



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

**Elektrotechnická fakulta
Katedra radioelektroniky**

Porovnání účinnosti moderních standardů pro kompresi videa

Performance comparison of advanced standards for video

bakalářská práce

Studijní program: Komunikace, multimédia a elektronika

Studijní obor: Multimediální technika

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

Oldřich Strnad

Praha 2014

České vysoké učení technické v Praze
Fakulta elektrotechnická
katedra radioelektroniky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Oldřich Strnad**

Studijní program:
Obor: Multimediální technika

Název tématu: **Porovnání účinnosti moderních standardů pro kompresi videa**

Pokyny pro vypracování:

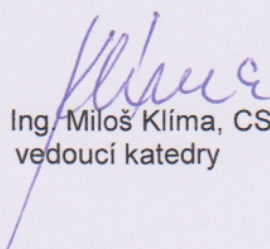
Vypracujte přehled nejnovějších standardů pro kompresi videa s důrazem na standard HEVC (High Efficiency Video Coding). Seznamte se s referenční implementací vybraných standardů a proveďte srovnání jejich účinnosti na testovacích videosekvencích.

Seznam odborné literatury:

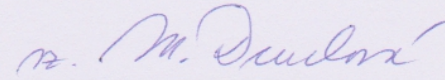
- [1] Richardson, Iain E. G.: The H.264 advanced video compression standard, Wiley, 2010.
- [2] Ohm, J.-R., Sullivan, G. J., Schwarz, H., Tan, T. K., Wiegand, T.: Comparison of the Coding Efficiency of Video Coding Standards - Including High Efficiency Video Coding (HEVC), IEEE Transactions on Circuits and Systems for Video Technology, vol.22, no.12, pp.1669-1684, Dec. 2012.
- [3] Bovik, A.: Handbook of Image and Video Processing, Elsevier Academic Press, New York, 2005.
- [4] Gonzalez, R. C., Woods, R. E.: Digital Image Processing, Prentice Hall, 2007.

Vedoucí: Ing. Karel Fliegel, Ph.D.

Platnost zadání: do konce zimního semestru 2013/2014


Prof. Ing. Miloš Klíma, CSc.
vedoucí katedry




prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 4. 2. 2013

Čestné prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem č.1/2009 o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Datum: 3. 1. 2014

.....

podpis

Poděkování:

Chtěl bych poděkovat mému vedoucímu Ing. Karlu Fliegelovi, Ph.D. za odborné vedení a poskytnuté informace při zpracování této bakalářské práce. Dále pak děkuji Mgr. Lukáši Janírkovi za poskytnutí věcných připomínek. Nakonec děkuji rodině za poskytnuté zázemí.

Anotace:

Bakalářská práce obsahuje přehled současných video standardů. Pro každý formát je uveden základní popis principu kódování, podporované rozlišení, snímkový kmitočet, typické přenosové rychlosti aj. Jedná se o formáty Apple ProRes, DNxHD (VC-3), Dirac, WMV 9 (VC-1), VP8, VP9, H.264/MPEG-4 AVC a podrobněji zpracovaný H.265/HEVC. V praktické části porovnávám kompresní účinnosti standardů H.264/MPEG-4 AVC, H.265/HEVC a VP9 podle objektivních metod pro měření kvality videa na vybraných video sekvencích.

Klíčová slova:

Video komprese, Transformace, Pohybová kompenzace, Intra predikce, Inter predikce, Pohybový vektor

Summary:

My bachelor's thesis contains an overview of current video standards. Basic description of coding procedures, supported resolution, picture frequency, typical data signaling rate etc. is provided for every format. This concerns the following formats: Apple ProRes, DNxHD (VC-3), