesností, následně je provedena interpolace vzorků snímku na požadovanou subpixelovou přesnost a nakonec dojde k vyhledání optimální pozice (vektoru) z původní pozice i z okolních subpixelových pozic.

5.7 ROZSAH HLEDÁNÍ SHODY BLOKU

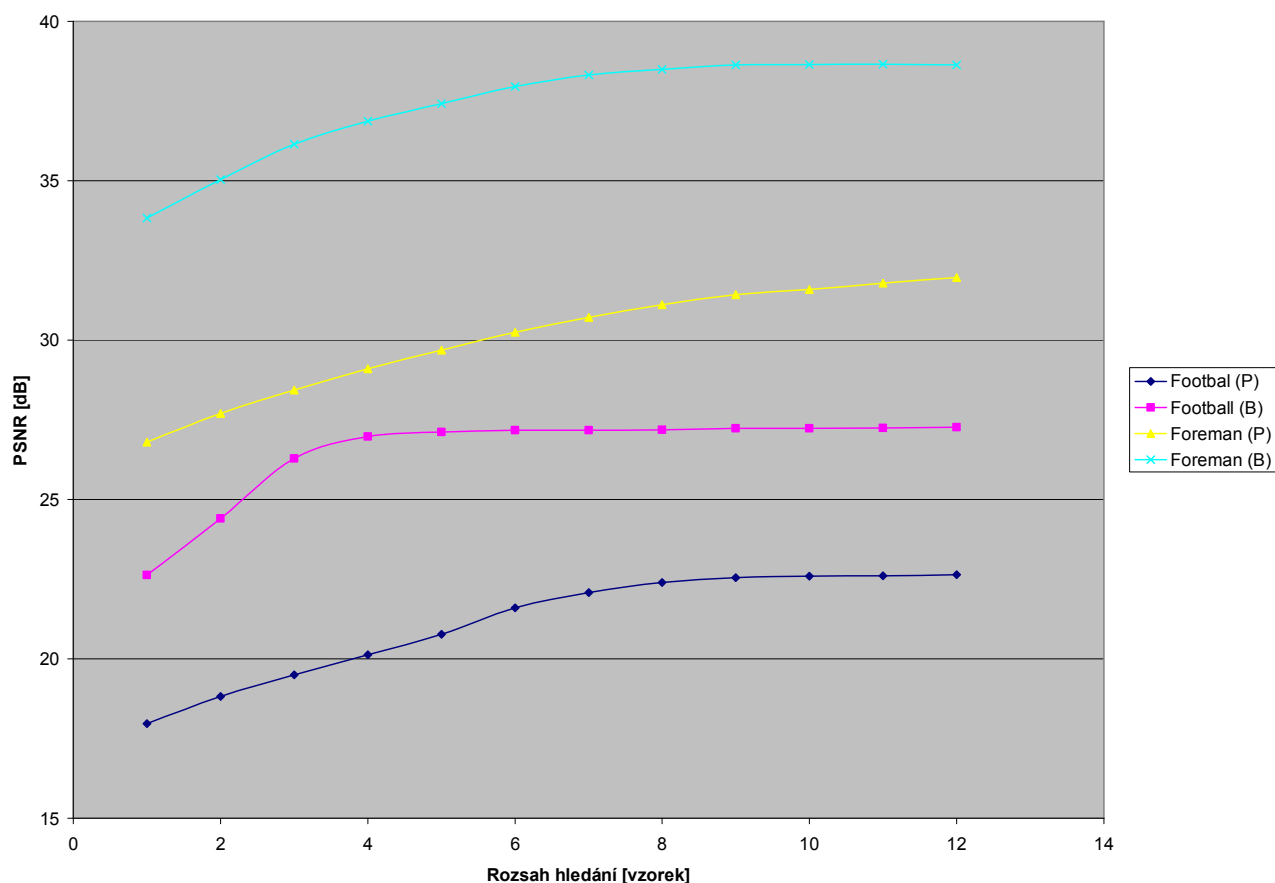
Při určování vektorů pohybu je kromě znalosti velikosti bloku, pro nějž je hledána nejlepší shoda, rovněž nutné znát oblast ve které má být tato shoda nalezena. Malá prohledávaná oblast neposkytuje možnost nalezení dostatečně odpovídajícího bloku u rychle se pohybujících objektů ve snímku a naopak příliš velká prohledávaná oblast zvyšuje požadavky na výpočetní výkon kodéru. Pro porovnání shody mezi prohledávaným a referenčním snímkem je v aplikaci použita metrika MAE, tj. střední absolutní chyba.

Význam vhodné volby prohledávacího parametru při Inter predikci s velikostí bloku 8x8 zobrazuje Obr. 5.7, na kterém je patrné, jak každé zvětšení prohledávané oblasti zásadně ovlivní hodnotu PSNR. Současně je ovšem třeba mít na zřeteli, že počet potřebných výpočtů, a tedy i čas, k nalezení těchto nejvíce odpovídajících vektorů pohybu, roste exponenciálně.

Následující Obr. 5.8 znázorňuje tutéž závislost, ovšem pro velikost bloku 16x16. Je patrné, že pro bloky 8x8 je získána vyšší hodnota PSNR, než pro bloky 16x16, při stejném prohledávacím parametru. Oba grafy byly vytvořeny s pomocí úplného prohledávacího algoritmu Full Search.



Obr. 5.7: Závislost PSNR na rozsahu prohledávané oblasti (bloky 8x8, kvantování AVC, QP=4)



Obr. 5.8: Závislost PSNR na rozsahu prohledávané oblasti (bloky 16x16, kvantování AVC, QP=4)

5.8 PŘESNOST VEKTORŮ POHYBU

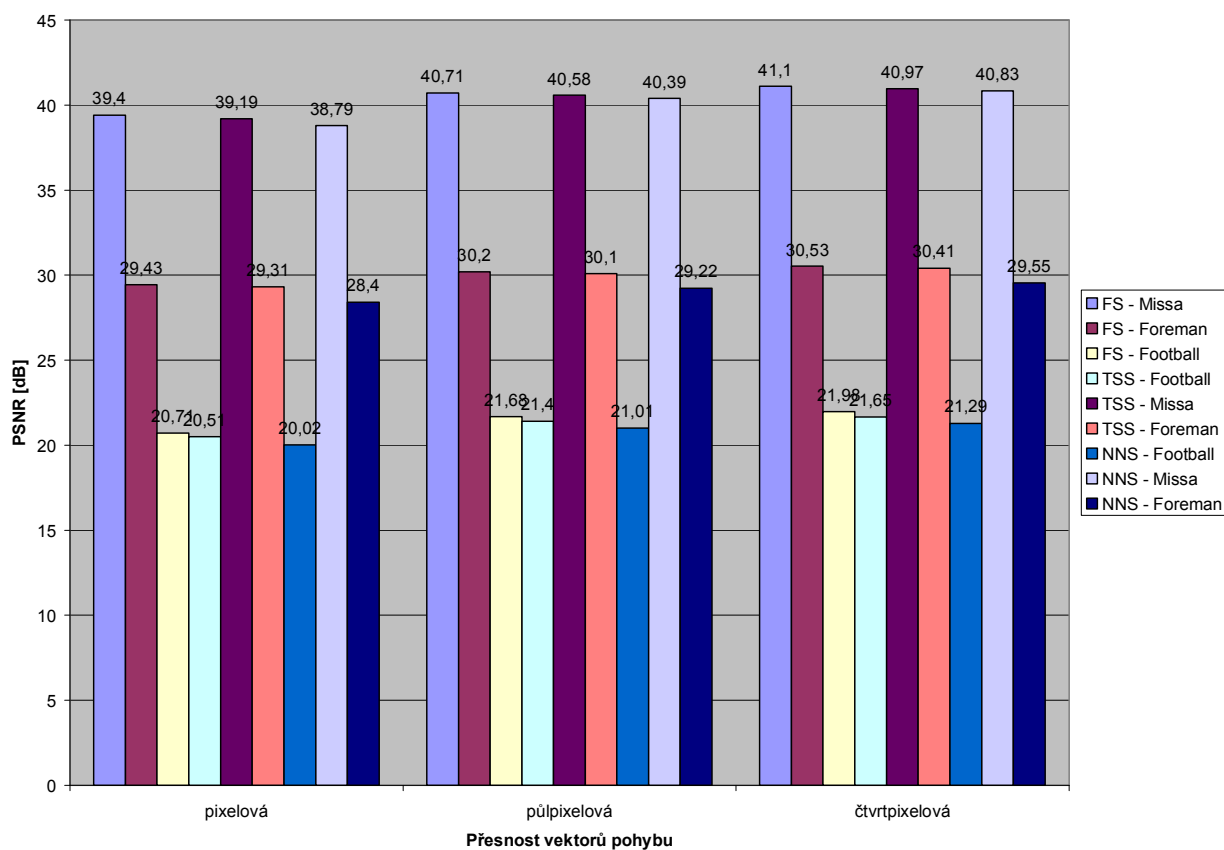
Určování vektorů pohybů se subpixelovou přesností patří mezi hlavní pilíře Inter predikce u snímků typu P a B. V realizované aplikaci lze volit mezi pixelovou, půlpixelovou a čtvrtpixelovou přesností vektorů pohybů pro bloky o velikosti 4x4, 8x8 a 16x16. Rovněž je možné vybrat typ prohledávacího algoritmu pro kompenzaci pohybu, případně upravit velikost prohledávacího parametru. Tyto uvedené volby poskytují mnoho kombinací, pomocí nichž mohou být vektory pohybu určeny.

Jestliže má být vyhodnocen vliv přesnosti vektorů pohybu na hodnotu PSNR, pak budou ostatní parametry ponechány beze změny a bude určeno jakého rozdílu bude dosaženo při pixelové a subpixelové přesnosti.

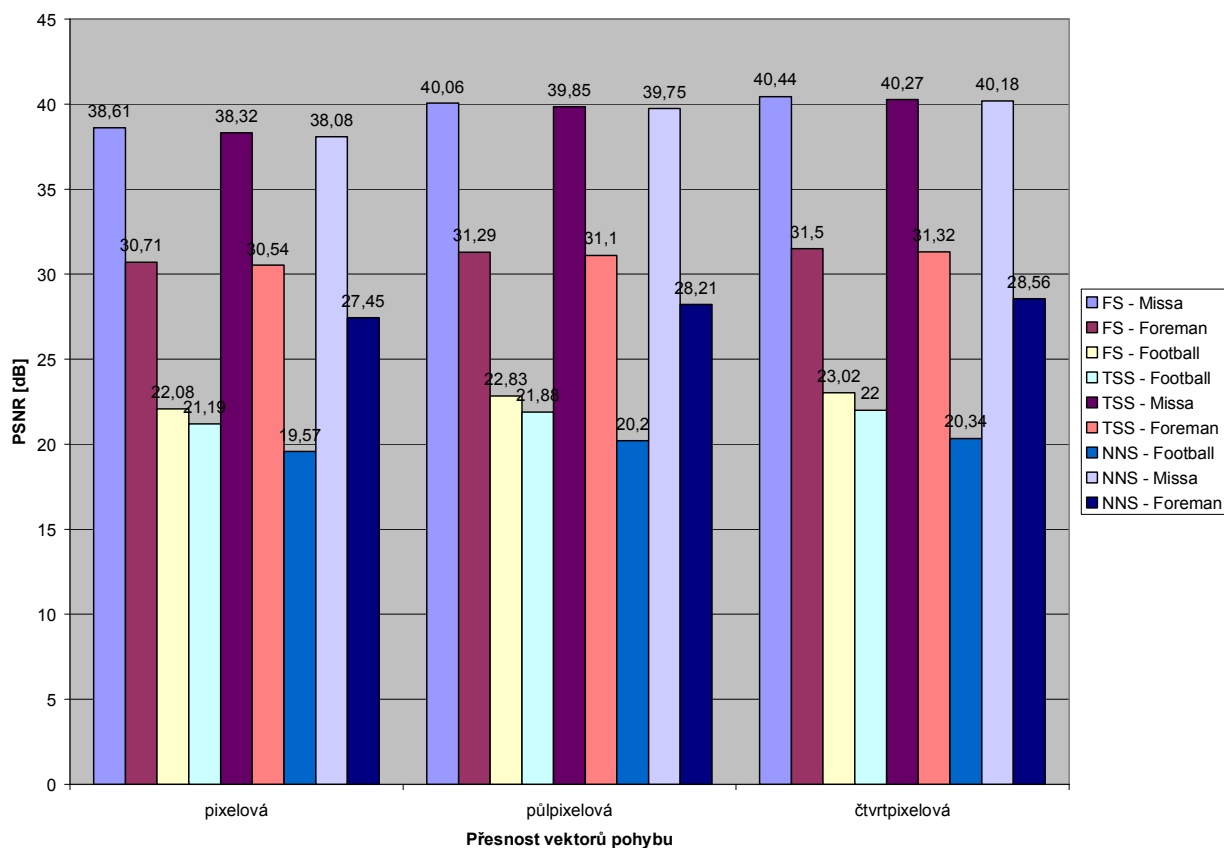
Na Obr. 5.9 je pro bloky o velikosti 8x8 znázorněna závislost parametru PSNR na přesnosti vektorů pohybu pro zvolenou trojici snímků, a to s různými typy prohledávacích algoritmů. Jak bylo možné očekávat, nejlepších výsledků dosahuje algoritmus Full Search, těsně následovaný algoritmem Three Step Search a Nearest Neighbours Search, s odstupy v řádu desetin dB.

Rovněž u porovnání mezi přesnostmi vektorů pohybu je výsledek dle očekávání. Nejlepší hodnoty podává čtvrtpixelová přesnost, dále pak půlpixelová a pixelová. Zde je odstup mezi hodnotami PSNR vyšší a pohybuje se v desetinách dB, často však kolem 1 dB.

Na Obr. 5.10 je znázorněno srovnání pro stejnou trojici snímků, se stejnými typy prohledávacích algoritmů, avšak pro bloky o velikosti 16x16. I zde se hodnoty PSNR u jednotlivých algoritmů a přesností liší o desetiny dB, obdobně jako tomu bylo u bloků 8x8.



