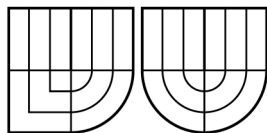


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A
KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV RADIOELEKTRONIKY



FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF RADIO ELECTRONICS



VÝUKOVÝ VIDEO KODEK

EDUCATIONAL VIDEO CODEC

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

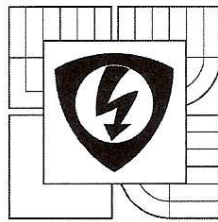
AUTOR PRÁCE
AUTHOR

Bc. Martin Dvořák

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. Martin Slanina, Ph.D.

BRNO, 2012



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav radioelektroniky

Diplomová práce

magisterský navazující studijní obor
Elektronika a sdělovací technika

Student: Bc. Martin Dvořák
Ročník: 2

ID: 106418
Akademický rok: 2011/12

NÁZEV TÉMATU:

Výukový video kodek

POKYNY PRO VYPRACOVÁNÍ:

Prostudujte základní principy komprimace obrazových signálů . Seznamte se s používanými technikami pro redukci zbytečnosti a nadbytečnosti v obrazovém signálu . Ve vybraném programovém prostředí (doporučen je Matlab) implementujte transformační kódování podle standardu MPEG -2. Vaši implementaci rozšiřte o další komprimační nástroje . Pokuste se implementovat i jednoduchou predikci a kompenzaci pohybu.

Proveďte analýzu kompresní účinnosti při použití různých komprimačních nástrojů a různých zdrojových videosekvencí. Zjistěte, které bloky nejvíce ovlivňují kvalitu dekomprimovaných videosekvencí.

DOPORUČENÁ LITERATURA:

[1] GONZALES, R. C., WOODS, R. E., EDDINS, S. L. Digital Image Processing using MATLAB. Upper Saddle River: Prentice Hall, 2004.

[2] RICHARDSON, I. E. G. Video Codec Design. Chichester: Wiley, 2002.

Termín zadání: 6.2.2012

Termín odevzdání: 18.5.2012

Vedoucí práce: Ing. Martin Slanina, Ph.D.

Konzultanti diplomové práce:

prof. Dr. Ing. Zbyněk Raida
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 realizovaný v rámci této diplomové práce byl finančně podpořen projektem
CZ.1.07/2.3.00/20.0007 **Wireless Communication Teams**
operačního programu **Vzdělávání pro konkurenceschopnost**.



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Finanční podpora byla poskytnuta Evropským sociálním fondem
a státním rozpočtem České republiky.

ABSTRAKT

Prvním cílem diplomové práce je prostudování základních principů komprimace obrazových signálů. Seznámení se s technikami používanými pro redukci zbytečnosti a nadbytečnosti v obrazovém signálu. Druhým cílem je, na základě těchto informací, realizovat jednotlivé komprimační nástroje v programovém prostředí Matlab a sestavit tak jednoduchý model video kodeku. Diplomová práce obsahuje popis realizace tří základních komprimačních bloků a sice - kódování uvnitř snímku, mezisnímkové kódování a kódování s proměnnou délkou slova - podle standardu MPEG-2.

KLÍČOVÁ SLOVA

Komprimační algoritmus, Video kodek, MPEG-2, Transformační kódování, Diskrétní kosinová transformace, DCT, Diskrétní vlnková transformace, DWT, Kvantování, Kvantizační parametr, Předpověď, Vektor pohybu, Skupina snímků, GOP, Entropické kódování, Huffmanovo kódování, Kódování délky běhu.

ABSTRACT

The first goal of diploma thesis is to study the basic principles of video signal compression. Introduction to techniques used to reduce irrelevancy and redundancy in the video signal. The second goal is, on the basis of information about compression tools, implement the individual compression tools in the programming environment of Matlab and assemble simple model of the video codec. Diploma thesis contains a description of the three basic blocks, namely - interframe coding, intraframe coding and coding with variable length word - according the standard MPEG-2.

KEYWORDS

Compression algorithm, Video codec, MPEG-2, Transform coding, Discrete cosine transform, DCT, Discrete wavelet transform, DWT, Quantization, Quantization parameter, Prediction, Motion vector, Group of images, GOP, Entropic coding, Huffman's coding, Run length Coding.

DVOŘÁK, M. *Výukový video kodek*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Ústav radioelektroniky, 2012. 65 s., 6 s příloh. Diplomová práce. Vedoucí práce: Ing. Martin Slanina, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svoji diplomovou práci na téma Výukový video kodek 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 práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne 18. května 2012

.....
podpis autora

PODĚKOVÁNÍ

Děkuji vedoucímu diplomové práce Ing. Martinu Slaninovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne 18. května 2012

.....
podpis autora

OBSAH

SEZNAM OBRÁZKŮ	viii
SEZNAM TABULEK	x
ÚVOD	1
1 ZÁKLADNÍ PRINCIPY KOMPRIMACE	2
<i>1.1 Blokové zapojení kodéru a dekodéru</i>	2
<i>1.2 Kódování uvnitř snímku</i>	5
1.2.1 Transformační kódování	5
1.2.1.1 Diskrétní kosinová transformace.....	6
1.2.1.2 Diskrétní vlnková transformace	7
1.2.2 Kvantování	9
<i>1.3 Mezišnímkové kódování</i>	10
1.3.1 Druhy snímků a jejich předpověď	10
1.3.2 Diferenciální pulsní kódová modulace.....	11
1.3.2.1 Vstupní signál DPCM	11
1.3.2.2 DPCM DC koeficientů	11
1.3.3 Odhad a kompenzace pohybu	12
1.3.3.1 Plné vyhledávání	12
1.3.3.2 N-krokové vyhledávání	13
1.3.3.3 Logaritmičké vyhledávání.....	14
1.3.4 Mezipixelový odhad a kompenzace pohybu	15
1.3.5 Skupina snímků	15
<i>1.4 Kódování s proměnnou délkou slova</i>	16
1.4.1 Cik-cak vyčítání	16
1.4.2 Kódování délky běhu	16
1.4.3 Entropické kódování	17
1.4.3.1 Huffmanovo kódování.....	17
2 METRIKY PRO MĚŘENÍ KVALITY	18
3 REALIZACE VIDEO KODEKU	20
<i>3.1 Kódování uvnitř snímku</i>	20
3.1.1 Zpracování vstupních dat	20
3.1.2 Diskrétní kosinová transformace.....	21
3.1.3 Diskrétní vlnková transformace	21
3.1.4 Kvantování a prahování	21
3.1.5 Dosažené výsledky I.....	22
3.1.5.1 Transformace a kvantování	22
3.1.5.2 Zpracování testovacího bloku	24
<i>3.2 Mezišnímkové kódování</i>	26
3.2.1 Plné vyhledávání	26
3.2.2 N-krokové vyhledávání	27
3.2.3 Logaritmičké vyhledávání.....	27
3.2.4 Vyhledávání s přesností ½ pixelu	28
3.2.5 Dosažené výsledky II	29
3.2.5.1 Vyhledávání s přesností celého pixelu	29

3.2.5.2 Vyhledávání s přesností ½ pixelu	33
3.3 Kódování s proměnnou délkou slova	35
3.3.1 Kódování délky běhu	35
3.3.2 Huffmanovo kódování.....	36
3.3.2.1 Kódování DC koeficientů.....	36
3.3.2.2 Kódování AC/DCT koeficientů	36
3.3.3 Dosažené výsledky III.....	37
3.3.3.1 Kódování délky běhu	37
3.3.3.2 Huffmanovo kódování.....	38
3.3.3.3 Kódování s proměnnou délkou slova	41
3.4 Výkonnost kodeku.....	43
3.4.1 Výkonnost nástrojů odhadu a kompenzace pohybu	44
3.4.2 Srovnání výkonnosti se standardem MPEG-2	48
3.5 Uživatelské rozhraní.....	49
3.5.1 Kodér	49
3.5.2 Dekodér	50
3.5.3 Výstupní informace	50
4 ZÁVĚR.....	52
SEZNAM LITERATURY	53
SEZNAM PŘÍLOH.....	55

SEZNAM OBRÁZKŮ

Obr. 1. Blokové zapojení kodéru standardu MPEG-2 [1].....	3
Obr. 2. Blokové zapojení dekodéru standardu MPEG-2 [1].....	4
Obr. 3. DCT bloku vzorků s malými rozdíly [1].....	6
Obr. 4. DCT bloku vzorků s velkými rozdíly [1].....	7
Obr. 5. Rozklad vstupního signálu do frekvenčních pásem [2].	8
Obr. 6. První stupeň diskrétní vlnkové transformace [2].	8
Obr. 7. Předpověď snímku typu P : dopředná (vlevo) a zpětná (vpravo).	10
Obr. 8. Obousměrná předpověď snímku typu B.	10
Obr. 9. Varianty vzorkování obrazového signálu YUV [2].	11
Obr. 10. DPCM DC koeficientů.....	11
Obr. 11. Princip odhadu pohybu.	12
Obr. 12. Vyhledávání rastrové (vlevo) a spirálové (vpravo) [2].	13
Obr. 13. Princip tříkrokového vyhledávání [2].	14
Obr. 14. Princip logaritmického vyhledávání [2].	14
Obr. 15. Bilineární interpolace hodnot vzorků v poloze $\frac{1}{2}$ pixelu.	15
Obr. 16. Skupina snímků standardu MPEG [3].	15
Obr. 17. Kódování s proměnnou délkou slova [9].	16
Obr. 18. Cik-cak vyčítání kvantovaných koeficientů [9].	16
Obr. 19. Systém měření strukturální podobnosti [19].	19
Obr. 20. Blokové schéma kódování uvnitř snímku – zpracování 1 snímku.....	20
Obr. 21. Originální snímek a testovací blok 8x8.....	22
Obr. 22. Kvantizační parametr 4 : DCT (vlevo) a DWT (vpravo).....	23
Obr. 23. Kvantizační parametr 8 : DCT (vlevo) a DWT (vpravo).....	23
Obr. 24. Kvantizační parametr 16 : DCT (vlevo) a DWT (vpravo).....	23
Obr. 25. Kvantizační parametr 31 : DCT (vlevo) a DWT (vpravo).....	23
Obr. 26. Závislost SSIM a PSNR na parametru bpp pro jasovou složku snímku.....	24
Obr. 27. Rozložení hodnot a energie signálu jasové složky testovacího bloku.	24
Obr. 28. Rozložení energie signálu po transformaci: DCT (vlevo) a DWT (vpravo).....	25
Obr. 29. Rozložení energie signálu po kvantování : DCT (vlevo) a DWT (vpravo).	25
Obr. 30. Rozložení hodnot chybového bloku : DCT (vlevo) a DWT (vpravo).	25
Obr. 31. Blokové schéma mezisnímkového kódování.	26
Obr. 32. Interpolace a zpracování vybrané oblasti.....	28
Obr. 33. Princip vyhledávání s přesností $\frac{1}{2}$ pixelu.....	28
Obr. 34. Testovací snímky ze sekvence Foreman : 66 (vlevo) a 69 (vpravo).....	29
Obr. 35. Rozdílový snímek – vlevo bez kompenzace, vpravo s kompenzací.	29
Obr. 36. Pole vektorů (vlevo) a kompenzovaný snímek 66 (vpravo).	30
Obr. 37. Závislost SAE na velikosti vyhledávacího okna.....	31
Obr. 38. Závislost doby zpracování na velikosti vyhledávacího okna.....	31
Obr. 39. Závislost sumy pohybových vektorů na velikosti vyhledávacího okna.....	32
Obr. 40. Mapa SAE pro předpověď makrobloku 189 snímku 69.....	32
Obr. 41. Závislost SAE na velikosti vyhledávacího okna.....	33
Obr. 42. Závislost doby zpracování na velikosti vyhledávacího okna.....	34
Obr. 43. Závislost sumy pohybových vektorů na velikosti vyhledávacího okna.....	34
Obr. 44. Blokové schéma kódování s proměnnou délkou slova.	35
Obr. 45. RLE - Redukce toku AC/DCT koeficientů.....	37
Obr. 46. RLE - Závislost redukce toku AC/DCT koeficientů na Q_p .	38
Obr. 47. Huffmanovo kódování - Redukce toku AC/DCT koeficientů.	39
Obr. 48. Huffmanovo kódování - Redukce toku DC koeficientů a vektorů pohybu.	40
Obr. 49. Huffmanovo kódování - Závislost redukce toku DC koef. snímku I na Q_p .	40
Obr. 50. Huffmanovo kódování - Závislost redukce toků AC/DCT koeficientů na Q_p .	41

Obr. 51. VLC - Závislost redukce toku dat snímku I/P na Q_p	42
Obr. 52. VLC - Redukce toku dat pro 50 snímků komprimované sekvence.	42
Obr. 53. Sekvence Foreman - identický kvantizační parametr.	45
Obr. 54. Sekvence Foreman - rozdílný kvantizační parametr.....	45
Obr. 55. Sekvence Mobile.....	46
Obr. 56. Sekvence News.	46
Obr. 57. Sekvence Container.....	47
Obr. 58. Sekvence Salesman.	47
Obr. 59. Srovnání implementace Matlab a standardu MPEG-2.....	48
Obr. 60. Náhled uživatelského rozhraní.....	49
Obr. 61. Náhled přehrávání dekomprimované sekvence.	51

SEZNAM TABULEK

Tab. 1. Přehled hlavních nástrojů kodéru standardu MPEG-2 [1].	2
Tab. 2. Huffmanova zdrojová redukce [10].	17
Tab. 3. Vytvoření Huffmanova kódu [10].	17
Tab. 4. Přehled velikostí vyhledávacího okna pro N-krokové vyhledávání.	27
Tab. 5. Přehled velikostí vyhledávacího okna pro logaritmické vyhledávání.	27
Tab. 6. Srovnání výkonnosti vyhledávacích algoritmů.	30
Tab. 7. Srovnání výkonnosti plného vyhledávání pro různou přesnost vyhledávání.	33
Tab. 8. Informace o snímcích komprimované videosekvence.	37
Tab. 9. Testovací videosekvence.	43
Tab. 10. Nastavení odhadu pohybu.	50

ÚVOD

Video kodek (**kodér** a **dekodér**), respektive algoritmus pro komprimaci a dekomprimaci obrazového signálu slouží ke snížení datového toku obrazového signálu, tím i velikosti komprimovaného videozáznamu a dále k dekódování (přehrání) videozáznamu na straně uživatele při co nejmenším viditelném zhoršení kvality obrazu.

Obrazové signály se komprimují z důvodu ušetření množství paměti potřebné k uchování těchto dat (např. záznamová média DVD, Blu-Ray, HD-DVD) a šířky pásma potřebné při jejich přenosu (např. digitální televizní přenos). Ke komprimaci obrazového signálu dochází při redukci redundantních (nadbytečných) a irelevantních (zbytečných) dat obrazového signálu.

Náplní semestrálního projektu je prostudovat základní principy používané pro komprimaci obrazových signálů. Kapitola 1 se věnuje popisu těchto principů, jedná se o kódování uvnitř snímku, což zahrnuje transformační kódování - vlnková či kosinová transformace a kvantování. Následuje popis mezisnímkového kódování - využití diferenciální pulsní kódové modulace pro přenos rozdílových snímků, zavedení odhadu a kompenzace pohybu pro zvýšení redukce dat rozdílových snímků. Popis používaných snímků ve standardu MPEG-2 a analýza několika vybraných vyhledávacích algoritmů. Poslední část této kapitoly se věnuje kódování s proměnnou délkou slova (cik-cak vyčítání frekvenčních koeficientu, kódování délky běhu a Huffmanovo kódování).

Kapitola 2 obsahuje popis metrik pro měření kvality - PSNR a SSIM index. Na jejich základě je porovnávána účinnost, jak jednotlivých komprimačních nástrojů, tak kodeku jako celku.

Kapitola 3 se věnuje samotné realizaci video kodeku, obsahuje stručný popis koster programu jednotlivých komprimačních nástrojů. Součástí jsou i výsledky simulací těchto nástrojů. U kódování uvnitř snímku se jedná o srovnání účinnosti diskrétní kosinové a vlnkové transformace a o detailní pohled na průběh transformace s kvantizací pro testovací blok. U mezisnímkové kódování zahrnují výsledky srovnání účinnosti jednotlivých vyhledávacích algoritmů a také srovnání vyhledávání s přesností poloviny a celého pixelu. Další část v této kapitole prezentuje účinnost, jak jednotlivých bloků kódování s proměnnou délkou slova, tak i kódování jako celku. Předposlední část obsahuje výsledky simulací kodeku na několika vybraných testovacích videosekvencích, zhodnocení výkonnosti jednotlivých nástrojů jako takových a i jejich chování pro rozdílnou dynamiku obrazu. V poslední části je pak uveden stručný popis grafického rozhraní a dostupných informací z kodeku.

1 ZÁKLADNÍ PRINCIPY KOMPRIMACE

Princip komprimace obrazového signálu spočívá v redukci redundantních (nadbytečných) a irelevantních (zbytečných) dat obrazového signálu. Redundantní data obrazového signálu jsou taková data, která uměle navyšují velikost, jejich potlačení či odstranění je tedy úprava bezztrátová. Irelevantní data jsou naopak data přímo spojená s obsahem obrazového signálu, ale díky nedokonalosti lidského vnímání je možné tyto data odstranit bez většího dopadu na výslednou kvalitu obrazového signálu, respektive kvality celého videozáznamu. Redukce irelevance je tedy úprava ztrátová.

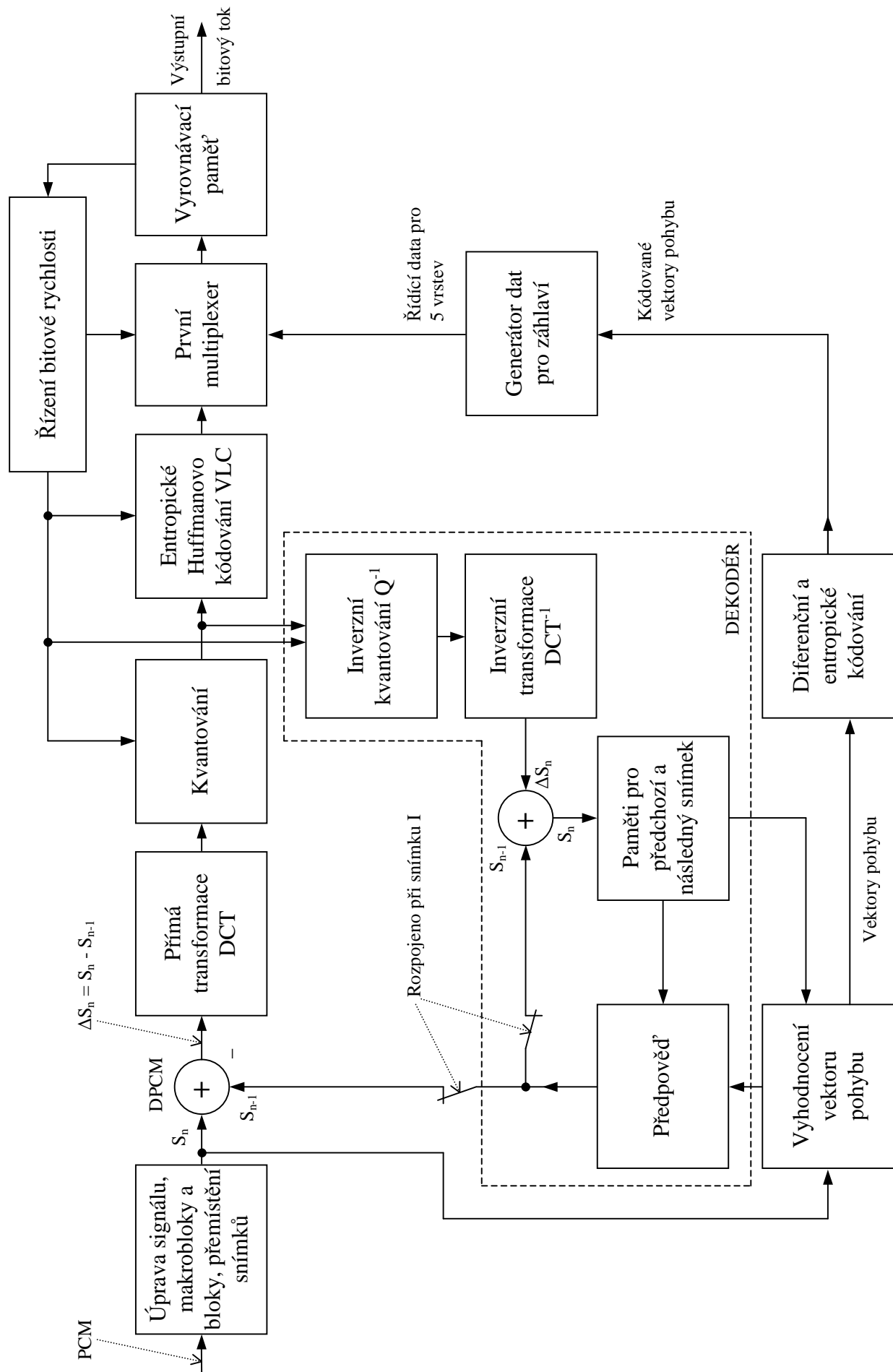
Existuje velká řada kodeků, které využívají celou škálu komprimačních nástrojů a různých přístupů k samotné komprimaci obrazového signálu (např. MPEG-4 Part 2: Visual vs. MPEG-4 Part 10: H.264/AVC nebo VC-1/SMPTE 421M). Nejrozšířenějším video kodekem je doposud MPEG-2 a popis principů komprimace bude tedy zahrnovat především komprimační nástroje tohoto standardu [6], [7].

1.1 Blokové zapojení kodéru a dekodéru

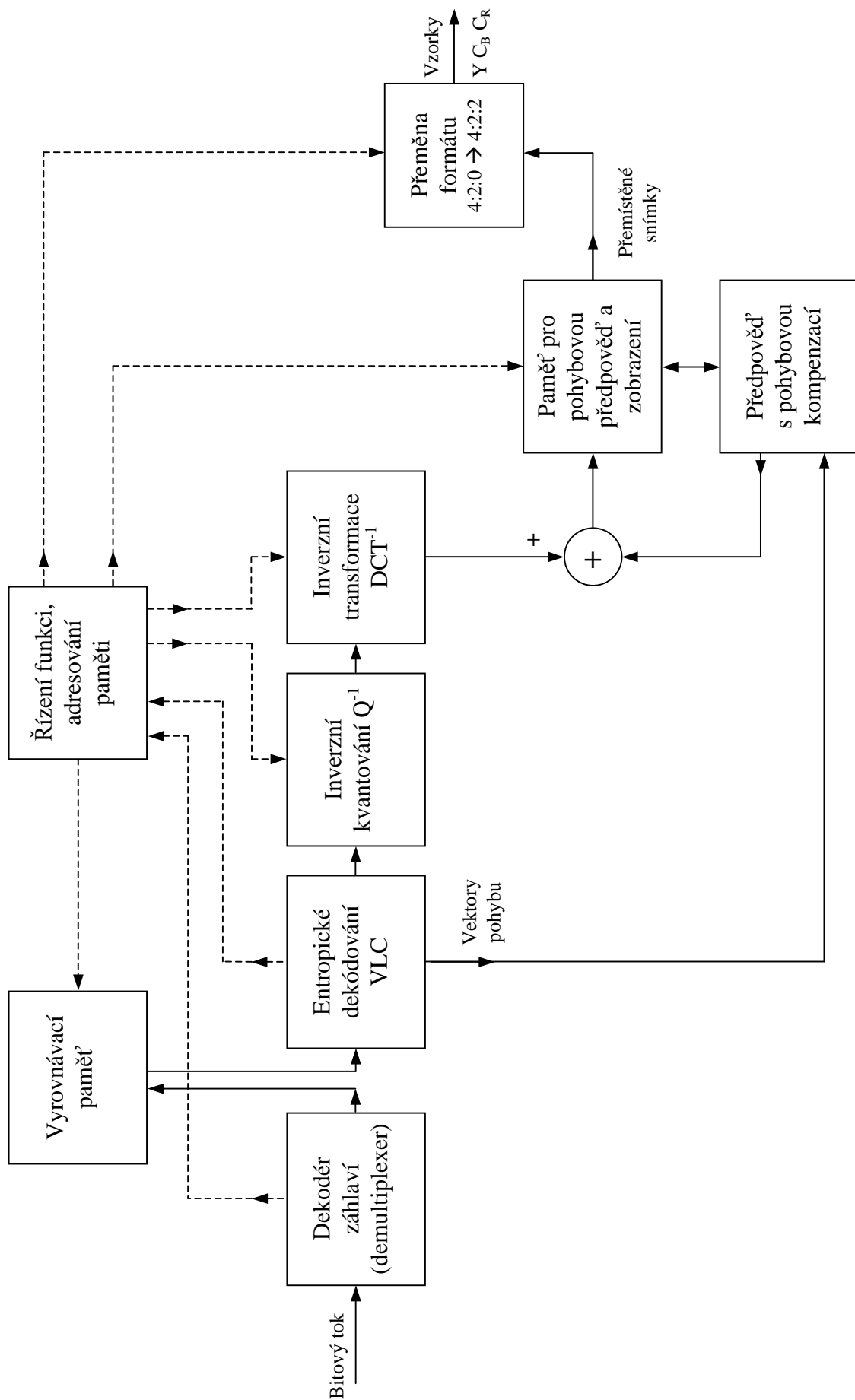
Pro návrh modelu video kodeku a jeho následnou realizaci se bude vycházet ze standardu MPEG-2 (ISO/IEC 13818), viz tab. 1. Zapojení kodéru standardu MPEG-2, viz obr. 1, lze rozdělit na dvě úrovně kódování a to: kódování uvnitř snímku (Intra-frame), kódování mezisnímkové (Inter-frame) a na úroveň zajišťující zpracování dat výstupní bitového toku [1].

Kódování uvnitř snímku je tvořeno bloky:	
Přímá transformace DCT (Diskrétní kosinová trans.)	Transformace obrazových vzorků v prostorového oblasti na frekvenční koeficienty ve frekvenční oblasti.
Kvantování	Dělení matice frekvenčních koeficientů kvantizační maticí.
Mezisnímkové kódování je tvořeno bloky:	
DPCM (Diferenciální pulsní kódová modulace)	Vytváří rozdíl mezi aktuálním a předchozím (následným) snímkem. Při snímku typu I rozdíl nevytváří.
Inverzí kvantování, inverzní transformace	Rekonstrukce rozdílového snímku, zpětné násobení kvantované matice a zpětná transformace DCT ⁻¹ .
Paměti pro předchozí a následný snímek	Do paměti se ukládá rekonstruovaný snímek a slouží jako reference pro předpověď.
Předpověď	Vytvoření předpovědi z vektorů pohybu a předešlého/následného snímku.
Vyhodnocení vektoru pohybu	Porovnávání aktuálního snímku s předchozím (následným), hledání stejných makrobloků, na základě jejich vzájemné pozice se vytváří vektor pohybu.
Bloky pro zpracování dat výstupního bitového toku:	
Entropické kódování (Kódování délky běhu, Huffmanovo kódování)	„cik cak“ čtení frekvenčních koeficientů a následné kódování.

Tab. 1. Přehled hlavních nástrojů kodéru standardu MPEG-2 [1].



Obr. 1. Blokové zapojení kodéru standardu MPEG-2 [1].



Obr. 2. Blokové zapojení dekodéru standardu MPEG-2 [1].

Na obr. 2 je uvedeno zapojení dekodéru standardu MPEG-2. Demultiplexer slouží k rozdělení dat záhlaví a obrazových dat, detekuje bity záhlaví v každém příchozím snímku. Tyto řídicí bity nastavují požadovanou kapacitu vyrovnávací paměti pro snímek a také přepínají paměť do stavu naplnění, od něhož začíná čtení a naplňování. Demultiplexer také předává základní informace bloku pro řízení funkcí, adresování paměti dekodéru [1].

Z vyrovnávací paměti se zakódovaná obrazová data posílají do dekodéru, kde probíhá entropické dekódování VLC. Data obsahující kódovací parametry (kvantizační matice, typ makrobloku) jsou předány bloku pro řízení funkcí. Užitečná obrazová data se po inverzním „cik cak“ čtení frekvenčních koeficientů doplní nulami a seřadí se do blokového uspořádání [1].

Takto seřazená data se inverzně kvantují příslušnou kvantizační maticí, následuje inverzní transformace DCT^{-1} . Transformovaná data se dostávají do dekodéru diferencíální pulsní kódové modulace [1].

První dekódovaný snímek je typu I, je uložen do paměti pro pohybovou předpověď a je použit jako referenční snímek pro dekódování snímků typu P. Ty jsou následně také uloženy do paměti a spolu se I slouží pro dekódování snímku typu B. Čtení při pohybové kompenzaci probíhá po makroblocích, adresy makrobloků se vypočítávají z dekódovaných vektorů pohybu. Paměť pro referenční snímky se využívá pro přemístění snímků do původního pořadí, které bylo na kódovací straně přemístěno pro účely předpovědi [1].

1.2 Kódování uvnitř snímku

Komprimace a dekomprimace obrazového signálu v rámci jednoho snímku je na straně kodéru realizována pomocí transformačního kódování (např. diskrétní kosinová transformace, diskrétní vlnková transformace), kvantování (lineární, nelineární, dynamické) a prahování kvantovaných koeficientů. Dekodér pak obsahuje blok pro inverzní transformaci a inverzní kvantování. Tyto bloky obsahuje i kodér, jejich činnost ale spočívá v rekonstrukci snímku pro další použití v rámci mezisnímkového kódování.

1.2.1 Transformační kódování

Transformační kódování je pomyslným srdcem většiny video kodeků a standardů. Jedná se o transformaci prostorově vyjádřených obrazových dat do jiné reprezentace. Existují dobré důvody pro transformaci obrazových dat tímto způsobem. Sousední vzorky prostorově vyjádřených obrazových dat jsou silně korelované (vzájemně se ovlivňují) a energie tak má tendenci být rovnoměrně rozložena po celém obrazu. V takovém případě je obtížné odstranit nebo potlačit data bez viditelné ztráty kvality obrazu [2].

Při vhodné volbě transformace je komprimace dat v transformované reprezentaci mnohem jednodušší. Po transformaci je energie koncentrována do malého množství významných hodnot, data už tedy nejsou korelována a odstranění či potlačení nadbytečných a zbytečných dat má minimální vliv na kvalitu obrazu. Dvě nejrozšířenější transformace užívané pro komprimaci obrazu jsou diskrétní kosinová transformace (DCT) a diskrétní vlnková transformace (DWT) [2].

1.2.1.1 Diskrétní kosinová transformace

Diskrétní kosinová transformace je obvykle aplikovaná na malé, pravidelné bloky vzorků (např. 8x8). DCT se ukázala jako obzvlášť odolná a je jádrem většiny současných komprimačních standardů pro obraz a video - JPEG, H.261, H.263, H.263+, MPEG-1, MPEG-2 a MPEG-4. Princip spočívá v transformaci prostorového rozložení hodnot vzorků obrazového signálu na spektrum jeho frekvenčních složek s příslušnými amplitudami, přičemž nulové hodnoty frekvenčních koeficientů a hodnoty blízké nule nepřenášíme. Tím dojde, spolu s použitím kvantizace, ke zmenšení velikosti přenášených dat, respektive výsledné přenosové rychlosti, s využitím redukce redundance a irelevance v signálu. [1], [3].

