



**ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE**

---

Fakulta Elektrotechnická  
Katedra Radioelektroniky

# **Porovnání účinnosti moderních standardů pro kódování videa**

## **Performance comparison of advanced video coding standards**

Bakalářská práce

Studijní program: Komunikace, multimédia a elektronika  
Studijní obor: Multimediální technika

Vedoucí práce: Ing. Karel Fliegel, Ph.D.

**Jindřich Adolf**

## **Poděkování**

Rád bych tímto poděkoval svému vedoucímu práce Ing. Karlu Fliegelovi, Ph.D. za cenné rady, čas který mi věnoval a hlavně za velmi vstřícný přístup.

## **Prohlášení autora práce**

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne.....

.....

Jindřich Adolf

## **Anotace**

Tato bakalářská práce se zabývá porovnáním a popisem současných standardů pro kódování videa, zejména H.264/AVC, HEVC a VP9. Pro vybrané standardy je popsán referenční software a možnosti jeho ovládání. V praktické části se práce zabývá srovnáním jednotlivých standardů na vybraných sekvencích pomocí objektivních kritérií pro porovnání kvality videa.

## **Klíčová slova**

H.264, H.265, AVC, HEVC, VP9, WebM, PSNR, SSIM, VIFP

## **Summary**

This work focuses on the comparison and description of the current standards for video encoding. Especially H.264/AVC, HEVC and VP9. Describes reference software and the possibility of control of selected standards. The practical part concentrates on the comparison of the variol standards in selected sequences using objective criteria to compare video quality.

## **Keywords**

H.264, H.265, AVC, HEVC, VP9, WebM, PSNR, SSIM, VIFP

## Obsah

1	Úvod.....	4
1.1	Zadání práce.....	4
1.2	Důvody komprese.....	4
1.3	Základní informace o digitálním videu .....	5
1.3.1	Základní parametry .....	5
1.3.2	YUV, barevný model a formát videa.....	6
1.3.3	Multimediální kontejner .....	8
1.4	Přehled nepoužívanějších standardů současnosti.....	8
1.5	Referenční Software .....	9
2	Standard H.264/AVC.....	10
2.1	Přehled .....	10
2.1.1	Základní informace.....	10
2.1.2	Vývojáři standardu.....	10
2.1.3	Základní blokové schéma standardu H.264.....	11
2.1.4	Princip kódování.....	11
2.2	Referenční software.....	14
2.2.1	Základní popis.....	14
2.2.2	Instalace a kompilace spustitelného souboru.....	14
2.2.3	Spuštění a ovládání programu .....	15
2.3	Parametry kodéru .....	16
2.3.1	Nastavení souborů .....	16
2.3.2	Ovládání kodéru.....	17
2.3.3	Řízení protiblokových filtrů .....	17
2.3.4	Vážené predikční parametry .....	17
2.3.5	Entropické kódování.....	18
3	Standard H.265/HEVC .....	19
3.1	Přehled .....	19
3.1.1	Základní údaje.....	19
3.1.2	Rozdíly mezi H.264/AVC a H.265/HEVC.....	19
3.1.3	Vývojáři standardu H.265.....	20
3.1.4	Základní blokové schéma standardu H.265.....	21
3.1.5	Princip kódování.....	21
3.2	Referenční software.....	23
3.2.1	Základní popis.....	23
3.2.2	Instalace a kompilace.....	23
3.2.3	Ovládání kodéru.....	24
3.3	Parametry kodéru .....	24
3.3.1	Nastavení souborů .....	24
3.3.2	Volba profilu a úrovně .....	25
3.3.3	Volba makrobloků .....	25
3.3.4	Parametry kódovací struktury.....	26
3.3.5	Parametry odhadu pohybu.....	26
3.3.6	Parametry výběru módu.....	27
3.3.7	Kvantizační parametry .....	28
3.3.8	Parametr entropického kódování .....	29
3.3.9	Parametry kódování snímků .....	29
3.3.10	Parametry protiblokových filtrů .....	30
3.3.11	Parametry nástrojů kódování .....	30
4	Standard VP9.....	31
4.1	Přehled .....	31
4.1.1	Základní údaje.....	31
4.1.2	Vývojáři standardu.....	31
4.1.3	Základní blokové schéma .....	31

4.1.4	Princip kódování.....	32
4.2	Referenční software.....	34
4.2.1	Popis.....	34
4.2.2	Kompilace referenčního software.....	34
4.2.3	Nastavení parametrů.....	34
4.2.4	Ovládání.....	36
5	Objektivní metody pro srovnání kvality videa .....	37
5.1	Základní údaje.....	37
5.2	MSE.....	38
5.3	PSNR.....	38
5.4	VIFP.....	39
5.5	SSIM.....	39
6	Srovnání kompresní účinnosti jednotlivých standardů.....	41
6.1	Postup srovnání účinnosti.....	41
6.2	Výběr sekvencí .....	42
6.2.1	Měření prostorových a časových informací TI a SI .....	42
6.2.2	Spatial Information SI.....	42
6.2.3	Temporal Information .....	42
6.2.4	Průběhy SI a TI jednotlivých sekvencí.....	43
6.2.5	Shnutí parametrů SI a TI všech sekvencí .....	44
6.3	Použitý software .....	45
6.4	Metody kódování .....	45
6.5	Volba parametrů .....	46
6.6	4K sekvence 4096x2048 s využitím parametru CRF.....	47
6.6.1	Popis sekvence.....	47
6.6.2	Nastavení jednotlivých enkodérů .....	47
6.6.3	Výsledné grafy .....	48
6.7	HD sekvence 1280x720 č.1.....	49
6.7.1	Popis sekvence.....	49
6.7.2	Parametry sekvence .....	49
6.7.3	Výsledné grafy .....	50
6.8	HD sekvence 1280x720 č.2 s využitím parametru CRF.....	51
6.8.1	Popis sekvence.....	51
6.8.2	Nastavení parametrů.....	51
6.8.3	Výsledné grafy .....	52
6.9	SD sekvence 704x576 s využitím parametru CRF .....	53
6.9.1	Popis sekvence.....	53
6.9.2	Nastavení enkodérů.....	53
6.9.3	Výsledné grafy .....	54
6.10	4K sekvence s využitím řízení bitového toku.....	55
6.10.1	Popis sekvence.....	55
6.10.2	Nastavení parametrů.....	55
6.10.3	Výsledné grafy .....	56
6.11	HD sekvence 1920x1080 s parametrem CRF.....	57
6.11.1	Popis sekvence.....	57
6.11.2	Parametry enkodérů.....	57
6.11.3	Výsledné grafy .....	58
7	Vyhodnocení výsledků .....	59
7.1	Výsledky pro referenční meze .....	59
7.1.1	Výsledky PSNR pro referenční mez 40 dB .....	59
7.1.2	Výsledky SSIM pro referenční mez 0,975.....	60
7.1.3	Výsledky VIFP pro referenční mez 0,65.....	60
7.2	Výsledky pomocí Bjontegaardovy metriky .....	61
8	Závěr .....	62
9	Zdroje.....	64

10	Přílohy .....	67
10.1	Tabulky naměřených hodnot jednotlivých sekvencí.....	67
10.1.1	Sekvence Traffic .....	67
10.1.2	Sekvence Kalendář.....	68
10.1.3	Sekvence Park Run .....	69
10.1.4	Sekvence Blue Sky .....	70
10.1.5	Sekvence Jachta.....	71
10.1.6	Sekvence Posádka .....	72

# 1 Úvod

## 1.1 Zadání práce

Zadáním práce bylo popsat standardy pro kódování videa, zejména pak H.264/AVC, H.265/HEVC a VP9, a dále seznámit se s referenčním softwarem dostupným pro jednotlivé kodéry. Na základě objektivních kritérií pro měření kvality komprimovaného videa a s využitím referenčního softwaru pak provést srovnání účinnosti kodéru nad databází testovaných videosekvencí.

## 1.2 Důvody komprese

Komprimací videa neboli zmenšením jeho bitového toku je nezbytné se zabývat z prostého důvodu - enormní velikostí potřebného datového objemu. Vezměme v potaz úvahu, že signál obrazu, který dostaneme ze senzorů, se skládá z jednotlivých bodů, (pixelů) a ty se skládají ze tří subpixelů. Zastupují základní barvy: R(červená), G(zelená), B(modrá). Video se skládá ze za sebou jdoucích obrazů. Bylo zjištěno, že při zobrazení 25 snímků během jedné vteřiny, vnímá lidský vizuální systém snímky jako plynulý obraz. Při 8bitovém kvantování na jeden subpixel snadno vypočteme potřebný datový objem pro uložení videa. Jako příklad si vezměme film, který má 120 minut, 25snímků za vteřinu a rozlišení 1920x1080. Z rovnice (1.1)

$$Br = w \cdot h \cdot sb \cdot b \cdot fps = 1920 \cdot 1080 \cdot 3 \cdot 8 \cdot 25 = 1,24 \text{ Gb/s} \quad (1.1)$$

Kde,  $w$  je šířka snímku,  $h$  je výška snímku,  $fps$  je počet snímků za vteřinu,  $sb$  počet subpixelů,  $b$  je počet bitů na vzorek, vyplývá, že velikost souboru by tedy byla cca 1,2 TB. Vzhledem k tak vysoké hodnotě je zřejmé, že takto ukládat video není v dnešní době možné. Díky metodám ztrátového kódování, znalosti funkce lidského vizuálního systému,



především jeho nedokonalostí, můžeme video velice podstatně zkomprimovat, až o několik řádů.

Vzhledem k rostoucímu počtu uživatelů internetu, stále většímu objemu uložených videí na internetu a poskytování nových služeb využívajících video, jako například Apple TV a jiné internetové půjčovny filmů, vzniká potřeba velkých přenosových rychlostí. Cílem moderních standardů videa je tak nejen zvýšit kvalitu ale nabídnout srovnatelnou kvalitu s nižším datovým tokem.

## 1.3 Základní informace o digitálním videu

### 1.3.1 Základní parametry

#### **Počet snímků za vteřinu**

Anglicky „Frame rate“ či frame per second. Často se používá zkratka fps.

Typická hodnota je 24, 25 a 30. Tento počet je již dostačující pro to, aby lidský mozek registroval sled obrazů jako plynulý pohyb. Tyto hodnoty vychází z historie analogového videa, kde se v Evropě pro nahrávání a projekci filmů využívalo 25 snímků, v USA 24 snímků za vteřinu.

Označení může být např. 30p nebo 30i. **p** (progressive) znamená, že se jedná o 30 celých snímků, oproti tomu **i** (interlaced) znamená, že se jedná o prokládané video. To je složeno z dvojnásobku, tedy 60 takzvaných půlsnímků, kdy se střídají při zobrazování liché a sudé řádky. To také vychází z historie, kde bylo toto řešení vhodnější z důvodu dosvitu luminoforu na CRT obrazovkách.

## Rozlišení a poměr stran

Rozlišení nám udává počet pixelů v obraze, poměr stran rozložení na sloupce a řádky. Historicky nejznámější pocházející z analogových standardů je PAL 768x576 tedy poměr 4:3, používaný v Evropě a NTSC 720x480 4:3, používaný v např. v USA. Jedná se o prokládané video, můžeme ho tedy označit 576i respektive 480i. Označované je také jako SD.

HD (High Definition) se označuje video, které má rozlišení 1280x720 (720p) nebo 1920x1080 (1080i nebo 1080p) pixelů.

4K UHD (Ultra High Definition) 3840x2160 (2160p) či 8K UHD 7680x4320 (4320p)

## Datový tok

Množství dat, použitých pro 1 vteřinu videa, běžně se udává v Mbit/s.

Pro Blu-ray se používá až 40 Mbit/s, 128-384 kbit/s pro videokonference, 9,8 Mbit/s pro DVD.

### 1.3.2 YUV, barevný model a formát videa

Tento model vznikl v době, kdy bylo potřeba vytvořit kompatibilní černobílé a barevné televizní vysílání. Vychází z toho, že tři barevné složky můžeme vyjádřit jako složku jasovou a dvě složky barev. Pro převod z formátu RGB použijeme matici (1.2)

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0,299 & 0,587 & 0,114 \\ -0,147 & -0,289 & 0,436 \\ 0,615 & -0,515 & -0,1 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad (1.2)$$

Kde Y je jasová složka, U a V jsou barevné složky, někdy uváděné také jako Cb, respektive Cr. R, G a B jsou původní tři základní barvy.

Formát se označuje číslem o počtu bitů, který připadá na jednu složku a symbolem A:B:C, kde A,B,C vyjadřují informaci o vzorkování jednotlivých složek. O poměru mezi jasovou a barevnými složkami (2:1 znamená 4 vzorky jasové na dva barevné)

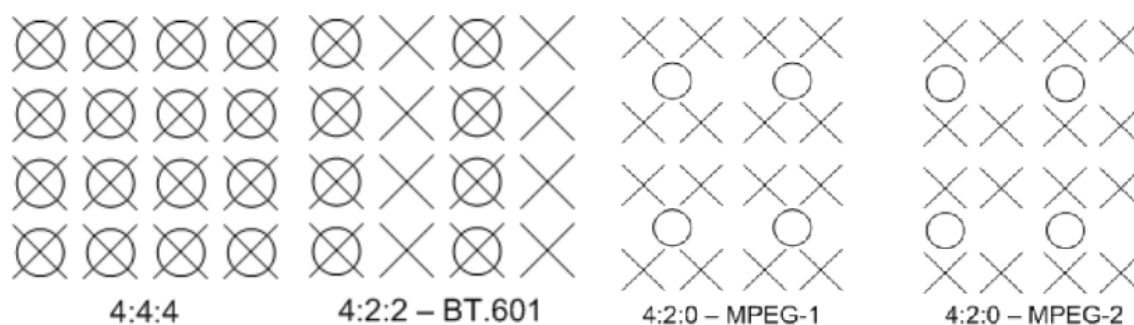
4:4:4 znamená, že jsou všechny složky vzorkovány stejně

4:2:2 znamená 2:1 vzorkování horizontálně, žádné vertikálně

4:2:0 znamená 2:1 vzorkování vertikálně a 2:1 vzorkování horizontálně

4:1:1 znamená 4:1 vzorkování vertikálně, žádné horizontálně

Z obrázku (1.1) je patrné jak je informace snímána, křížek znamená jasovou složku, kolečko barevnou.



Obr. 1.1 Vzorkování jasové a barevných složek [1]

V praktické části pracuji pouze s 8 bitovými formáty 4:2:0.

Znamená to, že každý pixel je vzorkován pro jasovou složku, a pro barevné každý čtvrtý. Velikost souboru je proto jasně daná. (např. pro 300 snímků 4096x2048 je to 3,77 GB). Data jsou v souborovém formátu „.yuv“ uskladněna po šesti bytech na každé dva pixely za sebou, zleva doprava, shora dolů. Je tedy nutné dát kodéru nebo přehrávači informaci o rozlišení souboru.[1]

### 1.3.3 Multimediální kontejner

Multimediální kontejner je datový soubor, obálka, ve kterém je zapouzdřen jeden nebo více multimediálních obsahů, např. video a dvě autostopy a titulky. Uchovává data o použitém kompresním standardu a obsahuje data ve formátu tímto standardem určeným.

Mezi nejznámější patří:

AVI, vyvinutý firmou Microsoft již v roce 1992, jako součást videa pro Windows, které umí přehrávat současně obraz i zvuk. V roce 1996 byl vylepšen skupinou Matrox OpenDML o další formáty.

MPEG-4 part 14 neboli MP4 s příponou .mp4, jedná se o mezinárodní standard vyvinutý skupinou MPEG, podporuje streamování přes internet.

QuickTime od Apple s příponou .mov nebo .qt, úspěšný formát s velkou podporou audio i video formátů jak pro zvuk, video tak pro efekty či titulky.

WebM pro video VP8 a VP9, založený na formátu Matroška.

VOB s příponou .vob pro DVD-Video

Flash Video s příponou .flv

Matroška s příponou .mkv .mka .mk3d .mks), moderní formát z roku 2002, dokáže pojmout více audiotop, podporuje kapitoly, rychlé vyhledávání v souboru, menu a je streamovatelný po internetu.[2][24]

## 1.4 Přehled nepoužívanějších standardů současnosti

Mezi nepoužívanější standardy patří Theora, volně šiřitelný ztrátový formát. Je založen na diskretní kosinově transformaci, je blokově orientovaný. Bloky jsou složeny z 8x8

pixelů a mají různé struktury, kterým se říká bloky, superbloky a makrobloky. Podporuje Intra-snímkové kódování a dopřednou predikci, nikoliv však zpětnou. Nepodporuje prokládané video a je omezen maximální hloubkou kvantování 8 bitů.[3]

Dalším volně šiřitelným standardem je Dirac vyvinutý výzkumným oddělením BBC. Dirac podporuje kompresi pro Ultra HDTV. Je intra-prediktivní a obsahuje pohybovou kompenzaci. Dirac Pro byl vnitřně použit BBC pro přenos HDTV obrazu při Olympijských hrách v roce 2008.[4]

SMPTE 421M, znám také jako VC-1 byl vyvinut firmou Microsoft a je podporován Blu-Ray disky. Jedná se také o kodek založený na diskrétní kosinově transformaci. Byl vytvořen pro herní konzoli Xbox 360 a je podporován i v konzoli Play Station 3. Má tři profily, Simple, Main a Advanced. Maximální datový tok je 135 Mbit/s a maximální rozlišení 2048x1536 při 24 snímcích za vteřinu.[5]

Mezi další patří VP8 od firmy Google, je předchůdcem standardu VP9 který je společně s H.264/MPEG part 10/AVC a H.265/HEVC popsán v práci podrobněji.

## 1.5 Referenční Software

Referenční software je ve většině případů volně dostupný kód, případně kompilace tohoto kódu, Je primárně určen k testování možností a funkcí daného standardu, případně k nahlášení možných chyb, vyskytujících se v kódu. Dává možnost nastavit všechny parametry daného standardu a otestovat tak chování při různém nastavení. Je stále ve vývoji a označuje se aktuálním pořadím verze. Na stránkách vývojářů je dostupný seznam změn oproti minulé verzi.