Obr. 5.9: Závislost PSNR na přesnosti vektorů pohybu (bloky 8x8, QP=4, prohl_par=3)

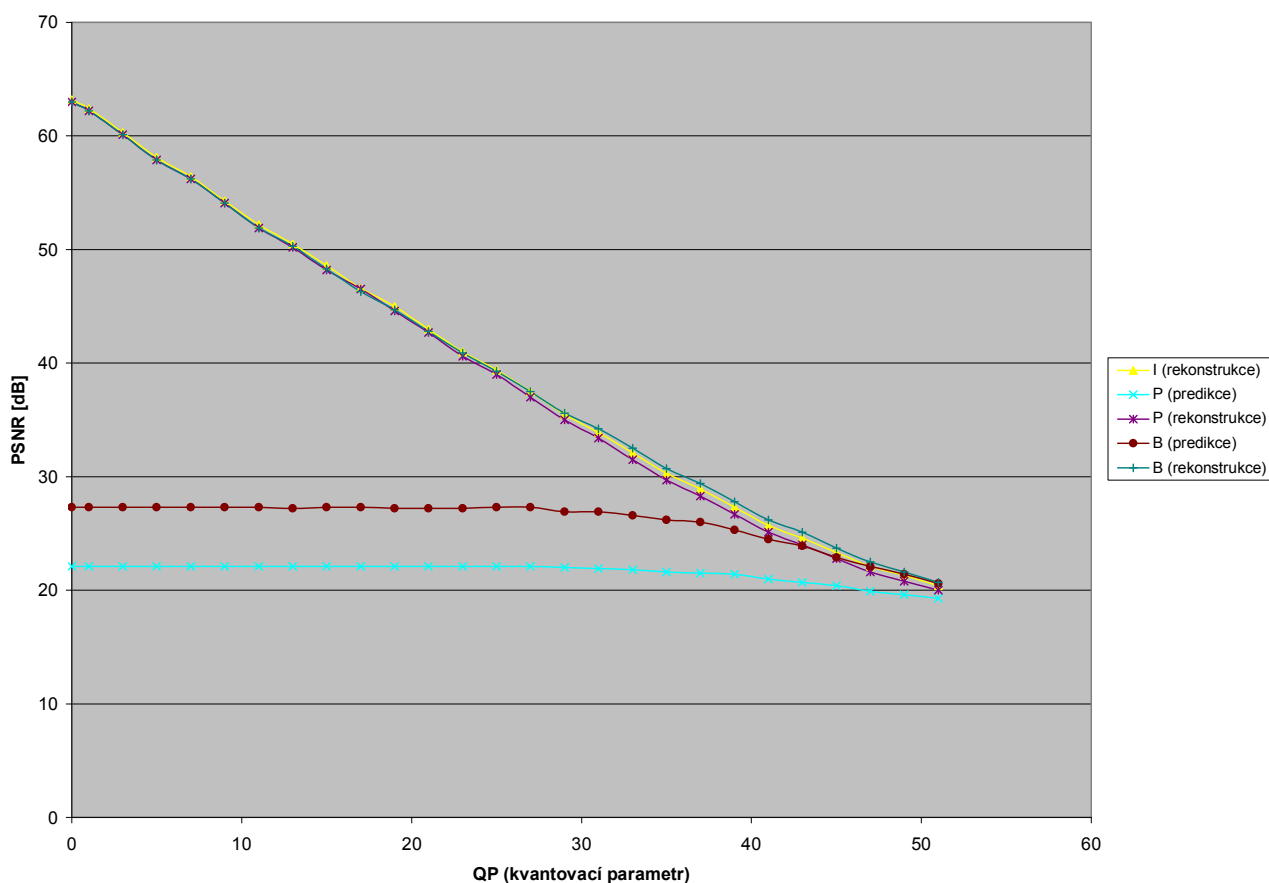


Obr. 5.10: Závislost PSNR na přesnosti vektorů pohybu (bloky 16x16, QP=4, prohl_par=7)

5.9 VLIV KVANTOVÁNÍ

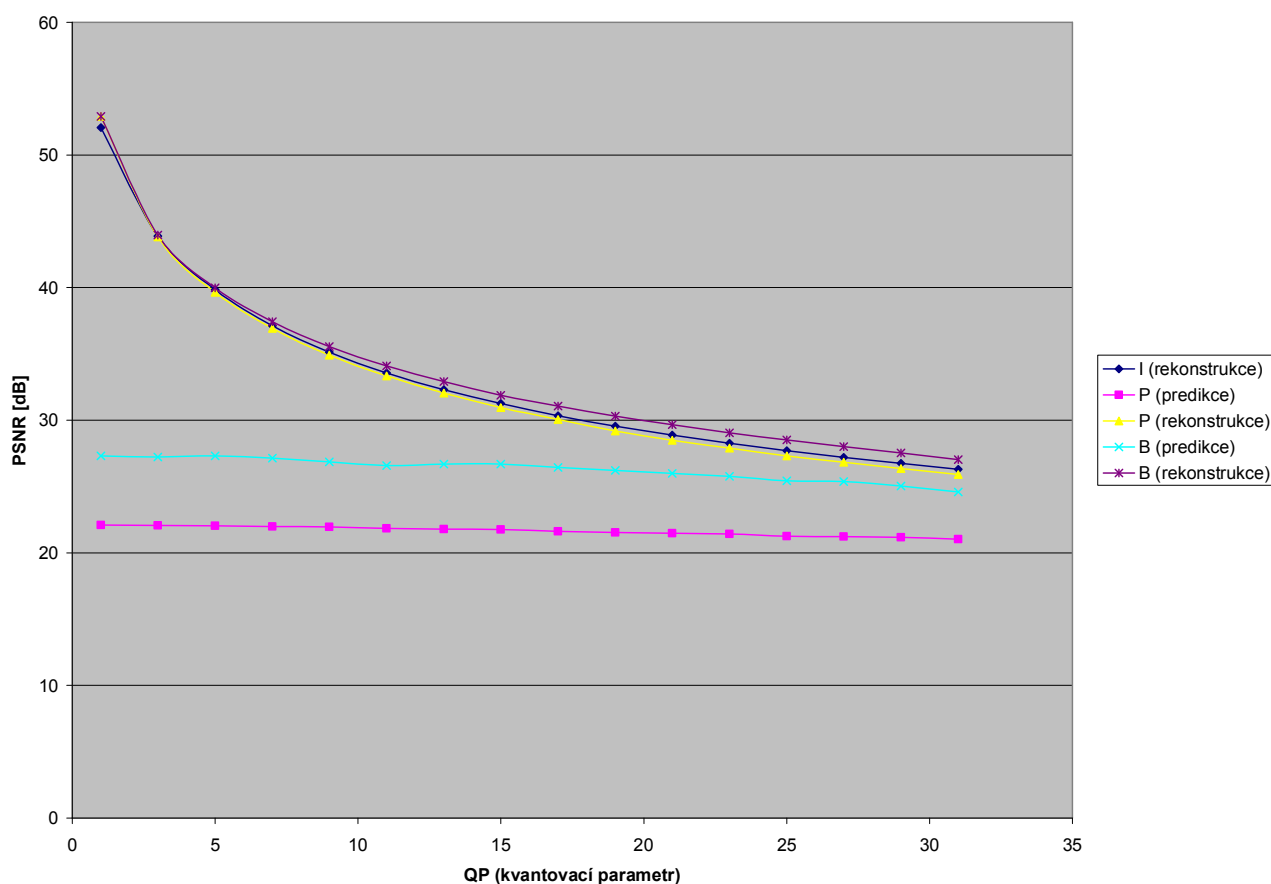
Kvantování je operace při níž dochází k zásadnímu ovlivnění množství přenášených dat použitím ztrátové komprese. V aplikaci lze zvolit předdefinované metody kvantování podle standardu MPEG-4 Visual, MPEG-4 AVC a pomocí ručního zadání kvantovacích matic. K nastavení kvantovacího kroku u standardů MPEG-4 slouží volba hodnoty QP , jenž je pro MPEG-4 Visual definována v rozsahu 1 – 31 a pro MPEG-4 AVC v rozsahu 0 – 51, tak jak je uvedeno v daných standardech a rovněž detailně popsáno v kapitole 3.3.6.2.