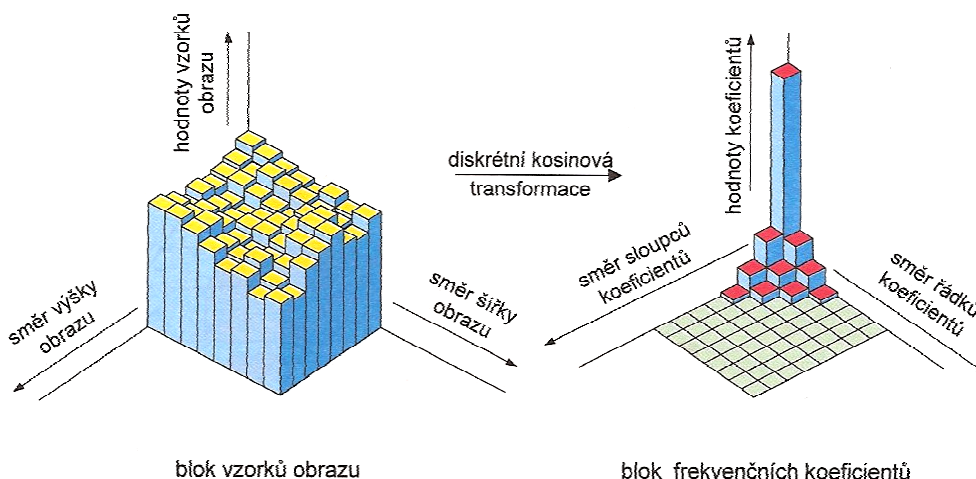
Koeficienty DCT, tj. funkce $G(u,v)$ se počítají pro blok $N=8$ (matice 8x8 prvků) z funkce $g(x,y)$ podle rovnice (1)

$$G(u,v) = \frac{1}{4} C(u)C(v) \sum_{x=0}^7 \sum_{y=0}^7 g(x,y) \cos\left(\frac{(2x+1)u\pi}{2N}\right) \cos\left(\frac{(2y+1)v\pi}{2N}\right), \quad (1)$$

kde u,v jsou souřadnice ve frekvenční oblasti,
 x,y jsou souřadnice v prostorové (časové) oblasti,

konstanty $C(u) = C(v) = 1/\sqrt{2}$ pro $u = 0, v = 0$ a
 $C(u) = C(v) = 1$ pro $u > 0, v > 0$.

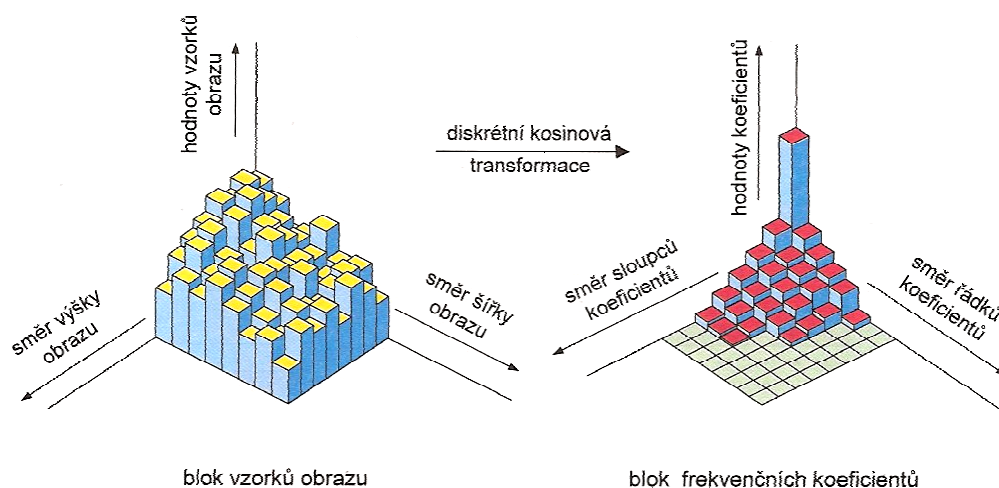
Koeficient $G(0,0)$, označovaný také jako DC koeficient, představuje stejnosměrnou hodnotu, tedy střední hodnotu celého transformovaného signálu. V této hodnotě je soustředěna největší energie obrazového signálu, neboť její velikost převyšuje amplitudy ostatních střídavých složek – AC koeficientů. Hodnoty frekvenčních AC koeficientů klesají, s postupně rostoucí frekvencí, od levého rohu doprava a směrem dolů. Čím méně vysokofrekvenčních složek se v obrazu vyskytuje, tím více nul nebo hodnot blízkých nule je ve výsledku. Každý z koeficientů $G(u,v)$ se počítá ze všech 64 vzorků jednoho bloku [1], [3].



Obr. 3. DCT bloku vzorků s malými rozdíly [1].

Při transformaci matice vzorků obrazového signálu na matici frekvenčních koeficientů se jedná o dvourozměrnou transformaci 2D DCT. Matice vzorků se, z důvodu technické náročnosti zpracování, rozděluje na bloky 8x8 prvků. Nedosáhne se tak maximální vytíženosti transformace, kdy celkový počet vynechaných frekvenčních AC koeficientů, které jsou nulové nebo blízké nulové hodnotě, není tak veliký jako u transformace přes celý snímek.

Tím dochází i ke zmenšení stupně komprimaci. Zpracování transformace přes celý snímek bylo časově velmi náročné [1], [3].



Obr. 4. DCT bloku vzorků s velkými rozdíly [1].

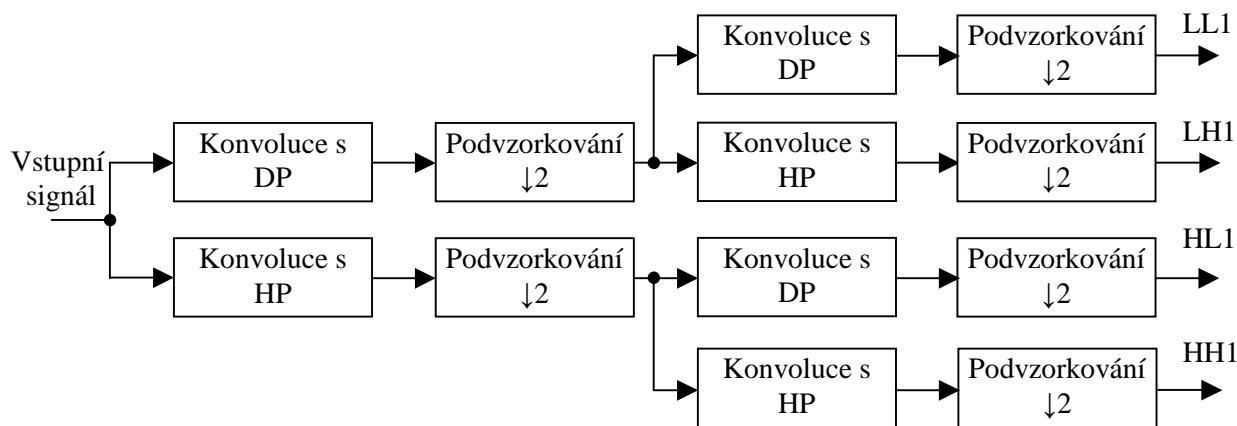
U jednolitých ploch, kde se nevyskytují vyšší frekvence, je tento způsob vhodný (nejvyšší koeficienty budou nulové). Obecně platí, že čím menší rozdíly v hodnotách vzorků, tím méně početná bude skupina frekvenčních AC koeficientů, viz obr. 3. V nejjednodušším případě, kdy jsou všechny vstupní hodnoty stejné, dostaneme po DCT pouze jeden koeficient – DC koeficient, obsahující veškerou energii zpracovávaného bloku. Naopak, obraz s prudkými změnami v hodnotách vzorků vytváří početnou skupinu frekvenčních AC koeficientů, viz obr. 4 [1], [3].

Diskrétní kosinová transformace (DCT) je úprava bezeztrátová, neboť jde pouze o převod vzorků z prostorové oblasti do oblasti frekvenční. Chyby vznikají následným zaokrouhlováním kvantovaných frekvenčních koeficientů, po inverzní transformaci (iDCT) tedy nedostaneme původní hodnoty vzorků obrazového signálu. Tím dojde ke zvýšení komprese, ale za cenu větší degradace kvality obrazu [1], [3].

1.2.1.2 Diskrétní vlnková transformace

Diskrétní vlnková transformace, stejně jako DCT, využívá skutečnosti, že aplikací transformace získáme rozložení frekvenčních koeficientů výhodné pro další zpracování. Jedním z klíčových rozdílů mezi uvedenými transformacemi je velikost oblasti, která je transformací zpracovávána. Zatímco DCT je aplikována na bloky o velikosti 8x8 vzorků, DWT je primárně určena pro zpracování celého snímku nebo větších oblastí čtvercového tvaru, neboť pro větší oblast poskytuje lepší kompresi než DCT. DWT používá například standard JPEG2000, MJPEG nebo video kodek Dirac [2], [13].

Základním principem dvourozměrné DWT obrazového signálu je rozklad vstupního signálu do frekvenčních pásem pomocí filtrů, viz obr. 5. Základní filtry jsou dolní a horní propust. Tvoří pár tzv. kvadraturních zrcadlových filtrů, které mají komplementární propustná pásma. Výstupy obou filtrů jsou podvzorkovány na polovinu vstupních vzorků. Horní propust poskytuje koeficienty tzv. detailů DWT, dolní propust koeficienty tzv. aproximace DWT. Díky podvzorkování je celkový počet koeficientů po jednom kroku filtrace stejný jako počet vstupních vzorků [2], [12].



Obr. 5. Rozklad vstupního signálu do frekvenčních pásem [2].

Obrazový signál reprezentující jednotlivé frekvenční pásma rozloženého vstupního signálu, viz obr. 6, obsahuje blok LL1, což je originální snímek filtrovaný dolní propustí a nese nejvíce energie. V pravém horním rohu se nachází blok HL1, který obsahuje vyfiltrované vertikální frekvence. V levém dolním rohu je blok LH1 obsahující horizontální frekvence. A nakonec v pravém dolním rohu blok HH1 prezentující zbytkové diagonální frekvence. Všechny bloky jsou podvzorkovány na polovinu jak ve vertikálním, tak horizontálním směru. Uvedená číslice 1 označuje stupeň DWT [2], [12].



Obr. 6. První stupeň diskretní vlnkové transformace [2].

Koeficienty aproximace lze dále analyzovat shodným rozkladem a obdržet tak další soubor koeficientů aproximace a detailů. Tak lze postupovat až do vyčerpání vstupní sekvence [12].

Ve standardech pro zpracování videa ovšem není diskretní vlnková transformace příliš rozšířená, jelikož ji nelze snadno rozšířit do časové oblasti. Proto i zmiňovaný standard Dirac umožňuje pouze kódování uvnitř snímku. Dalším rozdílem oproti nejrozšířenějším standardům je nutnost použití, pro korektní funkčnost, tzv. EZW (Embedded Zerotree Wavelet) kodéru a aritmetického kódování oproti kódování VLC u MPEG standardu. Tato implementace video kodeku je založena na standardu MPEG-2, proto bude pro zpracování transformovaných koeficientů využito informací (kvantizační matice, vyčítání koeficientů) ze zdroje [14] a dále bude aplikováno standardní kódování podle specifikace MPEG-2 [2].

1.2.2 Kvantování

Úkolem kvantování je zmenšení objemu dat a tím i snížení bitové rychlosti potřebné pro přenos či uchování obrazového signálu. Ke snížení dochází samotným kvantováním a také následným vyřazením nulových a nule blízkých frekvenčních koeficientů z přenosu. Kvantování je oproti transformaci úprava ztrátová a využívá nedokonalosti lidského zraku. Při dodržení povolené velikosti ztrát nevyhodnotí lidský zrak ztrátu jako škodlivou [1].

Kvantování AC střídavých frekvenčních koeficientů, u non-intra kódovaných bloků včetně stejnosměrných DC koeficientů, je definováno následujícím vztahem (2)

$$QDCT(i,j) = \text{round}\left(\frac{16 \cdot DCT(i,j)}{QM(i,j) \cdot Qp} + \frac{k}{2}\right), \quad (2)$$

kde i, j jsou souřadnice v bloku 8×8 ,
 $QDCT(i, j)$ je matice kvantovaných koeficientů,
 $DCT(i, j)$ je matice frekvenčních koeficientů,
 $QM(i, j)$ je kvantizační matice,
 Qp je kvantizační parametr,
 round je zaokrouhlení na celá čísla a
 identifikátor $k = 0$ pro intra bloky a
 $k = 1 \mid 0 \mid -1$ pro $DCT(i, j) > \mid < 0$ pro non-intra bloky.

Kvantování DC stejnosměrných koeficientů intra bloků se provádí vydělením hodnotou 8 (8. bitová přesnost) a zaokrouhlením. Po uvedené úpravě frekvenčních koeficientů je možné rozšířit tzv. mrtvou zónu kolem nuly za účelem vynulování maximálního počtu frekvenčních koeficientů. Protože malé odchylky kolem nuly jsou způsobovány také šumem, jejich potlačení obvykle zlepšuje subjektivní kvalitu obrazu. Tato úprava se označuje jako prahování [1], [4].

Standard MPEG-2 obsahuje celkem 4 typy kvantizačních matic, v podstatě může ale kodér vytvořit libovolnou kvantizační matici a spolu se signálem ji přenést do dekodéru. V případě vzorkování vstupního signálu 4:2:0 jsou použity 2 kvantizační matice. Kvantizační matice pro kvantování koeficientů snímku I, viz rovnice 3, a matice určená pro kvantování koeficientů snímků P a B, označovaná jako Flat16, obsahující pouze hodnoty 16. U vzorkování 4:4:4 a 4:2:2 jsou použity navíc 2 matice, pro jasovou a barevné složky [11].

Čísla kvantizační matice se obvykle zvětšují směrem k vyšším prostorovým kmitočtům v souladu se skutečností, že lidské oko je na ně méně citlivé a mohou tedy být kvantovány hruběji. Velikost hodnot kvantizační matice a hodnota kvantizačního parametru ovlivňujeme jak bitovou rychlost datového toku, tak i výslednou kvalitu dekomprimovaného obrazového signálu [1].

$$q_{\text{intra}} = \begin{bmatrix} 8 & 17 & 18 & 19 & 21 & 23 & 25 & 27 \\ 17 & 18 & 19 & 21 & 23 & 25 & 27 & 28 \\ 20 & 21 & 22 & 23 & 24 & 26 & 28 & 30 \\ 21 & 22 & 23 & 24 & 26 & 28 & 30 & 32 \\ 22 & 23 & 24 & 26 & 28 & 30 & 32 & 35 \\ 23 & 24 & 26 & 28 & 30 & 32 & 35 & 38 \\ 25 & 26 & 28 & 30 & 32 & 35 & 38 & 41 \\ 27 & 28 & 30 & 32 & 35 & 38 & 41 & 45 \end{bmatrix} \quad (3)$$

1.3 Mezisnímkové kódování

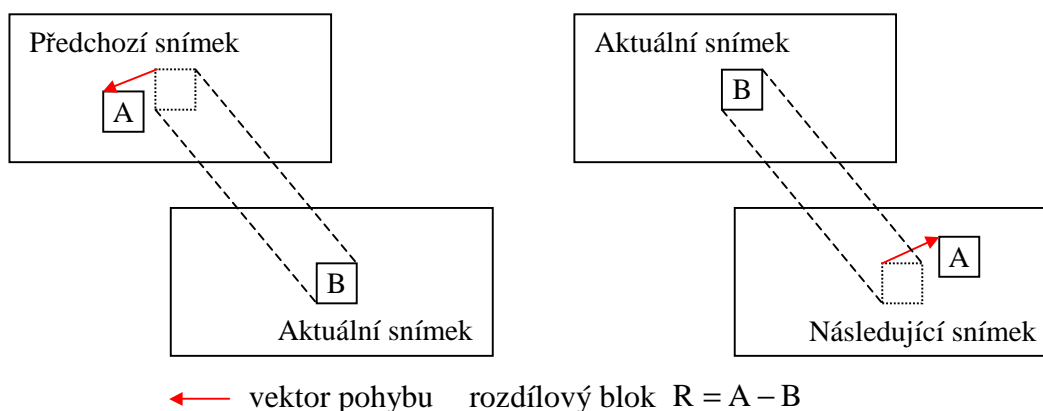
Je založeno na skutečnosti, že následující snímky jsou více či méně podobné snímkům přecházejícím. Není třeba tedy komprimovat celé snímky, ale pouze jejich rozdíl (předpověď mezi snímky), který se vytváří diferenciální pulsní kódovou modulací. Jedná se o redukci redundance v časové oblasti, tedy o bezztrátové kódování. Rozdíly v hodnotách vzorků dvou po sobě jdoucích snímků se vytváří na úrovni jednotlivých makrobloků. Spolu s transformačním kódováním tvoří tzv. hybridní komprimační kódování [1].

1.3.1 Druhy snímků a jejich předpověď

Standard MPEG-2 je založen na třech typech snímků:

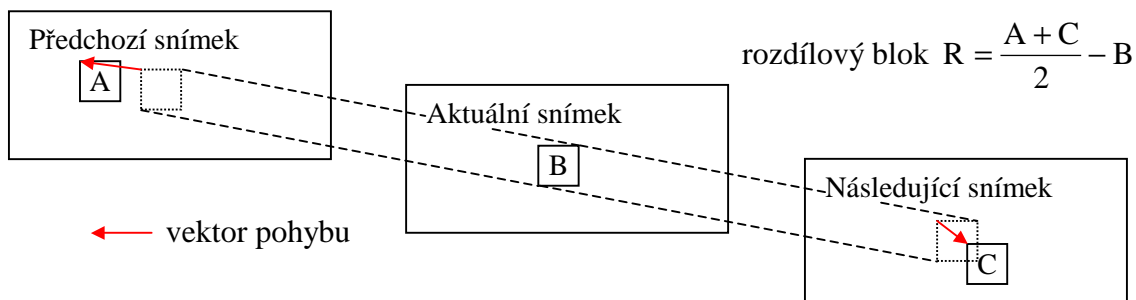
Snímek typu I (intra frame). Jedná se o tzv. referenční snímek, k jeho zakódování, respektive dekodování není potřeba žádného jiného snímku. Snímek se zpracovává přímo diskrétní kosinovou transformací. Slouží jako předpověď pro snímky typu P a B [1], [3].

Snímek typu P (forward or backward prediction), viz obr. 7. Pro kódování (dekódování) potřebuje jeden předcházející snímek typu I nebo P v případě, že se jedná o předpověď dopřednou (forward). Pro předpověď zpětnou (backward) naopak následující snímek, opět typu I nebo P. Při jednosměrné předpovědi, ať už zpětné nebo dopředné se bitová rychlost sníží asi dvakrát [1], [3].



Obr. 7. Předpověď snímku typu P : dopředná (vlevo) a zpětná (vpravo).

Snímek typu B (bidirectional prediction), viz obr. 8. Až osminásobné snížení bitové rychlosti se dosáhne obousměrnou předpovědí. Pro kódování a dekodování snímku typu B je potřeba předchozího a následujícího snímku, a to typu I nebo P. Kódovaný rozdíl je vytvořen jako rozdíl právě kódovaného snímku a průměru předchozího a následujícího snímku [1], [3].



Obr. 8. Obousměrná předpověď snímku typu B.

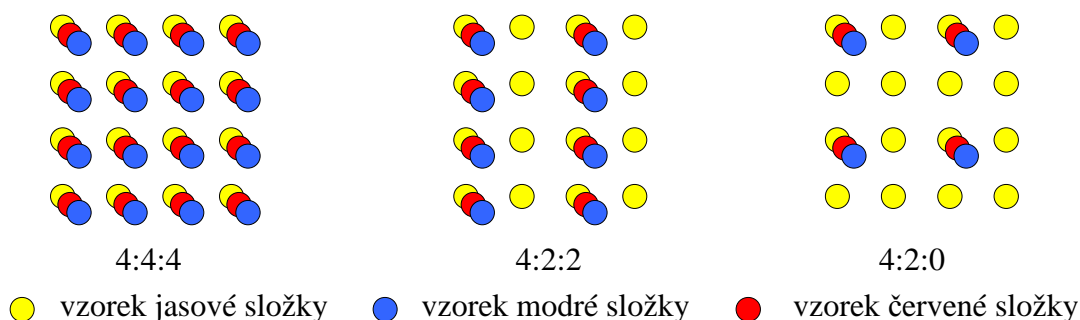
1.3.2 Diferenciální pulsní kódová modulace

DPCM vytváří rozdíl mezi aktuálním a předchozím či následným snímkem. Je výhodnější přenášet pouze rozdíl snímků, než úplné hodnoty vzorků každého snímku, tím dojde ke snížení objemu dat při přenosu. Při snímku typu I, který slouží jako referenční a kóduje se přímo bez jakékoliv předpovědi, se rozdíl nevytváří [2].

1.3.2.1 Vstupní signál DPCM

Vstupní signál DPCM jsou obrazová data reprezentována barevným modelem YCbCr, kde Y označuje jasovou (luminanční) složku, Cb a Cr pak barevné (chrominanční) složky modrou a červenou. Hlavní výhodou YCbCr oproti RGB je skutečnost, že barevné složky mohou být zastoupeny s nižším rozlišením než jasová složka, tím se snižuje množství dat potřebných k vyjádření těchto složek bez většího dopadu na vizuální kvalitu obrazu [2].

Obrazový signál YCbCr může být vzorkován, s poměrem složek Y:Cb:Cr, ve třech variantách, viz obr. 9, a to: 4:4:4, kdy na 4 vzorky jasového signálu připadají 4 vzorky od každé barevné složky. Další možnost je 4:2:2, kdy na 4 vzorky jasového signálu připadají 2 vzorky od každé barevné složky. A poslední varianta je vzorkování 4:2:0, kdy na 4 vzorky jasového signálu připadá 1 vzorek od každé barevné složky [2].

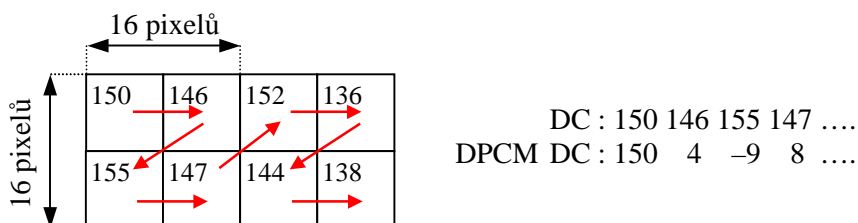


Obr. 9. Varianty vzorkování obrazového signálu YUV [2].

Tento popis variant je třeba uvést do správné podoby, protože zpracování dat se děje po blocích, respektive makroblocích, které mají rozměry 16x16 obrazových vzorků. Při vzorkování např. 4:2:2 se tedy bavíme o jednom makrobloku jasové složky (4x blok 8x8), dvěma blocích 8x8 červené složky a dvěma blocích 8x8 modré složky [2].

1.3.2.2 DPCM DC koeficientů

DPCM se mimo vytváření rozdílového snímků využívá i pro zpracování DC koeficientů u snímků I. Není, z hlediska úspory bitového toku, vhodné přenášet úplné hodnoty všech DC koeficientů. V ideálním případě se aplikací DPCM eliminují všechny vysoké hodnoty, kromě počáteční, viz obr. 10 [2].



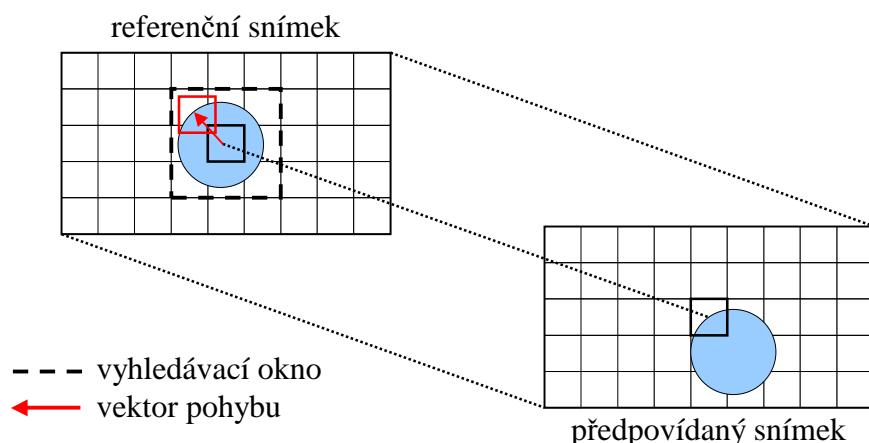
Obr. 10. DPCM DC koeficientů.

1.3.3 Odhad a kompenzace pohybu

Při použití pouze DPCM se vytvoří rozdíl v hodnotách vzorků na sobě si odpovídajících místech – makroblocích – sousedících snímků. Tím dojde ke snížení potřebného počtu bitů. Většího zvýšení úspory bitů dosáhneme použitím tzv. vektorů pohybu [1].

Princip, viz obr. 11, spočívá v přenášení informace o pohybu objektů za pomoci vektorů pohybu a rozdílů příslušných makrobloků namísto přenosu rozdílů hodnot makrobloků, které si odpovídají pozicí. Vyhledávání je hardwarově náročné a proto se vyhledává na úrovni makrobloků, tj. při zpracování makrobloku kódovaného snímku se vyhledává v předchozím či následujícím snímku makroblok se stejným obrazovým obsahem, jehož pozice je určena pohybovým vektorem o souřadnicích x , y . Hledání probíhá současně pro jasový i oba chrominanci signály, všem signálům pak náleží společný vektor pohybu [1].

Nalezení vhodných vektorů pohybu je označováno jako metoda odhadu pohybu (motion estimation) a existuje celá řada tzv. vyhledávacích algoritmů, některé z nich jsou popsány v následujících podkapitolách. Kompenzaci pohybu (motion compensation) je označována aplikace získaných pohybových vektorů na referenční snímek. Je ale zřejmé, že v reálném světě není pohyb vždy tak jednoznačný, objekt se mohl otočit nebo zmenšit. Proto je ještě proveden výpočet rozdílu mezi předpovídaným a kompenzovaným obrazem a ten je teprve pomocí DCT zakódován [1], [3].



Obr. 11. Princip odhadu pohybu.

1.3.3.1 Plné vyhledávání

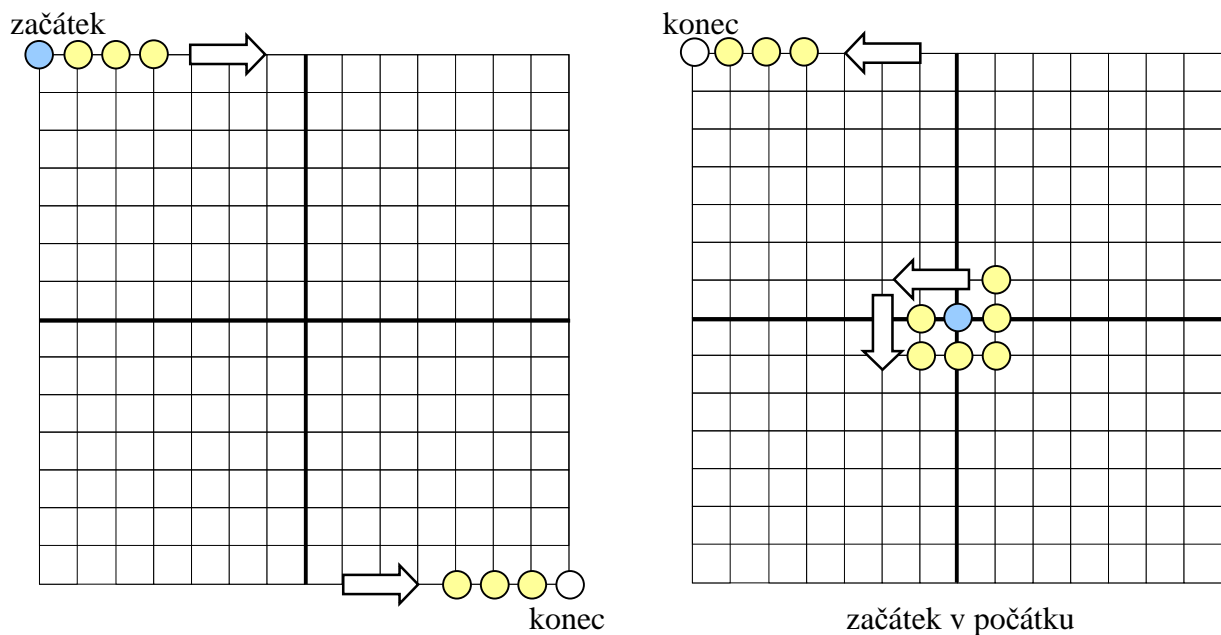
Pro nalezení makrobloku s největší shodou obrazu v referenčním snímku je teoreticky potřeba provést srovnání aktuálního makrobloku s každou možnou pozicí makrobloku v referenčním snímku. Takové srovnání je ovšem nepraktické z hlediska velkého množství výpočetních operací. V praxi se ovšem nejlepší shoda nalézá v blízkosti pozice aktuálního bloku v referenčním snímku. Oblast vyhledávání je pak definována velikostí vyhledávacího okna (search window) [2].

Plné vyhledávání porovnává aktuální makroblok a makrobloky v referenčním snímku v rámci zvoleného vyhledávacího okna a na základě výpočtu porovnávacího kritéria SAE (Sum of Absolute Errors), viz rovnice 4, vybere hodnotu s minimální chybou (nejlepší shodou obrazu) – minimum SAE. Souřadnice makrobloku s minimum SAE pak reprezentují vektor pohybu [2].

$$SAE = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |C_{ij} - R_{ij}|, \quad (4)$$

kde C je hodnota vzorku aktuálního makrobloku,
 R je hodnota vzorku v referenčním snímku a
 N je velikost makrobloku.

Plné vyhledávání může probíhat v rastrovém nebo spirálovém uspořádání, viz obr. 12. Spirálové uspořádání má jistou výpočetní výhodu při použití předčasného ukončení vyhledávání, protože nejlepší shoda je s největší pravděpodobností v blízkosti středu vyhledávací oblasti. Obecně je plné vyhledávání ze všech vyhledávacích algoritmů nejvíce výpočetně náročné, ovšem výhodou je, že vždy nalezne globální minimum prohledávané oblasti (minimum SAE) [2].



Obr. 12. Vyhledávání rastrové (vlevo) a spirálové (vpravo) [2].

1.3.3.2 N-krokové vyhledávání

N-krokové vyhledávání se řadí mezi algoritmy rychlého vyhledávání. Hlavní výhodou takového vyhledávání je redukce výpočetních operací oproti plnému vyhledávání a tedy i výrazné zrychlení celého algoritmu. Nevýhodou je skutečnost, že algoritmus prohledává vyhledávací oblast ve zvoleném počtu kroků a velikostí kroku, může se tedy stát, že algoritmus nenalezne globální minimum SAE, ale pouze lokální minimum. Velikost kroku S a vyhledávacího okna jsou dány vztahy:

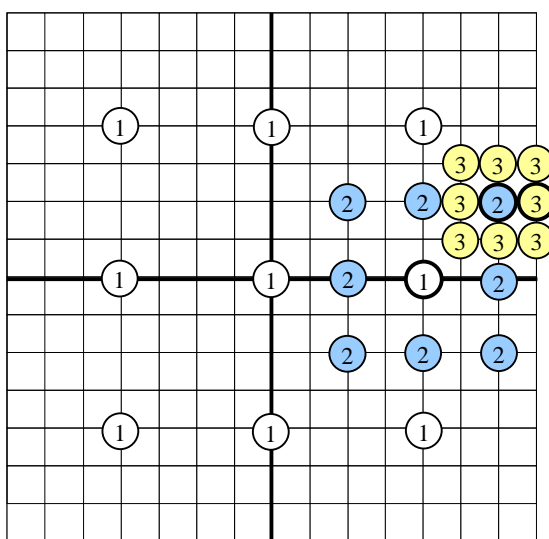
$$S = 2^{N-1}, \quad (5)$$

$$+/- (2^N - 1), \quad (6)$$

kde N je počet kroků.