## 2 Standard H.264/AVC

### 2.1 Přehled

#### 2.1.1 Základní informace

Standard H.264 je v této práci probrán nejpodrobněji, protože se jedná o nejrozšířenější formát videa, se kterým se každý uživatel setkává téměř denně. Jeho použití je velmi rozsáhlé, využívá ho například gigant Apple, Adobe ve svém Flash Playeru a mnozí další. Jeho využití není jen na webu ale i v pozemním televizním vysílání DVB-T či ve většině běžně dostupných záznamových zařízení, například videokamerách či bezpečnostních kamerách. V neposlední řadě je třeba zmínit záznam videa ve vysokém rozlišení na Blue-Ray discích. Na výše uvedených aplikacích je vidět jeho široká variabilita v datovém toku, pro web nízké rychlosti, naopak pro Blue-Ray disky velmi vysoké. V profilu Hi4444PP lze kódovat video až při 14bitové hloubce. Maximální rozlišení je 4096x2304 při 56,3 snímcích za vteřinu. Nejvíce snímků, tedy 172 při HD rozlišení 1920x1080. Obě dvě kombinace při největším možném datovém toku 720Mb/s.[22]

#### 2.1.2 Vývojáři standardu

Stejně jako předchozí verze standardů, byl i H.264 vyvinut skupinou ITU-T Video Coding Experts Group společně se skupinou ISO/IEC JTC1 Moving Picture Experts Group. Partnerství těchto dvou skupin je známo pod jménem Joint Video Team.[6]

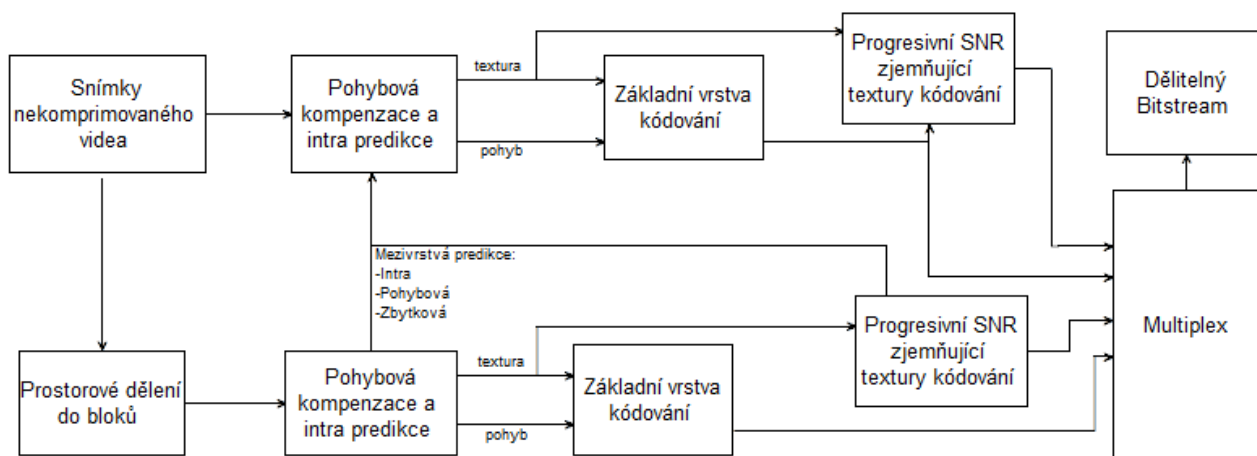
ITU (Internacional Telecommunication Union) je agentura OSN v oblasti telekomunikací. Standardizační sektor (ITU-T) je stálým orgánem ITU. Je zodpovědný za technické specifikace pro telekomunikační systémy, služby i tarifní zásady mezinárodního účtování.[18]

ISO/IEC JTC1 je společná technická komise dvou organizací: ISO (International Organization for Standardization) - Mezinárodní organizace pro normalizaci založené v roce 1947 a IEC (International Electrotechnical Commission) založená v roce 1906. Jejich úkolem je je vytvářet mezinárodně platné normy.[19][20]

MPEG je skupina expertů čítající více než 400 členů z 20 různých zemí založena roku 1988.[21]

VCEG byla založena roku 1994 pod jménem CCITT a to skupinou Specialists Group on Coding for Visual Telephony.[18]

### 2.1.3 Základní blokové schéma standardu H.264



Obr. č. 2.1- Blokové schéma kodéru H.264 [25]

Základní vrstva kódování- Diskrétní kosinova transformace, kvantizace

### 2.1.4 Princip kódování

Video je složeno z jednotlivých snímků, běžně se využívá 24 až 30 snímků za vteřinu. Kodér rozdělí jednotlivé snímky do skupin zvaných GOP (Group of Picture). Počet snímků ve skupině není pevný, průměrná hodnota je 12. Základním prvkem je Intra

Frame, zkráceně označovaným jako I-snímek. Je kódovaný formátem JPEG (Joint Photographic Expert Group). Obraz je rozdělen na takzvané makrobloky, o velikosti 16x16 jednotlivých bodů, neboli pixelů. Jednotlivé makrobloky jsou zvlášť transformovány do barevného prostoru YCrCb, kde je zmenšena informace o barvách na úkor jasů. Je to z toho důvodu, že HVS (Human visual system, čili lidský vizuální systém) vnímá změny jasů jinak než barvy, na jas je mnohem citlivější, je tedy výhodnější mít více informace právě o jasů. Zde už záleží nejen na pestrosti pixelů ale i na nastavení kodéru. Musíme vybrat kompromis mezi kvalitou obrazu a datovým tokem. Dochází zde ke kvantizaci hodnot jasů i barvy. Pokud bude snímek jednobarevný, například stěna, k velké ztrátě nedojde, rozdíl však bude znatelný u členitých povrchů jako například písek nebo trávník, důsledkem kvantizace dojde ke splynutí barev.

Ve skupině máme ještě dva druhy snímků a to P-snímky a B-snímky. P-snímky mají naprosto jiný způsob kódování, nesou mnohem méně informace. Udávají pouze změny kompletního I snímku, pokud bychom porovnali dva za sebou jdoucí snímky, zjistíme, že jsou skoro totožné, liší se pouze minimálně. B snímky fungují podobně, udávají změnu oproti I nebo P snímku, mají avšak výhodu v tom, že rozdíl může být jak proti předchozím snímkům, tak i budoucím. Kodér tedy propočítá, jestli je vzhledem k datovému toku výhodnější budoucí či minulý.

Jednotlivé bloky o velikosti 4x4 nebo 8x8, v závislosti na zvoleném profilu, jsou převedeny do frekvenční oblasti, bylo zjištěno, že HVS vnímá nízké frekvence více než vysoké, nejvíce tedy nejnižší frekvenci- stejnosměrnou složku, která nese informaci o průměrném jasů makrobloku, zde tedy dochází ke kvantizaci, parametr standardu H.264 má 51 možných stupňů, které jsou na logaritmické škále, čímž docílíme jemnějšího průběhu při nízkých frekvencích. Další možností snížení datového toku je fakt, že jednotlivé makrobloky, které se nachází vedle sebe, mají podobné hodnoty. Nemusíme tedy přenášet všechny vysoké hodnoty, ale opět si vystačíme s rozdílem oproti sousední hodnotě.

Další vlastností tohoto standardu je takzvané interkódování, jeho významem je popis co největšího počtu makrobloků pomocí pohybových vektorů, díky čemuž můžeme zachovat



věrnost pohybu objektů. Využívá se zde možnosti odvození pohybových vektorů z jiných snímků nežli z referenčního I snímku, tomuto algoritmu se říká Multi Frame Prediction. Vzniká tak kombinace předchozího I snímku a mezilehlých P snímků, tento snímek pak nese označení IDR. Nemá pravidelnou periodu, vygeneruje se, pokud je to možné v místě kamerového střihu. Průměrně se vyskytuje každých 300 snímků.

Velikost makrobloků u H.264 je v rozmezí 4x4 až 16x16, nemusí však jít nutně o čtverec. Pokud je to vhodné, rozdělí se obraz na bloky například 16 x 8. Přesnost pohybového vektoru je čtvrtina pixelu. Pohybový vektor má tři složky, dvě prostorové a jedna označuje pořadí referenčního snímku. Kodér potřebuje pro každý P snímek seznam možných referenčních snímků. Pro B snímek potřebuje seznamy dva. Dalším algoritmem je váhová predikce, každému bloku je přiřazen faktor pro změnu jasu a barvy. Tím ušetříme datový tok ve chvílích, kdy se obraz pouze stmívá nebo rozjasňuje. Velké úspory toku docílíme, ve chvílích kdy má obraz statické scény, makrobloky se opakují a je tak možné je přeskočit, označují se Skipped. Další možností zlepšení subjektivní kvality obrazu je použití neblokovaného filtru, ten dokáže blízke pixely, které se kvůli kvantizaci staly stejnými, pozměnit a dojem ze scény je tak příjemnější, při nízké hodnotě kvantizace je účinek téměř neznatelný, naopak při vysoké hodnotě kvantizačního parametru obraz naopak kazí, je tedy vhodný jen pro určité pásmo datového toku.

Při tolika nastavitelných parametrech není snadné vybrat správný poměr účinku jednotlivých algoritmů. Proto je zde RDO- Rate-distortion optimalization, algoritmus který ovlivňuje kombinaci zaplnění makrobloku, volí tak například počet referenčních snímků vůči pohybovým vektorům. Zjednodušeně se snaží dosáhnout nejlepší kvality obrazu při zadaném datovém toku. Řídí se parametrem PSNR, což je jeden z parametrů objektivního hodnocení kvality videa, podrobně je níže popsán a je stěžejním výsledkem praktické části této bakalářské práce.

Dalším algoritmem je PsyRDO, je to psychovizuální funkce schopna rozpoznat chyby videa, které jsou pro HVS vnímány citlivěji: dosáhne se tak pocitově kvalitnějšího obrazu. Na konci tohoto náročného procesu je ještě entropické kódování, závislé na výpočtu pravděpodobnosti výskytu binárních hodnot. U H.264 se používá převážně CABAC(Context based Adaptive Binary Arithmetic Coding), založen na podobném principu jako komprese ZIP, tedy možností nahradit

delší řady nul či jedniček jinými kombinacemi s menší bytovou délkou. Je však velmi náročný na hardware, nachází uplatnění u vyšších profilů. Pro nižší se používá jednodušší CAVLC (Context Adaptive Variable Length).[7]

## 2.2 Referenční software

### 2.2.1 Základní popis

Referenční software h.264 a HEVC se od sebe v mnohém neliší. Základní principy jsou stejné, je pouze třeba podotknout, že standard h.264 je na světě už delší dobu a tak je odladěno mnohem víc chyb a je napsána lepší dokumentace. Píši tedy podrobný postup na použití referenčního software pro h.264. Postup je přehlednější a jde s trochou představivosti a drobných obměn aplikovat i na referenční software HEVC. Obě verze existují jak pro linux tak pro Microsoft Windows. V této práci popisuji verzi pro MS Windows, pro kterou je vhodné mít nainstalované prostředí Microsoft Visual Studio. Pracoval jsem s verzí JM 18.5.

### 2.2.2 Instalace a kompilace spustitelného souboru

Na internetové stránce:

<http://iphome.hhi.de/suehring/tml/download/>

Stáhneme soubor "jm18.5.zip", ve kterém jsou všechny potřebné soubory ke kompilaci výsledného \*.exe souboru. Otevřeme soubor jm\_vc9.sln (číslo na konci souboru se liší v závislosti na vámi používané verzi MS Visual studio). Zde se nám nabídne několik částí projektu, ldecod, lence, rtp\_loss a rtpdump. Nás bude zajímat lencod. To je část projektu, kde je napsán kódér AVC. Pokud si jej rozklikneme, můžeme upravovat libovolnou část kódu. Pokud se spokojíme s tím, co napsali jiní, klikneme pravým tlačítkem myši na lencod a zvolíme „Build“. Pokud nemáme v kódu chybu a máme k dispozici všechny potřebné soubory (měly by být v zipu, co jsme v úvodu stáhli) vytvoří se nám do složky jm18.5\JM\bin soubor ldecod.exe.

## 2.2.3 Spuštění a ovládání programu

Přístup k programu je možný přes příkazový řádek MS Windows. Proto si jej spustíme (tlačítko Windows+R, a napsat cmd a enter). Pomocí příkazu cd se dostaneme do adresáře, ve kterém máme program uložený.

(V mém případě C:\BP264\jm18.5\JM\bin). Odtud můžeme program spustit.

Je zde několik druhů příkazů, jak program nastavovat a měnit.

-h	vytiskne použité parametry
-d	načte konfigurační soubor defenc.cfg (pokud není za parametrem zadán jiný)
-f	načte soubor curencM.cfg (pokud není za parametrem zadán jiný) a resetuje tak nastavené parametry enkodéru.
-p	nastavujeme jednotlivé parametry.

Příklad zadání:

```
lencod.exe -f curenc1.cfg -p InputFile="e:\data\container_qcif_30.yuv"  
-p SourceWidth=176 -p SourceHeight=144 -p FramesToBeEncoded=30
```

Tento sousled příkazů načte konfigurační soubor curenc1.cfg, jako videosoubor bude zpracovávat ten, který se nachází ve složce zadané cestou, šířka souboru je 176 pixelů, výška 144 a zakóduje se prvních 30 snímků. Toto je poněkud nepřehledné a nepohodlné, bude tak nejlepší vytvořit si vlastní konfigurační soubor (respektive upravit defaultní) a ten potom jednoduše načíst. V něm jsou jeho parametry celkem srozumitelně popsány a psány stylem

```
<jméno parametru> = <hodnota parametru> # Komentář
```

## 2.3 Parametry kodéru

### 2.3.1 Nastavení souborů

<i>InputFile</i>	Udává název a cestu originálního „yuv“ souboru
<i>InputHeaderLength</i>	Pokud má soubor hlavičku, zadá se tímto parametrem jeho velikost v bytech, pokud ne, zadá se nula
<i>StartFrame</i>	Zde se zadá pořadí snímku původního videa, od kterého začne kodér kódovat
<i>FramesToBeEncoded</i>	Počet snímků, které bude kodér kódovat
<i>FrameRate</i>	Počet snímků výsledného videa za vteřinu
<i>SourceWidth</i>	Šířka snímku původního videa
<i>SourceHeight</i>	Výška snímku původního videa
<i>SourceResize</i>	Parametr pro zvětšení/zmenšení cílového souboru ke zdroji
<i>OutputWidth</i>	Šířka snímku výsledného videa
<i>OutputHeight</i>	Výška snímku výsledného videa
<i>TraceFile</i>	vytvoří soubor „.txt“, který mapuje jednotlivé kroky kódování
<i>ReconFile</i>	Vytvoří „yuv“ soubor který vznikne zpětným převedením z komprimovaného souboru
<i>OutputFile</i>	Výsledný zakódovaný soubor ve formátu *.264, který je použitelný pro uložení do kontejneru (např. avi, mpeg, mkv,...)

### 2.3.2 Ovládání kodéru

ProfileIDC	Nastavení požadovaného profilu, na výběr 66 Baseline, 77 Main, 88 Extended, 100 High, 110 High, 122 High 4:2:2, 244 High 4:4:4, 44 CAVLC 4:4:4 intra
LevelIDC	Nastavení požadovaného levelu v závislosti na rozlišení, nebo frame-rate, možné hodnoty jsou 10,11,12,13,20,21,22,30,31,32,40,41,42,50,51
IntraPeriod	Perioda zařazení I-snímku, 0 znamená pouze první snímek
IDRPeriod	Perioda IDR snímku, chování je stejné jako u I-snímku
FrameSkip	Počet snímků, které mají být vynechány při kódování vstupní sekvence.
SearchRange	Nastavení vyhledávacího rozsahu pro Odhad pohybu, defaultní hodnota je 16

### 2.3.3 Řízení protiblokových filtrů

DFParametersFlag	Nastavení hodnoty elementu <code>deblocking_filter_control_present_flag</code> , defaultní hodnota je 0 (nenastaveno). Tento parametr nastaví stejné parametry protiblokových filtrů pro všechny kódované obrázky i přes to že kodér podporuje různé nastavení pro různé snímky
DFDisableRefIlice	Nastavení hodnoty elementu <code>disable_deblocking_filter_idc</code> pro I snímky patřící k referenčnímu snímku, Vyžaduje aby byl nastaven předchozí parametr

### 2.3.4 Vážené predikční parametry

WeightedPrediction	Umožňuje váhování predikce pro P snímky na základě stejnosměrných hodnot snímků, defaultně je vynuto, tedy hodnota 0 (pro zapnutí, hodnota 1)
WeightedBiprediction	Nastavení váhování predikce pro B snímky, defaultně vynuto, tedy 0, 1 pro explicitní váhovou predikci, 2 pro Implicitní váhovou predikci (založeno na POC tedy Picture Order Count)
UseWeightedReferenceME	Použití váhování při pohybové kompenzaci, defaultně zakázáno (0 nebo 1)

### 2.3.5 Entropické kódování

SymbolMode	Nastavení entropického kódování 0 pro CAVLC (defaultně nastaveno) 1 pro CABAC
ContextInitMetod	Pro nastavení CABAC, 0 Fixní (defaultně) 1 Adaptivní
OutFileMode	Souborový mód. 0 pro Annex B byte stream format (defaultně 1 pro RTP pakety)
PartitionMode	Dělení dat, 0 bez dělení (defaultně) 1 dělení na tři části pro jeden snímek

## 3 Standard H.265/HEVC

### 3.1 Přehled

#### 3.1.1 Základní údaje

Zkratka HEVC znamená High Efficiency Video Coding, tedy velmi efektivní kódování videa. Je přímým nástupcem standardu h.264. S ohledem zejména na snížení datového toku, který se zvyšujícím počtem internetových uživatelů a se stále rostoucím multimediálním obsahem internetu naráží na limity přenosových soustav. Rozdíl oproti h.264 je udávané průměrné snížení datového toku o polovinu při zachování stejné kvality, avšak za použití většího výpočetního výkonu, které je zatím pro rozsáhlé zavedení limitujícím a je třeba počkat na plošné zvýšení výkonu u přístupových zařízení, aby mohl být bez problémů zaveden a plně nahradil h.264. [8]

#### 3.1.2 Rozdíly mezi H.264/AVC a H.265/HEVC

HEVC je položen na stejné struktuře jako předchozí verze AVC avšak má mnoho přidávaných vylepšení:

Vzhledem k nově vznikajícím zobrazovacím jednotkám je zajímavá podpora až 8K UHD (ultra high definition) videa s rozlišením 8192x4320, další výhody jsou:

- Flexibilnější rozdělení bloků od velkých po malé
- Lepší flexibilita v predikčních módech a transformování velikostí bloků
- Mnohem důmyslnější interpolační a „protiblokové“ filtry
- Důmyslnější predikci a signalizaci módů a pohybových vektorů
- Funkce k podpoře efektivnějších paralelních zpracování [8]



Obr. č. 3.2 – Volba bloků H.264 a H.265 [27]

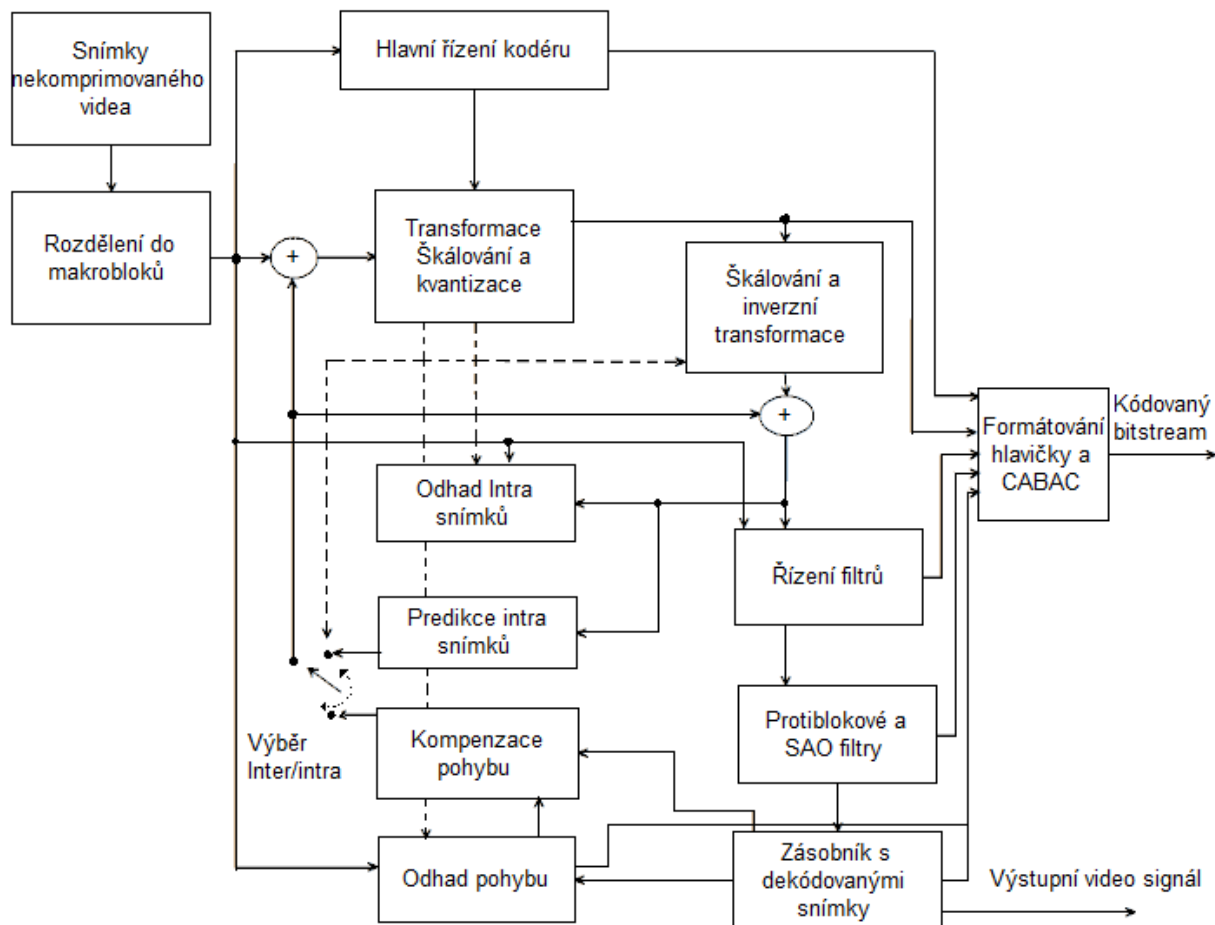
Z Obr. č. 3.2 je vidět, že H.265 volí bloky lépe, má větší rozpětí velikosti bloků.