Hodnoty kvantovacího kroku $Qstep$ jsou u MPEG-4 AVC definovány tím způsobem, že hodnota $Qstep$ zdvojnásobuje svou velikost pro každý násobek šesti parametru QP . Tato posloupnost uvedená v Tab. 3.4 má za následek, že hodnota PSNR se zvyšující se hodnotou parametru QP klesá přibližně lineárně. Při vysokých hodnotách kvantovacího kroku dochází k tak vysoké redukci reziduí, že hodnota PSNR snímku po rekonstrukci se začíná velmi přibližovat hodnotě PSNR predikovaného snímku, což je velmi zřetelné na Obr. 5.11, pro případ kvantování dle standardu MPEG-4 AVC. Pokles hodnoty PSNR u predikovaných snímku pro vyšší hodnoty QP je způsoben výraznou redukcí reziduí referenčního snímku I po provedení Intra predikce a kvantování.



Obr. 5.11: Závislost PSNR na kvantovacím parametru QP (sekvence Garden, vyhledávání TSS, kvantování dle MPEG-4 AVC, blok 16x16)

Při aplikování kvantování podle standardu MPEG-4 Visual je získán odlišný průběh grafu, s menší strmostí poklesu PSNR, viz Obr. 5.12, neboť i hodnoty kvantovacího kroku nedosahují takové hrubosti jako u MPEG-4 AVC. Hodnota kvantovacího kroku se pohybuje pro jasové DC koeficienty v rozmezí od 8 do 46 a pro ostatní hodnoty AC koeficientů přibližně v hodnotě dvojnásobku kvantovacího parametru QP , jak bylo uvedeno v kapitole 3.3.6.2 a v Tab. 3.5. Menší hrubost kvantování u nejvyšších hodnot QP tak nemá zásadní vliv ani na predikované snímky P a B, využívající jako reference predikovaného a kvantovaného snímku I.

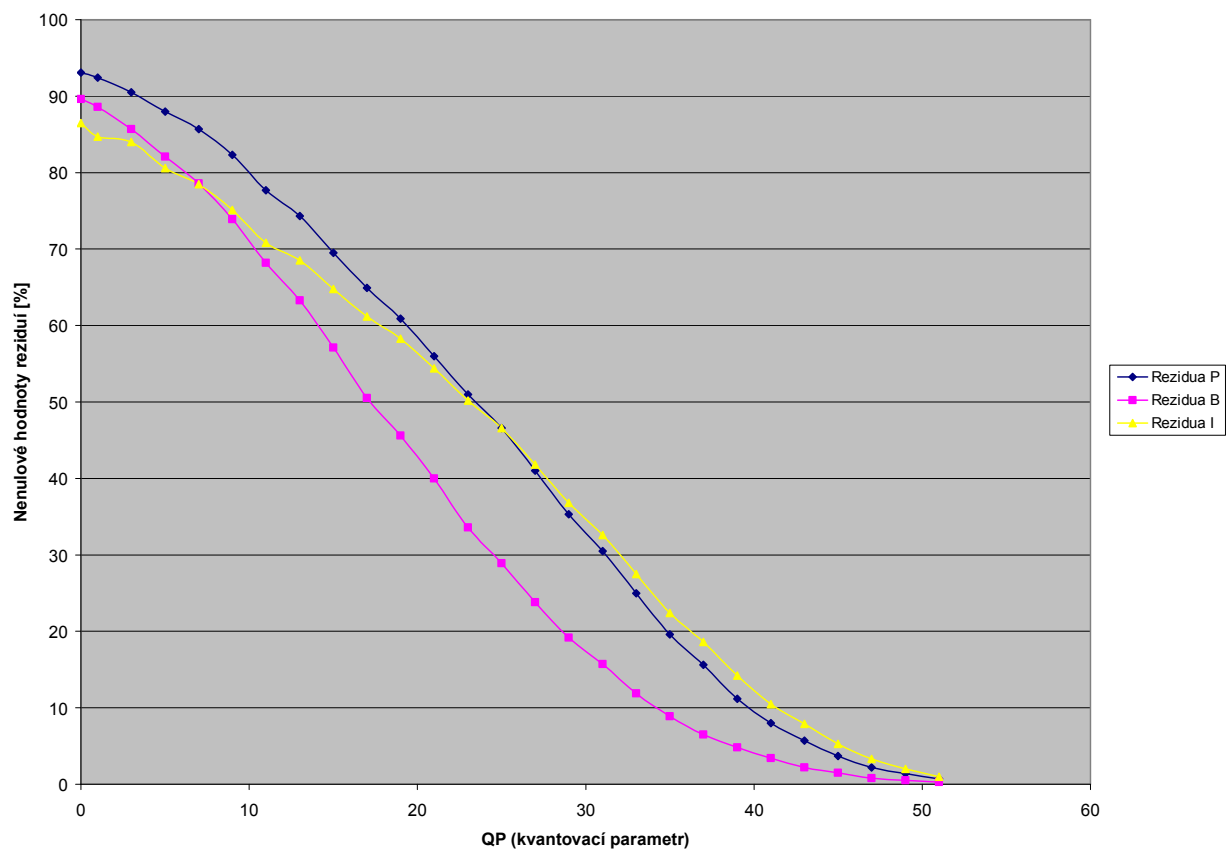


Obr. 5.12: Závislost PSNR na kvantovacím parametru QP (sekvence Garden, vyhledávání TSS, kvantování dle MPEG-4 Visual, blok 16x16)