Nejrozšířenější variantou N-krokového vyhledávání je tříkrokové, viz obr. 13. Vyhledávací okno je pro tři kroky +/- 7 pixelů. Prvním krokem vyhledávání je výpočet osmi hodnot SAE ve vzdálenosti +/- 4 pixely, viz rovnice 5, okolo počátku (0, 0) a včetně něj. Z vypočtených devíti hodnot se vybere hodnota s nejmenším SAE a její poloha se nastaví

jako nový počátek. Velikost kroku se zmenší na polovinu a celý proces se opakuje, dokud není velikost kroku rovna jedné [2].

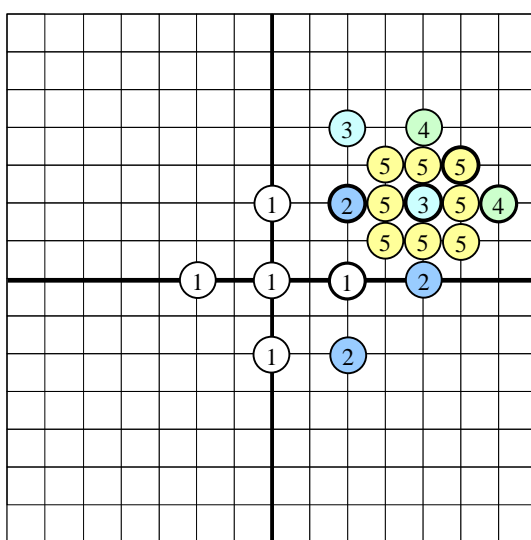


Obr. 13. Princip tříkrokového vyhledávání [2].

1.3.3.3 Logaritmicke vyhledávání

Logaritmicke vyhledávání, viz obr. 14, je další z řady algoritmů rychlého vyhledávání. Výhody a nevýhody jsou podobné jako u N-krokového vyhledávání, s rozdílem, že u logaritmickeho vyhledávání není pevně stanoven počet kroků, což může vést k většímu počtu výpočetních operací [2].

Prvním krokem vyhledávání je výpočet čtveřice hodnot SAE ve vzdálenosti S pixelů, kde S je počáteční velikost kroku, okolo počátku (0, 0) a včetně něj. Pozice vypočtených hodnot jsou ve tvaru '+'. Z vypočtených hodnot se vybere hodnota s nejmenším SAE a její poloha se nastaví jako nový počátek. Pokud se hodnota s minimem SAE nachází ve středu tvaru '+', tak je velikost kroku S zmenšena na polovinu. Tento postup se opakuje do doby, než se velikost kroku rovná jedné. Posledním krokem je výpočet osmi hodnot SAE obklopujících aktuální počátek, z těchto osmi hodnot a hodnoty v počátku je vybrána nejlepší shoda [2].

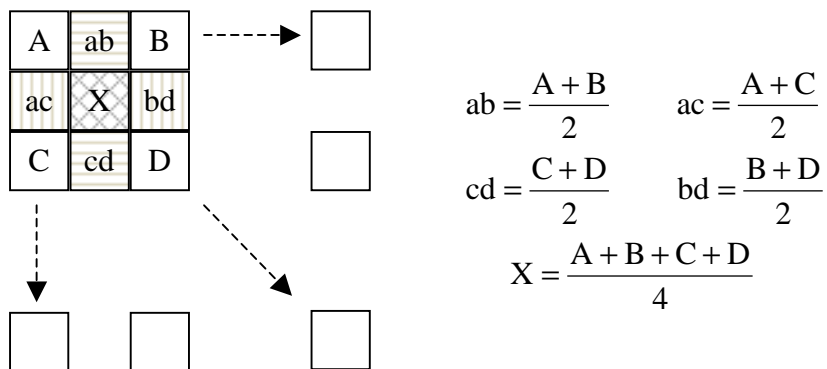


Obr. 14. Princip logaritmickeho vyhledávání [2].

1.3.4 Mezipixelový odhad a kompenzace pohybu

Mezipixelová kompenzace pohybu může poskytnout podstatně lepší kompresi než kompenzace, která pracuje s rozlišením celých pixelů, ovšem na úkor zvýšené složitosti. Jasové a chrominanční vzorky na pozici sub-pixelu, tj. mezi pixely, v referenčním obraze neexistují a proto je nutné je vytvořit pomocí interpolace z okolních vzorků obrazu.

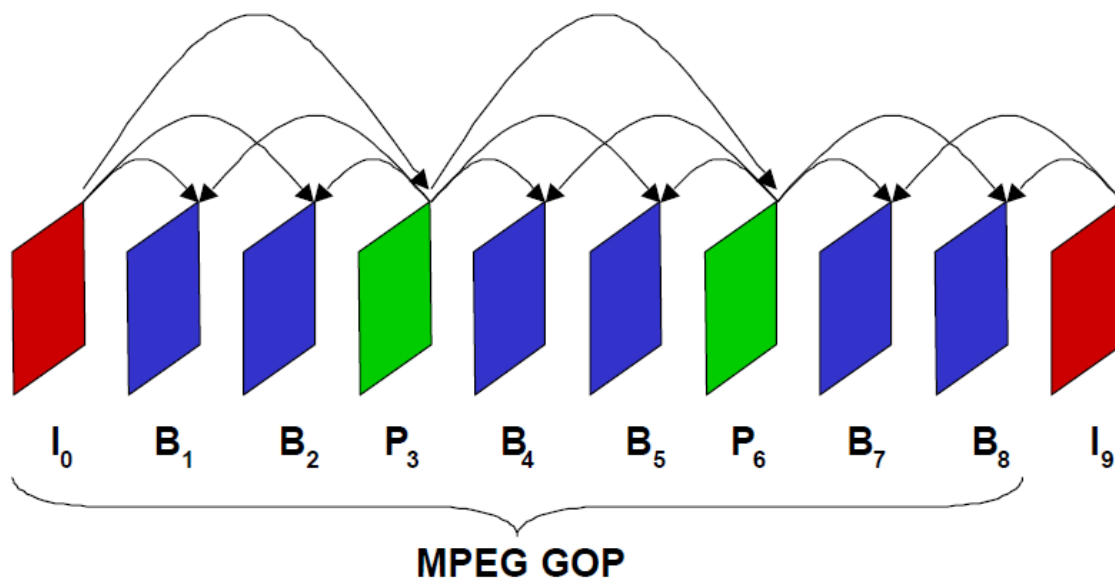
Existuje řada interpolačních funkcí – bikubická a bilineární interpolace u kodeku VC-1. Pro bilineární metodu, viz obr. 15, není důležitý sled operací, protože výsledek je definován jako výpočet ze čtyř nejbližších celých pixelů. U bikubické interpolace je sled operací naopak zásadní. U kodeku AVC jsou vzorky interpolované za pomoci FIR filtru [8].



Obr. 15. Bilineární interpolace hodnot vzorků v poloze ½ pixelu.

1.3.5 Skupina snímků

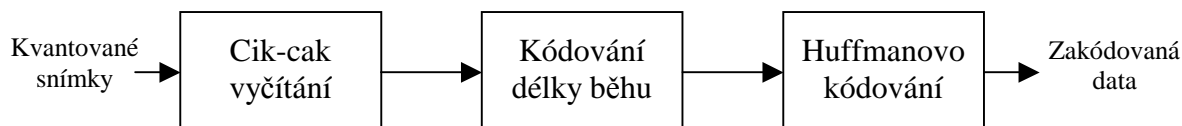
Struktura GOP, viz obr. 16, obsahuje snímky typu P, B a jeden snímek typu I, který je na začátku GOP. Ostatní snímky jsou dopočítávány na základě předpovědi. Strukturu lze popsat dvěma parametry: N, což je počet snímků v GOP, respektive délka GOP a pro MPEG-2 je hodnota délky 12, a M, což je vzdálenost snímků P. Snímek typu I je referenční snímek a nevyžaduje žádné dodatečné informace k jeho dekódování. Případné chyby v obraze jsou korigovány dalším snímkem typu I [3].



Obr. 16. Skupina snímků standardu MPEG [3].

1.4 Kódování s proměnnou délkou slova

Kódování s proměnnou délkou slova (VLC/Variable Length Coding) je poslední etapa komprimace signálu před samotným přenosem. Jedná se o kódování kvantovaných snímků a jedná se o bezeztrátovou úpravu signálu. Skládá se ze tří kroků: cik-cak vyčítání, kódování délky běhu (RLE/Run Length Encoding) a Huffmanovo kódování, viz obr. 17. Takto popsané kódování probíhá pouze pro AC koeficienty snímku I a DCT koef. snímků P/B, DC koef. snímku I a vektory pohybu jsou zpracovávány přímo Huffmanovým kódováním [9].



Obr. 17. Kódování s proměnnou délkou slova [9].

1.4.1 Cik-cak vyčítání

Před samotným kódováním dochází k vyčítání kvantovaných frekvenčních koeficientů tzv. cik-cak způsobem, viz obr. 18. Jedná se o vyčítání DCT koeficientů podle úhlopříčky od levého horního rohu do pravého dolního rohu matice. Stejnoseměrný koeficient DC se v případě snímku I přenáší samostatně. Tento způsob vyčítání je výhodný, neboť hodnoty koeficientů se zmenšují stejným směrem. Výstupem čtení je sériový tok dat, u něhož jsou od jistého koeficientu samé nuly [1].

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

Obr. 18. Cik-cak vyčítání kvantovaných koeficientů [9].

1.4.2 Kódování délky běhu

Vyčtené koeficienty se nekódují podle četnosti v toku dat, ale zavádí se skupiny, které jsou složeny z hodnoty koeficientu a počtu předcházejících nul. Skupina je potom dána dvěma symboly [1].

První symbol v sobě nese informaci o počtu nul a počtu bitů potřebných pro kódování frekvenčního koeficientu. Má-li například koeficient hodnotu 6 a předchází-li ho 5 nul, je první symbol 5/3. Hodnota 3 vychází ze skutečnosti, že pro vyjádření hodnoty 6 jsou zapotřebí 3 bity. Druhý symbol v přenášené skupině vyjadřuje hodnotu 6 v binární formě. Celá skupina je tedy 5/3/6, a v binární podobě 101|011|110 [1].

1.4.3 Entropické kódování

Entropie udává minimální počet bitů, který je potřeba pro vyjádření hodnoty vzorku v přenosu. Přiřazení bitů závisí na tom, s jakou pravděpodobností se hodnota vzorku vyskytuje v přenosu. Hodnotám s vyšší pravděpodobností výskytu se přiřazují slova, která jsou vyjádřena menším počtem bitů a naopak hodnotám, které se vyskytují málo, jsou přiřazena slova vyjádřena větším počtem bitů. Entropické kódování se také označuje jako statické [1].

1.4.3.1 Huffmanovo kódování

Standard MPEG-2 používá Huffmanovo kódování. Jedná se o tzv. prefixový kód, což znamená, že je jednoznačně dekódovatelný. Před samotným kódováním je nutné sestavit tabulku VLC kódu, která se vytvoří dvěma kroky. Prvním krokem je vytvoření statistiky četností výskytu každého symbolu v toku dat. Z této statistiky je odvozena pravděpodobnost výskytu symbolů. Druhým krokem je vytvoření binárního stromu na základě pravděpodobnosti a odvození Huffmanova kódu, viz tabulka 2, 3 [10].

Symbol	Pravděpodobnost	1	2	3	4
a1	0,4	0,4	0,4	0,4	0,6 0,4
a2	0,3	0,3	0,3	0,3	
a3	0,1	0,1	0,2	0,3	
a4	0,1	0,1	0,1		
a5	0,06	0,1			
a6	0,04				

Tab. 2. Huffmanova zdrojová redukce [10].

Symbol	Pravděpodobnost	Kód	1	2	3	4
a1	0,4	1	0,4 1	0,4 1	0,4 1	0,6 0
a2	0,3	00	0,3 00	0,3 00	0,3 00	0,4 1
a3	0,1	011	0,1 011	0,2 010	0,3 01	
a4	0,1	0100	0,1 0100	0,1 011		
a5	0,06	01010	0,1 0101			
a6	0,04	01011				

Tab. 3. Vytvoření Huffmanova kódu [10].

Ve standardu MPEG-2 je definována sada VLC tabulek, které byly ověřeny s určitou přesností v praxi. Pro kódování snímku I jsou definovány 3 tabulky. Tabulka B-12 pro DC koeficienty jasové složky Y, tabulka B-13 pro DC koeficienty chrominancních složek C a pro AC koeficienty tabulka B-15. Pro kódování DCT koeficientů snímku P a B slouží tabulka B-14. Vektory pohybu jsou kódovány tabulkou standardu MPEG-4 Part 2 Visual, která zahrnuje kódová slova i pro pohybové vektory s přesností 1/2 pixelu. B-12 [11], [15].

2 METRIKY PRO MĚŘENÍ KVALITY

PSNR (**P**eak **S**ignal-to-**N**oise **R**atio), neboli špičkový poměr signálu k šumu vyjadřuje poměr mezi maximální možnou energií signálu a energií šumu. PSNR se obvykle vyjadřuje v logaritmickém měřítku, jelikož spousta signálů má velmi široký dynamický rozsah. Hodnota PSNR se udává v decibelech a typické hodnoty pro kompresi videa jsou mezi 30dB až 50dB. Za přípustné hodnoty jsou pro bezdrátový přenos považovány hodnoty 20dB až 25dB. PSNR je definován rovnicí [16]:

$$\text{PSNR}_{\text{dB}} = 10 \cdot \log_{10} \left(\frac{\text{MAX}_I^2}{\text{MSE}} \right) = 20 \cdot \log_{10} \left(\frac{\text{MAX}_I}{\sqrt{\text{MSE}}} \right), \quad (7)$$

kde MAX_I je maximální hodnota pixelu obrazu a MSE (**M**ean **S**quared **E**rror) je střední kvadratická chyba. MSE se určí jako:

$$\text{MSE} = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - K(i, j)]^2, \quad (8)$$

kde I, K jsou černobílé obrazy a m, n jsou rozměry.

SSIM (The **S**tructural **S**IMilarity) index je navržen s cílem zlepšit tradiční metodami, jako je PSNR a MSE, které se ukázaly být v rozporu s vnímáním lidského oka. SSIM index je metoda pro měření podobnosti mezi dvěma obrazy, systém měření je na obr. 19. Tato metoda bere v potaz skutečnost, že lidské vidění je vysoce přizpůsobeno k extrahování strukturální informace. Index nabývá hodnot 0 až 1, kde 1 vyjadřuje shodné obrazy. U barevných obrazů se obvykle počítá jen na jasové složce Y . Hodnota SSIM je dána vztahem [17], [18]:

$$\text{SSIM}(x, y) = l(x, y)^\alpha \cdot c(x, y)^\beta \cdot s(x, y)^\gamma, \quad (9)$$

kde $l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}$ je funkce porovnání jasu, (10)

$c(x, y) = \frac{2\sigma_{xy} + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}$ je funkce porovnání kontrastu, (11)

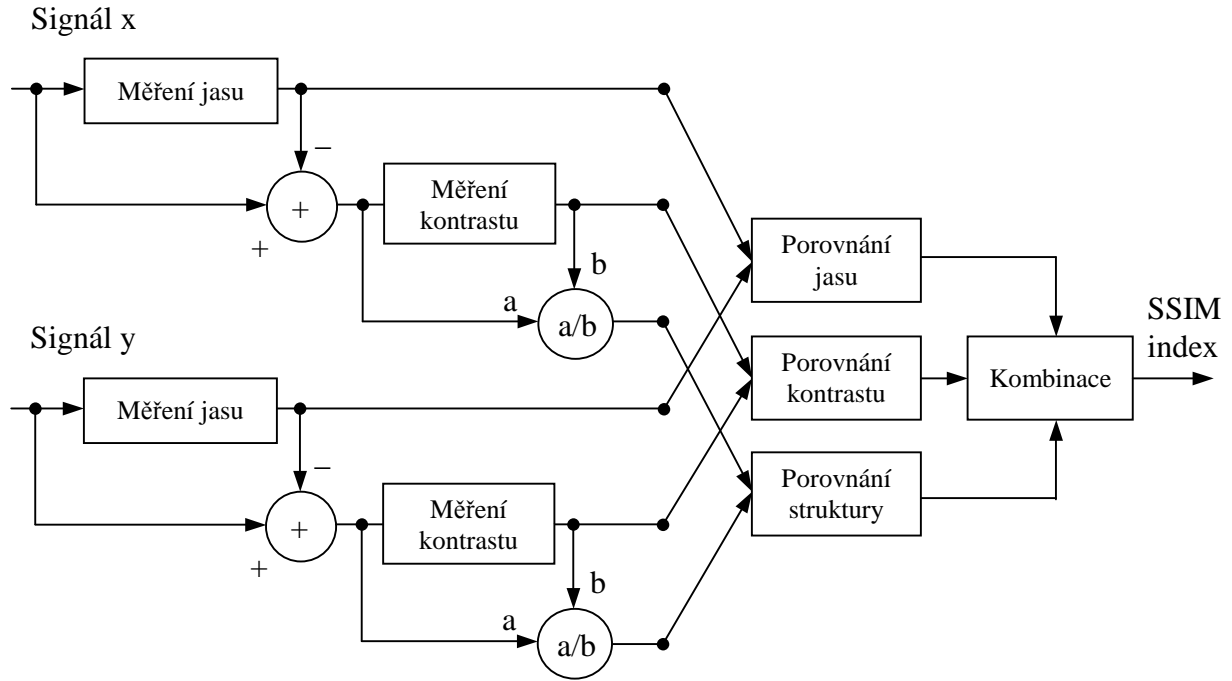
$s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}$ je funkce porovnání struktury (12)

Při hodnotách $\alpha = \beta = \gamma = 1$ a $C_3 = C_2 / 2$ dostáváme konkrétní měřítko kvality SSIM:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \quad (13)$$

kde α, β, γ - kladné konstanty sloužící k váhování jednotlivých porovnávacích funkcí,
 x, y - souřadnice,

μ_x, μ_y – vážený průměr x, y ,
 σ_x, σ_y – vážená variance x, y ,
 σ_{xy} – kovariance x a y ,
 $C_i = (K_i L)^2$, L je maximální hodnota pixelu v obrazu a K jsou malé konstanty. V práci jsou použité hodnoty $K_1 = 0,01$ a $K_2 = 0,03$.



Obr. 19. Systém měření strukturální podobnosti [19].

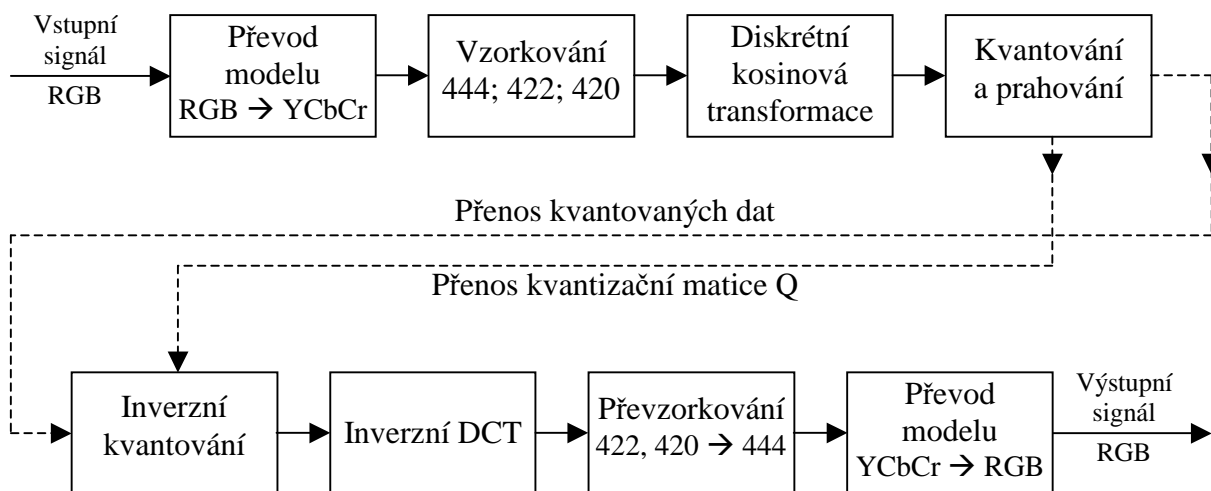
3 REALIZACE VIDEO KODEKU

K softwarovému řešení jednotlivých komponent video kodeku je využito programové prostředí Matlab (Matlab2009b verze 7.9.0.529). Veškeré výsledky obsažené v práci byly dosaženy na PC s konfigurací Intel Core2Duo 6300@1,86GHz, RAM 2GB, Windows 7 Professional 32bit SP1. Video kodek je tvořen zdrojovým a kanálovým kódováním. Zdrojové kódování je možné rozdělit na dvě části – kódování uvnitř snímku a mezisnímkové kódování. V následujících podkapitolách bude uveden základní popis zdrojového kódu jednotlivých komprimačních nástrojů a přehled dosažených výsledků.

3.1 Kódování uvnitř snímku

Zdrojový kód pro kódování uvnitř snímku odpovídá blokovému schématu na obr. 20. První čtyři bloky tvoří část kodéru, druhá sada čtyř bloků s inverzními operacemi tvoří část dekodéru.

Pro přesnost je třeba říct, že blok Převod modelu RGB \rightarrow YCbCr (YCbCr \rightarrow RGB) není součástí kódování, ale pro celkový přehled o funkci kodeku je třeba se o zpracování vstupních dat zmínit, viz kapitola 3.1.1. Přenos dat je v tomto případě chápán jako předávání hodnot mezi funkcemi v rámci několika programových souborů (*m-files*). Kapitola dosažené výsledky reprezentuje ověření funkce transformací, kvantování a dopad na kvalitu obrazu za použití různé hodnoty kvantizačního parametru.



Obr. 20. Blokové schéma kódování uvnitř snímku – zpracování 1 snímku.

3.1.1 Zpracování vstupních dat

Prvním krokem zpracování vstupních dat je samotné načtení videosekvence, definování parametrů výstupní videosekvence, volba vzorkování a zajištění kontinuálního zpracování jednotlivých snímků pomocí cyklu.

Před samotnou aplikací transformačního kódování, respektive před kompletním zpracováním obrazového signálu je potřeba převést barevný model RGB do modelu YCbCr, a podle zvoleného vzorkování podvzorkovat barevné složky Cb a Cr.

```
YCBCR = rgb2ycbcr(snimek_RGB); % Převod RGB → YCbCr.
```

3.1.2 Diskrétní kosinová transformace

Diskrétní kosinová transformace (m-file *koder_DCT_Q.m*) se provádí na blocích 8x8. Pro rychlejší zpracování se nevyužívá funkce *dct2*, ale funkce *dctmtx*, jejíž výstupem je matice DCT. Samotná transformace se pak provede vynásobením bloku 8x8 zleva DCT maticí a zprava transponovanou DCT maticí. Zpracování všech vzorků obrazu zajišťují dva cykly *for*, vždy se vybere blok 8x8, provede se transformace a uloží se do nové proměnné – matice – na odpovídající pozici. Transformace jasové složky a barevných složek probíhá odděleně z důvodu možného vzorkování barevných složek. Je potřeba dodat, že rozměry vstupního obrazu musí být dělitelné 8, respektive 16.

```
d = dctmtx(8); % Výpočet DCT matice 8x8.
for i = 1:8:radky % Posun po blocích 8x8.
    for j = 1:8:sloupce
        Y_DCT(blok 8x8) = d*(Y(blok 8x8))*d'; % 2D DCT bloku 8x8.
```

3.1.3 Diskrétní vlnková transformace

Diskrétní vlnková transformace (m-file *koder_DWT_Q.m*) má stejnou strukturu zpracování jako DCT, tzn. 2 cykly *for* pro zpracování snímku po blocích 8x8. Výpočet DWT bloku 8x8 se provádí funkcí *DWT* (m-file *DWT.m*).

```
Y_DWT(blok 8x8) = DWT(Y(blok 8x8)); % 2D DWT bloku 8x8.
```

Tato funkce obsahuje tři stupně transformace, každý stupeň je obstarán funkcí *dwt2* se vstupní vlnkou *db1* (Daubechies wavelet).

```
[cA1, cH1, cV1, cD1] = dwt2(data, 'db1'); % 2D DWT bloku 8x8.
dwtL1 = [cA1, cH1; cV1, cD1]; % Blok 8x8 prvního stupně DWT.
```

3.1.4 Kvantování a prahování

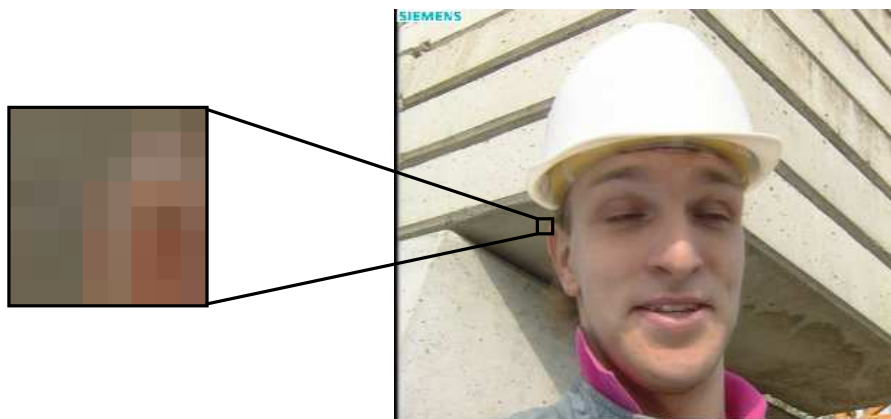
Kvantování probíhá ve stejných cyklech jako DCT/DWT, blok 8x8 se ihned po transformaci kvantuje na základě rovnice 2. Kvantizační parametr Q_p kontroluje míru kvantování, větší hodnoty znamenají větší míru kvantování. Kvantované bloky jsou následně prahovány funkcí *round*, která zaokrouhluje hodnoty vzorků na nejbližší celé číslo, tím dochází k eliminaci desetinných čísel, které potřebují k přenosu více bitů a také dochází k odstranění hodnot blízkých nule, což znamená další ušetření bitové rychlosti. Kvantizační matice použitá pro DWT je uvedena v rovnici 14 [14].

```
Y_Kvantovane(blok 8x8) = (16* Y_DxT(blok 8x8))./(Q*Qp)- ...
    (sign(Y_DxT(blok 8x8))/2) % Kvantování.
Y_Prahovane=round(Y_Kvantovane); % Prahování kvantovaných vzorků.
```

$$q_{DWT} = \begin{bmatrix} 8 & 7 & 8 & 8 & 34 & 34 & 34 & 34 \\ 7 & 7 & 8 & 8 & 34 & 34 & 34 & 34 \\ 8 & 8 & 12 & 12 & 34 & 34 & 34 & 34 \\ 8 & 8 & 12 & 12 & 34 & 34 & 34 & 34 \\ 34 & 34 & 34 & 34 & 55 & 55 & 55 & 55 \\ 34 & 34 & 34 & 34 & 55 & 55 & 55 & 55 \\ 34 & 34 & 34 & 34 & 55 & 55 & 55 & 55 \\ 34 & 34 & 34 & 34 & 55 & 55 & 55 & 55 \end{bmatrix} \quad (14)$$

3.1.5 Dosažené výsledky I

Pro ověření funkčnosti byl použit nekomprimovaný snímek ze sekvence Foreman, viz obr. 21, s rozměry 352x288 pixelů a bitovou hloubkou 24bitů, formát vstupního signálu 4:2:0. První část výsledků prezentuje dopad velikosti kvantizačního parametru na výslednou kvalitu snímku po dekomprimování a dále objektivní a subjektivní srovnání obou transformací. V druhé části výsledků je detailní ukázka zpracování testovacího bloku 8x8, opět pro obě transformace.



Obr. 21. Originální snímek a testovací blok 8x8.

3.1.5.1 Transformace a kvantování

Na obr. 22 je odpovídající dekomprimovaný snímek pro kvantizační parametr o velikosti 4. U DCT si lze všimnout jemného rozostření celého obrazu a při větším přiblížení i samotných bloků. U DWT naopak je kvalita srovnatelná s originálem, při přiblížení je možné si všimnout pixelů s ostrými přechody.

Dvojnásobný kvantizační parametr, viz obr. 23, má za následek mírné zhoršení kvality. Při standardním zobrazení, u DCT, lze oproti předešlému případu pozorovat občasný výskyt blokových artefaktů a celkové rozmazání obrazu. Naproti tomu u DWT je kvalita stále srovnatelná s originálem, vyskytuje se jen mírné „roapixelování“ obrazu.

Na obr. 24 a 25 je dekomprimovaný obraz při hodnotě 16, respektive 31, kvantizačního parametru. U DCT se oproti DWT kvalita zhoršuje výrazněji. V obraze se vyskytuje více a více blokových artefaktů. Při kvantizačním parametru 31 vykazuje ztrátu většiny detailů, jsou pozorovatelné pouze bloky. U DWT je na tom obraz o poznání lépe, jeho kvalita subjektivně odpovídá obrazu u DCT při kvantizačním parametru 16.

Pro objektivní srovnání transformací byl použit parametr bpp (**bit per pixel**) a PSNR, index SSIM pro měření kvality snímku. Na obr. 26 lze pak vidět, že DCT poskytuje lepší výsledky než DWT. Subjektivně srovnat lze dekomprimovaný snímek DCT při kvantizačním parametru 16 (0,3304 bitů/pixel) a DWT při kvantizačním parametru 31 (0,3309 bitů/pixel), při tomto porovnání je kvalita dekomprimovaného snímku u DCT o něco lepší než u DWT.

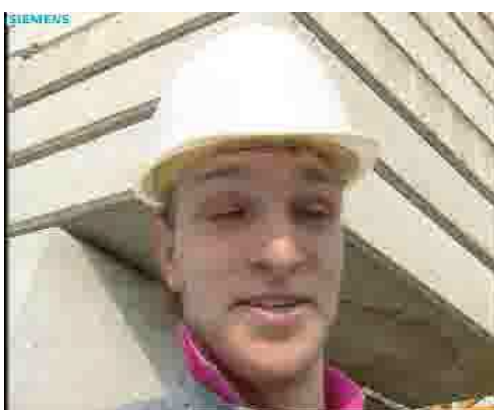
Jak už bylo v teoretické části zmíněno, DWT je výhodnější pro zpracování větších bloků či celého snímku a také je nutnost použití odpovídajících komprimačních nástrojů. Při implementaci samotné DWT do stávajícího konceptu, bez příslušných souvisejících nástrojů, nelze dosáhnout lepších výsledků než DCT.



Obr. 22. Kvantizační parametr 4 : DCT (vlevo) a DWT (vpravo).