### 3.1.3 Vývojáři standardu H.265

Tvůrci standardu je, stejně tak jako v případě H.264, spolupráce dvou standardizačních skupin, ITU-T Video Coding Experts Group (VCEG) a ISO/IEC Moving Picture Experts Group (MPEG). První verze standardu byla vydána na začátku roku 2013 a je stále ve vývoji. Pracuje se na dalších rozšířeních, například pro 3D video.



### 3.1.4 Základní blokové schéma standardu H.265



Obr. č. 3.1 Blokové schéma kodéru H.265 [26]

### 3.1.5 Princip kódování

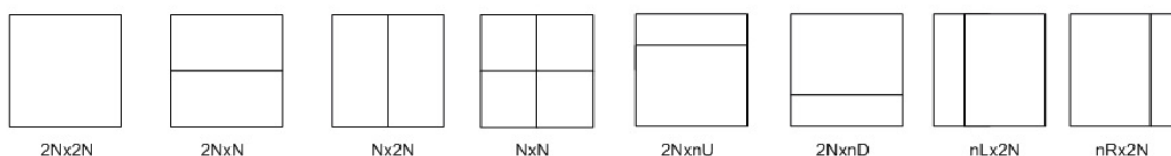
Standard HEVC je hybridní blokově-orientovaný kodek, založený na prostorové transformaci. Vstupní video snímek je nejprve rozdělen do bloků stejné velikosti, ty se nazývají makrobloky. Makroblok je rozdělen na menší bloky, aby mohla být provedena predikce. Jsou dva základní typy predikce: intra a inter. Intra-predikce pracuje přímo v aktuálním snímku a je založena na komprimovaných a dekódovaných datech pro předpovídaný blok. Inter-predikce se používá pro kompenzaci pohybu: pro predikci se používá podobná oblast předchozího snímku.

Koncept makrobloku v HEVC je představován kódovacími jednotkami CTU (Coding Tree Unit). CTU může mít velikost 16x16, 32x32 nebo 64x64. Zde je rozdíl oproti H.264 která

má pouze 16x16. Větší jednotky CTU umožňují zavedení rozvětveného dělení do menších kódových jednotek CU (coding unit). CU je nejspodnější částí stromové struktury CTU. CU obsahuje predikční jednotku PU (Prediction Unit) a transformační jednotku TU (Transform Unit).

TU je syntaktický prvek k uchování transformačních dat. CU může být rozdělena na TU ve stromové struktuře na nejmenší možnou. Povolené velikosti jsou 32x32, 16x16, 8x8 a 4x4 s ohledem na velikost DFT (Discrete Fourier Transform) matice.

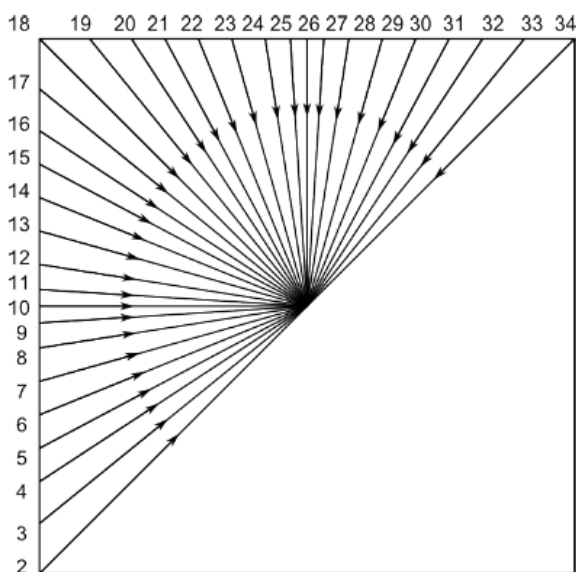
PU je syntaktický prvek který uchovává predikční data jako intra-predikční úhel nebo inter-predikční pohybový vektor. CU může obsahovat až čtyři predikční jednotky. Rozdělení CU na PU může být  $2N \times 2N$ ,  $2N \times N$ ,  $N \times 2N$ ,  $N \times N$ ,  $2N \times nU$ ,  $2N \times nD$ ,  $nL \times 2N$  a  $nR \times 2N$  viz tento obrázek:



Obr.3.3 Rozdělení CTU na CU H.265 [11]

Kde  $2N$  je velikost CU která je rozdělována. V intra-predikčním módu je povoleno pouze  $2N \times 2N$  PU.

HEVC má 35 Intra-predikčních módů: planar (mode 0), DC (mode 1) a 33 úhlových (mode 2-34) viz obrázek (3.4)



Obr 3.4 Predikční módy H.265 [11]

Na rozdíly, které dostaneme po odečtení predikovaných pixelů, provedeme frekvenční transformaci a kvantizaci. HEVC používá celočíselnou aproximaci diskrétní kosinové transformace a diskrétní sinové transformace (DCT II a DST VI). Transformace je určena pro matice s rozměry pixelů 32x32, 16x16, 8x8 a 4x4. DST se používá pro 4x4 intra kódování bloků. [11]

## 3.2 Referenční software

### 3.2.1 Základní popis

Jak již bylo zmíněno u popisu referenčního software H.264/AVC, tyto dva programy jsou si velmi podobné, kompilace se tedy provede obdobně. [9]

### 3.2.2 Instalace a kompilace

Referenční software lze stáhnout ze serveru:

[https://hevc.hhi.fraunhofer.de/svn/svn\\_HEVCSoftware/](https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/)

V našem případě použijeme vytvořený projekt pro prostředí MS Visual studio 9 s názvem a umístění: build/HM\_vc9.sln

### 3.2.3 Ovládání kodéru

Ovládání je obdobné, přes příkazový řádek. Najdeme cestu k souboru „TappEncoder.exe“ a spustíme jej,

Obecné zadávání vypadá takto:

```
TAppEncoder [-h] [-c config.cfg] [--parametr=hodnota]
```

Kde se nám při volbě `-h` zobrazí na obrazovku použití parametrů

Volbou `-c` a názvem souboru, zvolíme daný konfigurační soubor

Jednotlivé parametry a jejich hodnotu zadáváme takto: `--parametr=jeho hodnota`

## 3.3 Parametry kodéru

### 3.3.1 Nastavení souborů

InputFile	Zvolení požadovaného YUV souboru který budeme kódovat, musí být ve formátu 4:2:0, pokud je bitová hloubka větší než 8bitů, je každý vzorek kódován ve 2 bytech
BitstreamFile	zvolíme jméno výstupního zakódovaného videa, bitstreamu který program vytvoří
ReconFile	zvolíme název souboru rekonstruovaného videa po kompresi, výsledek bude ve formátu YUV
SourceWidth	zadáme šířku vstupního videa
SourceHeight	zadáme výšku vstupního videa

InputBitDepth	zadáme bitovou hloubku vstupního videa
InternalBitDepth	zadáme výstupní bitovou hloubku, pokud se liší od vstupní, je automaticky konvertována podle vzorce (3.1)
OutputBitDepth	zvolíme bitovou hloubku rekonstruovaného videa
FrameRate	zadáme počet snímků za vteřinu u vstupního videa
FrameSkip	zadáme počet snímků, které se od začátku vstupního videa přeskočí
FramesToBeEncoded	zadáme počet snímků, které budeme kódovat

$$IBD = \frac{Pel * 2^{InternalBitDepth}}{2^{InputBitDepth}} \quad (3.1)$$

### 3.3.2 Volba profilu a úrovně

Profile	zadání profilu, kterým bude kodér vytvářet výsledný bitstream, můžeme zvolit tyto varianty: none, main, main10, main-still-picture
Level	volba úrovně, kterou bude kodér vytvářet výsledný bitstream, můžeme zvolit tyto varianty: none, 1, 2, 2.1, 3, 3.1, 4, 4.1, 5, 5.1, 5.2, 6, 6.1, 6.2. Nemá význam, pokud jsme zvolili profil
Tier	volba stupně, kterou bude kodér vytvářet výsledný bitstream, možné hodnoty jsou: main, high

### 3.3.3 Volba makrobloků

MaxCUWidth	zvolíme maximální velikost šířky makrobloku, defaultně 64
MaxCUHeight	zvolíme maximální velikost výšky makrobloku, defaultně 64
MaxCUSize	zvolíme maximální velikost celého makrobloku
MaxPartitionDepth	zvolíme hloubku rozvětvení makrobloku, defaultně 4

QuadtreeTULog2MaxSize	zvolíme maximální velikost nejmenší části makrobloku, hodnota se zadává v logaritmické míře, defaultně 6, tedy dvojkový logaritmus 6 je 64
QuadtreeTULog2MinSize	zvolíme minimální velikost nejmenší části makrobloku, hodnota se zadává v logaritmické míře, defaultně 2, tedy dvojkový logaritmus 2 je 4
QuadtreeTUMaxDepthIntra	zvolíme hloubku větvení pro Intra frame
QuadtreeTUMaxDepthInter	zvolíme hloubku větvení pro Inter

### 3.3.4 Parametry kódovací struktury

IntraPeriod	volba periody IntraFrame, klíčových snímků, defaultně 1, to znamená nekonečnou periodu
DecodingRefreshType	zvolíme typ obnovení při dekódování u intra snímkové periody, defaultně 0, použije se I snímek, 1 použije se ten, který není IDR (pro otevřenou GOP), 2 použije se IDR (pro zavřenou GOP)
GOPSize	zvolíme velikost cyklické GOP struktury

### 3.3.5 Parametry odhadu pohybu

FastSearch	povolí či zakáže rychlé hledání pohybu, defaultně zapnuto, 0 metoda plného vyhledávání, 1 metoda rychlého vyhledávání
SearchRange	zvolíme rozsah hledání pro odhad pohybu, defaultně 96, rozsah hledání je určen podle prediktoru. Pohybové vektory odvozené podle odhadu pohybu tak mohou mít větší hodnotu než vyhledávacího rozsahu.
BipredSearchRange	defaultně nastaveno 4, volba vyhledávacího rozsahu použitá pro upřesnění bipredikce v odhadování pohybu
HadamardME	defaultně nastaveno, povolí či zakáže použití Hadamardovy transformace v odhadu pohybu „fractional_pel“
ASR	defaultně zakázáno, povolí či zakáže použití adaptivních vyhledávacích rozsahů, kde je pohybový vyhledávací rozsah dynamicky upraví podle rozdílu POC aktuálního a referenčního snímku

### 3.3.6 Parametry výběru módu

LambdaModifierN	defaultně 1.0, určuje hodnotu Lagrangiánu $\lambda$ , pro použití při optimalizace a výpočtu rychlosti kódování. N může být v rozsahu 0 až 7.
FEN	defaultně nenastaveno. Povoleno či zakázáno pro použití rychlého kódovacího módu. Pokud je povoleno, dojde u výpočtu SAD pro bloky s velikostí větší než 8, jsou linky považovány za řádky bloku.
FDM	defaultně nastaveno, povolí nebo zakáže používání rychlého kódování se sloučením $2N \times 2N$ . Pokud je povoleno, RD pro aktuálního kandidáta není vyhodnoceno, pokud merge_skip mód měl lepší volbu pro jednoho z předchozích kandidátů. Počet opakování použitých u obousměrného pohybového vektoru je snížena ze 4 na 1.
RDpenalty	defaultně hodnota 0, tedy nenastaveno. Vyloučí RD pro části makrobloku o rozměrech $32 \times 32$ pro intra snímky které nejsou intra. Povolení tohoto parametru může odstranit viditelnost hranic jednotlivých makrobloků kódovaného videa. Pro 0 není nastaveno, pro 1 je nastaveno, pro 2 vyloučí $32 \times 32$ .

### 3.3.7 Kvantizační parametry

QP	nastaví základní hodnotu kvantizačního parametru, defaultně 30
CbQpOffset/CrQpOffset	defaultně nastaveno 0. Posunutí kvantizačního parametru jasové složky k odvození kvantizačních parametrů složek Cb respektive Cr. Tyto možnosti odpovídají hodnotám cb_qp_offset a cr_qp_offset, které jsou přenášeny v PPS. Platné hodnoty jsou v rozsahu [12,12]
RDOQ	defaultně zapnuto, povolí nebo zakáže omezení toku optimalizované kvantizace.
RDOQTS	defaultně zapnuto, povolí nebo zakáže omezení toku optimalizované kvantizace pro části makrobloku, které byly transformačně přeskočeny.
DeltaQpRD	defaultně nastaveno 0, určuje maximální posun kvantizačního parametru při snímkové úrovni pro multi-pass kódování. Při kódování je každý snímek testován několikrát pomocí hodnoty v rozsahu [-DeltaQpRD, DeltaQpRD] a nejlepší hodnota je vybrána jako snímek QP.
dQPFile	určuje soubor obsahující seznam delt kvantizačních parametrů. N-tý řádek (kde n je nula pro první řádek) tohoto souboru odpovídá hodnotě kvantizačního parametru pro snímek s POC hodnota n.
AdaptiveQpSelection	určuje, zda bude vypočtena hodnota kvantizačního parametru pro jiné než 1 snímky na základě statistik dříve vytvořených snímků
RecalculateQPAccordingToLambda	defaultně není nastaveno, přepočítává hodnoty kvantizačních parametrů podle hodnot lambda. Nedoporučuje se povolovat ve všech případech pro intra.



### 3.3.8 Parametr entropického kódování

SBACRD	defaultně nastaveno, povoluje či zakazuje používání kódování při rozhodování ohledně datového toku.
--------	---

### 3.3.9 Parametry kódování snímků

SliceMode	defaultně nastaveno 0, ovládá metody dělení snímků ve spojení se SliceArgument. 0 samostatný snímek, 1 maximum makrobloků na snímek, 2 maximum bytů na snímek, 3 maximum bloků na snímek
SliceArgument	Určuje maximální počet makrobloků, bytů nebo bloků v závislosti na nastavení SliceMode
SliceSegmentMode	defaultně 0, určuje závislost kódování části snímku, 0 samotný snímek, 1 maximum makrobloků části snímku, 2 maximum bytů na část snímku, 3 maximum bloků na část snímku
SliceSegmentArgument	Určuje maximální počet makrobloků, bytů nebo bloků v závislosti na nastavení SliceSegmentMode
WaveFrontSynchrono	defaultně vypnuto, umožňuje používání konkrétní CABAC pravděpodobnostní synchronizace na začátku každého řádku makrobloku, za účelem vytvoření datového toku, který může být kódován či dekódován za použití jednoho nebo více jader.

### 3.3.10 Parametry protiblokových filtrů

LoopFilterDisable	defaultně vypnuto, povoluje či zakazuje deblokovací filtr in-loop
LFCCrossSliceBoundaryFlag	defaultně povoleno, povoluje či zakazuje použití filtru in_loop skrze hranice jednotlivých snímků.
DeblockingFilterControlPresent	defaultně vypnuto, povoluje či zakazuje přítomnost kontrolních parametrů protiblokového filtru, v nastavení parametrů snímku, v jeho hlavičce. Pokud je vypnuto, jsou použity defaultní neblokovací filtry.

### 3.3.11 Parametry nástrojů kódování

WeightedPredP	defaultně vypnuto, umožňuje použít váženou predikci v P_snímcích
WeightedPredB	defaultně vypnuto, umožňuje použít váženou predikci v B_snímcích
SignHideFlag	defaultně zapnuto, pokud je povoleno, udává každému koeficientu skupiny 4x4, pro které je počet koeficientů mezi první nenulovou a poslední nenulovou hodnotou podél snímané řady větší než 4, znaménkový bit prvního nenulového koeficientu nebude přenesen do bitového toku, ale bude odvozen podle parity.

## 4 Standard VP9

### 4.1 Přehled

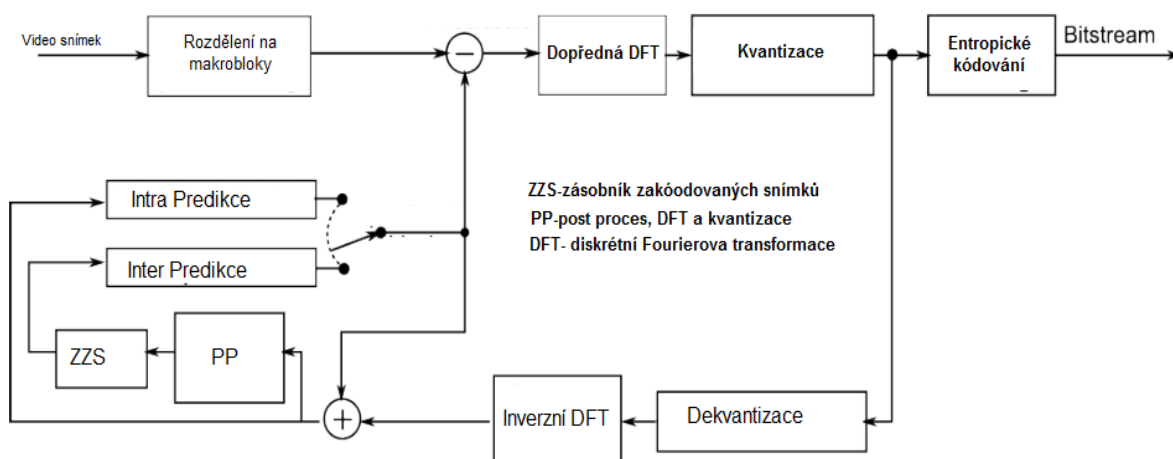
#### 4.1.1 Základní údaje

VP9 je moderní standard vyvíjený jako Open Source, tedy otevřený software, což znamená, že je plně dostupný jeho zdrojový kód zdarma i s licencí a možností kód jakkoliv modifikovat. Je přímým nástupcem standardu VP8, se kterým se můžeme setkat u služeb od firmy Google, například YouTube. Podpora byla přidána v druhé polovině roku 2013 i do webového prohlížeče Google Chrome. Snižuje údajně datový tok na polovinu při zachování stejné kvality oproti standardu VP8. Nabízí možnost Ultra HD videa. Použitý kontejner je založen na systému Matroška.[10]

#### 4.1.2 Vývojáři standardu

Tvůrcem standardu je firma Google, veškerý vývoj a nové úpravy jsou volně dostupné všem na webových stránkách [www.webmproject.org](http://www.webmproject.org). Tam je možné si stáhnout aktuální verzi kodéru a zjistit všechny nové události ohledně tohoto standardu [10].