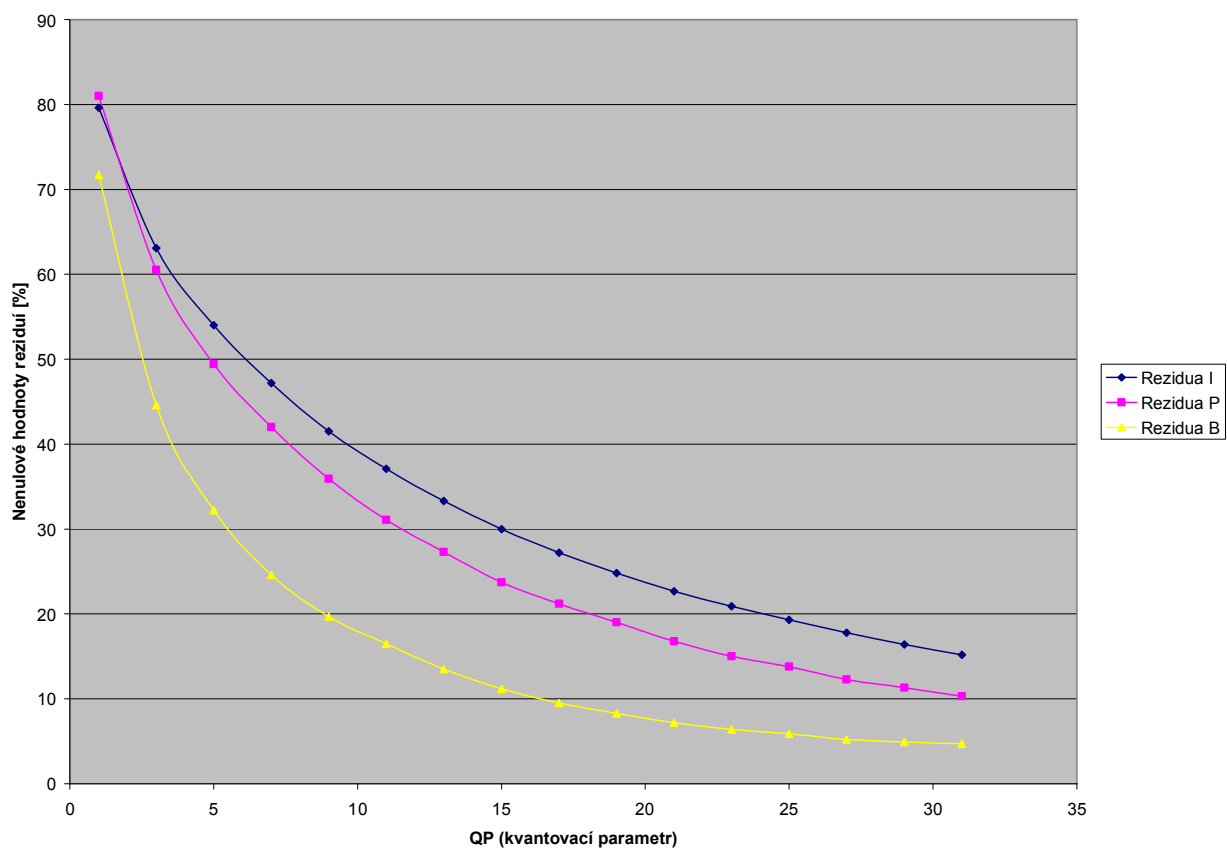
Vzhledem k tomu, že u obou standardů je využito jiné posloupnosti hodnot kvantovacího kroku a i jeho maximální hodnota se výrazně liší pro oba standardy, je zřejmé, že tento fakt bude mít významný vliv na množství reziduí, jenž budou i po kvantování dosahovat nenulových hodnot.

U standardu MPEG-4 AVC je nejvyšší hodnota kvantovacího kroku $QStep = 224$ (pro $QP = 51$). Lze očekávat, že takto vysokých hodnot reziduí u rozdílových snímků po provedení predikce bude dosahovat naprosté minimum koeficientů. Tuto domněnku zcela potvrzuje i Obr. 5.13, jenž ilustruje, jak množství nenulových reziduí klesá s rostoucí hodnotou QP .

Pro standard MPEG-4 Visual jsou typické nižší hodnoty kvantovacího kroku pro vyšší hodnoty QP , než je tomu u standardu MPEG-4 AVC, dále též jeho odlišný průběh v definovaném rozmezí, a proto i množství reziduí s nenulovou hodnotou po provedení kvantování klesá s odlišným průběhem. I pro nejvyšší hodnotu QP tak dosahuje množství zbytkových reziduí nezanedbatelného počtu, jak je patrné z Obr. 5.14.



Obr. 5.13: Závislost množství nenulových reziduí na QP (sekvence Garden, kvantování AVC)

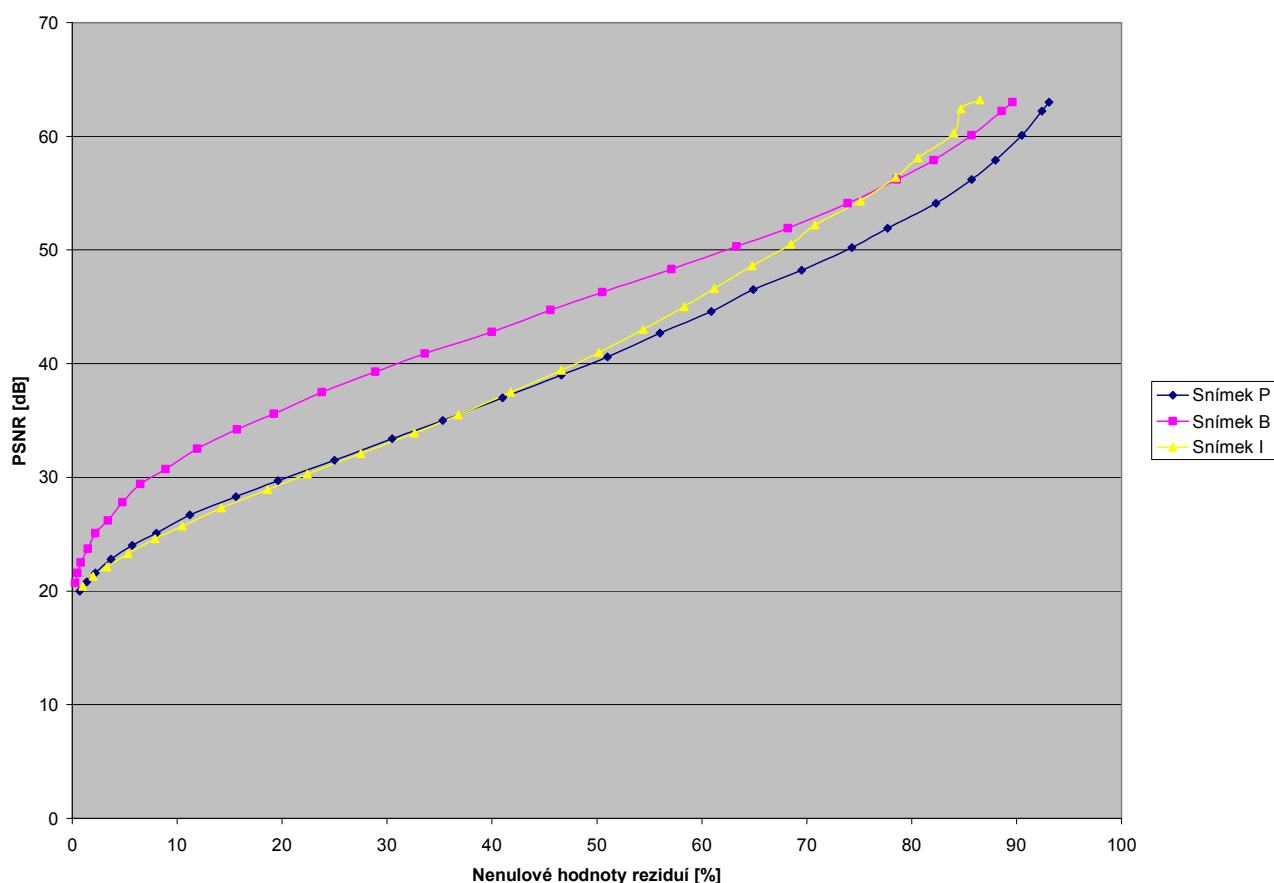


Obr. 5.14: Závislost množství nenulových reziduí na QP (sekvence Garden, kvantování Visual)