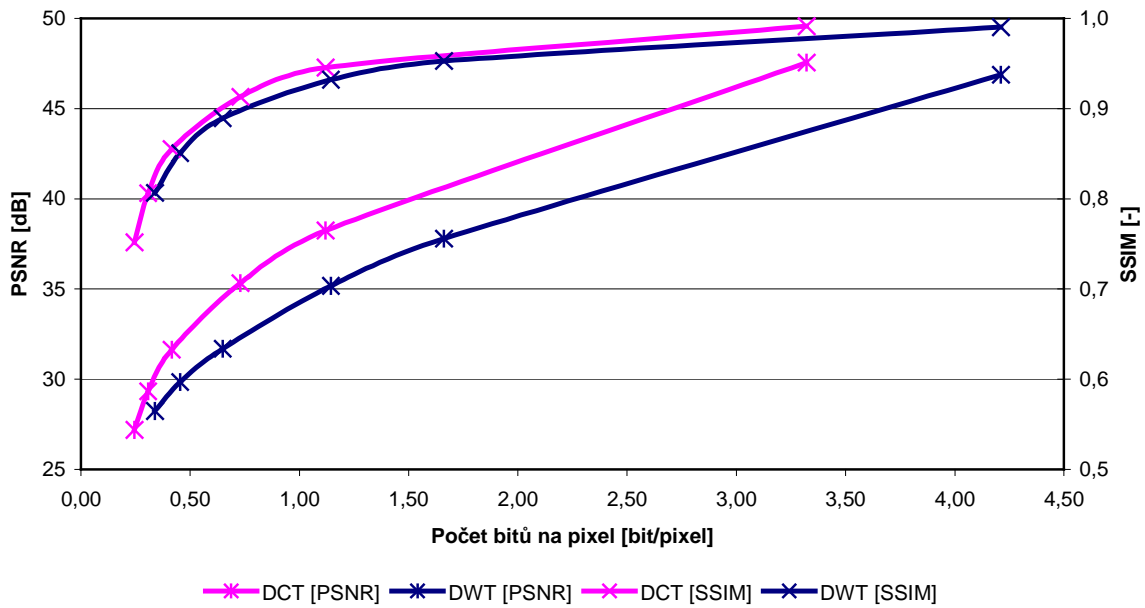
Obr. 23. Kvantizační parametr 8 : DCT (vlevo) a DWT (vpravo).



Obr. 24. Kvantizační parametr 16 : DCT (vlevo) a DWT (vpravo).



Obr. 25. Kvantizační parametr 31 : DCT (vlevo) a DWT (vpravo).

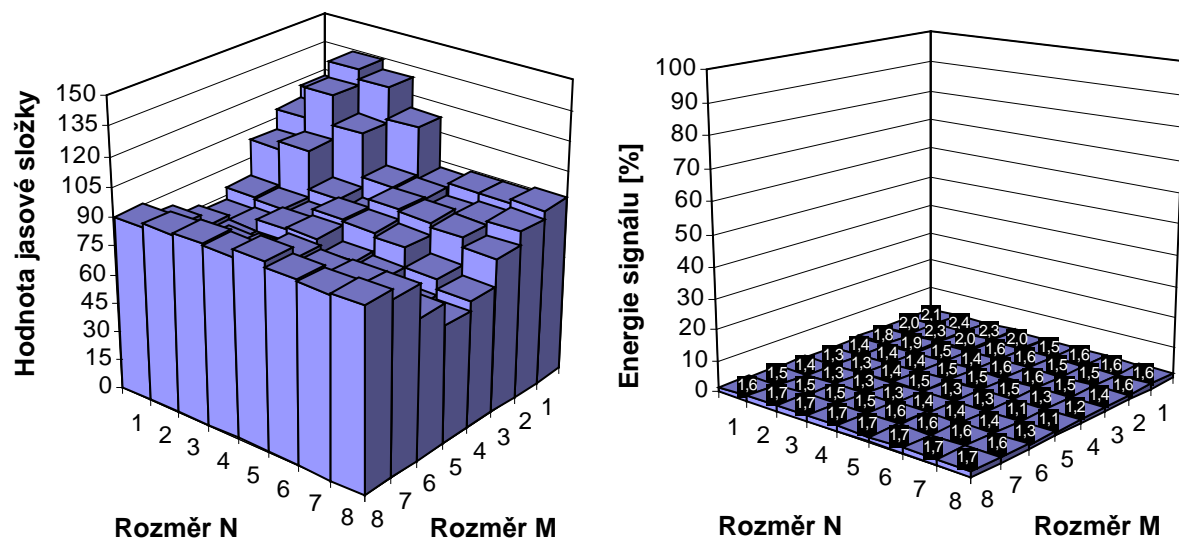


Obr. 26. Závislost SSIM a PSNR na parametru bpp pro jasovou složku snímku.

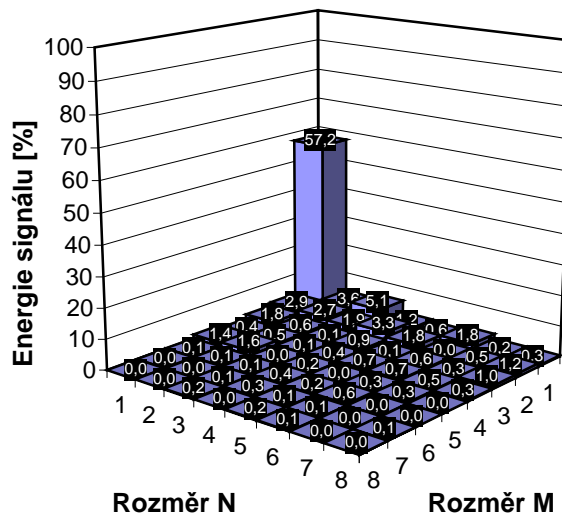
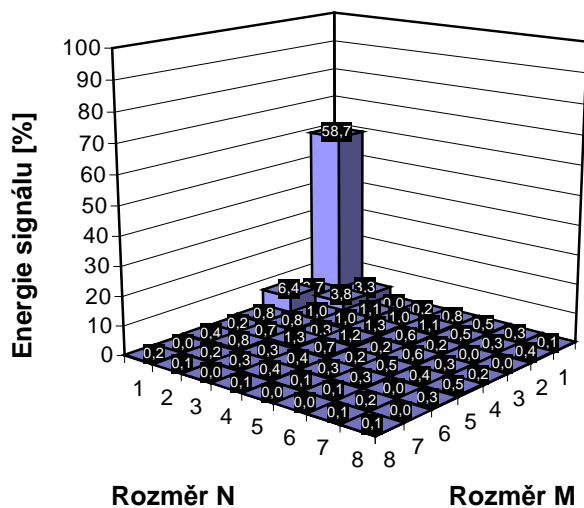
3.1.5.2 Zpracování testovacího bloku

Obr. 27 zobrazuje rozložení hodnot a procentuální rozložení energie signálu jasové složky testovacího bloku. V prostorové oblasti je energie rovnoměrně rozložena mezi všech 64 hodnot bloku.

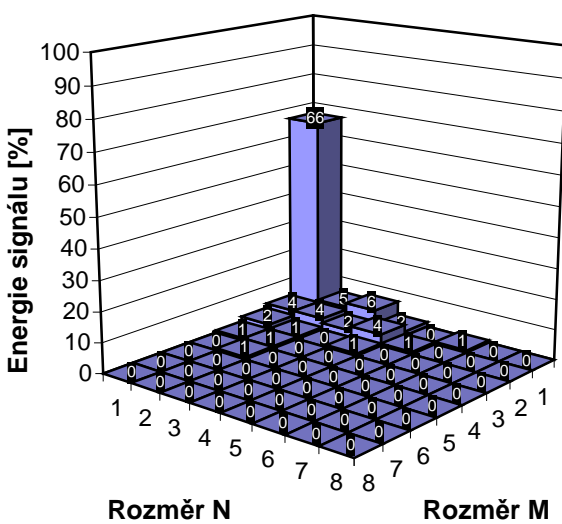
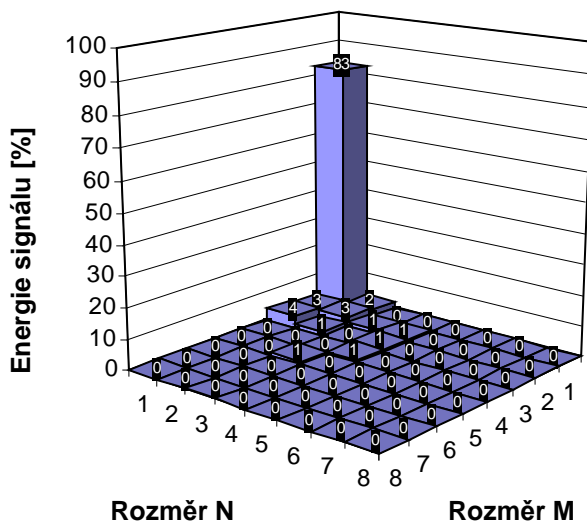
Po transformaci DCT/DWT, viz obr. 28, dojde ke změně rozložení energie a velká část energie je transformována na pozici DC koeficientu, 58,7% u DCT a 57,2% u DWT, a do jeho blízkého okolí. Po kvantování a prahování, viz obr. 29, vzroste energie DC koeficientu na 83% u DCT a na 66% u DWT. V tuto chvíli je na DC koeficientu a v jeho blízkém okolí (+2 pozice) více jak 97% energie testovacího bloku u DCT a přibližně 90% u DWT. Obr. 30 prezentuje rozložení hodnot chybového bloku, který představuje rozdíl hodnot originálního testovacího bloku a bloku po inverzním kvantování a transformaci DCT/DWT.



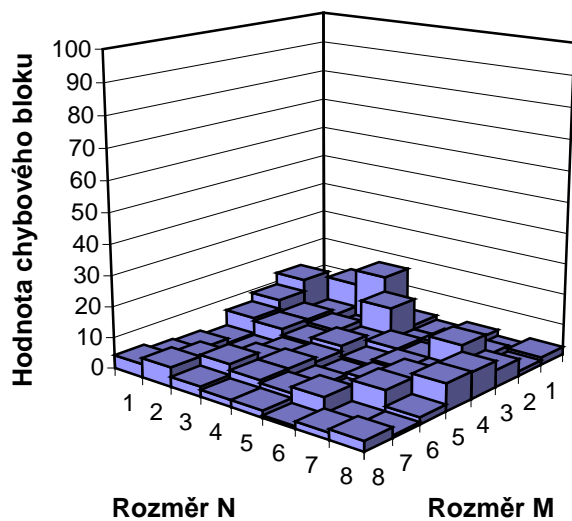
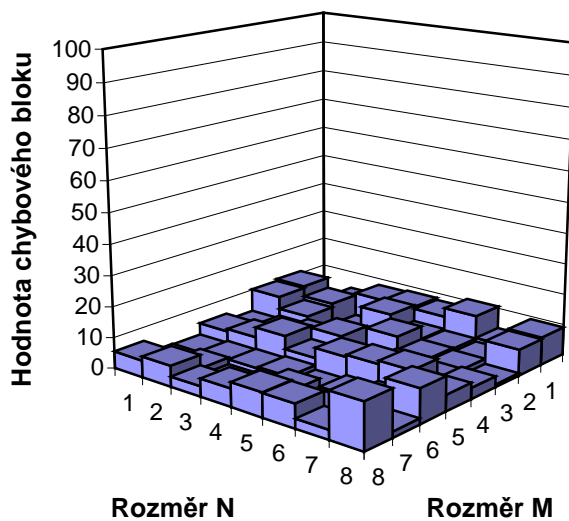
Obr. 27. Rozložení hodnot a energie signálu jasové složky testovacího bloku.



Obr. 28. Rozložení energie signálu po transformaci: DCT (vlevo) a DWT (vpravo).



Obr. 29. Rozložení energie signálu po kvantování : DCT (vlevo) a DWT (vpravo).

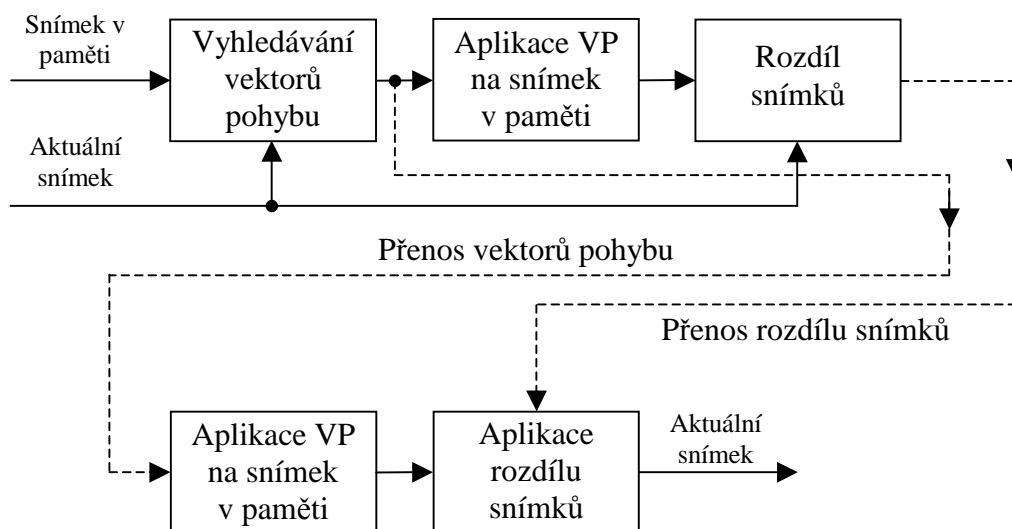


Obr. 30. Rozložení hodnot chybového bloku : DCT (vlevo) a DWT (vpravo).

3.2 Mezisnímkové kódování

Zdrojový kód pro mezisnímkové kódování vychází z blokového schématu na obr. 31. První tři bloky tvoří část kodéru, část dekodéru je tvořena dvěma bloky. Dekodér je výrazně jednodušší, jelikož neobsahuje vyhledávání vektorů pohybu.

Snímek v paměti je posunut vůči snímku aktuálnímu o jeden snímek vpřed, jedná-li se o předpověď zpětnou nebo o jeden snímek vzad, jde-li o předpověď dopřednou. Celý proces, jak v kodéru, tak dekodéru, probíhá po jednotlivých makroblocích. Vyhledávání vektorů pohybu má na starosti jeden ze tří vyhledávacích algoritmů, viz podkapitoly. Aplikací vektorů pohybu na snímek v paměti získáme snímek s minimálním rozdílem vůči aktuálnímu snímku. Rozdíl tohoto snímku a snímku aktuálního se spolu s vektory pohybu přenáší do dekodéru. Na straně dekodéru se obdobným způsobem zrekonstruuje aktuální snímek.



Obr. 31. Blokové schéma mezisnímkového kódování.

3.2.1 Plné vyhledávání

Plné vyhledávání (m-file *koder_predikce_FS.m*) je realizováno několika cykly *for*. První dvojice má na starosti zpracování snímku po jednotlivých makroblocích, další dvojice obstarává posun vyhledávacího okna. Aby prohledávaná oblast odpovídala zvolené velikosti vyhledávacího okna, je zapotřebí zvolenou velikost pro účely posunu zdvojnásobit. Povolená velikost vyhledávacího okna je ± 16 pixelů. Velikost kroku je vždy jeden pixel.

```
for m = 1:16:size(snímek,1) % Posun po makroblocích.
for n = 1:16:size(snímek,2)

for moffset = 0:(2*okno) % Posun vyhledávacího okna.
for noffset = 0:(2*okno)

for i = 0:15 % Výpočet SAE.
SAE = SAE + abs(Y2(i+m,0+n) - Y1(i+Ver,0+Hor));
SAE = SAE + abs(Y2(i+m,1+n) - Y1(i+Ver,1+Hor));
.
.
SAE = SAE + abs(Y2(i+m,15+n) - Y1(i+Ver,15+Hor));
```

Samotný výpočet SAE je realizován pouze jedním cyklem *for* z důvodu šetření výpočetního výkonu a tedy i času. V případě dvou cyklů *for* by byl každý ze 256 výpočtů doprovázen inkrementací a kontrolou čítače vnitřního cyklu *for* [2].

3.2.2 N-krokové vyhledávání

N-krokové vyhledávání (m-file *koder_predikce_NSS.m*) má obdobnou strukturu jako plné vyhledávání s tím rozdílem, že místo posunu vyhledávacího okna se vyhledávají makrobloky na pozicích teorií daného tvaru, což obstarává dvojice cyklů *for*. Algoritmus je omezen N-kroky – opět cyklus *for*, přičemž s každým krokem se velikost kroku zmenšuje. Touto podmínkou, viz rovnice 6, je dána velikost vyhledávacího okna, viz tabulka 4.

```
for NSS = 1:N                               % Cyklus pro N kroků (1, 2, 3, 4).
    for m = [-S,0,S]                         % Vyhledávání makrobloků na pozicích
        for n = [-S,0,S]                   % v požadovaném tvaru - viz teorie.
```

Jak plné, tak N-krokové vyhledávání obsahují sérii podmínek, které zamezují algoritmu prohledávat hodnoty mimo snímek.

Počet kroků N	Velikost vyhledávacího okna [+/- pixel]	Velikost počátečního kroku S [pixel]
1	1	1
2	3	2
3	7	4
4	15	8

Pozn. : Výpočty na základě rovnic 5 a 6

Tab. 4. Přehled velikostí vyhledávacího okna pro N-krokové vyhledávání.

3.2.3 Logaritmické vyhledávání

Toto vyhledávání (m-file *koder_predikce_LS.m*) vychází z vyhledávání N-krokového. Změna je pouze v nahrazení cyklu pro N kroků za smyčku *while*, neboť tento algoritmus prohledává oblast, která je dána počátečním krokem, viz tabulka 5, do doby, dokud není velikost kroku rovna jedné. V takovém případě se provede poslední prohledávání, krok se zmenší na polovinu a algoritmus se ukončí.

```
while(S>=1)                                 % Běh algoritmu, dokud není S<1.
```

Stejně jako předchozí vyhledávání obsahuje i vyhledávání logaritmické sérii podmínek pro zamezení výpočtu s hodnotami mimo snímek. Navíc obsahuje podmínky definující proměnný tvar, který se prohledává. Všechny tři algoritmy také obsahují kontrolu, zda má makroblok s minimálním SAE také minimální velikost pohybového vektoru. Pohybové vektory jsou ukládány do společné matice na pozici odpovídající příslušnému makrobloku. Na konci každého algoritmu je tato matice převedena na vektor a je tak připravena ke zpracování Huffmanovým kódováním.

Zvolená velikost počátečního kroku S [pixel]	Velikost vyhledávacího okna [+/- pixel]
1	1
2	3
4	7
8	15

Tab. 5. Přehled velikostí vyhledávacího okna pro logaritmické vyhledávání.

3.2.4 Vyhledávání s přesností 1/2 pixelu

Pokud je v hlavním vyhledávacím algoritmu zapnuta volba vyhledávání s přesností 1/2 pixelu, je přepínačem *switch* zpřístupněna funkce vyhledávání na sub-pixelových pozicích (m-file *koder_predikce_sub.m*).

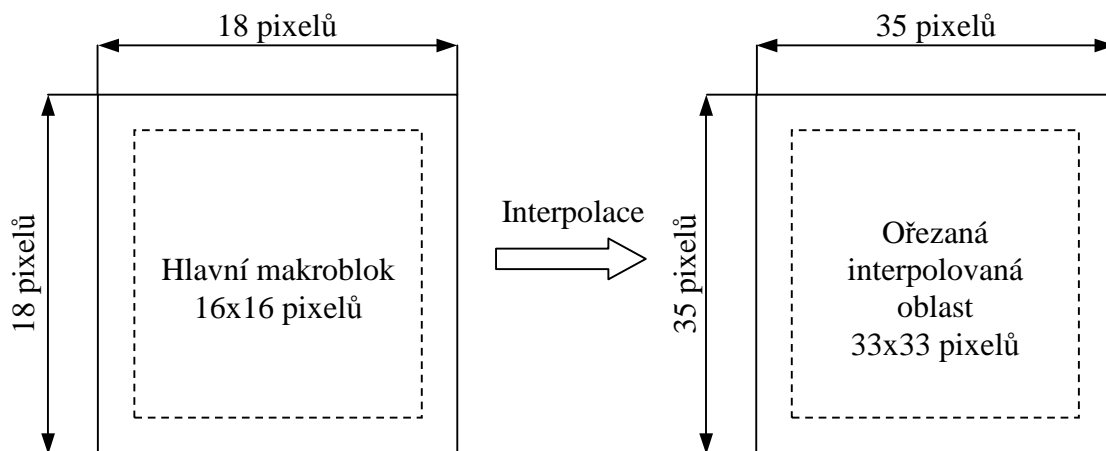
```
[i j delta] = koder_predikce_sub (oblast(Y1(MB)), Y2(MB), N);
```

Vstupem funkce je oblast, v jejímž středu je makroblok nalezený hlavním algoritmem, tedy makroblok s minimálním rozdílem vůči druhé vstupní proměnné – makrobloku předpovídaného snímku. Třetí vstupní proměnnou N je přesnost vyhledávání, respektive stupeň interpolace dané oblasti. Výstupem je vektor pohybu o hodnotách $i, j \in \{-0,5; 0; 0,5\}$. Tento vektor pohybu se následně přičte k hlavnímu pohybovému vektoru. Z tohoto vektoru se teprve odvozují vektory pohybu pro výpočet rozdílových makrobloků. Tato funkce také obsahuje volání funkce pro interpolaci makrobloku (m-file *interpolace.m*).

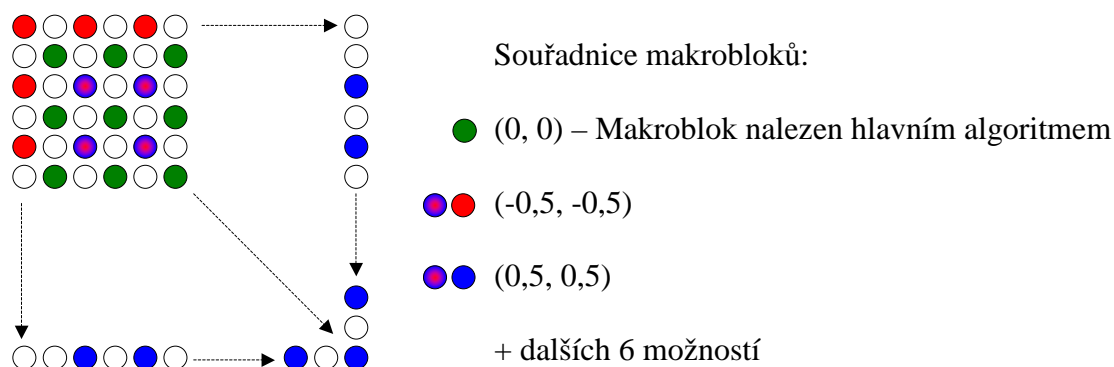
```
[int_oblast] = interpolace(oblast(Y1(MB)), N);
```

Interpolační funkce.

Bilineární interpolace, viz teorie, se provádí na oblasti o velikosti 18x18 pixelů, viz obr. 32. V této oblasti je celkově 9 makrobloků a její interpolací vznikne oblast o velikosti 35x35 pixelů. Z této oblasti se vybere střední část o velikosti 33x33, čímž se eliminuje prohledávání původních 8 makrobloků, na pozici celých pixelů, obklopujících hlavní makroblok. Vyhledávání probíhá plným vyhledáváním, viz obr. 33, tedy pro všechny možnosti interpolované oblasti. Celkový počet srovnávacích operací pro zpracování jedné oblasti je 9.



Obr. 32. Interpolace a zpracování vybrané oblasti.



Obr. 33. Princip vyhledávání s přesností 1/2 pixelu.

3.2.5 Dosažené výsledky II

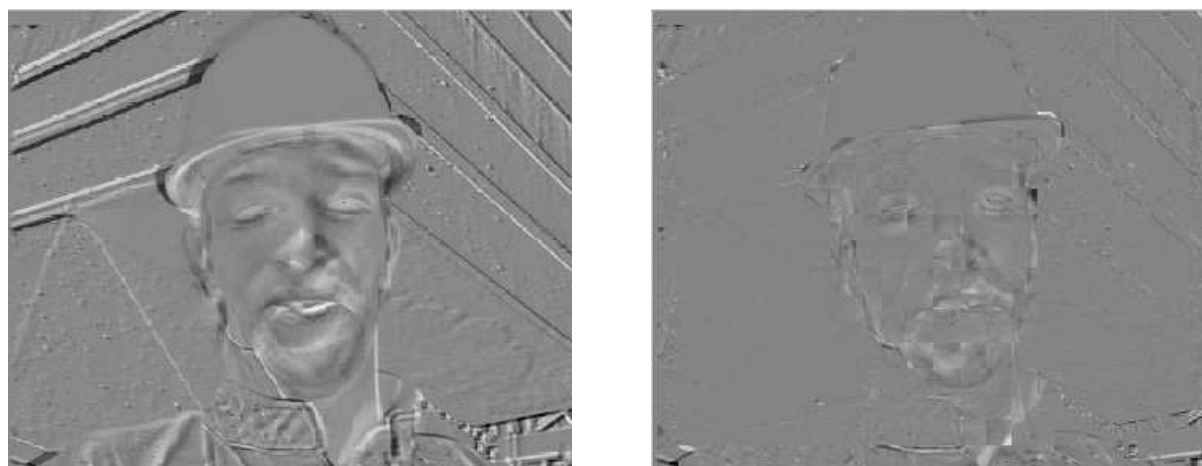
První část výsledků obsahuje porovnání popsaných vyhledávacích algoritmů při vyhledávání s přesností celého pixelu. A dále podrobnější pohled na předpověď. Ve druhé části je uvedeno srovnání odhadu pohybu s přesností celého a $\frac{1}{2}$ pixelu u plného vyhledávání. Výsledky prezentují rychlost a výkonnost jednotlivých algoritmů. Jako testovací snímky byly použity snímky 66 a 69, viz obr. 34, z videosekvence Foreman – 396 makrobloků na snímek. Snímek 66 je referenční a snímek 69 je předpovídaný, jedná se o předpověď dopřednou.



Obr. 34. Testovací snímky ze sekvence Foreman : 66 (vlevo) a 69 (vpravo).

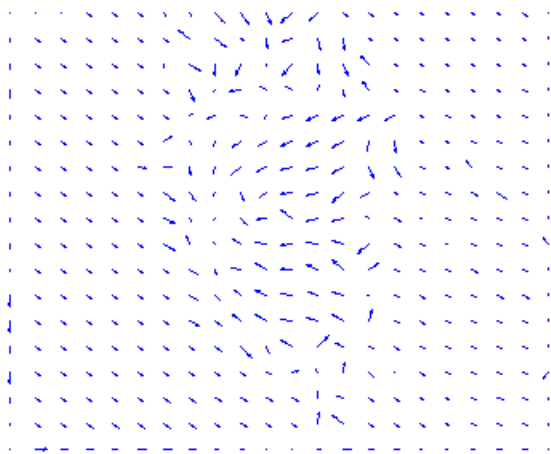
3.2.5.1 Vyhledávání s přesností celého pixelu

Rozdíl testovacích snímků bez kompenzace a s kompenzací, za použití plného vyhledávání s velikostí vyhledávacího okna ± 7 pixelů, lze vidět na obr. 35. Hodnotu rozdílového snímku lze vyjádřit hodnotou SAE, a pro snímek bez kompenzace je 1170701 [-], pro snímek s kompenzací je hodnota SAE více jak 2x menší a to 428255 [-], viz tabulka 6.



Obr. 35. Rozdílový snímek – vlevo bez kompenzace, vpravo s kompenzací.

Na obr. 36 je graficky znázorněno pole vektorů, po jeho aplikaci (provedení pohybové kompenzace) na snímek 66, dostaneme kompenzovaný snímek 66. Odečtením rozdílového snímku od kompenzovaného snímku 66 získáme předpovídaný snímek 69.



Obr. 36. Pole vektorů (vlevo) a kompenzovaný snímek 66 (vpravo).

Závislost SAE na velikosti vyhledávacího okna, viz obr. 37, ukazuje, že plné vyhledávání vždy lokalizuje globální minimum celé prohledávané oblasti, zatímco N-krokové a logaritmické vyhledávání mají díky pevně danému kroku tendenci se zachytit v lokálním minimu prohledávané oblasti, viz obr. 40. K zachycení v lokálním minimu přispívá i charakter jednotlivých snímků. V případě malého pohybu obsahu snímku a zvolení velké prohledávací oblasti je díky odpovídající velikosti kroku globální maximum přeskočeno už v prvním kroku. Z grafu vyplývá, že se situace může v takových případech dokonce zhoršovat, viz oblast +/- (10 až 15) pixelů, kdy se SAE zvyšuje.

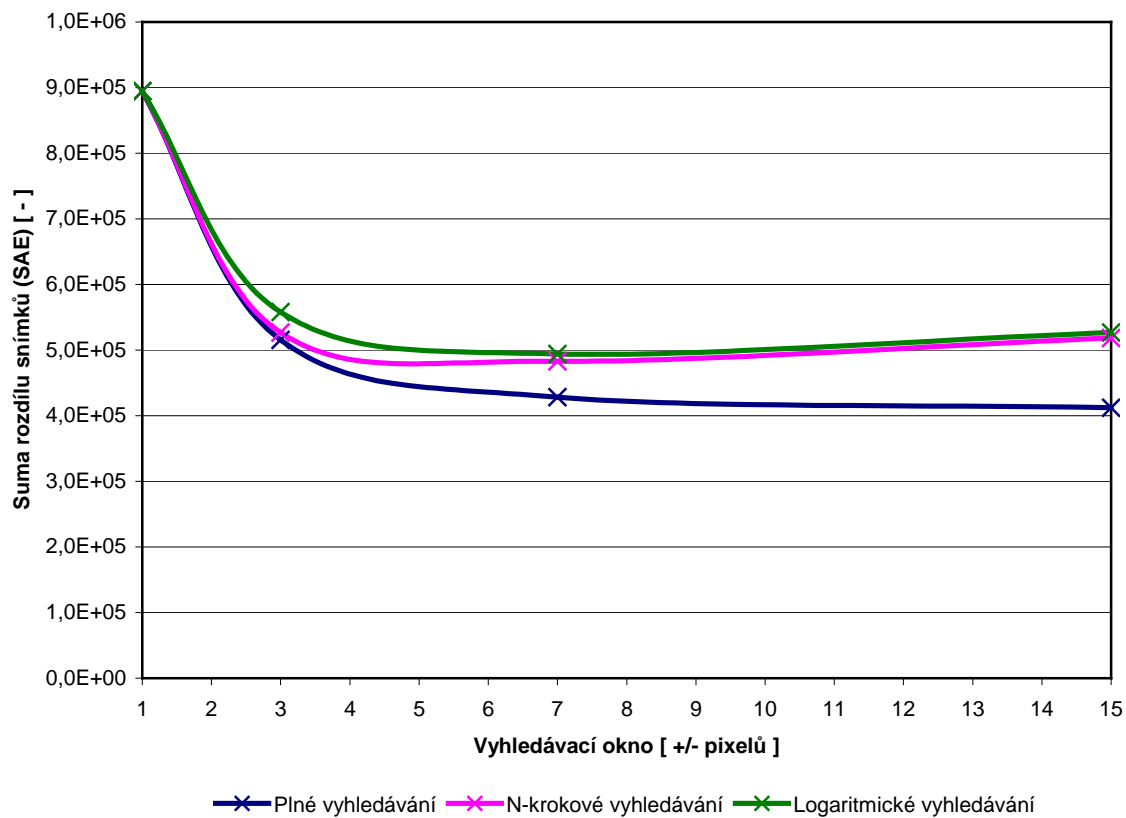
Doba zpracování, viz obr. 38, je u všech algoritmů přibližně stejná do oblasti +/- (3 až 4) pixelů. Od této hranice doba zpracování u plného vyhledávání se zvyšující se velikostí vyhledávacího okna, lineárně roste, do cca 2,6 sekund. Jak N-krokové, tak logaritmické vyhledávání se drží po celý rozsah do 0,4 sekund, přičemž N-krokové je o něco málo rychlejší. Tyto výsledky závisí na hardwaru a na jiných konfiguracích se mohou lišit. Počet srovnávacích operací je nejmenší pro logaritmické vyhledávání, což se ovšem projevuje i na sumě rozdílů snímků, kde je naopak logaritmické vyhledávání nejhorské.