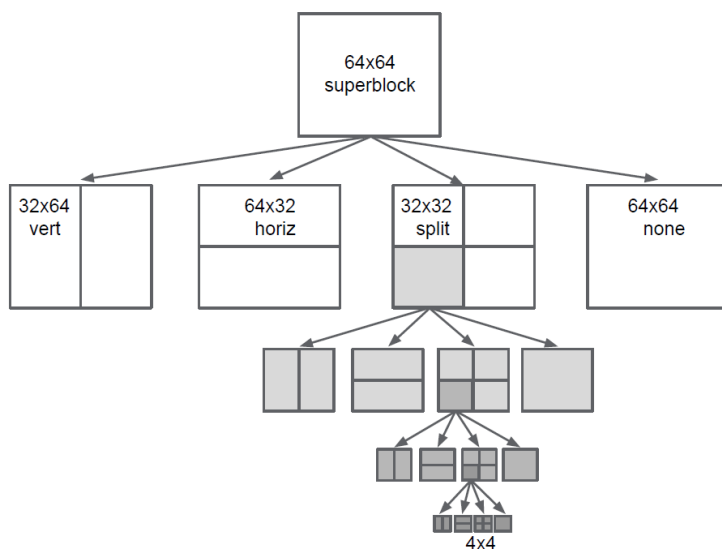
#### 4.1.3 Základní blokové schéma



Obr 5.1 Blokové schéma kodéru VP9 [11]

#### 4.1.4 Princip kódování

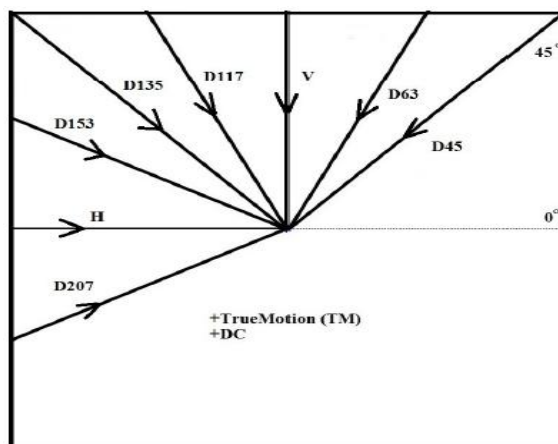
Kodér VP9 je v určitých ohledech velmi podobný standardu HEVC, jedná se také o hybridní blokově orientovaný kodek založený na prostorové transformaci. Koncept makrobloků je u VP9 reprezentován takzvaným superblokem. Jeho velikost je 64x64 pixelů a jeho dělení je velice podobné jako dělení CTU u standardu HEVC, také používá stromovou strukturu, sub-bloky můžou být rozděleny až na velikost 4x4. Viz tento obrázek:



Obr. 5.2 Rozdělení superbloku VP9 [28]

Každý sub-blok může být dále rozdělen na predikční bloky a transformační bloky. Na rozdíl od HEVC, každý sub-blok může být rozdělen na predikční bloky v intra módu. Mimoto jsou možné obdélníkové intra-predikční bloky. Intra-predikce se ve VP9 stále provádí na čtvercových plochách, tedy obdélníkové predikční bloky představují dva čtvercové bloky se stejným predikčním módem. Predikční dělení může být 2Nx2N, NxN, 2NxN nebo Nx2N. Kde 2Nx2N je velikost děleného bloku. Jako v případě HEVC, sub-bloky můžou být rozděleny do transformačních bloků ve stromové struktuře až po blok o rozměrech 4x4. Povolené matice DFT mají rozměry 32x32, 16x16, 8x8 a 4x4

VP9 má celkem 10 intra-predikčních módů: DC, True Motion (TM), a 8 úhlových módů (H, V, D207, D153, D135, D117, D63 a D45).



Obr 5.3 Predikční módy VP9 [11]

Na rozdíl po odečtení predikčních hodnot je provedena Diskrétní Fourierova Transformace a kvantizace. Transformační bloky mohou být 32x32, 16x16, 8x8, nebo 4x4 pixely. VP9 používá celočíselnou transformaci DCT II a DST II pro všechny transformační velikosti vyjma 4x4 kde se používá DST VI. VP9 zavádí podporu pro nový druh transformace Asymetric Discrete Sine Transform (ADST), která může být použita v kombinaci s určitými predikčními módy. Intra-predikční módy, které jdou z levé strany, mohou použít 1-D ADST v horizontálním směru v kombinaci s 1-D DCT ve vertikálním směru. Obdobně rozdílové signály intra-predikčních módů, které jdou od vrchní strany, mohou použít vertikálně 1-D ADST transformaci v kombinaci s horizontální 1-D DCT transformací.

Jako entropické kódování používá VP9 8-bitový, aritmetický nástroj známý z VP8 jako Bool-coder. Na rozdíl od AVC nebo HEVC, se pravděpodobnost VP9 bool-coderu nemění adaptivně se snímky. VP9 využívá dopředný kontext aktualizovaný prostřednictvím použití ukazatelů v záhlaví rámečku, kteří signalizují změny kódového obsahu na začátku každého snímku. Tyto pravděpodobnosti jsou uchovávány ve „frame kontext“. Dekodér udržuje čtyři z těchto kontextů a každý snímek specifikuje, který z nich bude použit v bitstramu.

Ve VP9 je pouze jedna fáze post procesu a to deblokovací filtr. Ten má za cíl snížit nejdříve viditelné přechody mezi vertikálními hranami superbloků, poté horizontálními. VP9 má 16,8,4 a 2-pixelově široké filtry s poloviční velikostí filtru na každé straně hranice. VP9 také obsahuje rovinný detektor v look filtru který rozpozná ploché oblasti a podle toho mění sílu a velikost filtru. [11]

## 4.2 Referenční software

### 4.2.1 Popis

Jak již bylo zmíněno, jedná se o otevřený software. Aktuální verze zdrojových kódů má název libvpx a číslo verze. Ke stažení je na oficiálním webu <http://www.webmproject.org/>. Je koncipován zejména pro prostředí Linux, je však možné jej zkompilovat i pro windows.[12]

### 4.2.2 Kompilace referenčního software

Kompilace je poněkud obtížnější než u předchozích standardů. Zejména pro to, že je software cílený pro uživatele Linux. S vhodnými emulačními programy Linuxového prostředí je ale možné program zkompilovat i ve Windows. Na domovské stránce projektu WebM je popsáno několik postupů jak toho docílit. Je zde popsán postup s využitím MS Visual Studio, kde je potřeba nainstalovat komponentu Yasm, nebo použitím programu MSYS který emuluje Linuxové prostředí. Případně je možné stáhnout již zkompilovanou verzi jako spustitelný soubor, avšak bez možnosti editace kódu.

### 4.2.3 Nastavení parametrů

Nastavení základních parametrů je stejné jako pro VP8, některé jsou pouze pro VP9.

Output	nastavení cesty a názvu výstupního souboru
codec	volba kodeku, pro vp9 se vyplní vpxenc-vp9
passes	volba mezi jedno či dvoupásmovým kódováním
limit	po n-tém snímku se zastaví kódování
skip	přeskočení prvních n snímků
i420	použití při načítání videa ve formátu i420
threads	počet použitých vláken
profile	volba mezi profilem 1 a 0
width	nastavení šířky videa
height	nastavení výšky videa
fps	počet snímků za vteřinu
lag_in_frames	0 až 25, maximální počet snímků na které se kodér „dívá dopředu“
target_bitrate	požadovaný bitrate
min_q	minimální kvantizační parametr
max_q	maximální kvantizační parametr
minsection_pct	minimální bitrate GOP
maxsection_pct	maximální bitrate GOP
kf_min_dist	minimální zvolení další klíčového snímku
kf_max_dist	maximální zvolení klíčových snímků
disable_kf	vypnutí řízení klíčových snímků
cpu_used	v rozmezí 16 až 16 defaultně 0

## 4.2.4 Ovládání

Ovládání je jako v předchozích referenčních programech přes příkazovou řádku

```
C:\FFMPEG\bin\vp9enc.exe -r 24 -s cif -i flowerORIG.yuv -vcodec vp9 -r 24 -
```

### **Příklad nastavení**

```
vp9enc input_1280_720_30fps.yuv -o output_vp8.webm \  
--codec=vp9 --i420 -w 1280 -h 720 -p 2 -t 4 \  
--good --cpu-used=1 --target-bitrate=2000 --end-usage=vbr \  
--auto-alt-ref=1 --fps=30000/1001 -v \  
--minsection-pct=5 --maxsection-pct=800 \  
--lag-in-frames=16 --kf-min-dist=0 --kf-max-dist=360 \  
--token-parts=2 --static-thresh=0 \  
--min-q=0 --max-q=60
```



# 5 Objektivní metody pro srovnání kvality videa

## 5.1 Základní údaje

Nejspolehlivějším vyhodnocováním kvality obrazu nebo videa je subjektivní měření kvality. Tedy přímé hodnocení lidmi, protože ten kdo bude daný obsah ve výsledku pozorovat je právě člověk. Tato metoda se nazývá MOS (Mean Opinion Score). Je tím spolehlivější, čím je větší počet pozorovatelů, účastníků měření, kteří hodnotí danou kvalitu. Tato metoda je ale velmi zdlouhavá a nákladná. Proto se zavedlo Objektivní měření kvality, které je levnější. Cílem výzkumu objektivního hodnocení kvality je navrhnout metriky kvality tak, aby automaticky předpovídaly vnímanou kvalitu obrazu a videa. Obecně řečeno, objektivní měření kvality může být použito pro tři aplikace.

K **monitorování** kvality obrazu pro řídicí systémy, například k nastavení snímacích systémů jako je kamera, k nastavení kvality a automatickému přizpůsobení pro získání nejlepší kvality. Nebo síťový server, pro zkoumání kvality přenášeného videa nebo řízení streamování videa.

K **optimalizaci** systémů, které zpracovávají video, jako např. algoritmy enkodéru pro nejlepší nastavení parametrů pro kódování videa, jako jsou filtry nebo řízení datového toku.

K **benchmarku** neboli systematickému porovnávání kvality systémů a algoritmů zpracovávajících video. A dává tak možnost porovnat, který vykazuje nejlepší výsledky. Právě tuto aplikaci využívám v této práci pro porovnání jednotlivých videostandardů.

Ve své práci používám pro srovnání tři metody: PSNR, SSIM a VIFP. V následující části jsou popsány.

## 5.2 MSE

Zkratka anglického termínu Mean Squared Error, tedy střední kvadratická chyba, je velmi jednoduché ji určit podle rovnice 6.1

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (I_{(i,j)} - K_{(i,j)})^2 \quad (6.1)$$

Kde  $m$  je počet sloupců,  $n$  počet řádků,  $I$  je referenční snímek a  $K$  je snímek po kompresi.

Postupně odečte každý pixel referenčního snímku od pixelu na stejné pozici v komprimovaném snímku, umocní se na druhou a přičte se k ostatním, výsledek se pak už jen podělí počtem pixelů a vznikne průměrná hodnota, tento výpočet se dělá zvlášť pro jasovou a zvlášť barevné složky, podle potřeby se buď určí hodnota pro jednotlivé složky, nebo se zprůměrují dohromady.[15]

## 5.3 PSNR

Zkratka anglického termínu peak signal-to-noise ratio, tedy špičkový poměr signálu k šumu. Je to nejpodstatnější parametr objektivního měření kvality obrazu a videa, je definováno rovnicí (6.2)

$$PSNR = 10 \cdot \log \left( \frac{MAX^2}{MSE} \right) [dB] \quad (6.2)$$

Kde  $MAX$  určuje počet bitů, které užijeme ke kódování videa (standardně 8bitů, tedy hodnota  $MAX$  je pak 255).  $MSE$  je střední kvadratická chyba. Vzhledem k velkému rozptylu hodnot se zavedl logaritmický parametr.  $PSNR$  je aproximací lidského vnímání kvality obrazu. Obecně platí, čím vyšší hodnota, tím lepší kvalita, nemusí to však platit vždy. Typické hodnoty pro ztrátové kódování videa s 8 bitovou hloubkou jsou v rozmezí mezi 30-50 dB, pro 16 bitovou hloubku pak 60-80 dB. [15]

## 5.4 VIFP

Z anglického termínu Visual Information Fidelity in Pixel Domain. Můžeme přeložit jako vizuální informace věrnosti v pixelu.

Hodnota je odvozena od vzájemné informace mezi vstupem a výstupem HVS (Lidského vizuálního systému). Porovnává informaci, kterou by mozek dokázal zpracovat z originálního obrazu s informací, kterou by mozek zpracoval z komprimovaného obrazu. Aplikuje tedy na každý obraz zvlášť sadu filtrů, které modelují chování ztráty informace lidským mozkem. A tyto hodnoty porovná. Filtry jsou voleny na základě výzkumu subjektivního porovnávání kvality. Používá se zde vlnková transformace. Visual information Fidelity je známé také pod pojmem Sheikh-Bovik Index, pojmenovaného podle jeho autorů. [16]

Hodnota může nabývat 0 až 1, kde 1 znamená přesnou shodu. [14]

## 5.5 SSIM

Z anglického termínu Structural SIMilarity index, můžeme přeložit jako index strukturální podobnosti, počítá se pouze na jasové složce. Je sestaven na základě vlastností lidského vizuálního systému, konkrétně že zrakový vjem je citlivý nejen na samotnou změnu obrazu ale i změnu jasu a kontrastu.

Je definován rovnicí (6.3)

$$SSIM(x, y) = [l(x, y)]^a \cdot [c(x, y)]^b \cdot [s(x, y)]^c \quad (6.3)$$

Kde  $a > 0$ ,  $b > 0$ ,  $c > 0$  a umožňují poměrně měnit hodnoty, standardně  $a=b=c$ .

$x, y$  je signál původního respektive kódovaného obrazu

Jednotlivé složky  $l, c, s$  jsou definovány zvlášť.

Jasová, I jako luminance (6.4)

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \quad (6.4)$$

Složka kontrastu, c jako kontrast (6.5)

$$c(x, y) = \frac{2\rho_x\rho_y + C_2}{\rho_x^2 + \rho_y^2 + C_2} \quad (6.5)$$

Složka s, structural components (6.6)

$$s(x, y) = \frac{2\rho_{xy} + C_3}{\rho_x\rho_y + C_3} \quad (6.6)$$

Kde  $\mu_x, \mu_y$  představují originální respektive kódovaný obraz.  $\rho_x, \rho_y$  představují standardní odchylku těchto signálů a  $\rho_{xy}$  je jejich kovariancí.

Konstanty  $C_1 = (K_1L)$ ,  $C_2 = (K_2L)$ ,  $C_3 = (K_3L)$ ,  $L=2^n$  (pro b bitů tedy 256)

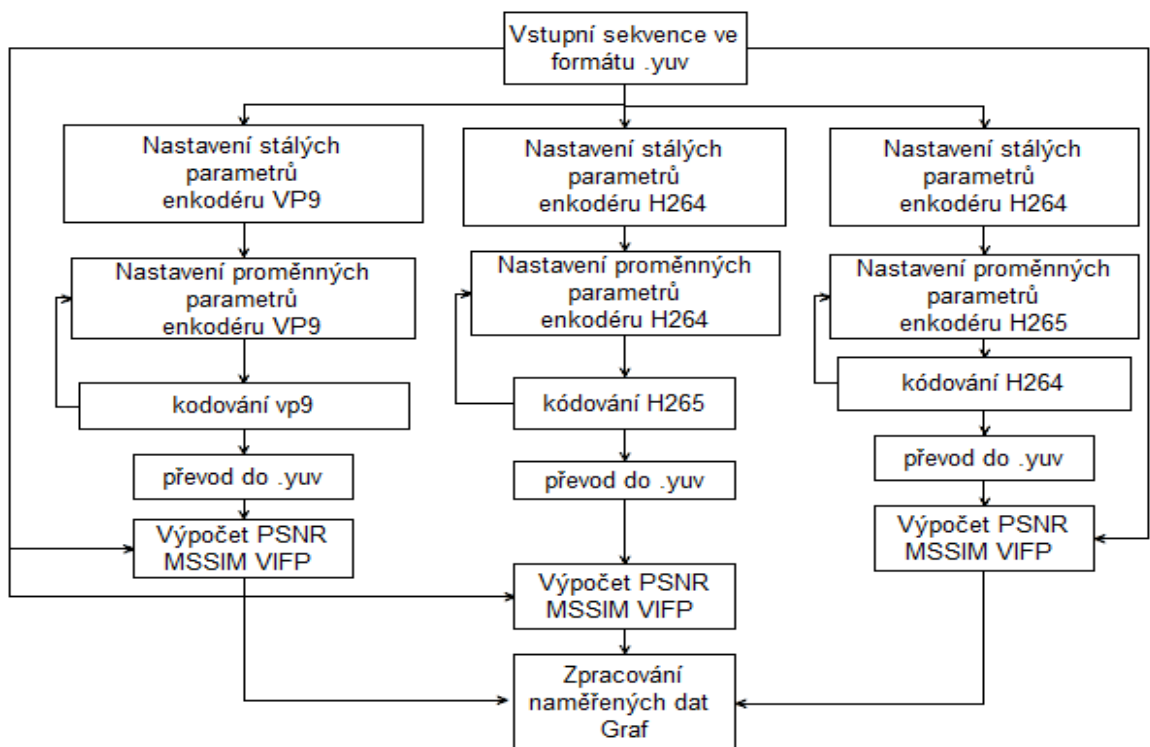
$n$ =počet bitů jasové složky,  $K_1 = 0,01$ ,  $K_2 = 0,02$   $K_3 = 0,03$

Hodnota SSIM může nabývat 0 až 1, kde 1 znamená přesnou shodu. [13]

# 6 Srovnání kompresní účinnosti jednotlivých standardů

## 6.1 Postup srovnání účinnosti

Základem je nekomprimovaná sekvence ve formátu yuv. Pro tu nastavíme parametry jednotlivých enkodérů, které zůstanou stejné, poté nastavujeme parametry proměnné. Sekvence postupně kódujeme. Zjistíme výsledný bitový tok. Každou sekvenci převedeme zpět do yuv. Pro každou takto vzniklou sekvenci vypočítáme parametry PSNR, VIFP a SSIM porovnáním s originální sekvencí. Na základě informací o bitovém toku a parametrech sestavíme grafy. Postup je na obr. č. 6.1.



Obr. 6.1- Postup srovnání účinnosti

## 6.2 Výběr sekvencí

Výběr sekvencí byl prováděn na základě jejich dostupnosti, např. sekvencí s UHD rozlišením není tolik s volně dostupnou licencí. Pro zjištění charakteru sekvence, bylo provedeno měření SI a TI parametrů.

### 6.2.1 Měření prostorových a časových informací TI a SI

Měření prostorových parametrů TI (Temporal Information) a časových parametrů SI (Spatial Information) získáme hodnotu pro každý snímek sekvence. Výsledkem je tedy časová závislost (časový průběh sekvence) na těchto parametrech. To může být užitečné třeba pro lepší odhadování stříhu ve videu. V grafech na následující straně jsou zobrazeny výpočty těchto parametrů pro všechny sekvence.

### 6.2.2 Spatial Information SI

Tento parametr nám udává míru prostorových detailů. Je většinou vyšší pro komplexnější scény. Je založen na Sobelově filtru. Každý snímek ( $F_n$ ) v čase  $n$  projde Sobelovým filtrem (Pouze jasová složka). Poté je vypočtena standardní odchylka každého pixelu oproti původnímu snímku. To je provedeno na každý snímek, maximální hodnota udává parametr SI, podle rovnice (7.1) [17]

$$SI = \max_{čas} \{std_{prostor}[Sobel(F_n)]\} \quad (6.1)$$

### 6.2.3 Temporal Information

Tento parametr udává míru časové změny videosekvence. Běžně je větší pro hodně pohyblivé scény. Je to maximum v čase  $\max_{čas}$  standardní odchylky v prostoru  $std_{prostor}$  z  $M_n(i, j)$  přes všechny  $i$  a  $j$  podle rovnice (7.2)

$$TI = \max_{čas} \{std_{prostor} [M_n(i, j)]\}$$

(6.2)

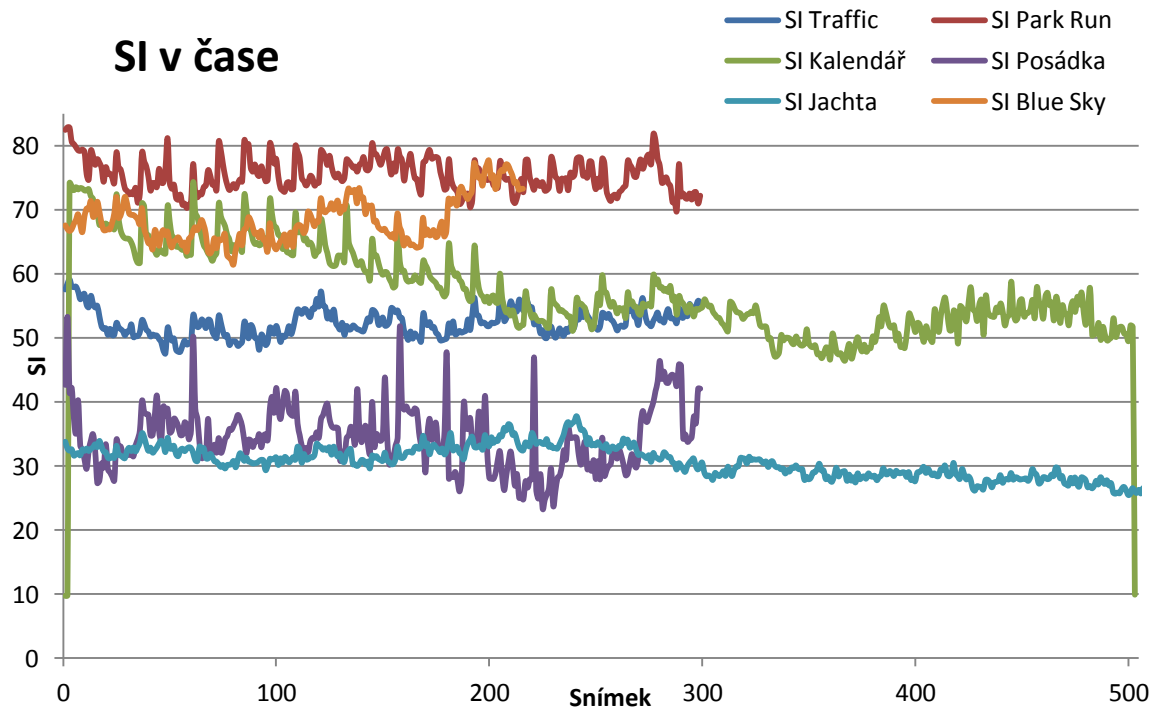
Kde  $M_n(i, j)$  je rozdíl mezi dvěma pixely na stejné pozici, dvou po sobě jdoucích snímků

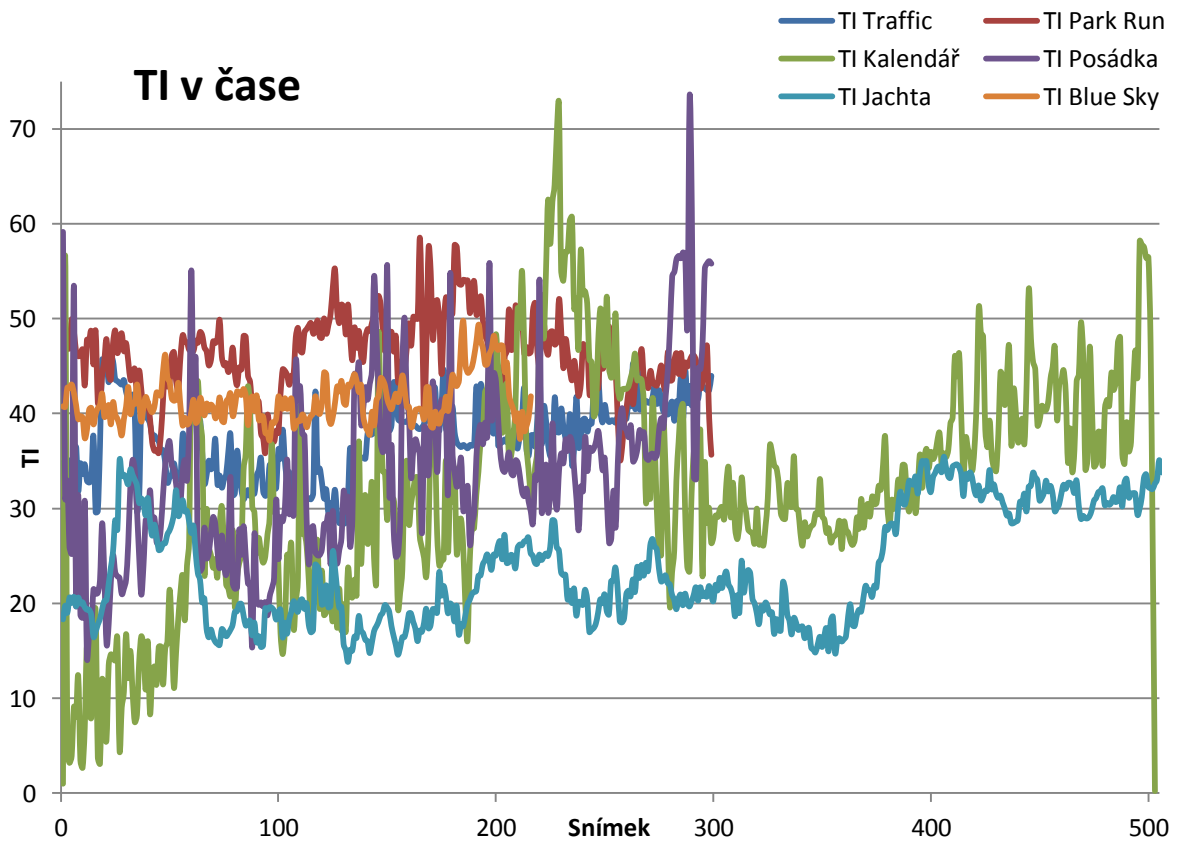
$$M_n(i, j) = F_n(i, j) - F_{n-1}(i, j)$$

(6.3)

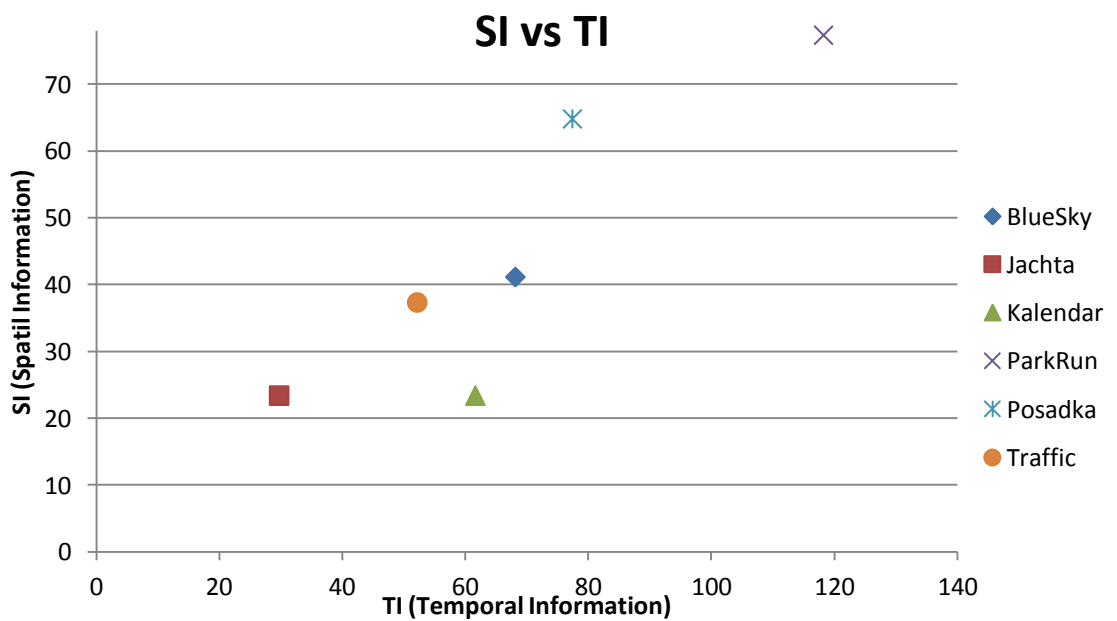
Kde  $F_n(i, j)$  je pixel i-tého řádku a j-tého sloupce snímku v čase. [17]

### 6.2.4 Průběhy SI a TI jednotlivých sekvencí





## 6.2.5 Shrnutí parametrů SI a TI všech sekvencí





V grafu „SI vs TI“ jsou uvedeny průměrné hodnoty sekvencí použitých v této práci pro porovnání jednotlivých standardů kódování videa.

	BlueSky	Jachta	Kalendar	ParkRun	Posadka	Traffic
SI	68,14	29,73	61,64	118,25	77,41	52,18
TI	41,16	23,41	36,38	77,36	64,82	37,35

Tab. 6.1 Průměrné parametry SI a TI testovaných sekvencí

### 6.3 Použitý software

Vzhledem k velmi vysoké hardwarové a časové náročnosti byla ke kódování použita volně dostupná utilita ffmpeg která obsahuje knihovny všech tří porovnávaných standardů. Kódování je však optimalizované a výrazně rychlejší než jednotlivé referenční software. Díky tomu, jsem mohl porovnat více sekvencí a provést kódování u jednotlivých sekvencí pro více hodnot. Jedná se o svobodný software vyvíjený pod systémem Linux, může být však zkompileován i pro Windows (použil jsem právě verzi pro Windows). Uživatelské rozhraní programu je stejné, jako u referenčních software, jedná se tedy také o práci v příkazovém řádku. Vzhledem k velkým počtům zadávaných údajů a dlouhému trvání kódování jsem použil dávkové soubory. Tento program jsem použil i pro zpětný převod do formátu yuv.[23]

Pro výpočet parametrů jsem použil program VQMT (Video Quality Measurement Tool ), program je volně k použití pro výzkumné účely a je ke stažení na <http://mmsp.epfl.ch/vqmt> buď jako zkompileovaný spustitelný soubor, nebo jako projekt pro Microsoft Visual Studio 2010 nebo pouze jako zdrojový kód. Ovládání programu je také přes příkazový řádek, výstupem programu je pak samostatný csv soubor pro každou zvolenou metriku.

### 6.4 Metody kódování

Pro výpočet jsem použil dvě základní metody kódování, VBR a CBR, které jsou podporovány všemi třemi standardy.

VBR (Variable Bit Rate Encoding), kde je nastavena konstantní kvalita videa po celou dobu, ale s nemožným odhadem datového toku, který závisí na charakteru videa. V prostředí FFmpeg se zadává jako parametr CRF.

CBR (Constant Bit Rate Encoding) který je koncipován tak, aby měl přibližně zadanou výslednou velikost.

Jak již bylo zmíněno v nadpisu, srovnání jednotlivých enkodérů jsem prováděl pomocí parametru CRF (Constant Rate Factor). Tato funkce je implementována ve všech třech testovaných enkodérech. Nastavení jednotlivých parametrů je tedy nejlepší možné, tedy velmi vhodné pro porovnání účinnosti. Pro h264 a h265 je nastavitelný na škále (0-51) kde 0 znamená bezztrátové kódování, tedy nejlepší možnou kvalitu a 51 nejhorší možnou kvalitu. U enkodéru VP9 to funguje obdobně, škála je však širší (0-63). Každému snímku je přidělen takový datový tok, který zaručí udržení požadované kvality.

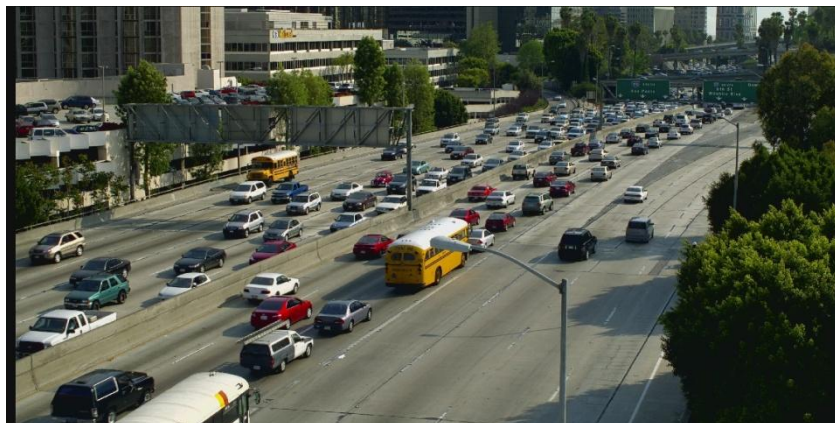
## 6.5 Volba parametrů

U metody VBR (variable Bit Rate) jednotlivé parametry jsem volil tak, aby neblokovaly nastavení hlavní parametr CRF, který podle jeho hodnoty, tak jak je popsáno v kapitole 6.4, drží kvalitu videa po celou dobu stejnou, tedy především kvantizační parametr jsem nechal v celém rozsahu. Volbu I snímku jsem nastavil defaultní, doporučenou jednotlivými manuály k referenčním software. I přes to že se ve VP9 liší od H265 a H264, po analýze videosekvencí je vidět, že volí stejné množství I snímků, například pro sekvenci Traffic 2 I snímky (z celkem 300). Hodnota Level u H264 a H265 je nastavena podle požadovaného rozlišení a snímkové frekvence pro dané kódéry. Volbu množství snímků v GOP (Group Of Picture) jsem nechal otevřenou, tedy aby si kódér vybral nejlepší možnost. Profil jsem volil nejlepší možný pro každý kódér.

U sekvence Jachta jsem použil metodu CBR, tedy konstantní bitový tok, kde jsem zvolil pro všechny tři kódéry frekvenci I-snímků na každý 60 snímek, tedy každou vteřinu. Ostatní parametry obdobně jako ve variantě VBR.

## 6.6 4K sekvence 4096x2048 s využitím parametru CRF

### 6.6.1 Popis sekvence



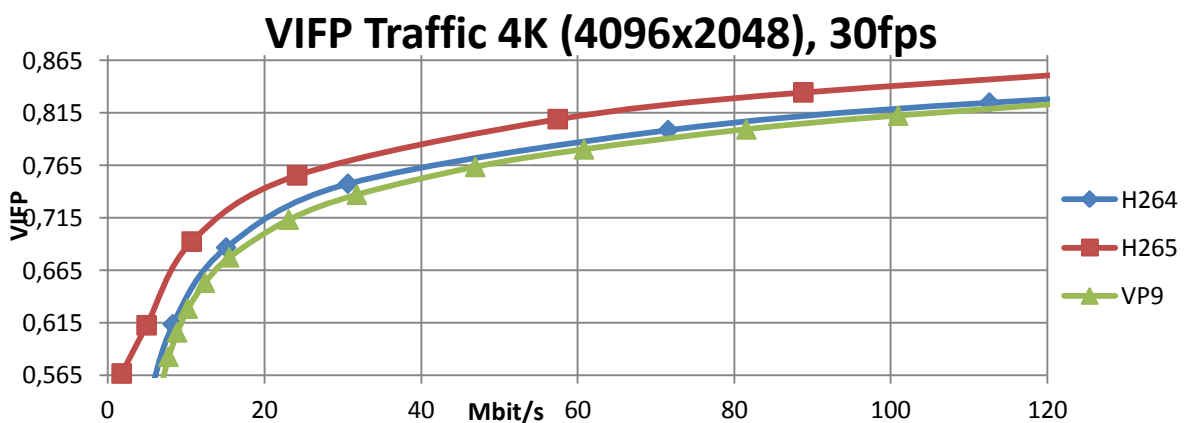
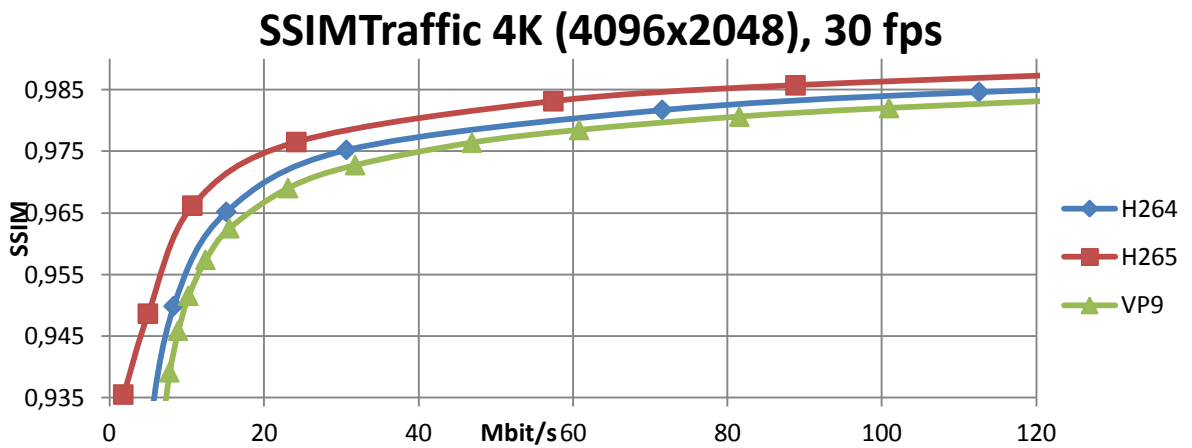
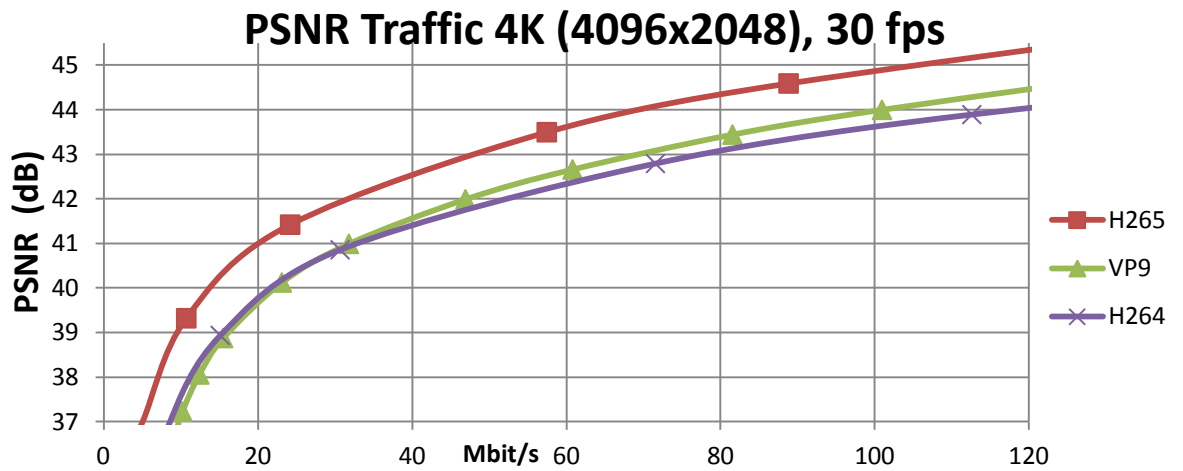
Obr. 7.2 Sekvence traffic

Název sekvence	Traffic
Rozlišení	4K UHD 4096x2048
Počet snímků za vteřinu	30, neprokládané
Celkový počet snímků	300
Formát	YUV 4:2:0
Velikost souboru	3,7 GB
Video Bitstream	2,96 Gbps

### 6.6.2 Nastavení jednotlivých enkodéru

<b>H.264</b>		<b>H265</b>		<b>VP9</b>	
Codec	libx264	Codec	libx265	Codec	libvpx-vp9
CRF	10 až 40	CRF	10 až 40	CRF	10 až 60
Profil	High	Profil	Main	Profil	0
Level	5.2	Level	6	kvalita	good
qmin	0	qmin	0	qmin	0
qmax	69	qmax	69	qmax	63
GOP	open GOP	GOP	open GOP	aut-alt-ref	1
preset	very slow				
Keyint	0-250	Keyint	0-250	Keyint	0-360

### 6.6.3 Výsledné grafy



## 6.7 HD sekvence 1280x720 č.1

### 6.7.1 Popis sekvence



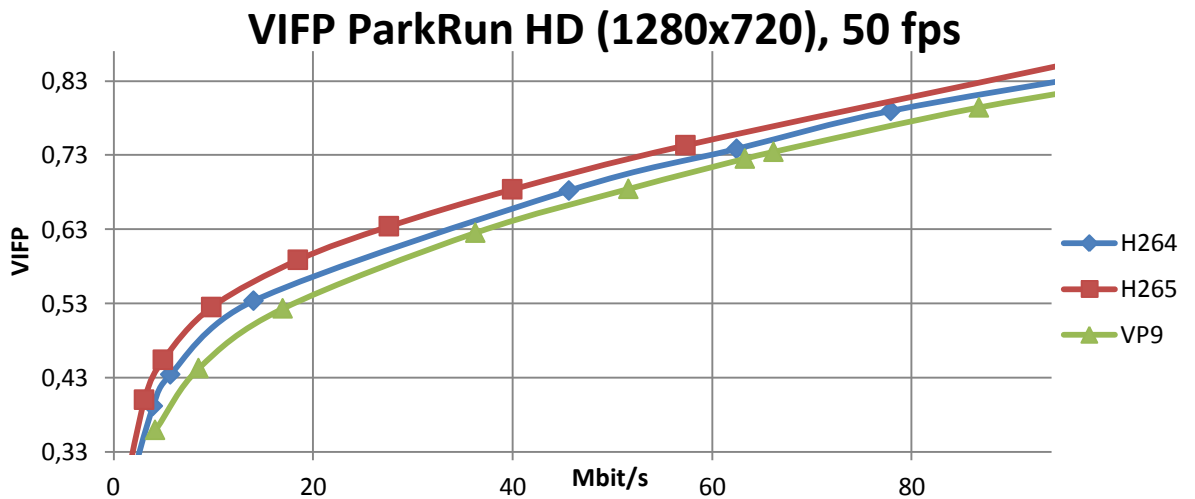
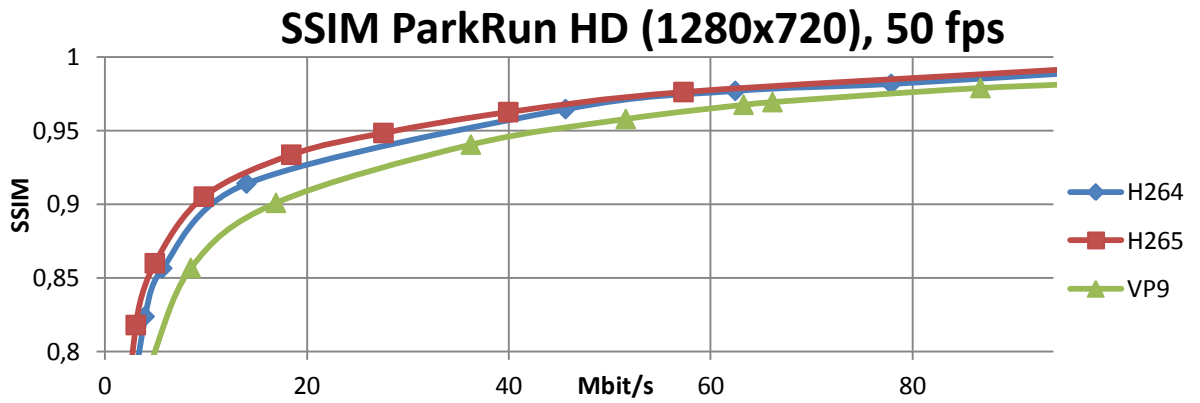
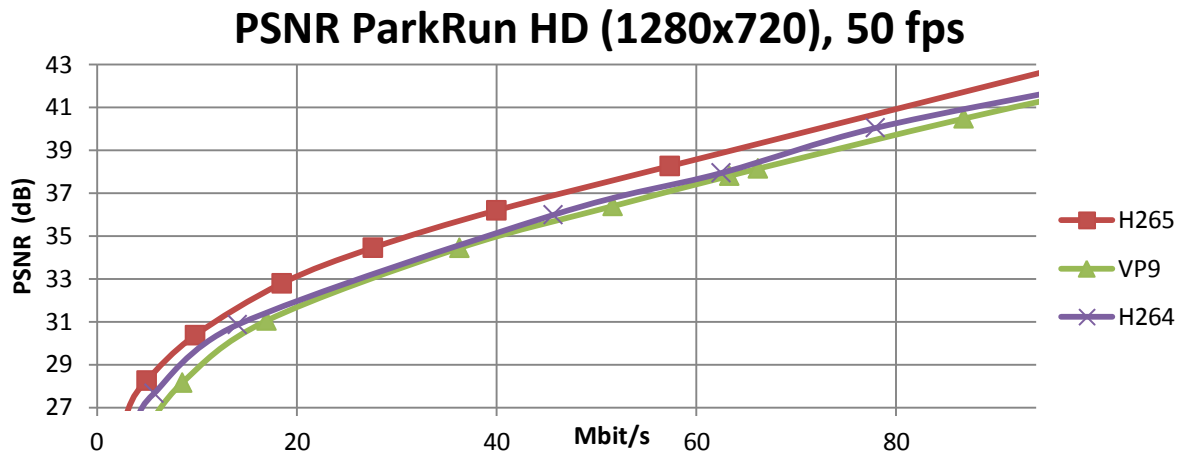
Obr. 7.3 Sekvence ParkRun

Název sekvence	ParkRun
Rozlišení	HD 1280x720
Počet snímků za vteřinu	50, neprokládané
Celkový počet snímků	300
Formát	YUV 4:2:0
Velikost souboru	405 MB
Video Bitstream	540 Mbit/s

### 6.7.2 Parametry sekvence

<b>H.264</b>		<b>H265</b>		<b>VP9</b>	
Codec	libx264	Codec	libx265	Codec	libvpx-vp9
CRF	10 až 40	CRF	10 až 40	CRF	10 až 60
Profil	High	Profil	Main	Profil	0
Level	4	Level	4.1	kvalita	good
qmin	0	qmin	0	qmin	0
qmax	69	qmax	69	qmax	63
GOP	open GOP	GOP	open GOP	aut-alt-ref	1
preset	very slow				
Keyint	0-250	Keyint	0-250	Keyint	0-360

### 6.7.3 Výsledné grafy



## 6.8 HD sekvence 1280x720 č.2 s využitím parametru CRF

### 6.8.1 Popis sekvence



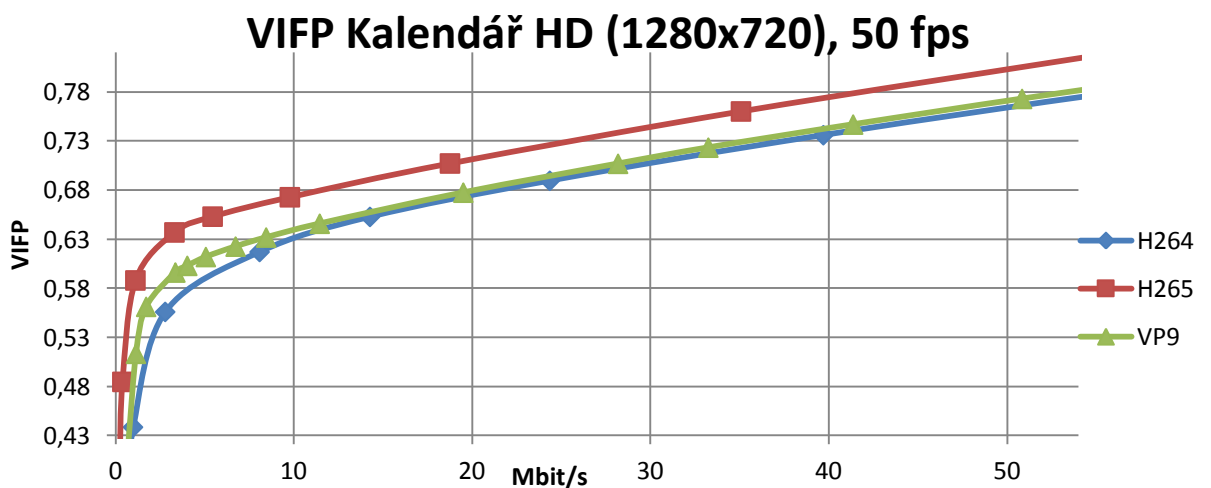
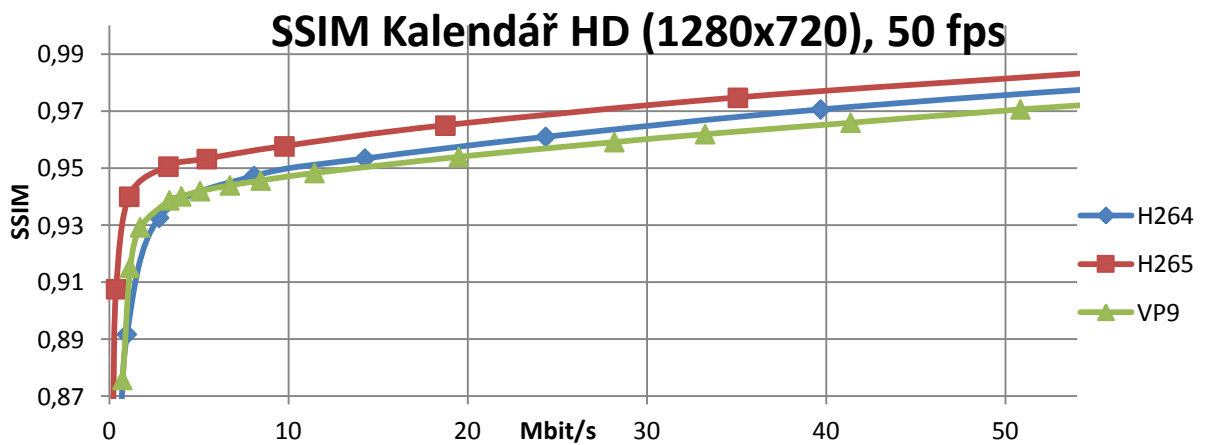
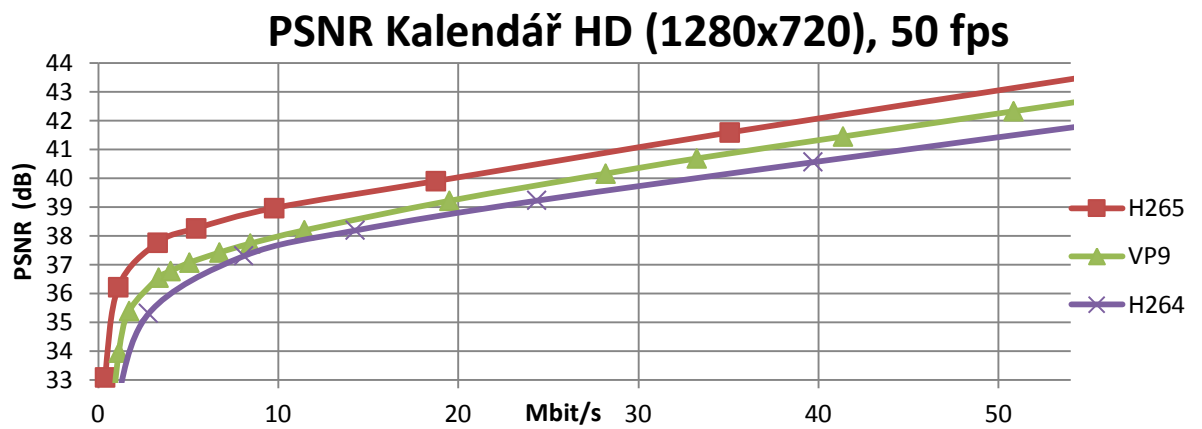
Obr. 7.4. Kalendář HD

Název sekvence	Kalendar
Rozlišení	HD 1280x720
Počet snímků za vteřinu	50, neprokládané
Celkový počet snímků	499
Formát	YUV 4:2:0
Velikost souboru	674 MB
Video Bitstream	540 Mbit/s

### 6.8.2 Nastavení parametrů

<b>H.264</b>		<b>H265</b>		<b>VP9</b>	
Codec	libx264	Codec	libx265	Codec	libvpx-vp9
CRF	10 až 40	CRF	10 až 40	CRF	10 až 60
Profil	High	Profil	Main	Profil	0
Level	4	Level	4.1	kvalita	good
qmin	0	qmin	0	qmin	0
qmax	69	qmax	69	qmax	63
GOP	open GOP	GOP	open GOP	aut-alt-ref	1
preset	very slow				
Keyint	0-250	Keyint	0-250	Keyint	0-360

### 6.8.3 Výsledné grafy





## 6.9 SD sekvence 704x576 s využitím parametru CRF

### 6.9.1 Popis sekvence



Obr. 7.5 Posadka SD

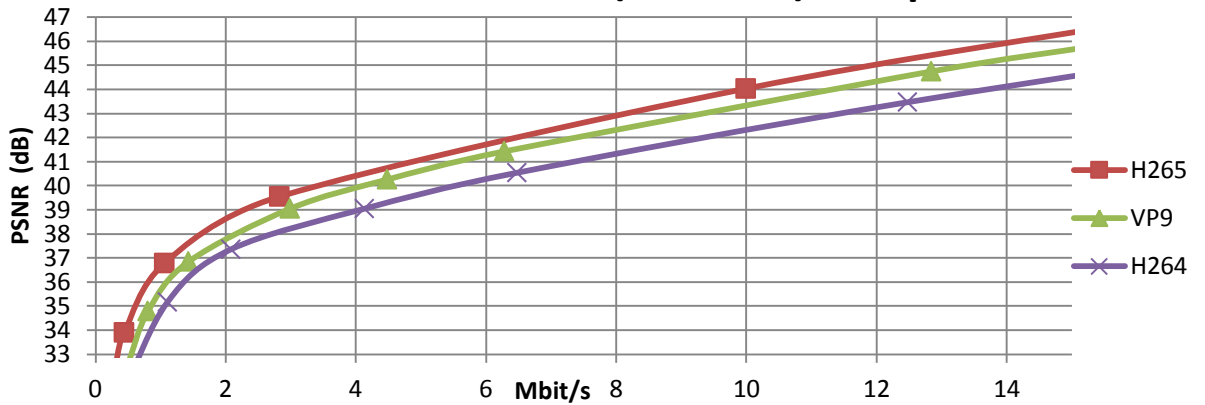
Název sekvence	Posadka
Rozlišení	SD 704x576
Počet snímků za vteřinu	30, neprokládané
Celkový počet snímků	300
Formát	YUV 4:2:0
Velikost souboru	178 MB
Video Bitstream	143 Mbit/s

### 6.9.2 Nastavení enkodéru

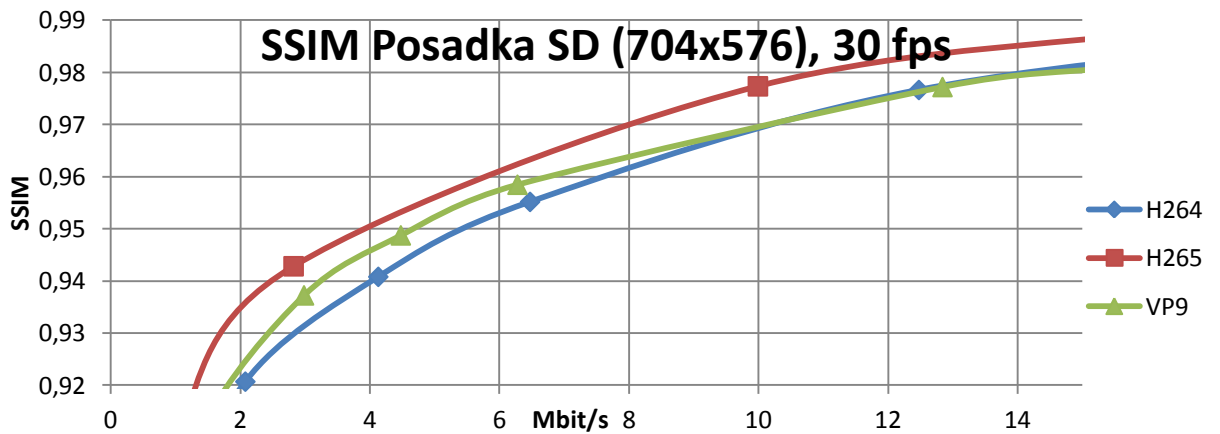
<b>H.264</b>		<b>H265</b>		<b>VP9</b>	
Codec	libx264	Codec	libx265	Codec	libvpx-vp9
CRF	10 až 40	CRF	10 až 40	CRF	10 až 60
Profil	High	Profil	Main	Profil	0
Level	3	Level	3	kvalita	good
qmin	0	qmin	0	qmin	0
qmax	69	qmax	69	qmax	63
GOP	open GOP	GOP	open GOP	aut-alt-ref	1
preset	very slow				
Keyint	0-250	Keyint	0-250	Keyint	0-360

### 6.9.3 Výsledné grafy

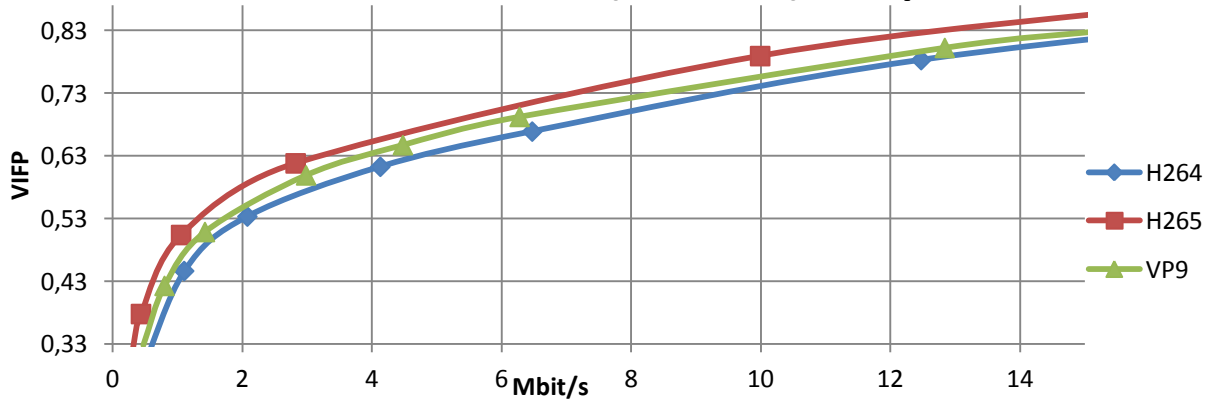
#### PSNR Posadka SD (704x576), 30 fps



#### SSIM Posadka SD (704x576), 30 fps



#### VIFP Posadka SD (704x576), 30 fps



## 6.10 4K sekvence s využitím řízení bitového toku

### 6.10.1 Popis sekvence



Obr. č. 7.6 Jachta 4K

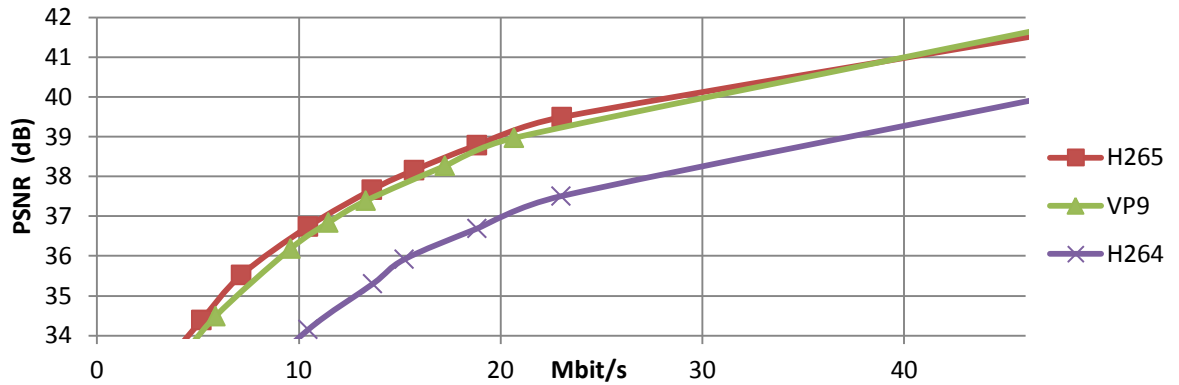
Název sekvence	Jachta
Rozlišení	4K UHD 3840x2160
Počet snímků za vteřinu	60, neprokládané
Celkový počet snímků	600
Formát	YUV 4:2:0
Velikost souboru	7,29 GB
Video Bitstream	5,83 Gbit/s