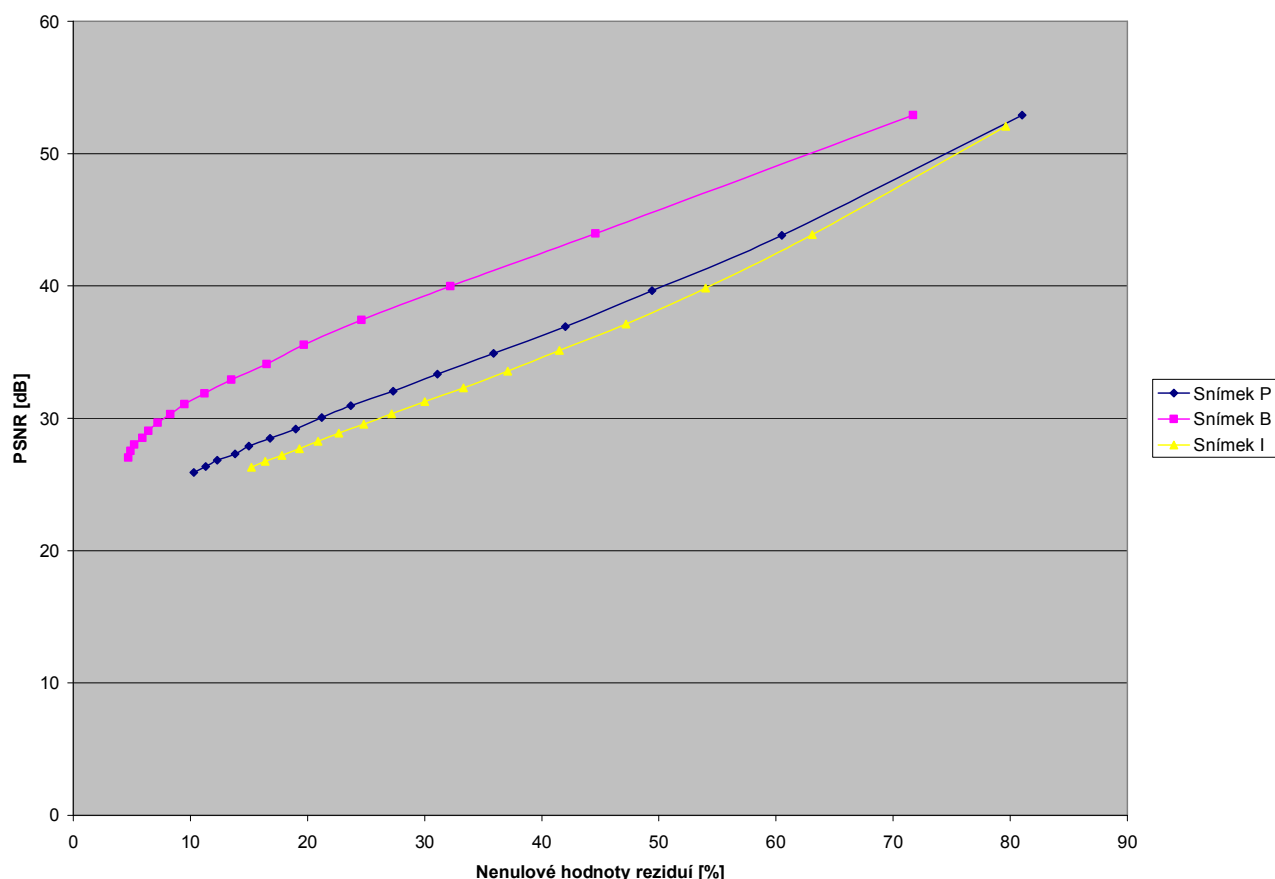
Z dříve uvedených faktů lze rovněž logicky usuzovat, že na základě znalosti množství nenulových reziduí pro určitou hodnotu QP a k nim odpovídající hodnotu PSNR rekonstruovaného snímku, je možné vyjádřit závislost hodnoty PSNR jednotlivých typů snímků na množství nenulových hodnot reziduí. Neboli, jak množství doplňkových informací ovlivní výslednou podobu rekonstruovaného snímku a tím i objektivní metriku hodnocení kvality.

Na Obr. 5.15 je uvedena závislost hodnoty PSNR na množství nenulových hodnot reziduí při použití kvantování dle standardu MPEG-4 AVC. Počáteční hodnota křivek vychází z bodu, kde metrika PSNR pro predikované snímky s téměř nulovým počtem nenulových reziduí má hodnotu 20 dB. Tak jak roste počet informací pro rekonstrukci snímku, tak současně roste i hodnota metriky PSNR. Nejlépe z tohoto srovnání vychází snímek typu B, jenž s výhodou využívá predikce ze dvou referenčních snímků.

Pro kvantování dle standardu MPEG-4 Visual je podoba grafu závislosti hodnoty PSNR na množství nenulových hodnot reziduí znázorněna na Obr. 5.16. Počátečním bodem křivek je místo, ve kterém je kvantovací parametr $QP = 31$ a kdy je tak k dispozici nejmenší počet nenulových reziduí, čemuž odpovídá hodnota PSNR rekonstruovaného snímku, vyznačená na ose y. I v tomto srovnání si mezi jednotlivými typy snímků nejlépe vede snímek typu B, jenž z nárůstu množství doplňkových informací pro predikovaný snímek, dokáže díky obousměrné predikci nejvíce vytěžít.



Obr. 5.15: Závislost PSNR na množství nenulových reziduí (sekvence Garden, kvantování dle MPEG-4 AVC)



Obr. 5.16: Závislost PSNR na množství nenulových reziduí (sekvence Garden, kvantování dle MPEG-4 Visual)

5.10 TRANSFORMACE

V kodérech standardů MPEG-4 se užívá DCT i celočíselná transformace založená na DCT, doplněná též užitím Hadamardovy transformace. Pro porovnání vlivu jednotlivých typů transformací byla v aplikaci umožněna volba čtyř druhů transformace. A to DCT pro bloky 8x8 a 4x4, dále též celočíselná transformace založená na DCT pro bloky 4x4 a tatáž transformace s využitím Hadamardovy transformace.

Pro vyhodnocení vlivu transformace na výslednou hodnotu PSNR bylo provedeno kódování a dekódování s ponecháním ostatních parametrů beze změny, pouze s obměnou typu transformace. Zjištěné hodnoty uvedené v Tab. 5.6 ukazují, že DCT aplikovaná na blocích 4x4 podává lepší výsledky při predikci a kompenzaci pohybu oproti DCT použité na blocích 8x8. Vzhledem k tomu, že použitím DCT na menších blocích jsou získány také menší hodnoty koeficientů reziduí, pak lze očekávat, že i použité kvantování bude mít výraznější dopad na redukci těchto koeficientů při rekonstrukci snímku. Redukce reziduí vlivem kvantování je tak zřetelně výraznější pro transformace na blocích o velikosti 4x4.