Závislost sumy pohybových vektorů na velikosti vyhledávacího okna, viz obr. 39, inverzně kopíruje průběh závislosti SAE, až na drobné výjimky, viz zachycení v lokálním minimu. Obecně platí (pokud uvažujeme dva výrazně se od sebe lišící snímky), že čím menší rozdíl snímků, tím větší suma pohybových vektorů. Z grafu lze vyvodit, že v případě těchto dvou snímků je N-krokové vyhledávání náchylné na větší velikost pohybových vektorů při větší velikosti vyhledávacího okna, v sumě je to o cca 800 více než u ostatních.

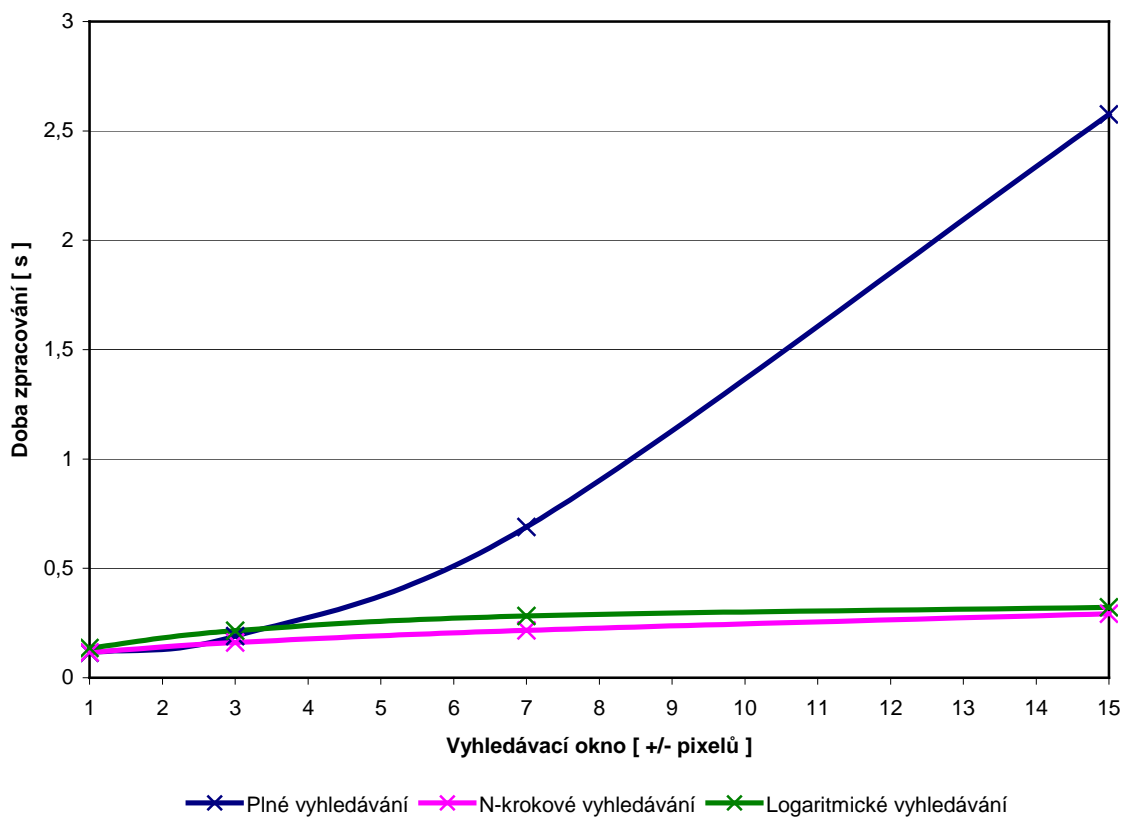
Vyhledávací okno [+/- pixelů]	Suma rozdílů snímků [v tisících]			Suma pohybových vektorů [-]			Srovnávací operace [-]		
	FS	NSS	LS	FS	NSS	LS	FS	NSS	LS
1	894	894	894	617	617	617	3328	3328	3328
3	515	527	558	1644	1620	1421	17760	6665	6669
7	428	482	494	2139	2387	1902	80896	10014	8983
15	412	518	526	2493	3323	2389	344256	13390	10182

Pozn. : FS – Plné vyhledávání
 NSS – N-Krokové vyhledávání
 LS – Logaritmické vyhledávání

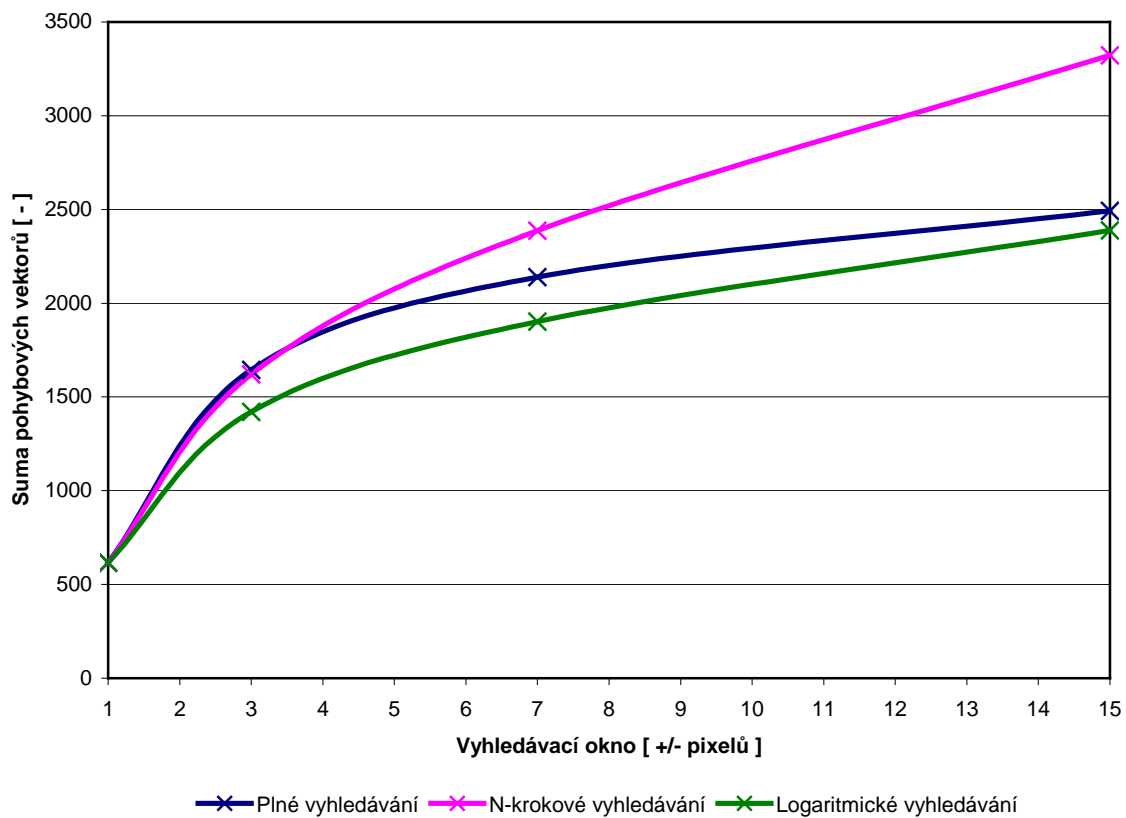
Tab. 6. Srovnání výkonnosti vyhledávacích algoritmů.



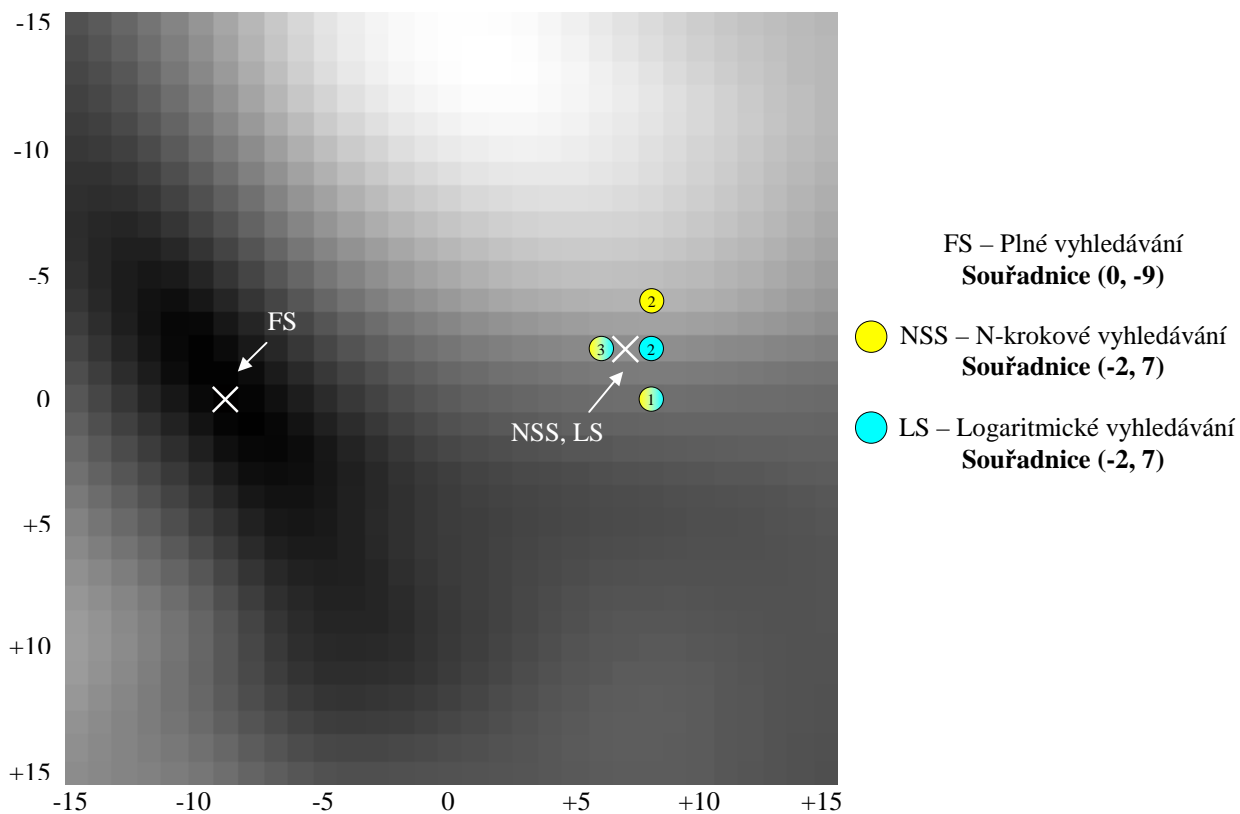
Obr. 37. Závislost SAE na velikosti vyhledávacího okna.



Obr. 38. Závislost doby zpracování na velikosti vyhledávacího okna.



Obr. 39. Závislost sumy pohybových vektorů na velikosti vyhledávacího okna.



Obr. 40. Mapa SAE pro předpověď makrobloku 189 snímku 69.

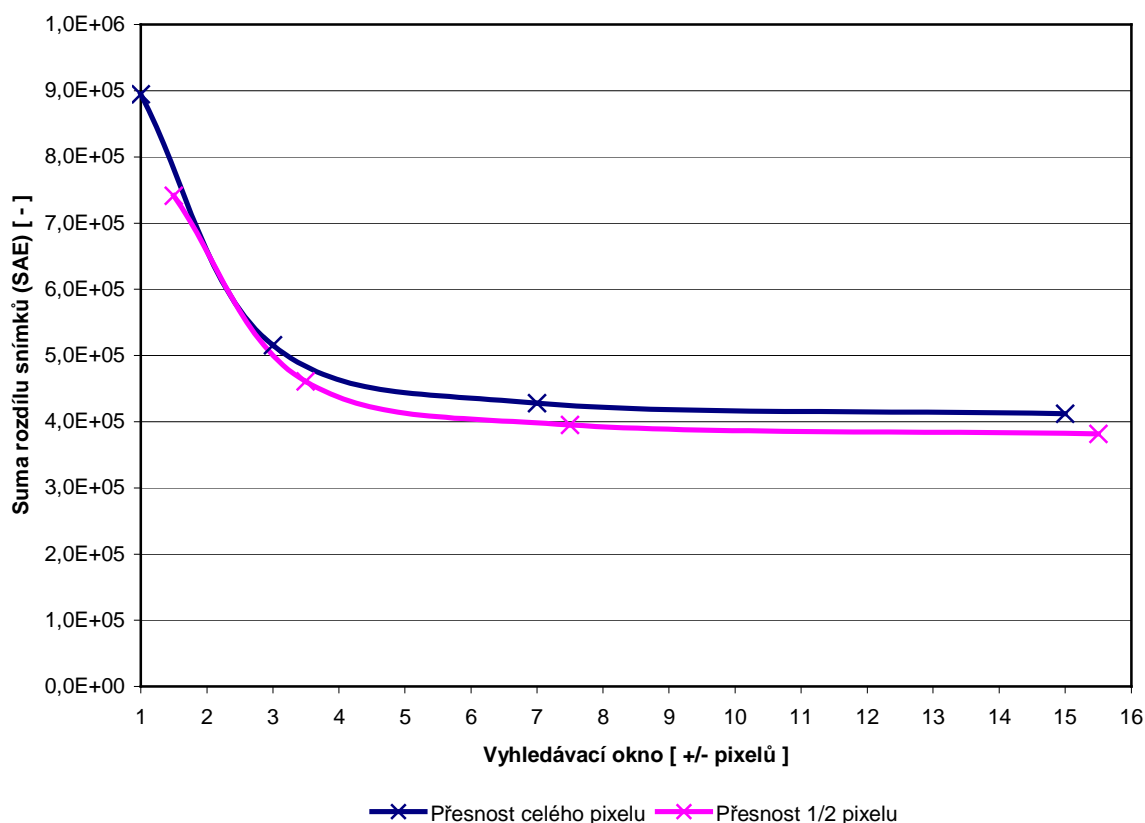
3.2.5.2 Vyhledávání s přesností 1/2 pixelu

Vyhledávání s přesností 1/2 snižuje, podle očekávání, sumu rozdílů snímků. V tomto případě, viz obr. 41, je rozdíl mezi přesností s celým pixelem a 1/2 pixelu znatelný od velikosti vyhledávacího okna cca +/- (3 až 4) pixelů. Od této hranice je hodnota SAE o zhruba 30000 až 50000 menší pro přesnost 1/2 pixelu. Doba zpracování, viz obr. 42, je pro obě přesnosti pro vyhledávací okno +/- (7 až 15) pixelů stejná, při menším okně je vyhledávání s přesností 1/2 pixelu pomalejší. Větší rozdíly by se projevíly například u HD snímků. Na obr. 43 je vynesena závislost sumy pohybových vektorů, díky velikosti vektorů v desetinných hodnotách není nárůst celkové sumy tak markantní jako u celých pixelů. V tabulce 7 je pro porovnání uvedena suma srovnávacích operací, přesnost 1/2 pixelu má o 3564 srovnávacích operací u každé velikosti vyhledávacího okna navíc.

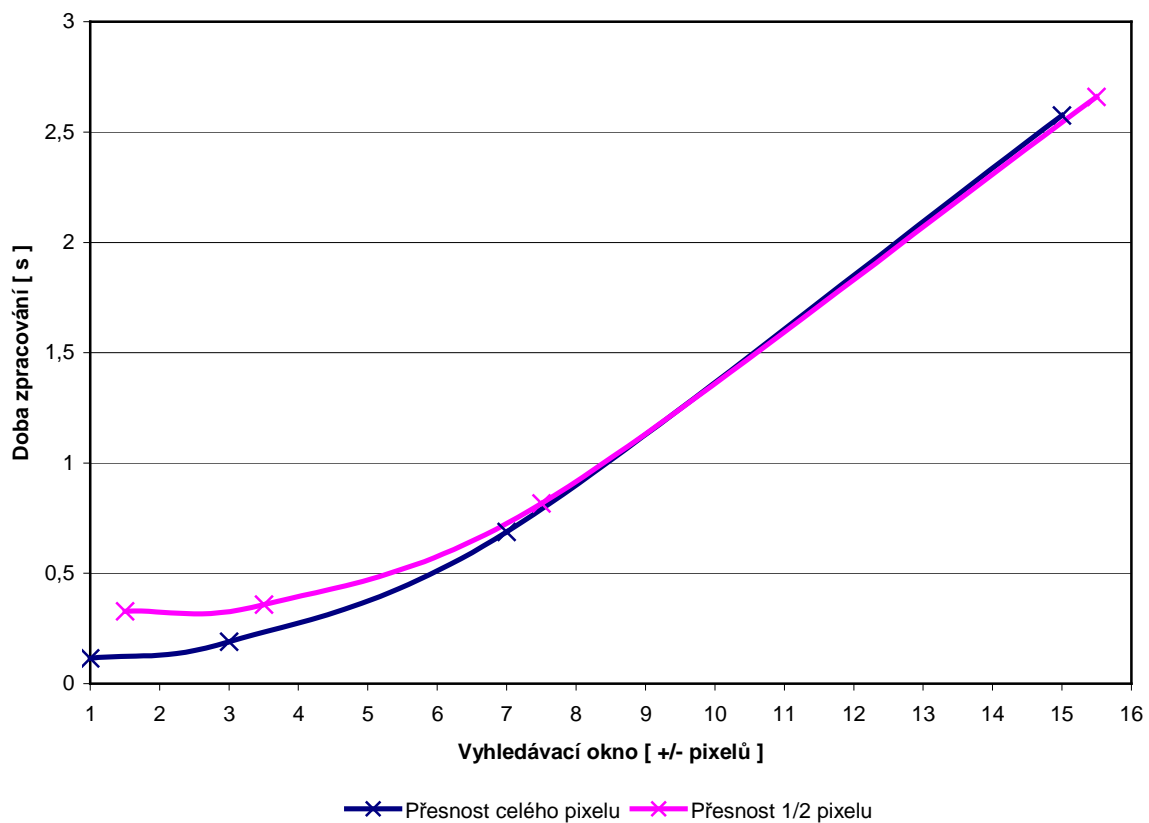
Vyhledávací okno [+/- pixelů]	Suma rozdílů snímků [v tisících]		Suma pohybových vektorů [-]		Srovnávací operace [-]	
	Fpx	1/2px	Fpx	1/2px	Fpx	1/2px
1	894	741	617	904	3328	6892
3	515	460	1644	1744	17760	21324
7	428	395	2139	2170,5	80896	84460
15	412	381	2493	2507,5	344256	347820

Pozn. : Fpx – Přesnost celého pixelu
1/2px – Přesnost 1/2 pixelu – vyhledávací okno +/- <0,5 až 15,5> pixelů

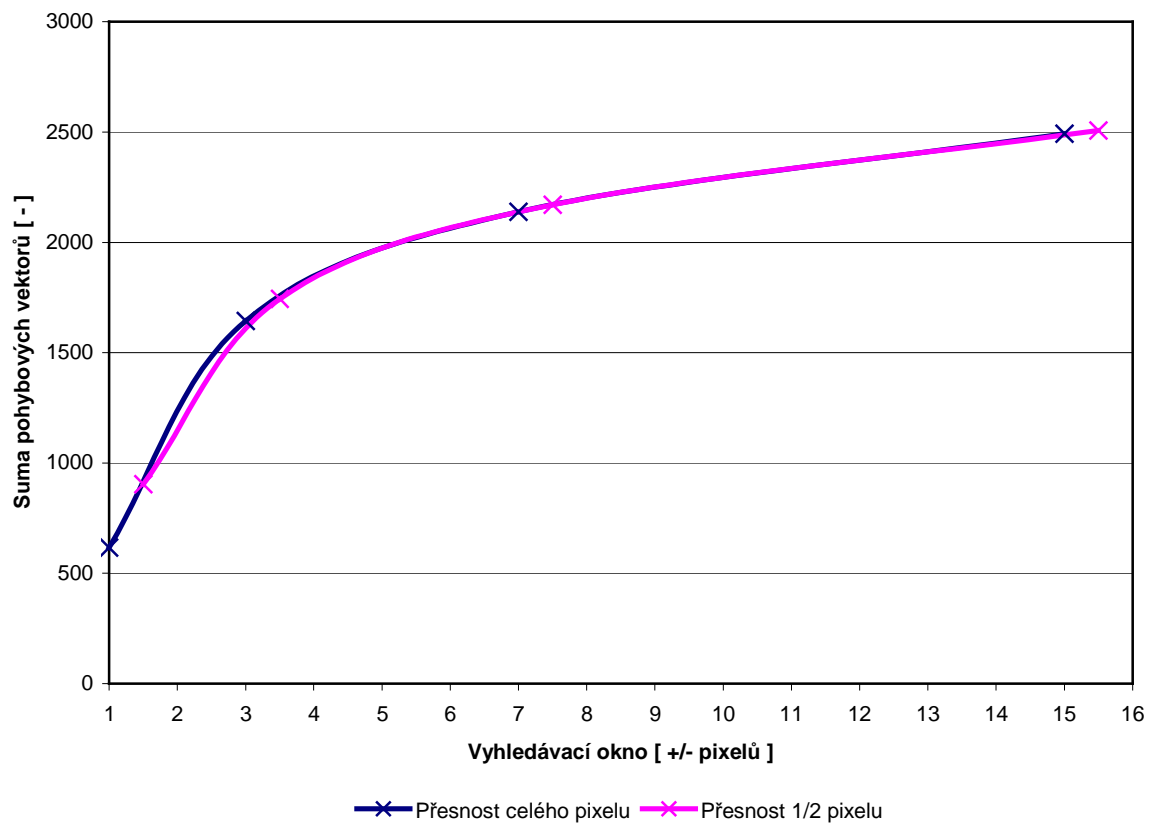
Tab. 7. Srovnání výkonnosti plného vyhledávání pro různou přesnost vyhledávání.



Obr. 41. Závislost SAE na velikosti vyhledávacího okna.



Obr. 42. Závislost doby zpracování na velikosti vyhledávacího okna.

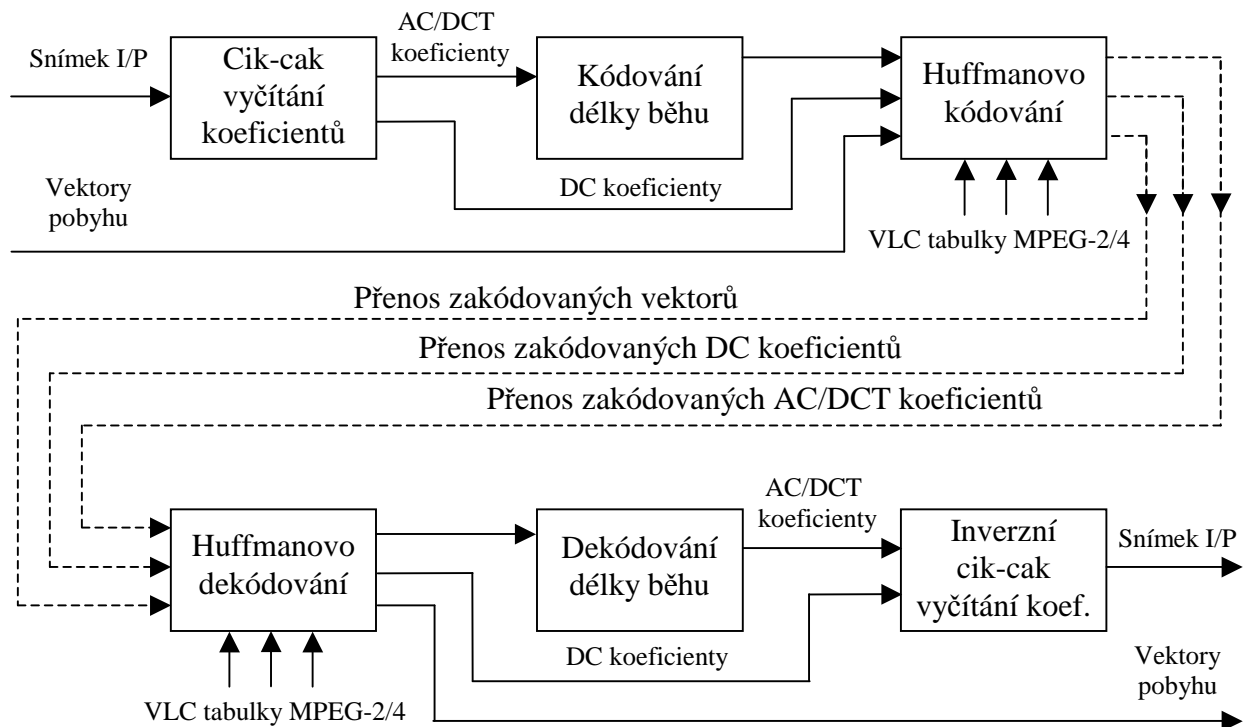


Obr. 43. Závislost sumy pohybových vektorů na velikosti vyhledávacího okna.

3.3 Kódování s proměnnou délkou slova

Zdrojový kód pro kódování s proměnnou délkou slova odpovídá blokovému schématu na obr. 44. První tři bloky tvoří část kodéru, část dekodéru je tvořena stejných počtem bloků, ovšem s inverzními funkcemi.

U snímku typu I se AC a DC koeficienty zpracovávají odděleně. Každý blok 8x8 DCT koeficientů se cik-cak vyčítáním převede na vektor 64 hodnot. Z toho první hodnota, odpovídající DC koeficientu, se kóduje přímo Huffmanovým kódováním. Na zbývajících 63 hodnot, odpovídajících AC koeficientům, se aplikuje kódování délky běhu a tento výsledek je zpracován Huffmanovým kódováním. U snímku typu P nebo B se blok DCT koeficientů převede na vektor 64 hodnot. Ten je celý zpracován kódováním délky běhu a následným Huffmanovým kódováním. Vektory pohybu se kódují přímo Huffmanovým kódováním.



Obr. 44. Blokové schéma kódování s proměnnou délkou slova.

3.3.1 Kódování délky běhu

Kódování délky běhu má za cíl zredukovat délku vektoru kvantovaných AC/DCT koeficientů za pomoci čítání běhu nul a následného vytvoření skupin složených z délky běhu nul a následující nenulové hodnoty. Pokud do konce vektoru zbývají samé nuly, je za poslední platnou skupinu zapsán identifikátor EoB (**E**nd of **B**lock). To je ilustrováno následujícím úryvkem kódu:

```

for i=1:size(data,1)                               % Smyčka zpracování vektorů.
    while y<=delka_bloku;                           % Smyčka zpracování vektoru.
        if sum(do konce bloku)==0 % Podmínka pro zápis NaN (EoB).

            { % Zápis identifikátoru NaN (EoB). }
        else
            while blok(y)==0; % Smyčka čítání běhu nul.

                { % Inkrementace čítače nul.
                  % Zápis skupiny: běh nul | hodnota. }
            
```

Zdrojový kód RLE (m-file *koder_RLE.m*) je tvořen cyklem *for* zajišťujícím načítání vektorů AC/DCT koeficientů. Smyčka *while* obstarává zpracování vybraného vektoru, délka vektoru je 63 pro AC koeficienty snímku I nebo 64 pro DCT koeficienty snímků P/B. Podmínka *if* kontroluje, zda nejsou do konce vektoru pouze nulové hodnoty. Pokud je podmínka splněná, je zapsán identifikátor EoB, respektive NaN (Not a Number). Pokud není splněna, následuje čítání běhu nul pomocí smyčky *while* a následný zápis skupiny.

3.3.2 Huffmanovo kódování

Huffmanovo kódování lze u standardu MPEG-2 rozdělit podle principu zpracování na kódování DC koeficientů a kódování AC/DCT koeficientů. VLC tabulky pro DC koeficienty obsahují VLC kódy, které definují počet bitů, jimiž lze vyjádřit hodnotu DC koeficientu. VLC tabulky pro AC/DCT koeficienty a vektory pohybu obsahují VLC kódy přímo pro hodnoty, které se vyskytují ve vektoru dat.

3.3.2.1 Kódování DC koeficientů

Zdrojový kód VLC pro DC koeficienty (m-file *koder_VLC_dc.m*) obsahuje cyklus *for* pro načítání DC koeficientů. Nulová hodnota DC koeficientu je kódována přímo VLC kódem, k čemuž slouží podmínka *if*, která testuje nulovou hodnotu. V opačném případě dochází za pomoci cyklu *for* k prohledávání VLC tabulky, při shodě je do vektoru zapsán VLC kód spolu s binárně vyjádřenou hodnotou DC koeficientu.

```

for i=1:length(data);           % Smyčka načítání DC koeficientů.
    if data(i)==0;              % Podmínka pro nulovou hodnotu.

        { % Zapsání VLC kódu do vektoru. }
    else
        for n=1:12;            % Smyčka prohledávání VLC tabulky.
            if isequal();      % Testování shody s VLC tabulkou.

                { % Zapsání kódu do vektoru. }
            end
        end
    end
end

```

3.3.2.2 Kódování AC/DCT koeficientů

Zdrojový kód VLC pro AC/DCT koeficienty (m-file *koder_VLC.m*) obsahuje smyčku *while* pro načítání RLE kódovaných dat. Podmínka *if* zajišťuje detekci hodnoty NaN (EoB), jinak se načte skupina ve formátu [běh nul | hodnota]. Následuje cyklus *for* pro prohledávání VLC tabulky, pokud je VLC kód nalezen, je zapsán do kódovaných dat. Pokud není pro skupinu VLC kód definován, vygeneruje se nový VLC kód ve formátu [únikový kód (6b) | binárně vyjádřená délka běhu (6b) | binárně vyjádřená hodnota (8b)], celkově tedy 20 bitů.

```

while i<=length(data);        % Smyčka procházení vektoru RLE dat.
    if isnan(data(i));        % Podmínka pro detekci NaN (EoB).
        { % Uložení řetězce NaN do skupiny. }
    else
        { % Načtení skupiny: běh nul | hodnota. }
    end;
    for n=1:113;              % Smyčka prohledávání VLC tabulky.
        if isequal();        % Testování shody s VLC tabulkou.

            { % Zápis VLC kódu do vektoru. }
        end
    end;
    elseif n==113 % Není definován VLC kód skupiny.
        { % Vygenerování a zápis VLC kódu do vektoru. }
    end
end

```

3.3.3 Dosažené výsledky III

První část výsledků reprezentuje účinnost kódování délky běhu, druhá část účinnost Huffmanova kódování a třetí část účinnost kódování jako celku. Pokud není uvedeno jinak, jsou dosažené výsledky získány ze snímků 1 až 50 komprimované sekvence Foreman s nastavením komprimace: vzorkování 4:2:0, plné vyhledávání, velikost vyhledávacího okna +/- 7 pixelů, přesnost vyhledávání na celé pixely, kvantizační parametr Qp 4, kvantizační tabulky MPEG-2, formát skupiny snímků IPPP PPPP PPPP.