### 6.10.2 Nastavení parametrů

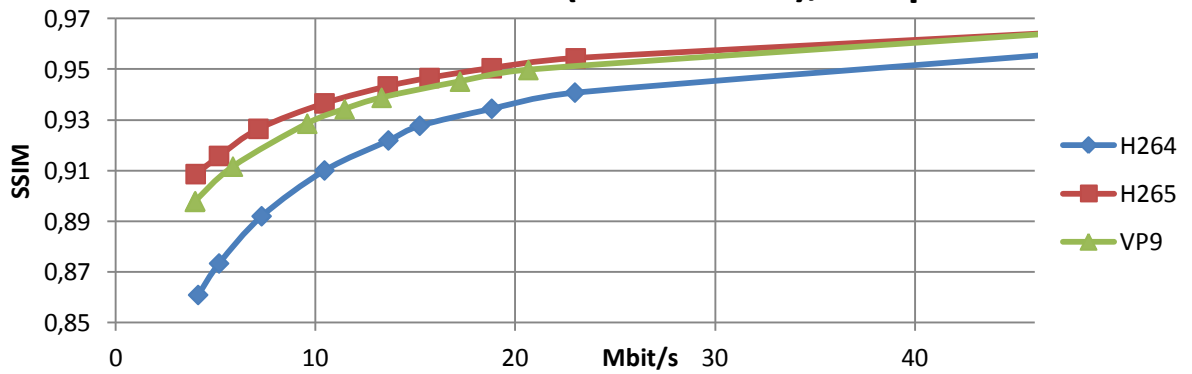
<b>H.264</b>		<b>H265</b>		<b>VP9</b>	
Codec	libx264	Codec	libx265	Codec	libvpx-vp9
CRF	10 až 40	CRF	10 až 40	CRF	10 až 60
Profil	High	Profil	Main	Profil	0
Level	5.2	Level	6	kvalita	good
qmin	0	qmin	0	qmin	0
qmax	69	qmax	69	qmax	63
GOP	open GOP	GOP	open GOP	aut-alt-ref	1
preset	very slow				
Keyint	60	Keyint	60	Keyint	60

### 6.10.3 Výsledné grafy

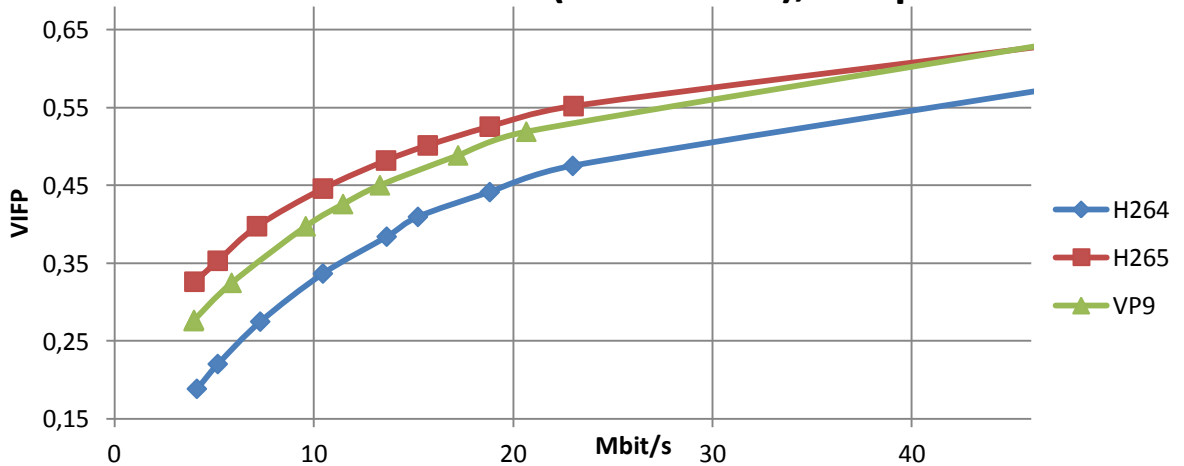
#### PSNR Jachta HD (3840x2160), 60 fps



#### SSIM Jachta HD (3840x2160), 60 fps



#### VIFP Jachta HD (3840x2160), 60 fps



## 6.11 HD sekvence 1920x1080 s parametrem CRF

### 6.11.1 Popis sekvence



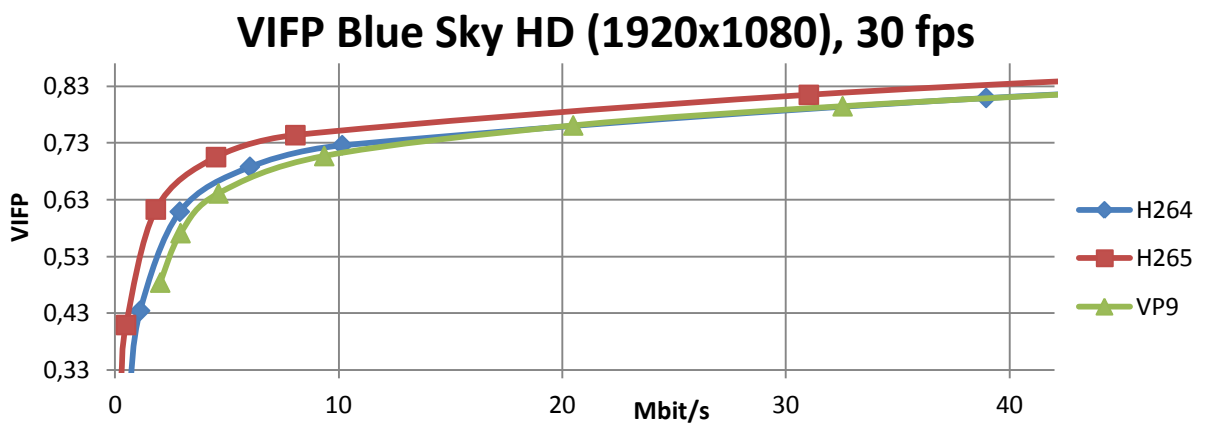
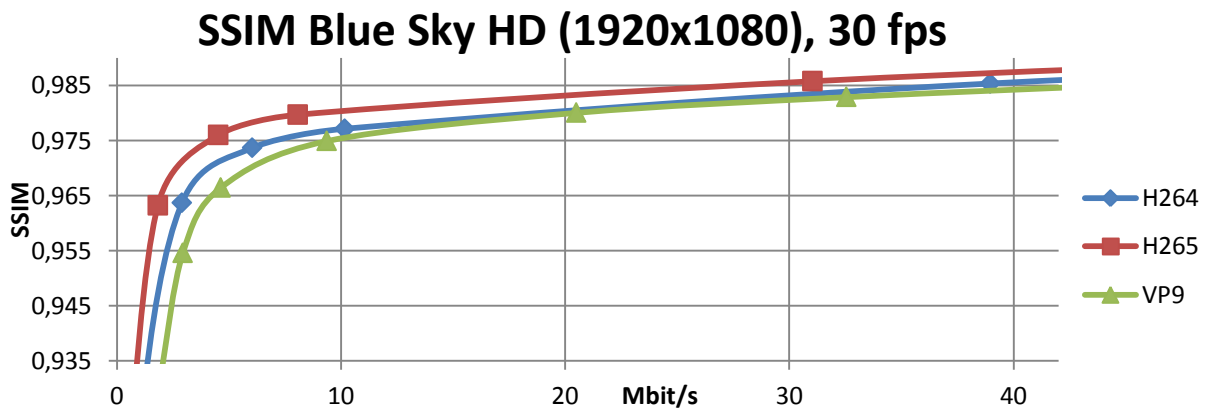
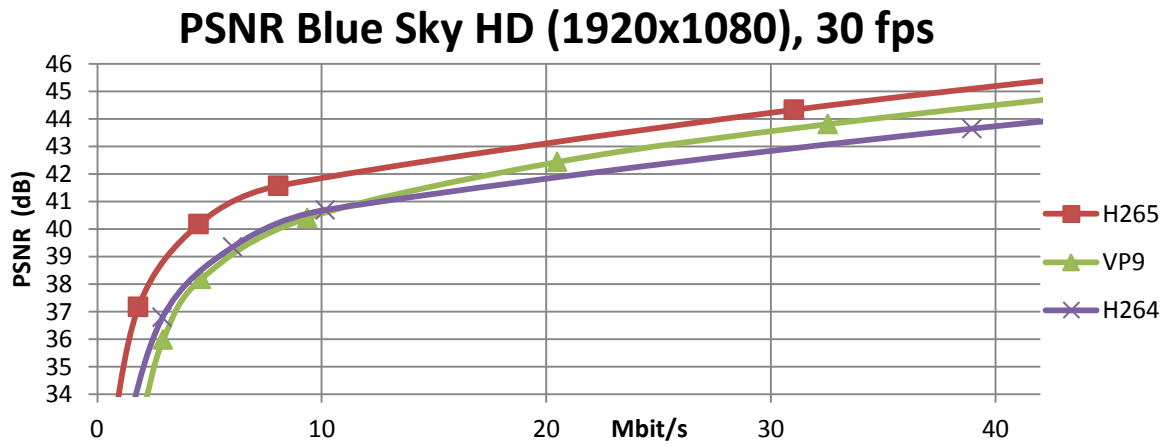
Obr 7.7 BlueSky HD

Název sekvence	Blue Sky
Rozlišení	HD 1920x1080
Počet snímků za vteřinu	30, neprokládané
Celkový počet snímků	217
Formát	YUV 4:2:0
Velikost souboru	659 MB
Video Bitstream	729 Mbit/s

### 6.11.2 Parametry enkodéru

<b>H.264</b>		<b>H265</b>		<b>VP9</b>	
Codec	libx264	Codec	libx265	Codec	libvpx-vp9
CRF	10 až 40	CRF	10 až 40	CRF	10 až 60
Profil	High	Profil	Main	Profil	0
Level	3.1	Level	3.1	kvalita	good
qmin	0	qmin	0	qmin	0
qmax	69	qmax	69	qmax	63
GOP	open GOP	GOP	open GOP	aut-alt-ref	1
presset	very slow				
Keyint	0-250	Keyint	0-250	Keyint	0-360

### 6.11.3 Výsledné grafy



## 7 Vyhodnocení výsledků

Vyhodnocení výsledků bylo provedeno pro referenční mez 40 dB u metriky PSNR, 0,975 pro metriku SSIM a 0,65 pro metriku VIFP, kde se nejčastěji pohybují požadavky na kvalitu. V druhé části pomocí tzv. Bjontegaardovy metriky. Výpočet průběhu byl proveden pouze u pěti sekvencí, kde bylo použito kódování metodou VBR (Variable Bit Rate). Tedy s ohledem na konstantní kvalitu po celou dobu sekvence. Sekvence Jachta, kde bylo použito kódování CBR (Constant Bit Rate) jsem v úvahu nevzal, výsledky vyšly zcela odlišně a to vzhledem k neprůhledné volbě některých parametrů a volbě parametru I-snímku s nastavením na hodnotu 60, tedy každou vteřinu. Pro bližší popis sekvencí a rozdílů v bitovém toku jsou k dispozici grafy a jim příslušné tabulky pro všechny naměřené hodnoty.

### 7.1 Výsledky pro referenční meze

#### 7.1.1 Výsledky PSNR pro referenční mez 40 dB

Na základě hodnot PSNR jsem stanovil průměrnou úsporu bitového toku a to při zvoleném PSNR vždy při hodnotě 40 dB, která se považuje za optimální hodnotu. Tyto výsledky jsou vidět v tabulce 7.1

Datové toky pro hodnoty 40dB PSNR				Úspora datového toku [%]	
	HEVC	VP9	AVC	HEVC vs VP9	HEVC vs AVC
traffic	14	22	21,5	36,36	34,88
kalendář	20	26,7	31	25,09	35,48
Park Run	72	82	78	12,20	7,69
Blue Sky	4,255	8	7,75	46,81	45,10
Posádka	3,3	4,1	5,6	19,51	41,07
				<b>28,00</b>	<b>32,85</b>

Tabulka 7.1

Z výsledků vyplývá, že průměrná úspora datového toku HEVC oproti VP9 je 28 % a úspora datového toku HEVC oproti jeho předchůdci AVC je 32,85%. Výsledky jednotlivých sekvencí se liší, největší úspory bylo dosaženo u sekvence Blue Sky (46,81 % resp. 45,1 %), naopak nejhorších u sekvence Park Run (12,2 % resp. 7,69%).

### 7.1.2 Výsledky SSIM pro referenční mez 0,975

Na základě hodnot SSIM jsem stanovil průměrnou úsporu bitového toku a to při zvoleném SSIM vždy při hodnotě 0,975, která se dá považovat za optimální hodnotu. Tyto výsledky jsou vidět v tabulce 7.2

	Datové toky pro hodnoty 0,975 SSIM			Úspora datového toku [%]	
	HEVC	VP9	AVC	HEVC vs VP9	HEVC vs AVC
Traffic	21	40	31	47,50	32,26
Kalendář	35	61	48	42,62	27,08
Park Run	55	78	59	29,49	6,78
Blue Sky	4,2	9,5	7	55,79	40,00
Posádka	9,3	12	11,8	22,50	21,19
				<b>39,58</b>	<b>25,46</b>

Tabulka 7.2

Z výsledků metriky SSIM dostáváme relativní úsporu HEVC oproti VP9 39,58%. Úsporu HEVC oproti AVC pouze 25,46 %. Obdobně jako u měření PSNR, bylo i zde dosaženo největší úspory u sekvence Blue Sky (55,79% respektive 40 %) naopak nejmenší u sekvence posádka (22,5 % respektive 25,46 %)

### 7.1.3 Výsledky VIFP pro referenční mez 0,65

Na základě hodnot VIFP jsem stanovil průměrnou úsporu bitového toku a to při zvoleném PSNR vždy při hodnotě 0,65, která se považuje za optimální hodnotu. Tyto výsledky jsou vidět v tabulce 7.3



Datové toky pro hodnoty 0,65 VIFP				Úspora datového toku [%]	
	HEVC	VP9	AVC	HEVC vs VP9	HEVC vs AVC
Traffic	6,8	12	11	43,33	38,18
Kalendář	5	12,5	13,5	60,00	62,96
Park Run	31	42	37,5	26,19	17,33
Blue Sky	2,7	5	4	46,00	32,50
Posádka	3,9	4,7	5,6	17,02	30,36
				<b>38,51</b>	<b>36,27</b>

Tabulka 7.3

Relativní úsporu datového toku na základě metriky VIFP vychází následovně. Úspora HEVC oproti VP9 je 38,51 %, HEVC oproti AVC s úsporou 36,27%. Nejlepších výsledků bylo dosaženo u sekvence Kalendář (60 % respektive 62,96 %) naopak nejhorší stejně jako u parametru SSIM u sekvence posádka (26,19 % respektive 17,33).

## 7.2 Výsledky pomocí Bjontegaardovy metriky

Bjontegaardova metrika umožňuje spočítat průměrný rozdíl mezi dvěma PSNR křivkami. Můžeme tedy využít všechny naměřené hodnoty a spočítat tak rozdíl v úspoře datového toku pro všechny úrovně nastavené kvality. K výpočtu byl použit program Matlab a algoritmus převzatý ze skriptu pro výpočet Bjontegaardovy metriky.[29]

Úspora v %	Traffic	Kalendář	ParkRun	BlueSky	Posádka	Průměr
HEVC vs. VP9	40,8	48,11	28,16	47,39	22,3	<b>37,352</b>
HEVC vs. H264	41,72	61,77	28,75	45,81	38,34	<b>43,278</b>
Rozdíl v dB	Traffic	Kalendář	ParkRun	BlueSky	Posádka	
HEVC vs. VP9	1,33	1,34	1,48	1,7	0,82	<b>1,334</b>
HEVC vs. H264	1,86	2,16	1,23	2,06	1,66	<b>1,794</b>

Výpočet umožňuje získat výsledky jak pro průměrnou hodnotu rozdílu PSNR v decibelech, tak relativní hodnoty rozdílů datových toků. Průměrná úspora datového toku podle Bjontegaardovy metriky je tedy HEVC oproti VP9 37,4 %. HEVC oproti H.264 potom 43,3 %.

## 8 Závěr

V úvodu práce jsem pro snadnější porozumění tématu rozebral základní parametry digitálního videa, v dalších částech jsem se věnoval popisu tří zkoumaných standardů H.264/AVC, H.265/HEVC a WebM/VP9. K jednotlivým standardům jsem popsal jejich referenční software, možnosti jeho sestavení a ovládání.

V praktické části jsem se věnoval srovnání jednotlivých standardů a to za použití programu FFMpeg pro kódování sekvencí. Jednotlivé referenční software jsem použil pro jejich popis, při testování kódování se ukázalo, že vzhledem k použitému hardware a výběru sekvencí s velmi vysokým rozlišením, trvá kódování velice dlouho, i přes použití optimalizovaného software FFMpeg trvalo kódování testovaných šesti sekvencí desítky hodin, za použití referenčních software by tak nebylo možné naměřit dostatečné množství hodnot a sekvencí.

Zkoumal jsem metriky PSNR, SSIM a VIFP, tak, jak je popsáno v části, která se věnuje objektivním metodám pro srovnání kvality videa. Pro tyto metriky jsem stanovil referenční mez, při které jsem zkoumal rozdíly v datových tocích pro dosažení určité meze, kterou pokládám za uživatelsky přijatelný standard. Je zajímavé, jak různě vycházejí hodnoty jednotlivých metrik, což je tedy potvrzení toho, že se nemůžeme zabývat pouhou metrikou PSNR a odkazovat se tak na dobré výsledky nového standardu pouze pomocí nich.

Aby se daly jednotlivé kodéry přesněji porovnat, použil jsem Bjontegaardovu metriku, která bere v potaz celé PSNR křivky. Hodnoty vyšly mírně odlišně, ale jako nejrelevantnější považuji právě je. Nelze tedy říct, že se výsledkem práce je komplexní srovnání uvedených standardů, jak znělo zadání práce, ale pouze pro tuto implementaci, kterou jsem použil, zejména pak to, že jsem nepoužil pro výpočty právě jednotlivé referenční software. A drobné odchylky tak může vykazovat i řízení programu FFMpeg. Také počet testovaných sekvencí není veliký a výsledky s větším počtem sekvencí by byly přesnější, avšak i to jsem musel vzít v potaz vzhledem k velké časové a hardwarové náročnosti. Nicméně i přes tento fakt si myslím, že výběr sekvencí je různorodý, co se

týče prostorových a časových detailů, tak i vzhledem k rozlišení. Výsledky tak tedy za porovnání standardů považovat s jistou rezervou můžeme.

Všechny dílčí výsledky jsou přehledně zpracovány v tabulkách a grafech. Pokud stručně shrnu výsledky své bakalářské práce, můžu říct, že standard H.265/HEVC je výrazně efektivnější nežli jeho předchozí verze H.264/AVC, dokonce i než standard WebM/VP9. VP9 podle Bjontegaardovy metriky vykazuje lepší výsledky než H.264/AVC. Je však potřeba upozornit na to že tato metrika se zabývá pouze měřením PSNR, protože při měření VIFP i SSIM při zvolené referenční mezi, vykazoval lepší výsledky mezi těmito dvěma standardy právě H.264/AVC.

Jako důležité považuji zmínit mnohem vyšší výpočetní náročnost nových standardů WebM/VP9 a H.264/HEVC oproti H.264, je proto třeba brát v potaz konkrétní použití pro námi požadovanou aplikaci. Na druhou stranu, vývoj H.264 byl již ukončen a zbylé dva standardy budou v budoucnosti jistě výpočetně lépe optimalizovány.