U celočíselné transformace založené na DCT je patrný mírný pokles hodnoty PSNR oproti DCT transformaci na téže velikosti bloku. Tento rozdíl lze přisoudit drobné diferenci mezi koeficienty produkované „pravou“ DCT a transformací založenou na DCT. Tento rozdíl je způsoben snahou produkovat koeficienty pokud možno shodné s DCT, avšak pomocí celočíselné, výkonově podstatně méně náročné, transformace, viz kapitola 3.3.6 a [4].

Při porovnání celočíselné transformace založené na DCT a téže transformace doplněné užitím Hadamardovy transformace, je možné pozorovat kladný vliv Hadamardovy transformace na hodnotu PSNR. Tento fakt je možné přisoudit vlivu koncentrace významných DC koeficientů makrobloku do bloku 4x4 a jejich následné transformaci, která umožňuje uchovat energii významně popisující podobu makrobloku, viz kapitola 3.3.6.3 a [4], což má pozitivní vliv i na snímky, využívající Intra predikovaného snímku s Hadamardovou transformací, jako referenčního.

Tab. 5.6: Hodnoty PSNR dle typu transformace (sekvence Foreman, kvantování AVC, QP = 20, TSS, bloky predikce 16x16)

Hodnoty PSNR dle typu transformace	DCT 8x8	DCT 4x4	Celočíselná DCT 4x4	Celočíselná DCT 4x4 + Hadamard
I rekonstruovaný	44,2446 dB	44,2578 dB	44,2208 dB	44,5133 dB
P kompenzovaný	30,4274 dB	30,4422 dB	30,4395 dB	30,458 dB
P rekonstruovaný	44,0098 dB	43,8829 dB	43,8368 dB	43,8598 dB
B kompenzovaný	37,5238 dB	37,5491 dB	37,533 dB	37,5318 dB
B rekonstruovaný	44,2911 dB	44,2351 dB	44,2079 dB	44,2342 dB

ZÁVĚR

V této diplomové práci bylo mým cílem prostudovat možnosti komprese videosignálu a seznámit se s užívanými kodeky. Detailněji byly prostudovány a popsány techniky použité ve standardech MPEG. V kapitolách 1, 2 a 3 jsem se nejprve zabýval charakteristickými vlastnostmi videa a schopnostmi a citlivostí lidského zrakového systému (HVS). V návaznosti na zjištěné poznatky byly popsány důvody a principy odstranění nadbytečnosti (redundance) a zbytečnosti (irrelevance) v obrazovém signálu. Podrobně byly popsány jednotlivé, nejčastěji používané, metody ztrátového a bezztrátového kódování, které byly následně v praktické části i programově zpracovány. Dále byly v teoretické části uvedeny kompresní standardy a kodeky z rodiny MPEG a H.26x. Nebyla vynechána ani část věnovaná způsobům posuzování kvality obrazového signálu.

Při studiu standardů bylo pozorováno, že ačkoliv je mezi sebou dělí řada let vývoje, mnoho technik a principů zůstalo napříč těmito standardy shodných, nebo velmi podobných. Lze však říci, že ač jsou si jednotlivá bloková schémata kodérů a dekodérů velmi podobná, je přesto zřejmé, že k pokroku samozřejmě došlo, a to především uvnitř jednotlivých funkčních bloků. U standardů rodiny MPEG je to patrné zejména z definice profilů a úrovní, kde jsou vzájemné odlišnosti opravdu významné. Postupným vývojem tak došlo ke zvýšení efektivity komprese, avšak při zachování kvality obrazu. Proto je nutné aby byly schopnosti kodérů a dekodérů jednoznačně definovány, neboť jedině tak může být zajištěna kompatibilita zařízení různých výrobců a jednoznačný popis jejich schopností.

Hlavním jádrem této práce je prozkoumání a realizace funkcí, jenž simulují činnost funkčních bloků kodéru a dekodéru u standardů MPEG. Konkrétně se tedy jedná o metody ztrátového a bezztrátového kódování. Mezi ztrátové patří především vstupní podvzorkování snímků, lépe řečeno jejich barvonosných složek. Dále též kvantování, které je rozhodující operací při regulaci výstupního bitového toku kodéru a má rovněž největší vliv na kvalitu obrazového signálu.

Mezi bezztrátové lze zařadit transformace, především tedy DCT, která provádí transformaci vzorků z prostorové do frekvenční oblasti. Dále také pohybovou predikci, pomocí níž jsou nalezeny vektory pohybu v obrazové sekvenci. Samotné zpracování chyby predikce, neboli rozdílu skutečného a predikovaného snímku, však již ztrátové je, neboť podstupuje operaci kvantování.

Před vstupem do vysílací paměti dochází u reálného kodéru k entropickému kódování kvantovaného signálu i vektorů pohybu, během něž je vstupní informace vyjádřena menším počtem bitů, a to zejména použitím RLE a VLC. Vzhledem k tomu, že nedochází k ovlivnění výstupního signálu, nebylo programové zpracování tohoto bloku nezbytné.

V praktické části této práce, jenž je popsána v kapitole 4, byly vytvořeny v prostředí Matlab funkce, simulují činnost funkčních bloků kodéru a dekodéru. Tyto funkce byly implementovány do grafického uživatelského rozhraní. Na základě uživatelsky zadaných vstupních parametrů se provádí dané algoritmy a následně se zobrazí výstupy jednotlivých funkčních bloků z průběhu kódování i dekódování obrazu. Nezbytným vstupem funkcí, jimiž je kódování a dekódování prováděno, je rovněž obrazová sekvence snímků v podobě bitové mapy ve formátu RGB.

Činnost jednotlivých funkcí je zastřešena programovým souborem *kodek.m*, který rovněž obsahuje definici uživatelského rozhraní a který volá potřebné funkce a průběžně zobrazuje grafické i numerické výstupy, jenž byly uvedeny v kapitole 4, popisující praktickou realizaci. Všechny realizované funkce byly odladěny v prostředí Matlab, ve verzi 7.4.0.287 (R2007a) a jsou uloženy na přiloženém CD. Součástí přiloženého CD je také adresář obsahující použité testovací sekvence snímků.

Zhodnocení výstupů realizovaných funkcí, za použití uživatelsky nastavitelných parametrů kódování a dekódování, je detailně popsáno v kapitole 5. Tato kapitola obsahuje řadu tabulek a grafů, jenž poskytují čtenáři představu o složitosti a komplexnosti problematiky komprese obrazového signálu. Namátkou lze jmenovat zhodnocení vlivu podvzorkování a proměnlivosti obrazové scény, dále též vliv snímkové frekvence, kvantování, transformací a rozsahu oblasti hledání vektorů pohybu i jejich subpixelové přesnosti. Dále byl diskutován přínos i výpočetní náročnost Intra a Inter predikce i vliv množství reziduí na rekonstruovaný obrazový signál.

Závěrem, při zpětném hodnocení této práce, lze říci, že šíře možností a variability jednotlivých kodérů a dekodérů, a to včetně možností modifikací parametrů jednotlivých funkčních bloků daných zařízení je natolik široká, že by při detailnějším zpracování a zkoumání měla tato práce mnohem větší rozsah. V kapitolách 4 a 5 tak byly popsány především hlavní momenty při kódování a dekódování obrazového signálu a rovněž byly zhodnoceny zásadní vlivy jednotlivých technik a algoritmů na výstupní signál, ale i na výpočetní náročnost pro provádění těchto operací. Detailnější zkoumání této problematiky by bylo zcela jistě přínosné, neboť množství různých pohledů na tuto problematiku z jiných úhlů, nabízí řadu podnětů pro další zkoumání.