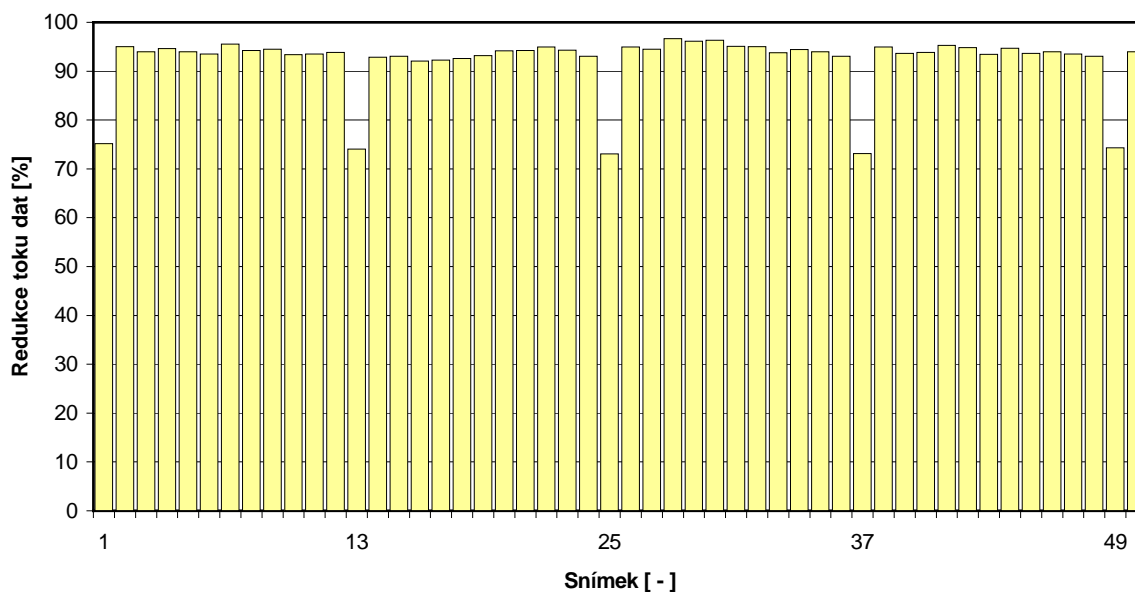
Snímek	Data	Počet bloků 8x8	Délka bloku	Počet hodnot	Bitů na hodnotu (reference)
P	DCT	2376	64	152064	8 až 10
	MV	-	2x396	792	7
I	AC	2376	63	149688	8 až 10
	DC Y	-	1584	1584	9
	DC Cb	-	396	396	9
	DC Cr	-	396	396	9

Tab. 8. Informace o snímcích komprimované videosekvence.

Tabulka 8 obsahuje sloupec bitových referencí pro výpočet účinnosti Huffmanova kódování. Referenční hodnota pro AC/DCT koeficienty je určována dynamicky na základě jejich velikosti. V bitové referenci je započítán i jeden bit pro vyjádření znaménka.

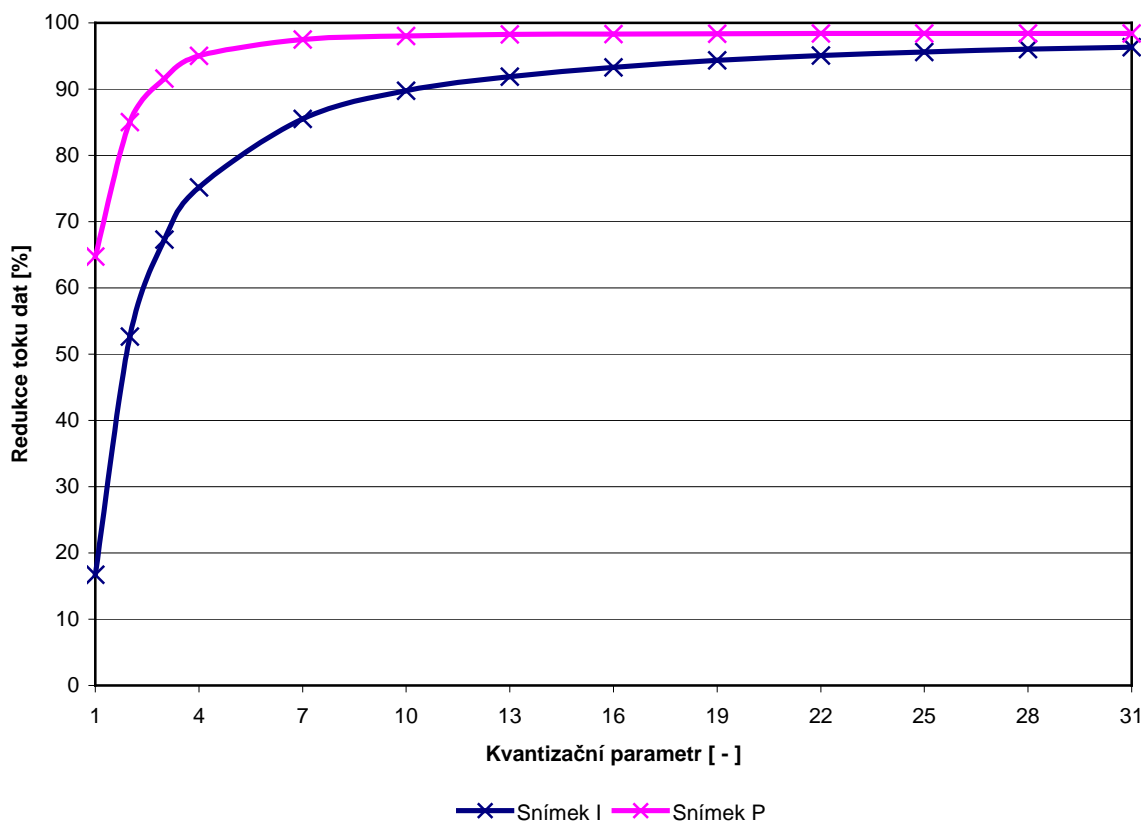
3.3.3.1 Kódování délky běhu

Na obr. 45 je zobrazena redukce toku AC/DCT koeficientů u jednotlivých snímků komprimované sekvence. Redukce toku dat se vztahuje k velikosti AC/DCT dat snímku I/P, uvedené v tabulce 8. Zobrazená čísla na ose x udávají pozici snímku I. Pro toto nastavení komprimace je průměrná redukce toku dat pomocí RLE 74% u snímku I, u snímku P je pak o cca 20% větší, tedy 94% - tzn. zredukování toku dat o toto procento. Z těchto hodnot lze usoudit, že kódování délky běhu má výrazný vliv na celkovou redukci bitového toku.



Obr. 45. RLE - Redukce toku AC/DCT koeficientů.

Obr. 46 prezentuje závislost redukce toku AC/DCT koeficientů na kvantizačním parametru. Účinnost kódování je u snímku P (65% a více) znatelně větší než u snímku I (17% a více), především u menších hodnot kvantizačního parametru. Větší kvantizační parametr má za následek větší redukci nenulových hodnot v kvantovaných datech. Větší výskyt nulových hodnot znamená výrazné zredukování toku dat, jelikož se sekvence nul nahradí hodnotou, která vyjadřuje počet nul.



Obr. 46. RLE - Závislost redukce toku AC/DCT koeficientů na Q_p .

3.3.3.2 Huffmanovo kódování

Jako referenční hodnota, ke které se vztahuje výpočet účinnosti Huffmanova kódování, byla použita hodnota binárně vyjádřeného AC/DC/DCT/MV toku dat. Pro DC koeficienty 9 bitů, pro vektory pohybu 7 bitů a pro AC/DCT koeficienty se na základě velikosti koeficientů reference měnila v určitém rozsahu, viz tabulka 8. S DC koeficienty a vektory pohybu se počítalo přímo, s AC/DCT koeficienty až po kódování délky běhu.

Celková účinnost kódování DC koeficientů složky Y/Cb/Cr snímku I, pohybových vektorů u snímku P a AC/DCT koeficientů snímku I/P byla vypočítána na základě následujících rovnic

$$\text{huff_DC} = \left(1 - \frac{\text{DC_huff} + \text{DC_sign}}{\text{DC_data} \cdot 9b} \right) \cdot 100, \quad (14)$$

$$\text{huff_AC/DCT} = \left(1 - \frac{\text{AC/DCT_vlc} + \text{AC/DCT_sign}}{\text{AC/DCT_data} \cdot (\text{dynamicka_reference})} \right) \cdot 100, \quad (15)$$

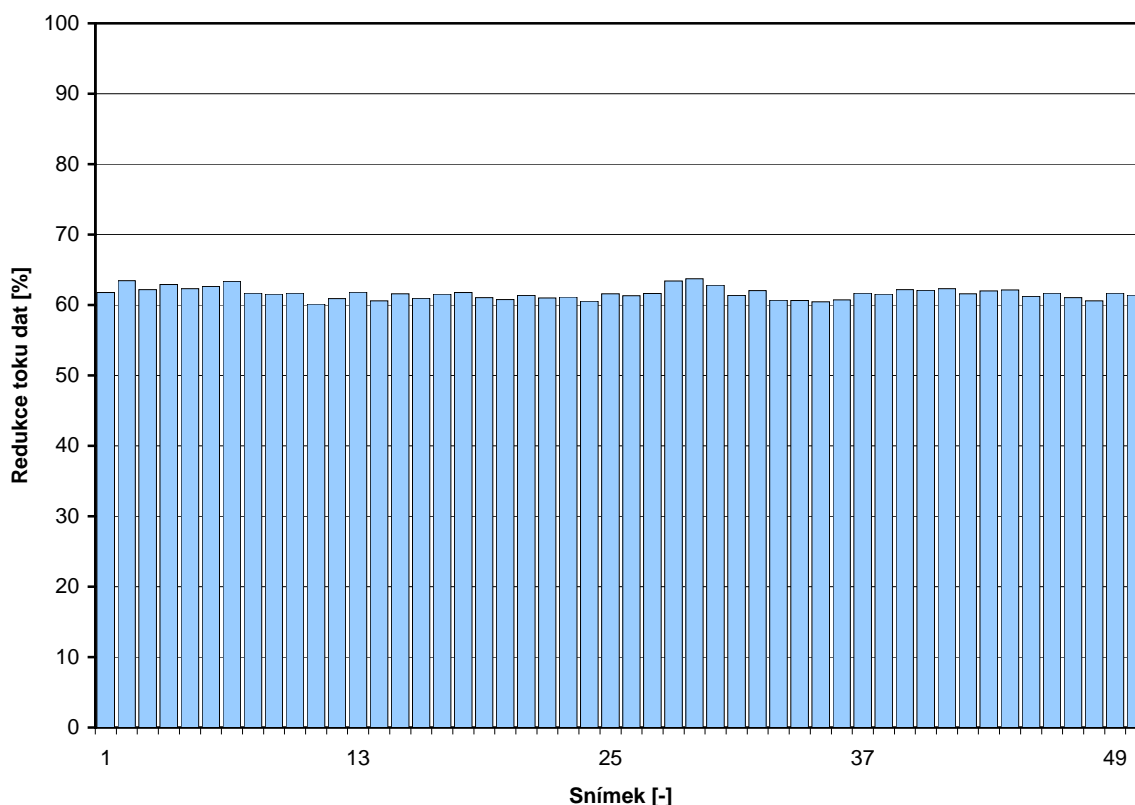
$$\text{huff_MV} = \left(1 - \frac{MV_huff}{MV_data \cdot 7b} \right) \cdot 100, \quad (16)$$

kde huff_DC , huff_AC/DCT , huff_MV jsou účinnosti kódování v procentech, AC/DCT_vlc jsou data zpracována RLE a Huffmanovým kódováním, DC_Huff , MV_huff jsou data zpracována pouze Huffmanovým kódováním, DC_sign , AC/DCT_sign jsou vektory znaménkových bitů a DC_data , AC/DCT_data , MV_data jsou velikosti příslušných dat.

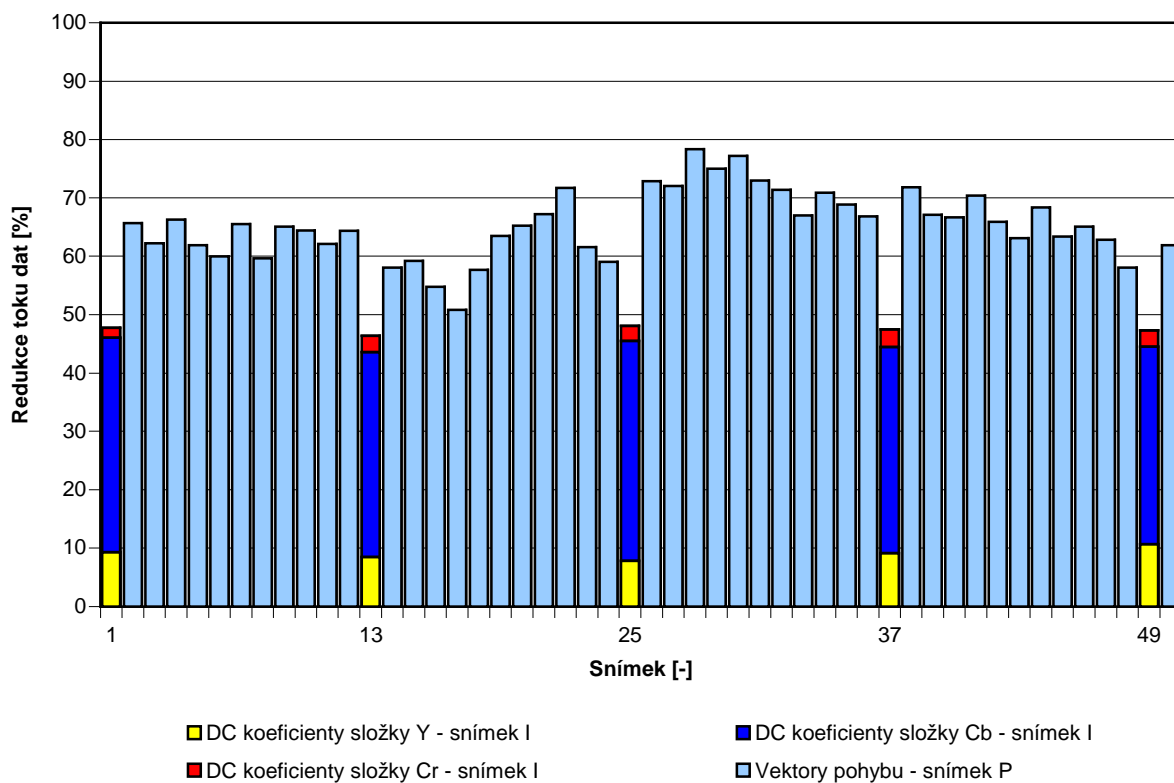
Na obr. 47 a 48 je barevně vyznačeno o kolik procent jsou zredukovány jednotlivé toky AC/DC/DCT koeficientů a vektorů pohybu u jednotlivých snímků komprimované videosekvence. Průměrná redukce toku dat AC/DCT koeficientů je o 62%, vektorů pohybu o 64%, redukce toku dat DC koeficientů složky Y je o 9% a složky Cb (Cr) o 45% (47%).

Na obr. 49 je zobrazena závislost redukce toků DC koeficientů složky Y/Cb/Cr jednoho snímku I na kvantizačním parametru. Redukce toku DC koeficientů není závislá na velikosti kvantizačního parametru, jelikož se DC koeficienty kvantují vždy stejnou hodnotou, bez vstupu kvantizačního parametru do výpočtu. Redukce toku dat je tedy stále konstantní.

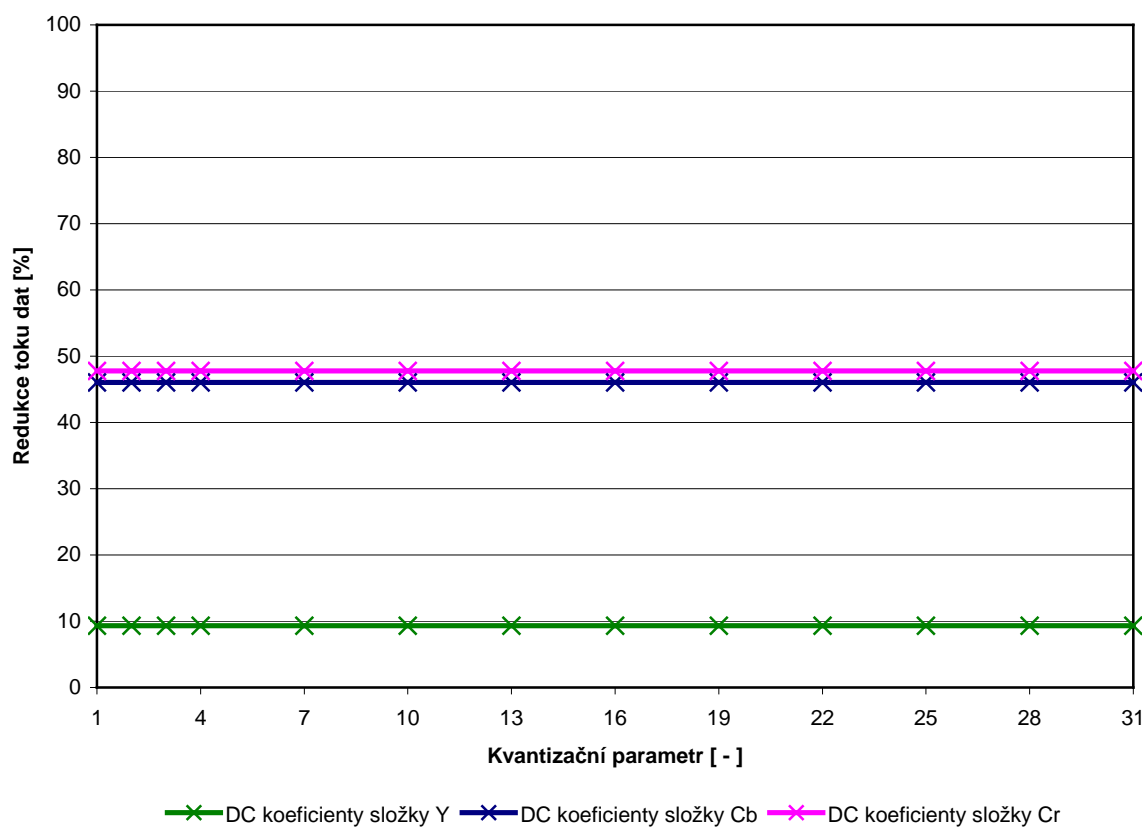
Průběhy závislostí redukce toku AC/DCT koeficientů snímku I/P na kvantizačním parametru jsou vidět na obr. 50. Prvotní klesání účinnosti u snímku I je zapříčiněno změnou bitové referenční hodnoty. Zbytek průběhu je téměř konstantní a průměrná hodnota redukce je 62%. Účinnost u snímku P se s rostoucím kvantizačním parametrem zvyšuje, až do hodnoty cca 75%. Lepší průběhy závislostí získáme při použití VLC tabulek generovaných na základě aktuálních RLE kódovaných dat. Redukce se sice zvýší, ale nutnost přenášet generované VLC tabulky na stranu dekodéru je značně nevýhodné z důvodu paměťové náročnosti.



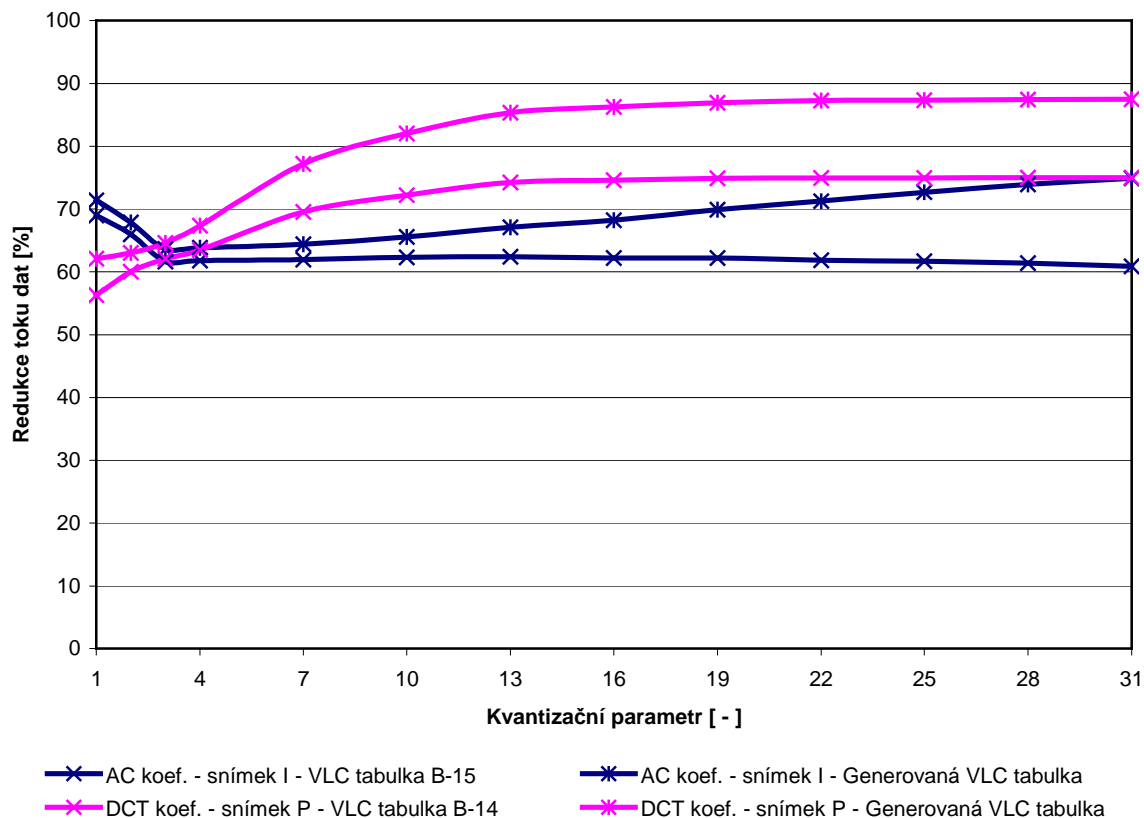
Obr. 47. Huffmanovo kódování - Redukce toku AC/DCT koeficientů.



Obr. 48. Huffmanovo kódování - Redukce toku DC koeficientů a vektorů pohybu.



Obr. 49. Huffmanovo kódování - Závislost redukce toku DC koef. snímku I na Q_p .



Obr. 50. Huffmanovo kódování - Závislost redukce toků AC/DCT koeficientů na Qp.

3.3.3.3 Kódování s proměnnou délkou slova

Obr. 51 a 52 znázorňují celkovou účinnost kódování s proměnnou délkou slova. První graf prezentuje redukci toku dat u snímku I/P v závislosti na kvantizačním parametru. Účinnost kódování je u snímku P ztelně větší než u I snímku, především v rozsahu hodnot 1 až 7 kvantizačního parametru. S rostoucím kvantizačním parametrem se opět zvyšuje účinnost kódování a to až do hodnoty 97% pro snímek I a 99% pro snímek P. Na druhém grafu je zobrazena redukce toku dat pro jednotlivé snímky komprimované videosekvence. Průměrná redukce dat činí 96%.

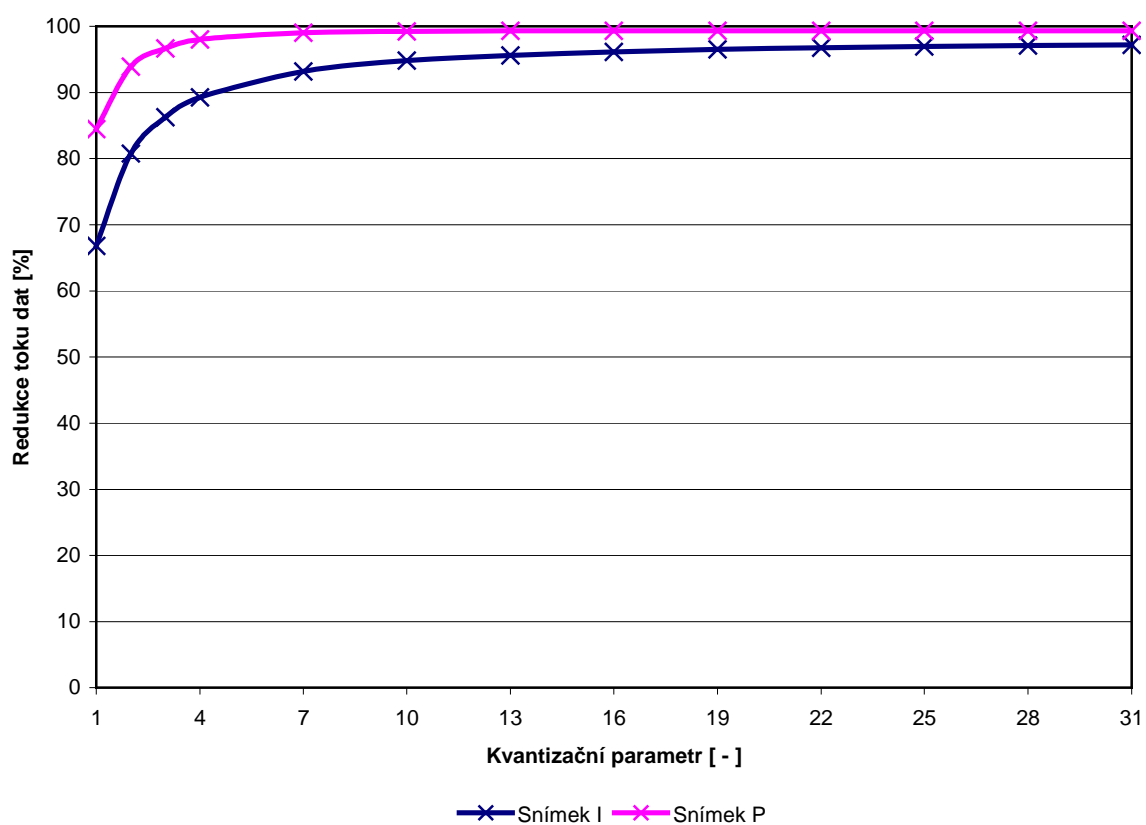
Jako reference slouží nekomprimovaný snímek o vzorkování 4:2:0. Velikost dat jednotlivých komponent snímku I/P jsou uvedeny v tabulce 8. Celková účinnost pro snímek I/P byla vypočítána na základě rovnic

$$rtd_I = \left(1 - \frac{AC_vlc + AC_sign + DC_huff + DC_sign}{(AC_data + DC_data) \cdot 8b} \right) \cdot 100, \quad (17)$$

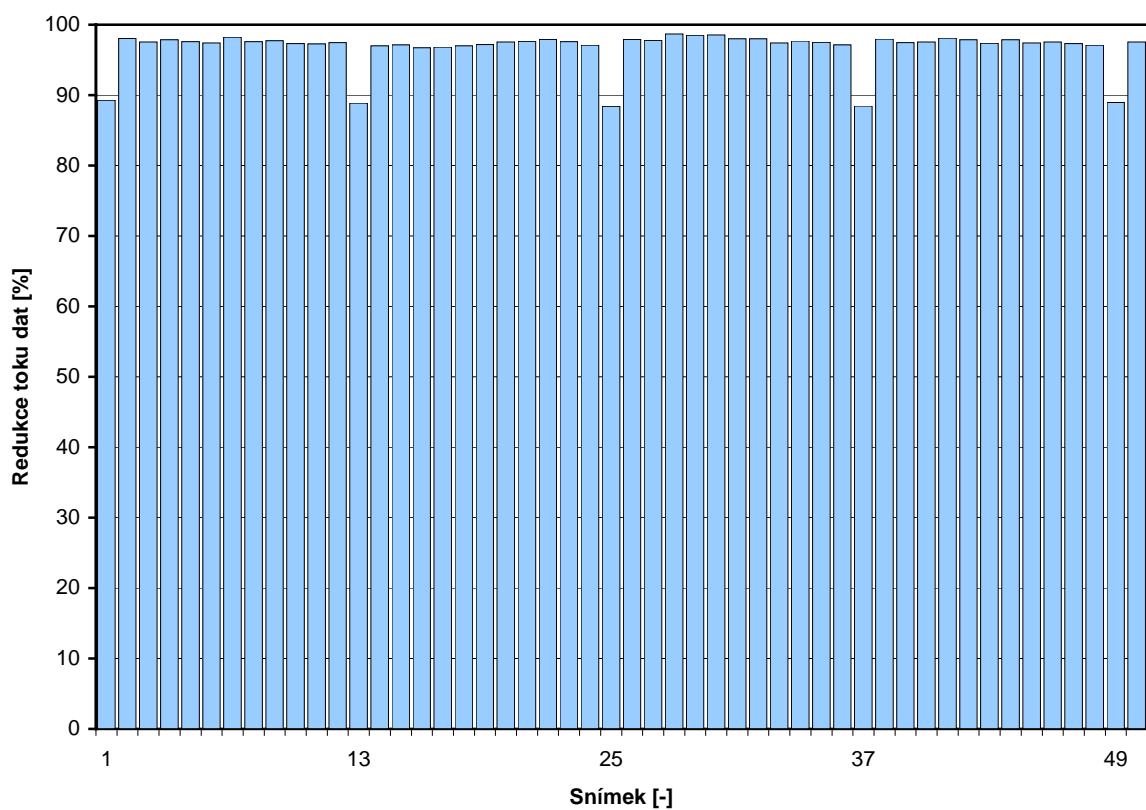
$$rtd_P = \left(1 - \frac{DCT_vlc + DCT_sign + MV_huff}{DCT_data \cdot 8b + MV_data \cdot 7b} \right) \cdot 100, \quad (18)$$

kde rtd_I , rtd_P jsou redukce toků dat snímků I a P v procentech, AC_vlc , DCT_vlc jsou data zpracována RLE a Huffmanovým kódováním, DC_Huff , MV_huff jsou data zpracována pouze Huffmanovým kódováním, DC_sign , AC_sign , DCT_sign jsou vektory znaménkových bitů a

AC_data, DC_data, DCT_data, MV_data jsou velikosti příslušných dat.



Obr. 51. VLC - Závislost redukce toku dat snímku I/P na Q_p .








Obr. 52. VLC - Redukce toku dat pro 50 snímků komprimované sekvence.

3.4 Výkonnost kodeku

Zpracování obrazu, v takovém rozsahu jako je simulace video kodeku, je v prostředí Matlab velice náročné na paměť RAM a také na dobu zpracování. Video kodek je proto primárně určen pro zpracování sekvencí s malým rozlišením a menším počtem snímku, což ale na simulaci nástrojů bohatě postačuje. Doporučená velikost RAM je alespoň 4GB, což také znamená nutnost použití 64-bitového operačního systému. Pro zrychlení komprimace a dekomprimace je vhodný vícejádrový procesor (optimálně 4 jádra), jelikož prostředí Matlab dovoluje zapnutí podpory zpracování s více jádry. Používání kodeku v jiné verzi, než je Matlab 2009 není doporučeno z důvodu možného pádu díky chybám v syntaxi v případě nižší verze nebo nepřítomnosti využívaných funkcí (jsou odstraněny) ve vyšších verzích.

Výkonnost jednotlivých nástrojů a jejich nastavení byla otestována na skupině testovacích videosekvencí [20] s různým obsahem a charakterem obrazu, tabulka 9 obsahuje informace o sekvencích a jejich podrobnější popis.

Sekvence	Informace	Náhled	Popis
Foreman	352x288 30 snímků za sekundu 300 snímků		Nahodilý a rychlý pohyb postavy, pohyb kamery → pozadí není statické, konec sekvence obsahuje rychlou změnu scény.
Container	352x288 30 snímků za sekundu 300 snímků		Rovnoměrný a pomalý pohyb objektu po dobu trvání celé sekvence, stejně tak pozadí (voda) vykazuje rovnoměrný a pomalý pohyb.
Mobile	352x288 30 snímků za sekundu 300 snímků		Středně rychlý a rovnoměrný pohyb několika objektů (vlak, kalendář, míč), pozadí se pomalu pohybuje zleva doprava.
News	352x288 30 snímků za sekundu 300 snímků		Statické pozadí, rychlý a nahodilý pohyb uprostřed scény (balet), mírný pohyb postav (ústa).
Salesman	352x288 30 snímků za sekundu 449 snímků		Statické pozadí, scéna obsahuje středně rychlý a nahodilý pohyb postavy (ruce, hlava) a objektu.