## 9 Zdroje

- [1] MICROSOFT. About YUV Video (Windows): Video Media Types [online]. [cit. 2014-05-12]. Dostupné z: [http://msdn.microsoft.com/en-us/library/windows/desktop/bb530104\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/bb530104(v=vs.85).aspx)
- [2] MATROSKA. Matroska Media Container: What is Matroska ? [online]. [cit. 2014-05-12]. Dostupné z: <http://www.matroska.org/technical/whatis/index.html>
- [3] XIPH.ORG FOUNDATION. Theora Specification [online]. 2011 [cit. 2014-05-12]. Dostupné z: <http://diracvideo.org/download/specification/dirac-spec-latest.pdf>
- [4] Dirac Specification. Dirac Specification [online]. 2008 [cit. 2014-05-12]. Dostupné z: <http://diracvideo.org/download/specification/dirac-spec-latest.pdf>
- [5] VC-1 Technical Overview. LOOMIS, Jay. MICROSOFT CORPORATION. Microsoft [online]. 2007 [cit. 2014-05-12]. Dostupné z: [http://www.microsoft.com/windows/windowsmedia/howto/articles/vc1techoverview.aspx#VC1\\_Profiles\\_and\\_Levels](http://www.microsoft.com/windows/windowsmedia/howto/articles/vc1techoverview.aspx#VC1_Profiles_and_Levels)
- [6] Joint Video Team. ITU. ITU-T [online]. Dostupné z: <http://www.itu.int/en/ITU-T/studygroups/com16/video/Pages/jvt.aspx>
- [7] MANDAU, MARKUS. H.264: Jeden standard pro všechna videa. H.264: Jeden standard pro všechna videa. 2009, roč. 2009, č. 4. Dostupné z: <http://www.chip.cz/soubory/dokumenty/afd9e4715dbac6b26fcdecb87cc67dfe.pdf>
- [8] SULLIVAN, Gary J., Jens-Rainer OHM, Woo-Jin HAN a Thomas WIEGAND. Overview of the High Efficiency Video Coding (HEVC) Standard. IEEE Transactions on Circuits and Systems for Video Technology [online]. 2012, vol. 22, issue 12, s. 1649-1668 [cit. 2014-05-12]. DOI: 10.1109/TCSVT.2012.2221191. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6316136>
- [9] BOSSEN, Frank, David FLYNN a Karsten SUHRING. JOINT COLLABORATIVE TEAM ON VIDEO CODING. HM Software Manual [online]. 2013 [cit. 2014-05-12]. Dostupné z: [https://hevc.hhi.fraunhofer.de/svn/svn\\_HEVCSoftware/branches/HM-9.2-dev/doc/software-manual.pdf](https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/branches/HM-9.2-dev/doc/software-manual.pdf)

- [10] The WebM Project: About WebM. THE WEBM PROJECT. WebM project [online]. [cit. 2014-05-12]. Dostupné z: <http://www.webmproject.org/about/>
- [11] SHARABAYKO, Maxim P., Oleg G. PONOMAREV a Roman I. CHERNYAK. Intra Compression Efficiency in VP9 and HEVC. *Intra Compression Efficiency in VP9 and HEVC* [online]. 2013 [cit. 2014-05-12]. Dostupné z: <http://www.m-hikari.com/ams/ams-2013/ams-137-140-2013/sharabaykoAMS137-140-2013.pdf>
- [12] WEBM PROJECT HOSTING. WebM Project [online]. [cit. 2014-05-12]. Dostupné z: <http://code.google.com/p/webm/downloads/lists>
- [13] YANG, Richard Dosselmann and Xue Dong. A formal assessment of the structural similarity index [online]. Regina: University of Regina. Department of Computer Science, 2008 [cit. 2014-05-12]. ISBN 07-731-0649-9. Dostupné z: <http://www.cs.uregina.ca/Research/Techreports/2008-02.pdf>
- [14] SHEIKH, Hamid R. a Alan C. BOVIK. A VISUAL INFORMATION FIDELITY APPROACH TO VIDEO QUALITY ASSESSMENT. [online]. [cit. 2014-05-12]. Dostupné z: <http://202.114.89.42/resource/pdf/415.pdf>
- [15] Digital video image quality and perceptual coding. Editor H Wu, K Rao. Boca Raton: Taylor, 2006, xxxv, 600 s. ISBN 08-247-2777-0.
- [16] SESHADRINATHAN, Kalpana a Alan C. BOVIK. Image and Video Quality Assessment - Introduction, Why Do We Need Quality Assessment?, Why is Quality Assessment So Hard?. [online]. The University of Texas at Austin [cit. 2014-05-12]. Dostupné z: <http://encyclopedia.jrank.org/articles/pages/6759/Image-and-Video-Quality-Assessment.html>
- [17] Methodology for the subjective assessment of video quality in multimedia applications: Question ITU-R 102/6. Methodology for the subjective assessment of video quality in multimedia applications: RECOMMENDATION ITU-R BT.1788 [online]. 2007 [cit. 2014-05-12]. Dostupné z: [http://www.itu.int/dms\\_pubrec/itu-r/rec/bt/R-REC-BT.1788-0-200701-!!!PDF-E.pdf](http://www.itu.int/dms_pubrec/itu-r/rec/bt/R-REC-BT.1788-0-200701-!!!PDF-E.pdf)
- [18] INTERNACIONAL TELECOMUNICATION UNION. ITU in brief: About ITU-T [online]. [cit. 2014-05-12]. Dostupné z: <http://www.itu.int/en/ITU-T/about/Pages/default.aspx>
- [19] International Electrotechnical Commission: About the IEC. [online]. [cit. 2014-05-12]. Dostupné z: <http://www.iec.ch/about/?ref=menu>
- [20] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. About ISO [online]. [cit. 2014-05-12]. Dostupné z: <http://www.iso.org/iso/home/about.htm>

- [21] Discover MPEG. [online]. [cit. 2014-05-13]. Dostupné z: <http://mpeg.chiariglione.org/about>
- [22] H.264 Profiles and Levels: Inside and Outside Media Coder. [online]. [cit. 2014-05-13]. Dostupné z: <http://blog.mediacoderhq.com/h264-profiles-and-levels/>
- [23] About FFmpeg. [online]. [cit. 2014-05-13]. Dostupné z: <http://www.ffmpeg.org/about.html>
- [24] Multimedia Container Formats. [online]. [cit. 2014-05-13]. Dostupné z: [http://www.divxland.org/en/article/15/multimedia\\_container\\_formats#.U3J89YGSySo](http://www.divxland.org/en/article/15/multimedia_container_formats#.U3J89YGSySo)
- [25] Image and Video Compression. [online]. [cit. 2014-05-18]. Dostupné z: <http://www.cmlab.csie.ntu.edu.tw/~blacksmith/portfolio/compression.htm>
- [26] The TV Studio Becomes a System. [online]. [cit. 2014-05-18]. Dostupné z: <http://www.altera.com/technology/system-design/articles/2013/tv-studio-system.html>
- [27] 3D Cine Cast. [online]. [cit. 2014-05-18]. Dostupné z: <http://3dcinecast.blogspot.cz/2013/04/wtf-is-h265-aka-hevc.html>
- [28] Technical Overview of VP9. [online]. [cit. 2014-05-18]. Dostupné z: <http://files.meetup.com/9842252/Overview-VP9.pdf>
- [29] File Exchange: Matlab Central. [online]. [cit. 2014-05-20]. Dostupné z: <http://www.mathworks.com/matlabcentral/fileexchange/27798-bjontegaard-metric/content/bjontegaard.m>

# 10 Přílohy

## 10.1 Tabulky naměřených hodnot jednotlivých sekvencí

### 10.1.1 Sekvence Traffic

Sekvence Traffic					
jmeno souboru	velikost(KB)	bitrate[Mbit/s]	PSNR	SSIM	VIFP
traffic4K265crf30	2247	1,798	35,926	0,936	0,567
traffic4K265crf28	6228	4,982	36,965	0,949	0,613
traffic4K265crf24	13422	10,738	39,318	0,966	0,692
traffic4K265crf20	30260	24,208	41,418	0,977	0,756
traffic4K265crf16	71866	57,493	43,492	0,983	0,809
traffic4K265crf14	111069	88,855	44,586	0,986	0,835
traffic4K265crf10	234689	187,751	46,886	0,990	0,884
traffic4Kvp9crf60	7931	6,345	34,145	0,924	0,537
traffic4Kvp9crf58	8510	6,808	34,740	0,930	0,555
traffic4Kvp9crf54	9702	7,762	35,668	0,939	0,583
traffic4Kvp9crf50	11090	8,872	36,449	0,946	0,606
traffic4Kvp9crf46	12780	10,224	37,222	0,952	0,628
traffic4Kvp9crf42	15536	12,429	38,057	0,957	0,653
traffic4Kvp9crf38	19402	15,522	38,874	0,963	0,677
traffic4Kvp9crf32	28912	23,130	40,121	0,969	0,713
traffic4Kvp9crf28	39783	31,826	40,987	0,973	0,737
traffic4Kvp9crf24	58686	46,949	41,986	0,976	0,764
traffic4Kvp9crf20	76051	60,841	42,656	0,978	0,780
traffic4Kvp9crf16	101976	81,581	43,439	0,981	0,799
traffic4Kvp9crf14	126211	100,969	43,995	0,982	0,812
traffic4Kvp9crf10	196065	156,852	45,334	0,985	0,842
traffic4K264crf40vs	2710	2,168	30,219	0,857	0,353
traffic4K264crf36vs	4002	3,202	32,451	0,897	0,442
traffic4K264crf32vs	6238	4,990	34,692	0,927	0,530
traffic4K264crf28vs	10394	8,315	36,881	0,950	0,614
traffic4K264crf24vs	18919	15,135	38,933	0,965	0,687
traffic4K264crf20vs	38359	30,687	40,855	0,975	0,748
traffic4K264crf16vs	89486	71,589	42,789	0,982	0,799
traffic4K264crf14vs	140776	112,621	43,888	0,985	0,825
traffic4K264crf10vs	319997	255,998	46,526	0,990	0,881

## 10.1.2 Sekvence Kalendář

Sekvence Kalendář					
jmeno souboru	velikost(KB)	bitrate[Mbit/s]	PSNR	SSIM	VIFP
kalendar265crf34	293	0,235	30,761	0,867	0,402
kalendar265crf30	494	0,396	33,079	0,908	0,485
kalendar265crf24	1405	1,126	36,212	0,940	0,588
kalendar265crf20	4136	3,315	37,757	0,951	0,637
kalendar265crf18	6807	5,457	38,256	0,953	0,653
kalendar265crf16	12211	9,788	38,960	0,958	0,673
kalendar265crf14	23413	18,768	39,898	0,965	0,707
kalendar265crf10	43773	35,089	41,588	0,975	0,760
kalendar264crf12	67741	54,301	43,465	0,983	0,815
kalendar264crf34	725	0,581	29,779	0,846	0,357
kalendar264crf30	1236	0,991	32,042	0,892	0,439
kalendar264crf24	3489	2,797	35,309	0,933	0,556
kalendar264crf20	10094	8,091	37,310	0,947	0,617
kalendar264crf18	17815	14,281	38,197	0,953	0,653
kalendar264crf16	30395	24,365	39,234	0,961	0,690
kalendar264crf14	49535	39,707	40,570	0,971	0,736
kalendar264crf12	75188	60,271	42,293	0,980	0,790
kalendar264crf10	103814	83,218	44,193	0,987	0,840
kalendarvp9xxcrf60	925	0,741	31,441	0,876	0,426
kalendarvp9xxcrf50	1443	1,157	33,940	0,915	0,513
kalendarvp9xxcrf40	2158	1,730	35,385	0,929	0,561
kalendarvp9xxcrf36	4203	3,369	36,546	0,939	0,596
kalendarvp9xxcrf34	5035	4,036	36,774	0,940	0,603
kalendarvp9xxcrf32	6335	5,078	37,068	0,942	0,612
kalendarvp9xxcrf30	8414	6,745	37,420	0,944	0,623
kalendarvp9xxcrf28	10554	8,460	37,732	0,946	0,632
kalendarvp9xxcrf26	14302	11,465	38,192	0,948	0,646
kalendarvp9xxcrf22	24347	19,517	39,216	0,954	0,677
kalendarvp9xxcrf18	35173	28,195	40,159	0,959	0,707
kalendarvp9xxcrf16	41490	33,259	40,688	0,962	0,723
kalendarvp9xxcrf14	51621	41,380	41,448	0,966	0,747
kalendarvp9xxcrf12	63439	50,853	42,327	0,971	0,773
kalendarvp9xxcrf10	77978	62,507	43,399	0,976	0,803



### 10.1.3 Sekvence Park Run

jmeno souboru	velikost(KB)	bitrate[Mbit/s]	PSNR	SSIM	VIFP
parkRunHD265crf40	136	0,181	20,591	0,510	0,158
parkRunHD265crf36	364	0,485	22,399	0,623	0,229
parkRunHD265crf28	2306	3,075	26,851	0,818	0,400
parkRunHD265crf26	3722	4,963	28,249	0,860	0,454
parkRunHD265crf23	7360	9,813	30,367	0,905	0,525
parkRunHD265crf20	13867	18,489	32,789	0,934	0,589
parkRunHD265crf18	20711	27,615	34,452	0,948	0,634
parkRunHD265crf16	29986	39,981	36,197	0,963	0,684
parkRunHD265crf14	43009	57,345	38,266	0,976	0,743
parkRunHD265crf10	70803	94,404	42,619	0,991	0,850
parkRunHDvp9crf60	3130	4,173	25,498	0,786	0,359
parkRunHDvp9crf50	6394	8,525	28,156	0,857	0,443
parkRunHDvp9crf40	12724	16,965	31,053	0,901	0,523
parkRunHDvp9crf30	27210	36,280	34,449	0,941	0,625
parkRunHDvp9crf25	38718	51,624	36,402	0,958	0,685
parkRunHDvp9crf21	47476	63,301	37,816	0,968	0,725
parkRunHDvp9crf20	49626	66,168	38,155	0,969	0,735
parkRunHDvp9crf14	65075	86,767	40,492	0,979	0,794
parkRunHDvp9crf10	80293	107,057	42,568	0,984	0,839
parkRunHD264crf42	396	0,528	20,301	0,525	0,152
parkRunHD264crf36	1021	1,361	22,990	0,692	0,262
parkRunHD264crf30	2965	3,953	26,437	0,824	0,392
parkRunHD264crf28	4275	5,700	27,659	0,857	0,434
parkRunHD264crf23	10533	14,044	30,866	0,914	0,534
parkRunHD264crf16	34244	45,659	35,993	0,965	0,682
parkRunHD264crf14	46857	62,476	37,952	0,977	0,738
parkRunHD264crf20	58445	77,927	40,055	0,982	0,790
parkRunHD264crf10	75613	100,817	42,214	0,991	0,843

## 10.1.4 Sekvence Blue Sky

Sekvence Blue Sky					
jmeno souboru	velikost(KB)	bitrate[Mbit/s]	PSNR	SSIM	VIFP
blueskyHD265crf40	251	0,278	28,954	0,875	0,315
blueskyHD265crf36	449	0,497	31,493	0,913	0,409
blueskyHD265crf28	1654	1,829	37,171	0,963	0,613
blueskyHD265crf23	4089	4,522	40,178	0,976	0,705
blueskyHD265crf20	7297	8,070	41,568	0,980	0,744
blueskyHD265crf14	28063	31,037	44,330	0,986	0,815
blueskyHD265crf10	52848	58,449	46,812	0,990	0,868
blueskyHDvp9crf60	1833	2,027	33,364	0,934	0,484
blueskyHDvp9crf50	2655	2,936	35,987	0,955	0,571
blueskyHDvp9crf40	4188	4,632	38,188	0,966	0,641
blueskyHDvp9crf30	8466	9,363	40,403	0,975	0,707
blueskyHDvp9crf20	18533	20,497	42,438	0,980	0,761
blueskyHDvp9crf14	29429	32,548	43,812	0,983	0,795
blueskyHDvp9crf10	42681	47,205	45,145	0,986	0,825
blueskyHD264crf42	594	0,657	28,011	0,870	0,291
blueskyHD264crf36	1018	1,126	31,862	0,926	0,434
blueskyHD264crf28	2622	2,900	36,795	0,964	0,609
blueskyHD264crf23	5459	6,038	39,348	0,974	0,688
blueskyHD264crf20	9191	10,165	40,689	0,977	0,726
blueskyHD264crf14	35232	38,966	43,639	0,985	0,809
blueskyHD264crf10	71386	78,952	46,522	0,992	0,875

## 10.1.5 Sekvence Jachta

Sekvence Jachta					
jmeno souboru	velikost(KB)	bitrate[Mbit/s]	PSNR	SSIM	VIFP
jachta265m4	5015	4,012	33,687	0,909	0,326
jachta265m5	6475	5,180	34,390	0,916	0,353
jachta265m7	8939	7,151	35,523	0,927	0,398
jachta265m10	13076	10,461	36,740	0,937	0,446
jachta265m13	17035	13,628	37,665	0,943	0,482
jachta265m15	19651	15,721	38,157	0,947	0,502
jachta265m18	23541	18,833	38,785	0,950	0,526
jachta265m22	28798	23,038	39,489	0,954	0,552
jachta265m50	65299	52,239	42,023	0,966	0,647
jachtavp9m4	4961	3,969	33,259	0,898	0,276
jachtavp9m5	4993	3,994	33,281	0,898	0,277
jachtavp9m7	7347	5,878	34,487	0,912	0,325
jachtavp9m10	11998	9,598	36,185	0,929	0,398
jachtavp9m13	14339	11,471	36,840	0,934	0,426
jachtavp9m15	16652	13,322	37,390	0,939	0,450
jachtavp9m18	21554	17,243	38,266	0,945	0,489
jachtavp9m22	25836	20,669	38,971	0,950	0,519
jachtavp9m50	57790	46,232	41,643	0,964	0,629
jachta264m4	5168	4,134	30,203	0,861	0,189
jachta264m5	6478	5,182	31,117	0,873	0,221
jachta264m7	9142	7,314	32,591	0,892	0,275
jachta264m10	13076	10,461	34,154	0,910	0,337
jachta264m13	17076	13,661	35,302	0,922	0,384
jachta264m15	19035	15,228	35,919	0,928	0,410
jachta264m18	23541	18,833	36,691	0,934	0,442
jachta264m22	28747	22,998	37,504	0,941	0,475
jachta264m50	64859	51,887	40,477	0,959	0,594

## 10.1.6 Sekvence Posádka

Sekvence Posádka					
jmeno souboru	velikost(KB)	bitrate[Mbit/s]	PSNR	SSIM	VIFP
posadkaSD265crf40	125	0,100	29,129	0,766	0,181
posadkaSD265crf32	406	0,325	32,811	0,841	0,328
posadkaSD265crf30	551	0,441	33,905	0,863	0,378
posadkaSD265crf25	1325	1,060	36,786	0,911	0,504
posadkaSD265crf20	3534	2,827	39,552	0,943	0,618
posadkaSD265crf14	12493	9,994	44,038	0,977	0,789
posadkaSD265crf10	21571	17,257	47,359	0,989	0,879
posadkaSDvp9crf60	593	0,474	32,440	0,841	0,328
posadkaSDvp9crf50	1005	0,804	34,793	0,882	0,423
posadkaSDvp9crf40	1785	1,428	36,853	0,912	0,509
posadkaSDvp9crf30	3735	2,988	39,055	0,937	0,599
posadkaSDvp9crf25	5603	4,482	40,267	0,949	0,647
posadkaSDvp9crf20	7854	6,283	41,436	0,958	0,692
posadkaSDvp9crf10	16055	12,844	44,756	0,977	0,802
posadkaSDvp9crf8	19225	15,380	45,820	0,981	0,830
posadkaSD264crf36	498	0,398	31,258	0,816	0,273
posadkaSD264crf28	1384	1,107	35,168	0,891	0,447
posadkaSD264crf24	2603	2,082	37,359	0,921	0,533
posadkaSD264crf20	5168	4,134	39,048	0,941	0,613
posadkaSD264crf18	8097	6,478	40,543	0,955	0,669
posadkaSD264crf14	15599	12,479	43,475	0,977	0,783
posadkaSD264crf10	25973	20,778	46,908	0,989	0,876