POUŽITÁ LITERATURA

- [1] PUŽMANOVÁ, R. *Moderní komunikační sítě A-Z*. Computer Press, Brno 2006. ISBN 80-251-1278-0
- [2] KUROSE, J., ROSS, K. *Computer Networking*. Addison-Wesley, New York 2009. ISBN 978-0-13-607967-5
- [3] ŠKORPIL, V. *Přístupové a transportní sítě*. Skriptum FEKT VUT, Brno 2004.
- [4] RICHARDSON, Iain E. G. *H.264 and MPEG-4 Video Compression*. John Wiley & Sons Ltd. 2003. ISBN 0-470-84837-5
- [5] HANZO, L., CHERRIMAN, P., STREIT, J. *Video Compression and Communications, 2nd edition*. John Wiley & Sons Ltd. 2007. ISBN 978-0-470-51849-6
- [6] ČSN EN ISO/IEC 368601: *Audiovizuální technika - Terminologie digitálního signálu*
- [7] ČSN EN ISO/IEC 11172: *Informační technologie – Kódování pohyblivých obrazů včetně doprovodného zvuku pro číslicový záznam do rychlosti 1,5 Mbit/s*
- [8] ČSN EN ISO/IEC 13818: *Informační technologie – Obecné kódování pohyblivých obrazů a doprovodné zvukové informace*
- [9] ČSN EN ISO/IEC 14496: *Informační technologie – Kódování audiovizuálních objektů*
- [10] VÍT, V. *Televizní technika – přenosové barevné soustavy*. BEN – technická literatura. 1997. ISBN 80-86056-04-X
- [11] ČÍKA, P. *Multimediální služby*. Skriptum FEKT VUT, Brno, 2007.
- [12] RAJMÍČ, P., SYSEL, P., SCHIMMEL, J. *Grafické a multimediální procesory*. Skriptum FEKT VUT, Brno, 2010.
- [13] ČÍKA, P. *Multimédia*. Přednášky k předmětu. FEKT VUT, Brno, 2007.
- [14] Úvod do techniky CCD čipů [online]. 2012 [cit. 2012-10-23]. Dostupné z WWW: <http://ccd.mii.cz/art?id=303&lang=405>
- [15] ZAPLATÍLEK, K., DOŇAR, B. *MATLAB pro začátečníky*. BEN – technická literatura. 2003. ISBN: 80-7300-095-4
- [16] ZAPLATÍLEK, K., DOŇAR, B. *MATLAB – tvorba uživatelských aplikací*. BEN – technická literatura. 2004. ISBN: 80-7300-133-0
- [17] Cisco Visual Networking Index: Forecast and Methodology, 2011-2016 [online], 2012 [cit. 2012-10-27]. Dostupné z WWW: http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360_ns827_Networking_Solutions_White_Paper.html
- [18] Morseova abeceda – Morseovka [online], 2012 [cit. 2012-10-27]. Dostupné z WWW: <http://morseovka.pitevna.cz/>
- [19] PRCHAL, J., ŠIMÁK, B., *Digitální zpracování signálů v telekomunikacích*. Vydavatelství ČVUT, 2001. ISBN 80-01-02149-1
- [20] NĚMEC, K. *Datová komunikace*. Skriptum FEKT VUT, Brno, 2007.
- [21] MPEG LA - The Standard for Standards [online]. 2012 [cit. 2012-10-27]. Dostupné z WWW: <http://www.mpegla.com>

- [22] ŠPAČEK, M. *Hierarchické kódování pohyblivých obrazů*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011
- [23] MathWorks - MATLAB and Simulink for Technical Computing [online]. 2012 [cit. 2012-11-05]. Dostupné z WWW: <http://www.mathworks.com>
- [24] BARJATYA, A. *Block Matching Algorithms for Motion Estimation* [online]. 2005. [cit. 2012-11-05]. Dostupné z WWW: <http://www.mathworks.com/matlabcentral/fileexchange/8761-block-matching-algorithms-for-motion-estimation>
- [25] CIPR Sequences [online]. 2002 [cit. 2012-11-15]. Dostupné z WWW: <http://www.cipr.rpi.edu/resource/sequences/>
- [26] GHANDI, M., GHANBARI, M., *The H.264/AVC Video Coding Standard for the Next Generation Multimedia Communication* [online]. 2004 [cit. 2013-02-23]. Dostupné z WWW: <http://mahdi.ghandi.ws/papers/iaee04.pdf>
- [27] CALAFATE, C. M. T., MALUMRES, M. P., *Evaluation of the H.264 codec* [online]. 2003 [cit. 2013-03-09]. Dostupné z WWW: http://atc.umh.es/gatcom/Ficheros/Articulos/H.264_int_report.pdf
- [28] RAJA, G., MIRZA, M. J., *In-loop Deblocking Filter for H.264/AVC Video* [online]. 2006 [cit. 2013-03-15]. Dostupné z WWW: <http://www.eurasip.org/Proceedings/Ext/ISCCSP2006/defevent/papers/cr1261.pdf>

SEZNAM ZKRATEK

AC	Alternating Current
ASP	Advanced Simple Profile
AVC	Advanced Video Coding
BAP	Binary Alpha Plane
CABAC	Context Adaptive Binary Arithmetic Coding
CAVLC	Context Adaptive Variable Length Coding
CCD	Charge Coupled Device
CCITT	Telegraph and Telephone Consultative Committee
CIF	Common Interchange Format
CODEC	Coder – Decoder
CPB	Constrained Parameters Bitstream
DC	Direct Current
DCT	Discrete Cosine Transform
DPCM	Differential Pulse-code Modulation
DFT	Discrete Fourier Transform
DWT	Discrete Wavelet Transform
FIR	Finite Impulse Response
FS	Full Search
HBV	High Bitrate Video
HDTV	High Definition Television
HVS	Human Visual System
IDCT	Inverse Discrete Cosine Transform
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
ITU	International Telecommunication Union
JPEG	Joint Photographic Experts Group
JVT	Joint Video Team
KLT	Karhunen L��ve Transform
LZW	Lempel-Ziv-Welch
MAE	Mean Absolute Error
MOS	Mean Opinion Score
MPEG	Motion Picture Experts Group
MSE	Mean Square Error
NAL	Network Abstraction Layer
NMSE	Normalized Mean Square Error
NNS	Nearest Neighbours Search
PAL	Phase-Alternation Line
PSNR	Peak Signal to Noise Ratio
QCIF	Quarter Common Interchange Format
QSIF	Quarter Source Input Format
RGB	Red, Green, Blue
RGBA	Red, Green, Blue, Alpha
RLE	Run-length Encoding
SAE	Sum of Absolute Errors
SIF	Source Input Format
SNR	Signal to Noise Ratio
TSS	Three Step Search

VCEG	Video Coding Experts Group
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 | Adresář s funkcemi pro realizaci kodeku | (\kodek) |
| A.2 | Tento dokument ve formátu PDF | (DP_Spacek_119338.pdf) |
| A.3 | Tento dokument ve formátu DOC | (DP_Spacek_119338.doc) |
| A.4 | Adresář s použitými obrazovými sekvencemi | (\sekvence) |