Tab. 9. Testovací videosekvence.

3.4.1 Výkonnost nástrojů odhadu a kompenzace pohybu

V této části jsou uvedeny výsledky poskytující přehled o výkonnosti (závislost PSNR na bitové rychlosti) základních nástrojů (P, B snímky) pro různý obsah a charakter scény. Každý graf obsahuje pětici křivek:

- Pouze I snímky (bez odhadu a kompenzace pohybu),
- P snímky, vyhledávání s přesností celého pixelu (Full pel),
- P snímky, vyhledávání s přesností ½ pixelu (Half pel),
- B snímky, vyhledávání s přesností celého pixelu (Full pel) a
- B snímky, vyhledávání s přesností ½ pixelu (Half pel).

Výsledky byly získány při nastavení komprimace: vzorkování 4:2:0, kvantizační tabulky MPEG-2, plné vyhledávání, velikost vyhledávacího okna +/- 7 pixelů, srovnávací kritérium SAE, vyhledávání v prostorové doméně. Při zapnutí podpory B snímků byla zvolena možnost tří po sobě jdoucích B snímků.

Z globálního pohledu, bez rozdílu charakteru obrazu, lze z následujících grafů, viz obr. 54 až 58, vyvodit výrazné zvýšení kvality dekomprimované sekvence při zapnutí podpory P snímků oproti přenosu pouze I snímků. Jednosměrná předpověď obrazu je tedy jeden z klíčových nástrojů komprimace obrazu. Lepších výsledků, této předpovědi, je dále dosaženo vyhledáváním s přesností ½ pixelu. Zvýšení kvality není ale tak dramatické jako u P snímků samotných. Další ušetření velikosti bitového toku při zachování kvality je dosaženo obousměrnou předpovědí obrazu - B snímky a opět možností přesnějšího vyhledávání.

Z pohledu jednotlivých sekvencí je jasné, že rozdíl mezi P a B snímky se zvyšuje spolu s dynamičtějším obsahem obrazu. U sekvence News, kde je scéna z velké části statická, pouze s malou oblastí rychlého pohybu, je rozdíl velmi malý. O něco větší rozdíl ve výkonnosti je u sekvence Container s rovnoměrným a pomalým pohybem. U těchto sekvencí je také dosažena vysoká kvalita při výrazně menší bitové rychlosti než u obsahově výrazně pohyblivějších sekvencí. Výsledky sekvence Salesman, která obsahuje střední pohyb přes nemalou část obrazu, pokračují v trendu zvětšujících se rozdílů mezi P a B snímky. Následuje skupina velmi dynamických sekvencí (Foreman, Mobile). Z těchto průběhů je možné vypočítat jistou linearitu křivek u rovnoměrného pohybu (Mobile) oproti nahodilému pohybu (Foreman).

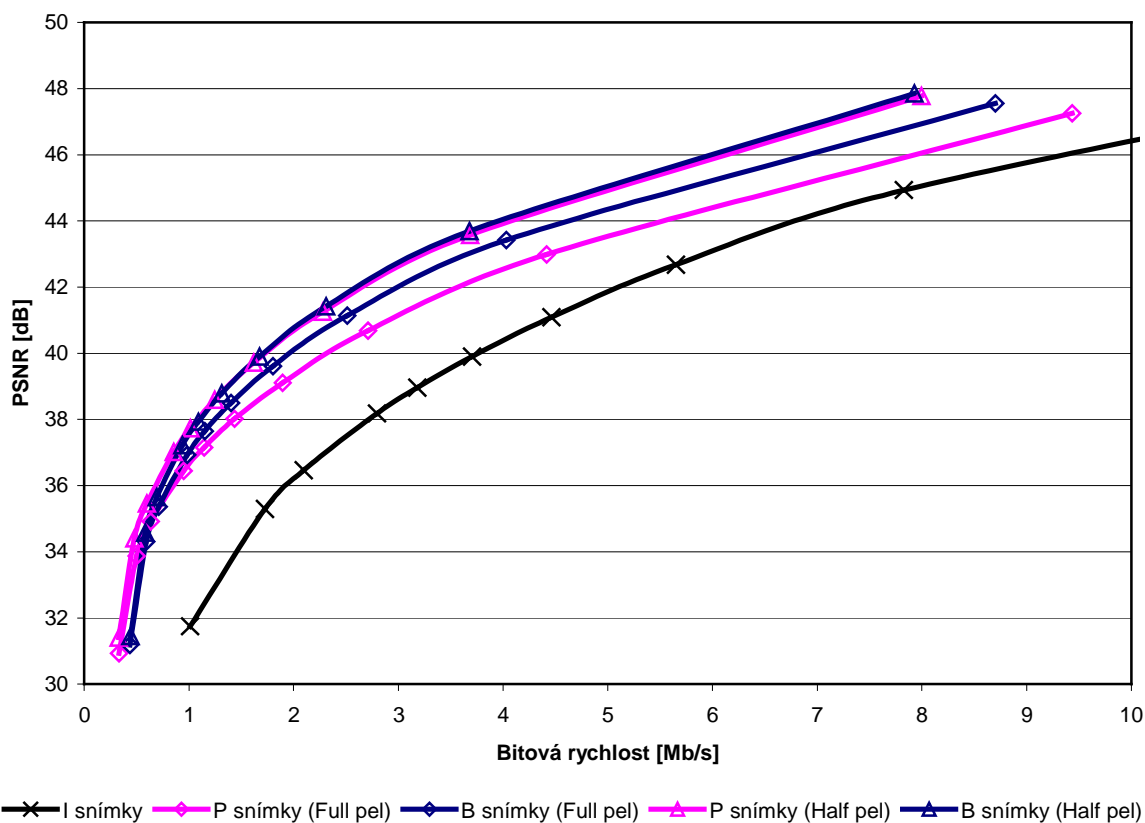
Z příložených průběhů lze také vypočítat, že je dosahováno poměrně malých hodnot bitových rychlostí. To je zapříčiněno rozdílným kvantizačním parametrem pro každý typ snímku. Kvantizační parametr QP pro snímek P, respektive B, je odvozen od kvantizačního parametru snímku I podle rovnic

$$QP_P = QP_I \cdot 1,1 \text{ a} \tag{19}$$

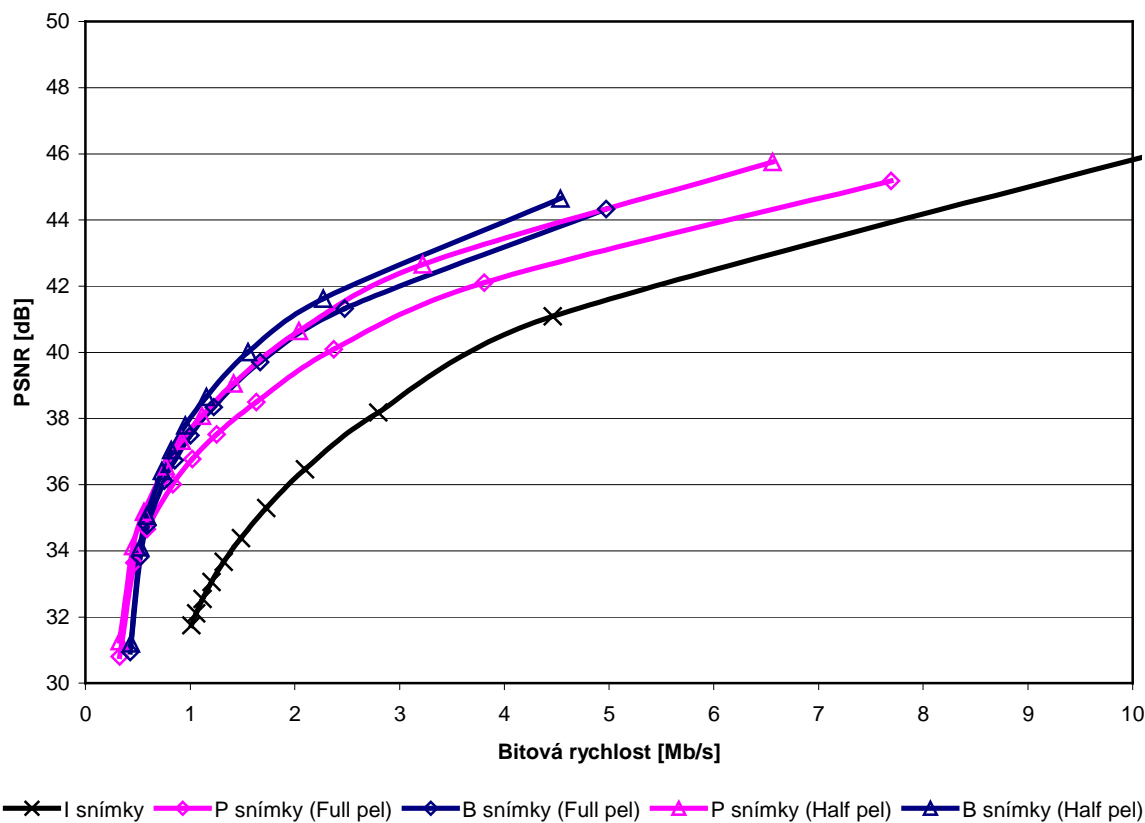
$$QP_B = (QP_I \cdot 1,5) + 1. \tag{20}$$

Při nastavení identického kvantizačního parametru pro každý snímek dostaneme průběhy pokrývající i vyšší bitové rychlosti, viz obr. 53 v porovnání s obr. 54. Tímto nastavením se ovšem zmenší rozdíly mezi jednotlivými nástroji, jmenovitě rozdíl mezi P a B snímky s přesností vyhledávání ½ pixelu není tak dobře pozorovatelný jako u nastavení s rozdílným kvantizačním parametrem. V tomto případě navíc vykazuje nastavení se stejným kvantizačním parametrem o něco nižší výkonnost u snímků B při nižších bitových

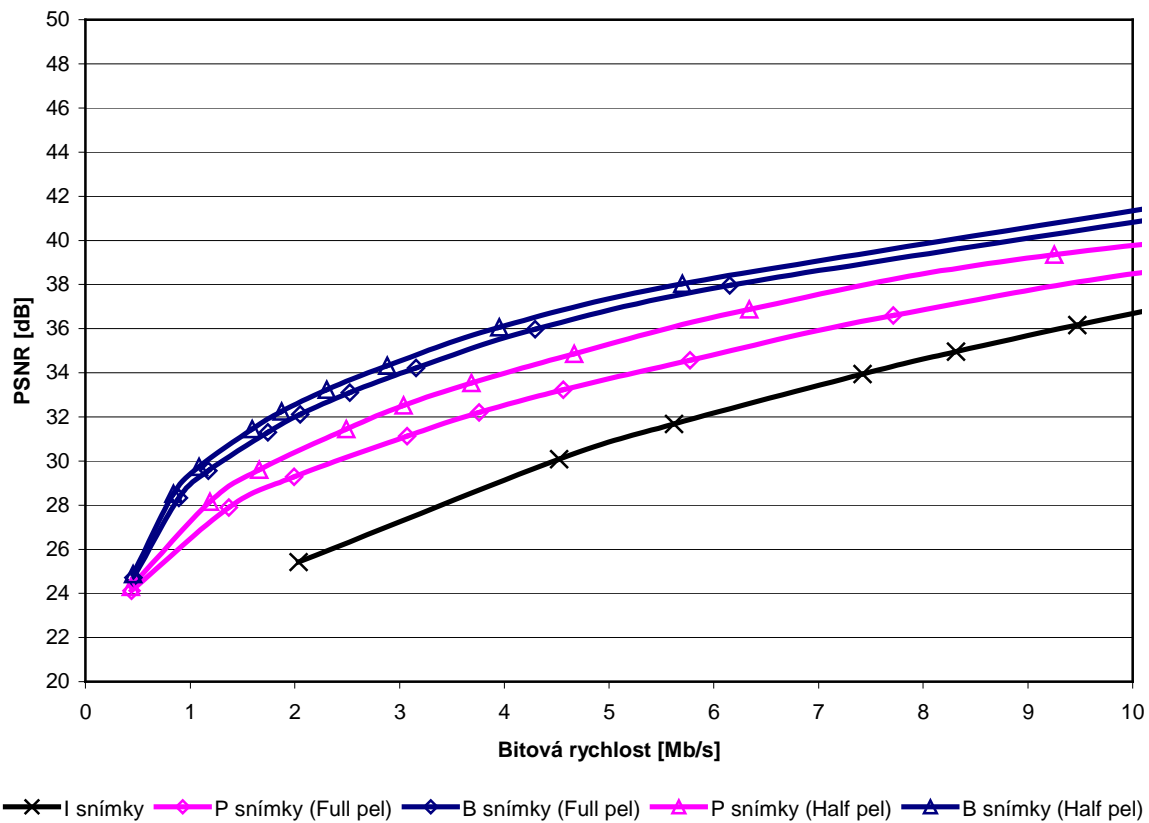
rychlostech. Lze tedy říci, že modifikace kvantizačního parametru snímku P a B poskytuje možnost dalšího ovlivňování výsledné kvality u konkrétní hodnoty bitové rychlosti.



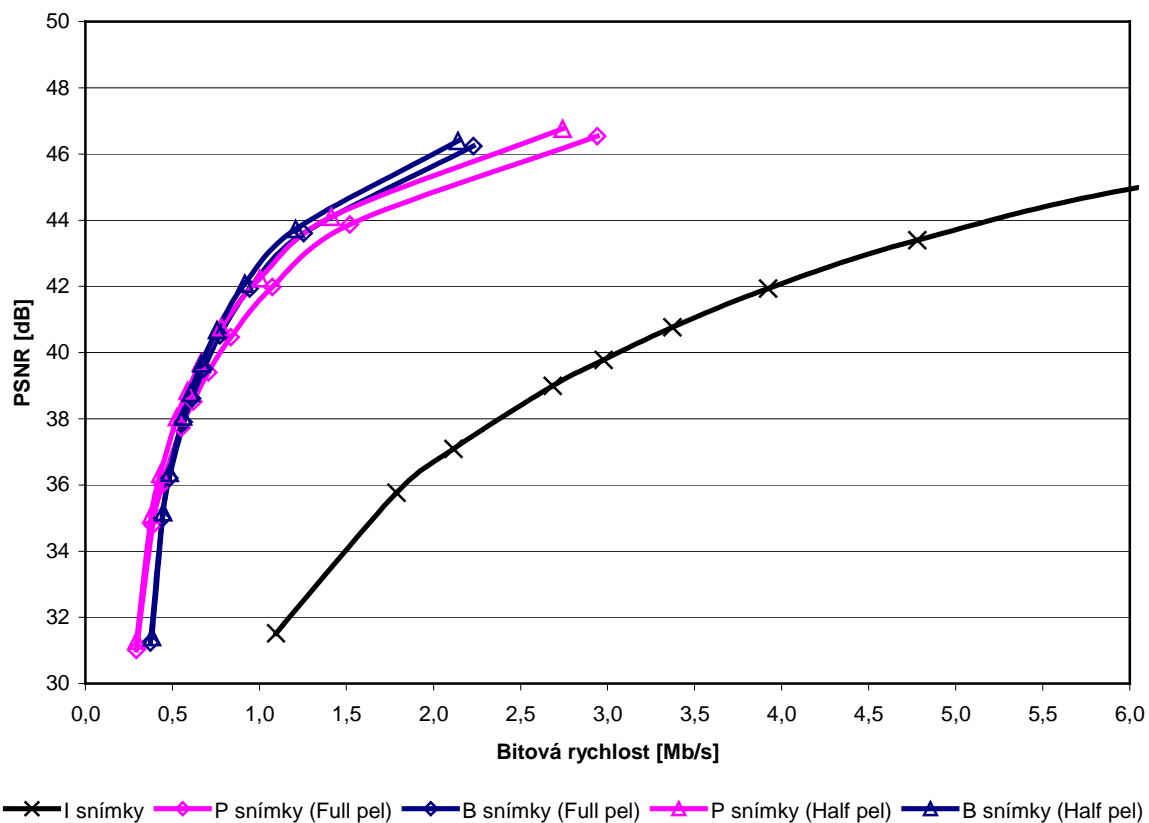
Obr. 53. Sekvence Foreman - identický kvantizační parametr.



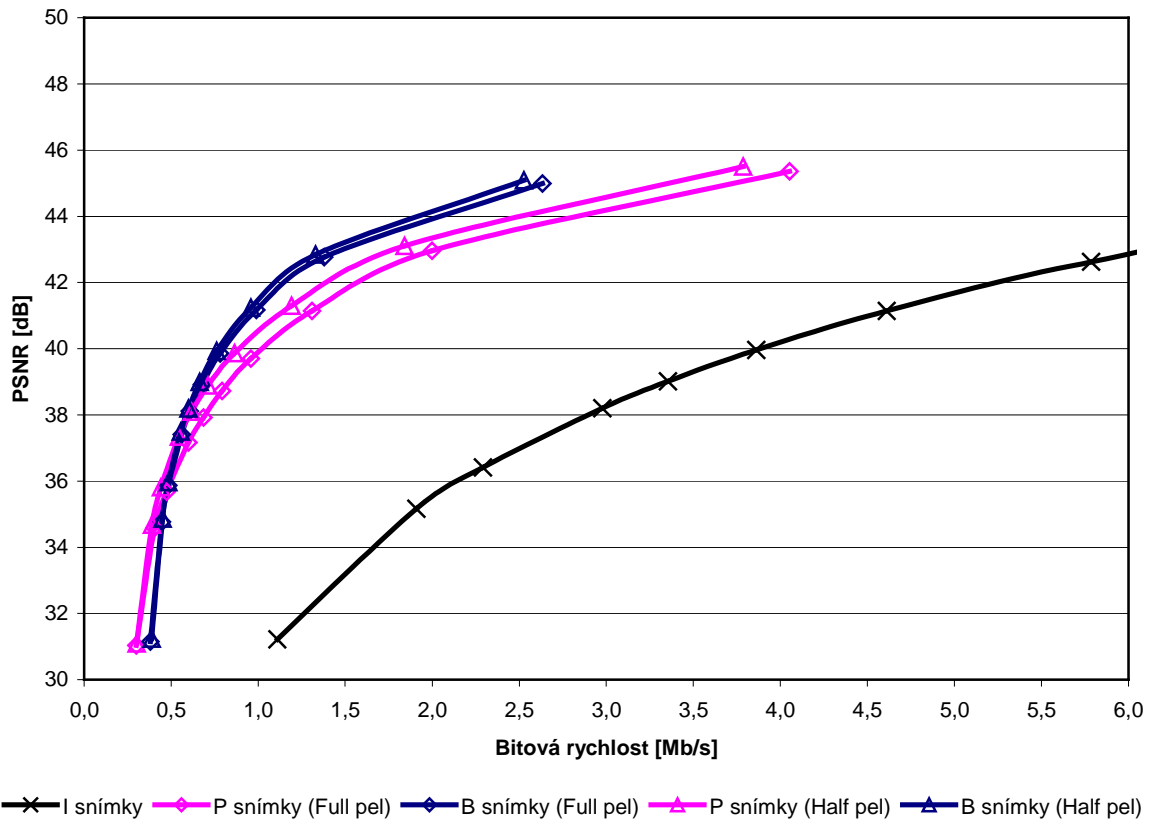
Obr. 54. Sekvence Foreman - rozdílný kvantizační parametr.



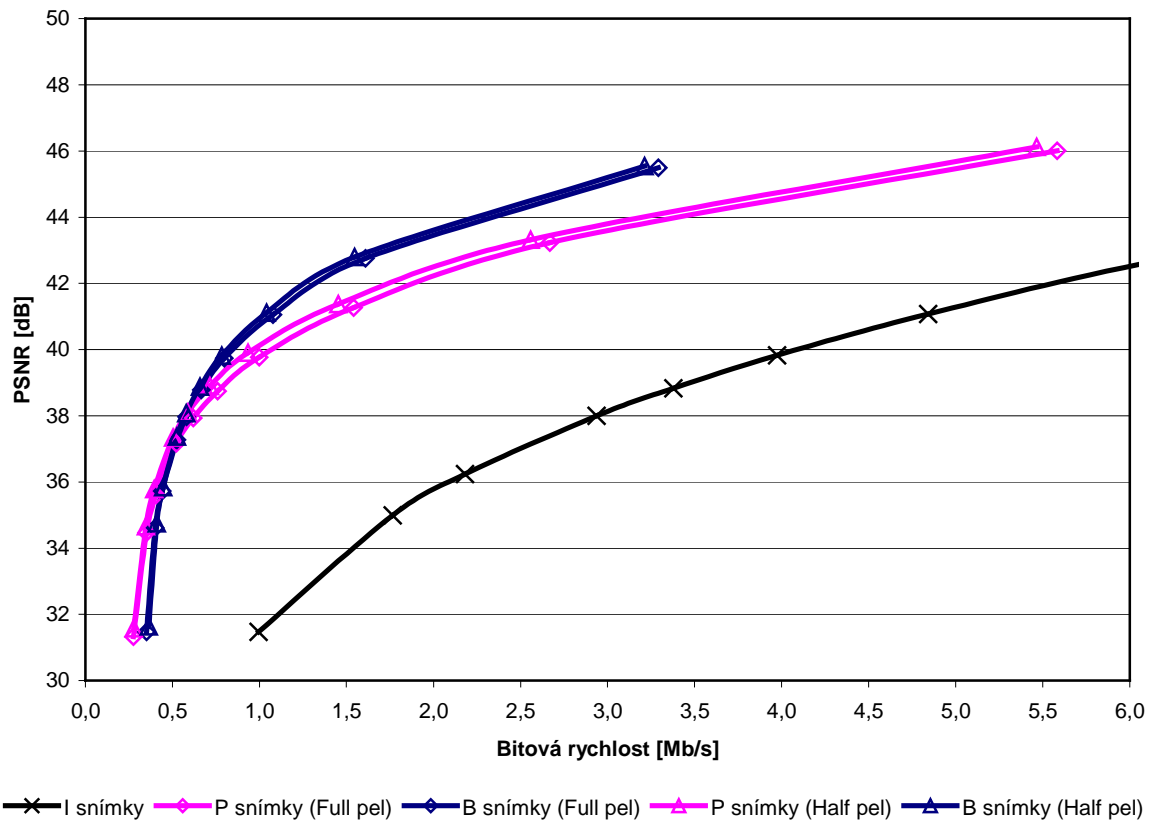
Obr. 55. Sekvence Mobile.



Obr. 56. Sekvence News.



Obr. 57. Sekvence Container.



Obr. 58. Sekvence Salesman.

3.4.2 Srovnání výkonnosti se standardem MPEG-2

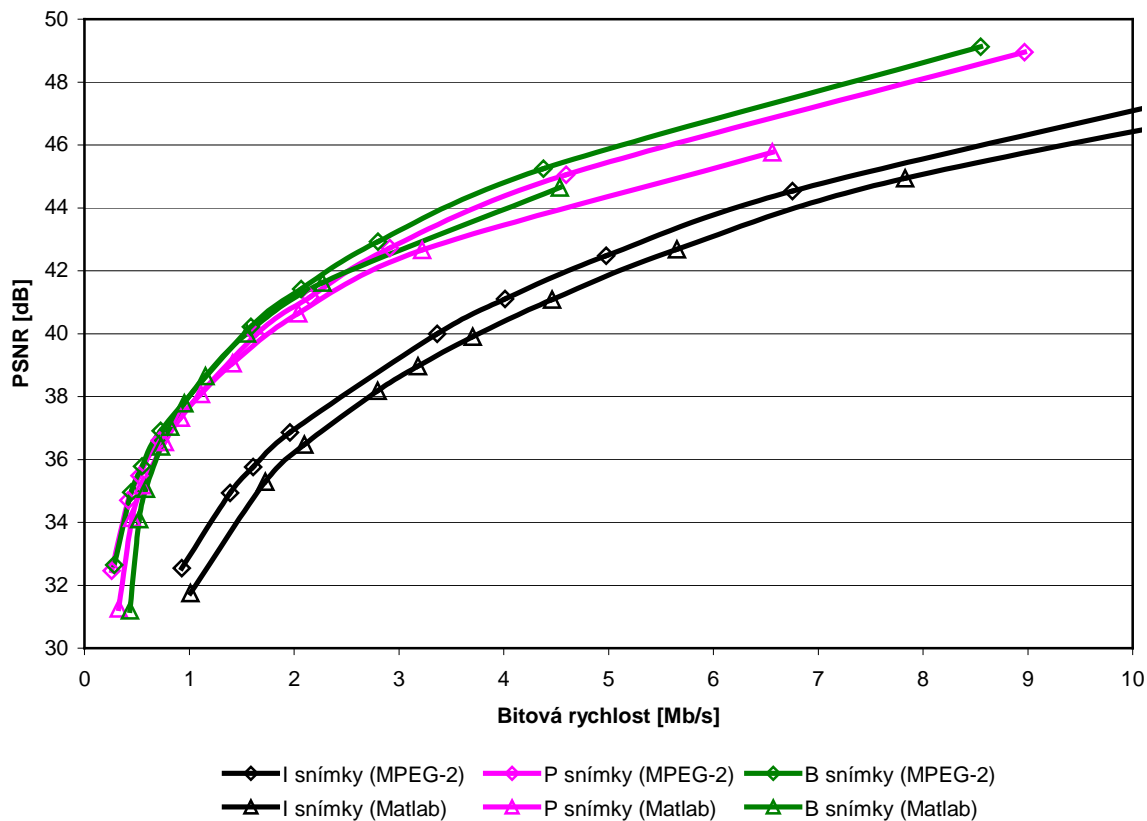
Pro komprimaci standardem MPEG-2 byla využita aplikace FFmpeg 0.5 [21] se standardním nastavením komprimace. Nastavení kvantizačního parametru bylo potlačeno příkazy `-qmin` a `-qmax` pouze na jednu hodnotu tak, aby bylo eliminováno kvantování makrobloků různou hodnotou a porovnání kodeků bylo co nepřesnější, za co nejshodnějších podmínek. Použitá syntaxe příkazu pro komprimaci pouze I snímků:

```
-vcodec mpeg2video -qmin 1 -qmax 1 -intra -r 30
```

Syntaxe příkazu pro komprimaci P a B snímků, v případě komprimace pouze s P snímky bylo nastavení parametru `-bf 0`. V případě B snímků pak `-bf 3`, což indikuje tři po sobě jdoucí B snímky.

```
-vcodec mpeg2video -g 12 -bf 0/3 -qmin 1 -qmax 1 -mbd 0 -r 30
```

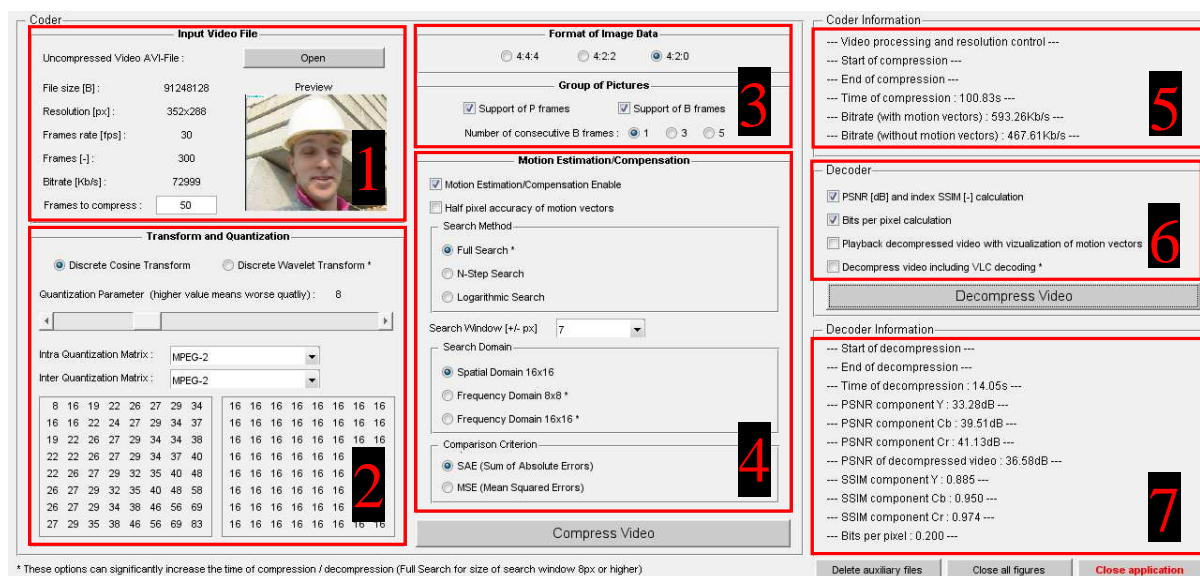
Z obr. 59 vyplývá, že výkonnost kodeku implementovaného v prostředí Matlab se blíží výkonnosti standardu MPEG-2. Na nižších bitových rychlostech je výkonnost velmi podobná, s vyšší bitovou rychlostí se rozdíly zvyšují, což je ovšem dáno i jistou volností v nastavení kvantizačního parametru pro P a B snímek u Matlab implementace, jak je již vysvětleno v předešlé kapitole 4.4.1. Hlavním důsledkem tohoto srovnání je především fakt, že realizace video kodeku je funkční a chová se v rámci mezí podle teoretických předpokladů.



Obr. 59. Srovnání implementace Matlab a standardu MPEG-2.

3.5 Uživatelské rozhraní

Uživatelské rozhraní, viz obr. 60, je vytvořeno za pomoci GUI Builder programu Matlab. Rozhraní se skládá celkem ze 7 oblastí a spodní části obsahující 3 tlačítka – Vymazání pomocných souborů (*Delete auxiliary files*), zavření všech grafů (*Close all figures*) a ukončení aplikace (*Close application*). Prvních 5 oblastí zahrnuje vstup, nastavení a výstupní informace kodéru. Oblasti 6 a 7 pak nastavení a výstupní informace dekodéru. Volby nastavení označené symbolem * jsou výpočetně náročné a komprimace (dekomprimace) se může časově velmi prodloužit.



Obr. 60. Náhled uživatelského rozhraní.

3.5.1 Kodér

Oblast 1, vstupní část (*Input Video File*), umožňuje otevření a zobrazení informací o nekomprimované videosekvenci v kontejneru *.avi. Mezi zobrazované informace patří velikost videosekvence v bajtech (*File size*), rozlišení v pixelech (*Resolution*), snímkování (*Frame rate*), počet snímků (*Frames*), bitová rychlost v kb/s (*Bitrate*) a náhled sekvence (*Preview*). V této oblasti části se také nastavuje počet snímku ke komprimaci (*Frames to compress*).

Nastavení transformace a kvantování (*Transform and Quantization*) je součástí oblasti 2. Na výběr je ze dvou transformací – diskrétní kosinová (*Discrete Cosine*) a diskrétní vlnková (*Discrete Wavelet*) transformace. Kvantizačního parametr (*Quantization Parameter*) lze nastavovat v rozsahu hodnot 1 až 31. Výběr kvantizační matice je doprovázen jejím zobrazením. Pro intra kódované snímky (*Intra Quantization Matrix*) je k dispozici standardní matice MPEG-2 a sada 4 matic, založených na modelu lidského vnímání, pro velmi nízkou, nízkou, vysokou a velmi vysokou bitovou rychlost. Pro inter kódované snímky (*Inter Quantization Matrix*) je nabídka matic totožná, navíc se zde nachází kvantizační matice MPEG-4 Part 2 (XviD/DivX).

V oblasti 3 lze nastavit vzorkování obrazových dat (*Format of Image Data*), s volbami 4:4:4, 4:2:2 a 4:2:0, a formát skupiny snímků (*Group of Pictures*). K dispozici je možnost zapnutí P snímků (*Support of P frames*), B snímků (*Support of B frames*) a počet po sobě následujících B snímků (*Number of consecutive B frames*) s volbou 1, 3 nebo 5.

Největší část, oblast 4, se věnuje nastavení odhadu a kompenzaci pohybu (*Motion Estimation/Compensation*). První možností je vypnutí/zapnutí této oblasti (*Motion Estimation/Compensation Enable*), při vypnutí jsou rozdílové snímky vytvořeny pouze jako jednoduchý rozdíl odpovídajících snímků. Další možností je nastavení vyhledávání s přesností poloviny pixelu (*Half pixel accuracy of motion vectors*). Tabulka 10 obsahuje další možnosti nastavení této oblasti.

Vyhledávací metoda	Search Method	Vyhledávací okno [+/- pixelů]
		Search Window [+/- px]
Plné vyhledávání	Full Search	1 až 16
N-krokové vyhledávání	N-Step Search	1, 3, 7 a 15
Logaritmické vyhledávání	Logarithmic Search	1, 3, 7 a 15
Vyhledávání v doméně	Search Domain	Velikost bloku [v pixelech]
		Block Size [px]
Prostorová doména	Spatial Domain	16x16
Frekvenční doména	Frequency Domain	8x8 nebo 16x16
Srovnávací kritérium		Comparison Criterion
Suma absolutních chyb		SAE (Sum of Absolute Errors)
Střední kvadratická chyba		MSE (Mean Squared Errors)

Tab. 10. Nastavení odhadu pohybu.

V oblasti 5, výstupní informace kodéru (*Coder Information*), se zobrazuje informace o zpracování vstupní videosekvence, začátku a konci komprimace, době komprimace a bitové rychlosti komprimovaného videa, bez a včetně pohybových vektorů.

3.5.2 Dekodér

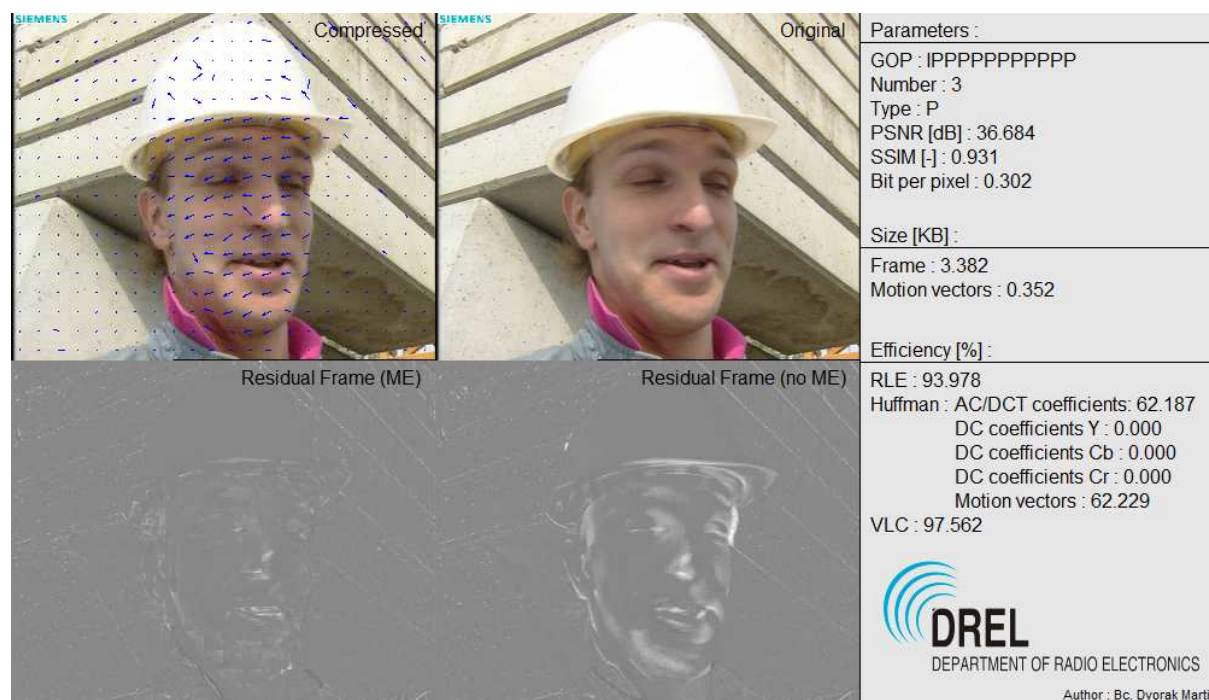
Oblast 6 obsahuje volbu dekomprimace videosekvence včetně dekódování VLC (*Decompress video including VLC decoding*). Tato volba je standardně vypnutá z důvodu vysoké výpočetní náročnosti a tedy i celkového času dekomprimace. Další 3 možnosti se nevztahují přímo k dekomprimaci, ale k funkcím provedeným po dekomprimaci. Jedná se o výpočet PSNR a indexu SSIM (*PSNR [dB] and index SSIM [-] calculation*), počtu bitů na pixel (*Bits per pixel calculation*) a přehrání dekomprimované sekvence s vizualizací pohybových vektorů a vypočtených parametrů (*Playback decompressed video with visualization of motion vectors*).

Poslední oblast, oblast 7, zobrazuje výstupní informace dekodéru. Zobrazované informace jsou začátek a konec dekomprimace, čas dekomprimace. V případě zapnutých výpočtů v nastavení dekodéru se zobrazují informace o PSNR, indexu SSIM a počtu bitů na pixel videosekvence.

3.5.3 Výstupní informace

Informace zobrazované v oblastech 5 a 7 podávají přehled o parametrech videosekvence jako celku. Pro zobrazení parametrů jednotlivých snímků a dalších detailních informací je potřeba v nastavení dekodéru zapnutí volby přehrání dekomprimované sekvence. Po dekomprimaci a výpočtu zvolených parametrů dojde k přehrání sekvence, viz obr. 61. Takto zobrazované

snímky jsou ukládány do adresáře [název_sekvence_playback], který je vytvořen ve stejném adresáři jako originální a dekomprimovaná videosekvence.



Obr. 61. Náhled přehrávání dekomprimované sekvence.

Levá část snímku obsahuje náhled snímku komprimované sekvence s vizualizací pohybových vektorů (*Compressed*), náhled originálního snímku (*Original*) a dva rozdílové snímky – s pohybovou kompenzací (*Residual Frame ME*) a bez (*Residual Frame no ME*).

Pravá část obsahuje výpis parametrů (*Parameters*). Jsou to : formát GOP, číslo snímku (*Number*), typ snímku (*Type*), parametry PSNR, index SSIM a BPP (*Bits per pixel*). Následuje výpis velikosti snímku (*Size Frame [KB]*) a pohybových vektorů (*Motion vectors [KB]*). Poslední zobrazovanou informací je účinnost kódování (*Efficiency [%]*). Účinnost kódování délky běhu (*RLE*), účinnost Huffmanova kódování (*Huffman*) AC/DCT koeficientů, DC koeficientů jednotlivých složek a vektorů pohybu. Nakonec účinnost kódování jako celku (*VLC*).

4 ZÁVĚR

V diplomové práci byly popsány základní principy komprimace obrazového signálu a to, s výjimkami, na základě standardu MPEG-2. Výjimkou je v tomto případě diskrétní vlnková transformace, kterou využívají standardy JPEG2000, video kodek Dirac. Teoretický popis lze rozdělit na tři hlavní komprimační části - kódování uvnitř snímku, mezisnímkové kódování, kódování s proměnnou délkou slova. V tomto pořadí byla zpracována i kapitola 4, která obsahuje realizaci a simulace jednotlivých bloků video kodeku.

Prvním realizovaným blokem je kódování uvnitř snímku, tedy transformační kódování a kvantování. Ze srovnání kosinové a vlnové transformace lze vyvodit, že kosinová transformace podává v tomto modelu video kodeku lepší výsledky než transformace vlnková, což ovšem, jak je zmíněno v teoretické části a u simulace, není překvapivé, protože vlnková transformace není vhodná pro implementaci do tohoto typu modelu video kodeku.

Druhým realizovaným blokem je mezisnímkové kódování. Z porovnávaných vyhledávacích algoritmů je pro největší redukci dat rozdílových snímků nejvýhodnější algoritmus plného vyhledávání, a to z důvodu, že vždy najde globální minimum prohledávané oblasti. V jeho neprospěch je ovšem fakt, že pro větší velikosti vyhledávacího okna je zpracování časově výrazně náročnější než pro zbývající logaritmické a N-krokové vyhledávání. Pro plné vyhledávání byla navíc provedena simulace pro vyhledávání s přesností $\frac{1}{2}$ pixelu. Zvýšení přesnosti vyhledávání přináší další redukci dat, ovšem také zvýšení časové náročnosti, obzvláště při menších velikostech vyhledávacího okna.

Třetím realizovaným blokem je kódování s proměnnou délkou slova, které se skládá z několika částí. První částí je kódování délky běhu, tímto kódováním se dosahuje redukce délky toku dat o 20% až 98% v závislosti na nastaveném stupni kvantizace a také na typu snímku. Druhou částí je Huffmanovo kódování, tímto kódováním se samostatně zpracovávají pohybové vektory, DC koeficienty u snímku I, AC a DCT koeficienty snímku I, respektive snímku P. Procentuální vyjádření redukce toků jednotlivých komponent jsou uvedeny v příslušné podkapitole. Nejdůležitější částí je ovšem účinnost kódování jako celku a pro testovanou videosekvenci byla 68% až 99%, opět v závislosti na velikost kvantizačního parametru a na typu snímku. Kódování délky běhu nul bych označil za klíčovou část kódování, bez něj by totiž nebylo možné efektivně použít Huffmanovo kódování.

Nakonec byl kompletní model video kodeku otestován na několika testovacích video sekvencích s různou dynamikou obrazu a také srovnána jeho výkonnost se standardem MPEG-2. Ze simulací a ze srovnání výkonnosti lze vyvodit, že model video kodeku pracuje podle teoretických předpokladů a také, že se přibližuje výkonností standardu MPEG-2.

Současná verze video kodeku zahrnuje většinu základních nástrojů standardu MPEG-2 a implementace v prostředí Matlab poskytuje prostor pro další modifikaci a přidávání dalších nástrojů z vyšších standardů, jako například nástroje z H.264/AVC. Mimo výukové účely je možné použít model i pro výzkumné účely, například za účelem vylepšení algoritmů odhadu pohybu, zdokonalení kódování pohybových vektorů či testování nových způsobů entropického kódování.

SEZNAM LITERATURY

- [1] VÍT, V. *Televizní technika, přenosové barevné soustavy*. Praha: BEN technická literatura, 1997. 719 stran. ISBN 80-86056-04-X.
- [2] RICHARDSON, I. E. G. *Video Codec Design: Developing Image and Video Compression Systems*. Chichester: Wiley, 2002. 301 stran.
- [3] *MPEG compression technique* [online]. [cit. 18. března 2011]. Dostupný z WWW: http://vsr.informatik.tu-chemnitz.de/~jan/MPEG/HTML/mpeg_tech.html.
- [4] JAHODA, R. *Kodeky tajemství zbavené* [online]. [cit. 18. března 2011]. Dostupný z WWW: http://www.tvfreak.cz/art_doc-73A9DA2913B7BD3C125727C00592A37.html.
- [5] PLANINŠIĆ, P., MOHORKO, J., CUCEJ, Ž. *Komprimiranje slik* [online]. [cit. 18. března 2011]. Dostupný z WWW: <http://www.sparc.uni-mb.si/diplome/s6i.pdf>.
- [6] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, *Information technology* [online]. [cit. 21. dubna 2011]. Dostupný z WWW: http://www.iso.org/iso/iso_catalogue/catalogue_tc/.
- [7] MICROSOFT CORPORATION, *VC-1 Technical Overview* [online]. [cit. 21. dubna 2011]. Dostupný z WWW: <http://www.microsoft.com/windows/windowsmedia/howto/articles/vc1techoverview.aspx>.
- [8] KALVA, H., LEE, J. *The VC-1 and H.264 Video Compression Standards for Broadband Video Services*. Springer, 2008.
- [9] PILLAI L. *Variable Length Coding* [online]. [cit. 24. října 2011]. Dostupný z WWW: http://www.xilinx.com/support/documentation/application_notes/xapp621.pdf.
- [10] GONZALES, R. C., WOODS, R. E., EDDINS, S. L. *Digital Image Processing using MATLAB*. Upper Saddle River: Prentice Hall, 2004.
- [11] INTERNATIONAL ORGANISATION FOR STANDARDISATION. *Coding of Moving Pictures and Associated Audio, Recommendation H.262, ISO/IEC 13818-2*. Draft International Standard, 25 March 1994.
- [12] BALAN, V., CONDEA, C. *Wavelets and Image Compression*. Telecommunication Standardization Sector of ITU [online]. [cit. 18. listopadu 2011]. Dostupný z WWW: <http://www-scf.usc.edu/~hbalan/wavelets.pdf>.
- [13] *Dirac Specification. Version 2.2.3: Dirac* [online]. [cit. 18. listopadu 2011] Dostupné z WWW: <http://diracvideo.org/download/specification/dirac-spec-latest.pdf>.
- [14] ROGUE, B., SALVADO, J. *A Comparative Study on JPEG-Like and EZW Based Image Coders* [online]. [cit. 18. listopadu 2011] Dostupné z WWW: <http://repositorio.ipcb.pt/bitstream/10400.11/101/1/461-131.pdf>.

- [15] INTERNATIONAL ORGANISATION FOR STANDARDISATION. *Coding of Audio Visual Objects, Part 2 Visual, ISO/IEC 14496-2*. Draft International Standard, Second Edition 01 January 2001.
- [16] WIKIPEDIE. *PSNR* [online]. [cit. 26. listopadu 2011]. Dostupný z WWW: <http://en.wikipedia.org/wiki/PSNR>
- [17] WIKIPEDIE. *SSIM* [online]. [cit. 26. listopadu 2011]. Dostupný z WWW: <http://en.wikipedia.org/wiki/SSIM>
- [18] BROOKS, A., PAPPAS N. *Structural similarity quality metrics in a coding context: exploring the space of realistic distortions* [online]. [cit. 26. listopadu 2011]. Dostupný z WWW: www.eecs.northwestern.edu/~pappas/papers/brooks_hvei06.pdf
- [19] SHETH, B. *Structural Similarity Index* [online]. [cit. 26. listopadu 2011]. Dostupný z WWW: www-ee.uta.edu/Dip/Courses/EE5359/Project_proposals2008/Bishwa/final%20presentation_mm.ppt
- [20] ARIZONA STATE UNIVERSITY. *YUV 4:2:0 Video Sequences*. [online]. [cit. 12. dubna 2012]. Dostupný z WWW: <http://trace.eas.asu.edu/yuv>
- [21] FFMPEG TEAM. *FFmpeg*. [online]. [cit. 1. května 2012]. Dostupný z WWW: <http://ffmpeg.org>

SEZNAM PŘÍLOH

A	DVD médium	56
B	Seznam souborů video kodeku	57
C	Návrh laboratorní úlohy	59

A DVD médium

Matlab soubory video kodeku

(kodek\...)

- Viz příloha B

Sada testovacích sekvencí 352x288

(288p30\...)

- foreman.avi
- container.avi
- mobile.avi
- news.avi
- salesman.avi
- highway.avi
- mthr_dotr.avi
- paris.avi

Sada testovacích sekvencí 352x288

(288p50\...)

- crowdrun.avi
- ducks.avi
- intotree.avi
- oldtowncross.avi
- parkjoy.avi

Sada testovacích sekvencí 720x576

(576p50\...)

- crowdrun.avi
- ducks.avi
- intotree.avi
- oldtowncross.avi
- parkjoy.avi

B Seznam souborů video kodeku

Soubor	Popis
bin_dec.m / dec_bin.m	Převod binárního čísla na dekadické a inverzní operace
cikcak.m	Cik-cak vyčítání frekvenčních koeficientů
dekoder.m	Dekomprimace komprimovaných obrazových dat
dekoder_DCT_Q.m	Inverzní diskrétní kosinová transformace a inverzní kvantování
dekoder_dekodovani_I.m	Entropické dekódování (Huffman + RLE) snímku I (AC+DC koeficienty)
dekoder_dekodovani_PB.m	Entropické dekódování (Huffman + RLE) snímku P a B (DCT koeficienty + vektory pohyhu)
dekoder_dpcm_mv.m	Dekódování diferenciálního kódování pohybových vektorů
dekoder_DWT_Q.m	Inverzní diskrétní vlnková transformace a inverzní kvantování
dekoder_predikce.m	Rekonstrukce snímku z rozdílového snímku a referenčního snímku s pomocí pohybových vektorů s přenosti celého pixelu
dekoder_predikce_sub.m	Rekonstrukce snímku z rozdílového snímku a referenčního snímku s pomocí pohybových vektorů s přenosti 1/2 pixelu (jasová složka)
dekoder_preskladani.m	Přeskládání snímků při podpoře B snímků na straně dekodéru
dekoder_RLE.m	Dekodér kódování délky běhu
dekoder_rozdil_C.m	Rekonstrukce snímku z rozdílového snímku a referenčního snímku s pomocí pohybových vektorů s přenosti 1/2 pixelu (chrominanční složky)
dekoder_rozdil_snimek.m	Sestavení snímku B ze dvou zrekonstruovaných snímků
dekoder_snimek_B.m	Dekódování snímku B (zdrojové + entropické)
dekoder_snimek_I.m	Dekódování snímku I (zdrojové + entropické)
dekoder_snimek_P.m	Dekódování snímku P (zdrojové + entropické)
dekoder_VLC.m	Dekodér Huffmanova dekódování AC a DCT koeficientů snímku I,P a B
dekoder_VLC_dc.m	Dekodér Huffmanova dekódování DC koeficientů snímku I
dekoder_VLC_mv.m	Dekodér Huffmanova dekódování pohybových vektorů snímku P a B
DWT.m	Funkce diskrétní vlnkové transformace
EduVideoCodec.fig	Grafické rozhraní
EduVideoCodec.m	Hlavní skript grafického rozhraní
format_dat_I.m	Formátování dat pro cik-cak vyčítání koeficientů a diferenciální kódování DC koeficientů (snímek I)
format_dat_P.m	Formátování dat pro cik-cak vyčítání koeficientů (snímek P a B)
generovani_gop.m	Generování GOP na základě zapnutých nástrojů.
iDWT.m	Funkce inverzní diskrétní vlnkové transformace
interpolace.m	Interpolace oblasti pro vyhledávání s přeností 1/2 pixelu
inverzni_cikcak.m	Inverzní cik-cak vyčítání frekvenčních koeficientů
inverzni_prerovnani.m	Přerovnání 4x(8x8) bloků do makrobloku 16x16
KoDek.m	Hlavní skript obsahující nastavení kodéru a dekodéru (programová verze grafického rozhraní)
koder.m	Komprimace obrazových dat
koder_DCT_Q.m	Diskrétní kosinová transformace a kvantování
koder_dpcm_mv.m	Diferenciální kódování pohybových vektorů
koder_DWT_Q.m	Diskrétní vlnková transformace a kvantování

koder_kodovani_I.m	Entropické kódování (Huffman + RLE) snímku I (AC+DC koeficienty), výpočet účinnosti jednotlivých kódování
koder_kodovani_PB.m	Entropické kódování (Huffman + RLE) snímku P a B (DCT koeficienty + vektory pohyhu), výpočet účinnosti jednotlivých kódování
koder_predikce_C.m	Vytvoření předpovídaného snímku (chrominanční složky)
koder_predikce_FS.m	Algoritmus plného vyhledávání, vytvoření předpovídaného snímku (jasová složka)
koder_predikce_LS.m	Algoritmus logaritmického vyhledávání, vytvoření předpovídaného snímku (jasová složka)
koder_predikce_NSS.m	Algoritmus N-krokového vyhledávání, vytvoření předpovídaného snímku (jasová složka)
koder_predikce_sub.m	Vyhledávání v interpolovaném bloku, vytvoření předpovídaného snímku (jasová složky)
koder_preskladani.m	Preskládání snímků při podpoře B snímků na straně kodéru
koder_RLE.m	Kodér kódování délky běhu
koder_rozdil_snimek.m	Dynamické sestavení rozdílového snímku B ze dvou předpovídaných snímků
koder_snimek_B.m	Kódování snímku B (zdrojové + entropické)
koder_snimek_I.m	Kódování snímku I (zdrojové + entropické)
koder_snimek_P.m	Kódování snímku P (zdrojové + entropické)
koder_VLC.m	Kodér Huffmanova dekódování AC a DCT koeficientů snímku I,P a B
koder_VLC_dc.m	Kodér Huffmanova dekódování DC koeficientů snímku I
koder_VLC_mv.m	Kodér Huffmanova dekódování pohybových vektorů snímku P a B
kontrola_rozmeru.m	Kontrola rozměrů vstupní videosekvence, rozměry musí být dělitelné 16, jinak dojde k ořezání na vhodné rozměry
kvant_param.m	Výběr kvantizačního parametru
porovnavaci_krit.m	Porovnávací funkce SAE a MSE
prehravac.m	Přehrávání dekomprimované sekvence s parametry
prerovnani.m	Přerovnání makrobloku 16x16 do bloků 4x(8x8).
rekonstrukce_dat_I.m	Rekonstrukce snímku I z dekódovaných dat
rekonstrukce_dat_P.m	Rekonstrukce rozdílových snímků P a B z dekódovaných dat
ssim_psnr_sekvence.m	Hlavní skript pro výpočet parametru PSNR a indexu SSIM
ssim_snimek.m	Výpočet indexu SSIM
vstup.m	Převod snímku z RGB na YUV, vzorkování snímku
vypocet_parametry.m	Hlavní skript pro výpočet parametrů, zobrazování grafů
vypocet_velikost_bpp.m	Výpočet velikosti snímků a počtu bitů na pixel
vystup.m	Převod snímku z YUV na RGB, převzorkování snímku

C Návrh laboratorní úlohy

Cíle měření

Tato laboratorní úloha se věnuje zdrojovému kódování obrazového signálu. Student by měl získat přehled o základních komprimačních principech používaných pro zpracování obrazu a jejich účinnosti a dopadu na kvalitu dekomprimovaného videosekvence. K tomuto účelu je k dispozici aplikace simulující kodér a dekodér (KoDek) standardu MPEG-2 realizovaná v prostředí Matlab.

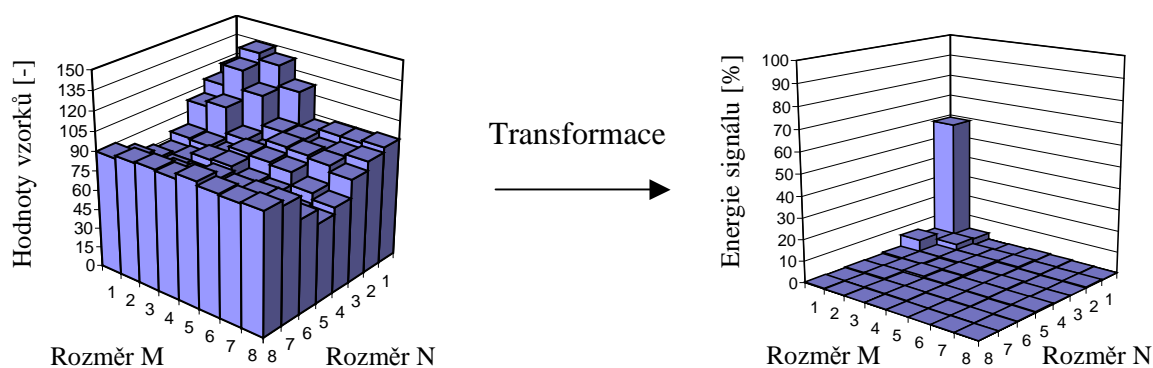
Zadání

1. Srovnajte velikost nekomprimovaného RGB snímku a komprimovaného YUV snímku se vzorkováním dat 4:2:2 a 4:2:0.
2. Pro dvě testovací videosekvence s různým charakterem obsahu proveďte komprimace s různým nastavením kodéru. Srovnajte dopad použitých nástrojů na výslednou kvalitu videosekvence a dále účinnost nástrojů pro rozdílný obsah videosekvencí.
3. Srovnajte účinnost vyhledávání s rozdílnou přesností pohybových vektorů.
4. Prohledněte si velikost střídavých koeficientů snímku I/P a velikost DC koeficientů (luminanční a chrominanční složky) snímku I před a po aplikaci DPCM.

Úvod

Obrazové signály se komprimují z důvodu ušetření množství paměti potřebné k uchování těchto dat (např. záznamová média DVD, Blu-Ray, HD-DVD) a šířky pásma potřebné při jejich přenosu (např. digitální televizní přenos). Ke komprimaci obrazového signálu dochází při redukci redundantních (nadbytečných) a irelevantních (zbytečných) dat obrazového signálu.

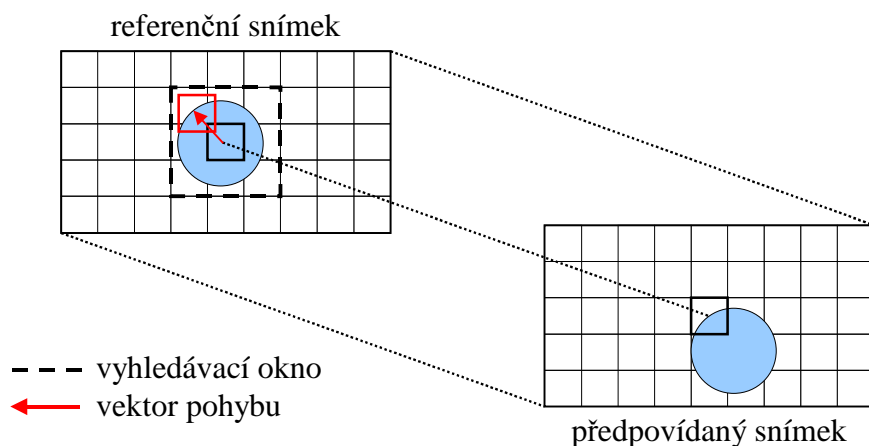
Zpracování obrazového signálu je rozděleno do několika částí. První část je tzv. kódování uvnitř snímku, kam patří transformační kódování (např. diskretní kosinová transformace) a kvantování. Diskretní kosinová transformace převádí vzorky z prostorové oblasti do oblasti frekvenční, čímž dochází k přesunutí střední energie bloku na pozici DC koeficientu. Energie ostatních koeficientů (AC) se zmenšuje a to směrem k pravému dolnímu rohu bloku, viz obr. 1, což odpovídá zvyšující se frekvenci. Díky nedokonalosti lidského zraku a jeho necitlivosti na vyšší frekvence (details obrazu) lze z přenosu vynechat nulové a nule blízké frekvenční koeficienty. K tomuto účelu slouží kvantizace, která zmenšuje velikost frekvenčních koeficientů. Kvantované frekvenční koeficienty jsou získány vydělením frekvenčních koeficientů kvantizační maticí a následným zaokrouhlením na celá čísla.



Obr. 1. Prostorová oblast (vlevo) a frekvenční oblast po transformaci (vpravo).

Další částí je tzv. mezisnímkové kódování. To je založeno na skutečnosti, že sousední snímky jsou si více či méně podobné. Není třeba tedy přenášet celé snímky, ale pouze jejich rozdíl. Přenosem rozdílových snímků se zmenší množství dat pro přenos, výrazně větší redukce je ovšem dosaženo zavedením předpovědi mezi snímky. Princip, viz obr. 2, spočívá v přenášení informace o pohybu objektů za pomoci vektorů pohybu a rozdílů příslušných bloků.

Nalezení vhodných vektorů pohybu je označováno jako metoda odhadu pohybu (motion estimation) a existuje celá řada tzv. vyhledávacích algoritmů (plné, N-krokové, logaritmicke vyhledávání, atd.).



Obr. 2. Princip odhadu pohybu.

Standard MPEG-2 využívá 3 typy snímků. **Snímek typu I**, k jehož kódování, respektive dekódování není potřeba žádného jiného snímku. Slouží jako referenční snímek pro ostatní typy snímků. **Snímek typu P**, pro kódování (dekódování) potřebuje jeden předcházející snímek typu I nebo P v případě, že se jedná o předpověď dopřednou (forward). Pro předpověď zpětnou (backward) naopak následující snímek, opět typu I nebo P. **Snímek typu B** využívá obousměrnou předpověď. Pro kódování a dekódování snímku typu B je potřeba předchozího a následujícího snímku, a to typu I nebo P. Rozdílový snímek je vytvořen jako rozdíl právě kódovaného snímku a průměru předchozího a následujícího snímku.

Poslední částí je tzv. kódování s proměnnou délkou slova a zahrnuje cik-cak vyčítání frekvenčních koeficientů, kódování délky běhu nul a Huffmanovo kódování. Cik-cak způsob vyčítání koeficientů, viz obr. 3, je výhodný, neboť velikost kvantovaných frekvenčních koeficientů se zmenšují stejným směrem. Výstupem vyčítání je vektor přeskládaných koeficientů, u něhož jsou od jistého koeficientu samé nuly.

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

Obr. 3. Cik-cak vyčítání kvantovaných koeficientů.

Na vektor koeficientů je následně aplikováno kódování délky běhu nul. Tímto kódováním se zavádí skupiny o dvou symbolech. První symbol označuje počet nul před

nenulovým koeficientem, druhý symbol pak hodnotu nenulového koeficientu. Pokud do konce vektoru zbývají samé nuly, je místo skupiny zapsán symbol EoB (End of Block).

Skupiny jsou nakonec zakódovány pomocí Huffmanova kódování. Huffmanovo kódování pracuje na základě entropie, která udává minimální počet bitů, který je potřeba pro vyjádření hodnoty vzorku v přenosu. Přiřazení bitů závisí na tom, s jakou pravděpodobností se hodnota vzorku vyskytuje v přenosu. Hodnotám s vyšší pravděpodobností výskytu se přiřazují slova, která jsou vyjádřena menším počtem bitů a naopak hodnotám, které se vyskytují málo, jsou přiřazena slova vyjádřena větším počtem bitů. Ve standardu MPEG-2 je definována sada tabulek, obsahující binární kódy pro skupiny, které byly ověřeny s určitou přesností v praxi.

Poznámky k vypracování

1. Výpočet proveďte s uvažovaným rozlišením snímku 352x288 a 24 bity na vzorek.
2. Při srovnání výkonnosti nástrojů je potřeba pro každé nastavení kodéru provést kódování na několika kvalitativních hladinách (cca 5) pro dostatečné vykreslení průběhu. Nastavení kodéru:
 - Pouze I snímky
 - Podpora P snímků
 - Podpora B snímků (3 po sobě jdoucí B snímky)

Kvalitu dekomprimované videosekvence lze měnit kvantizačním parametrem v rozsahu 1 až 31. Pro tento bod měření je doporučeno volit kvantizační parametr v rozsahu 1 až 13. Ostatní nastavení volte s ohledem na časovou náročnost komprimace a během jednotlivých komprimací je neměňte.

3. Pro jeden vybraný průběh a videosekvenci z předešlého bodu (P nebo B snímky) opakujte komprimaci s nastavením přesnosti pohybových vektorů $\frac{1}{2}$ pixelu.
4. Pro nastavení komprimace slouží skript **KoDek.m**, pomocí breakpointů ve skriptech **format_dat_I.m** a **format_dat_P.m** lze pozastavit zpracování.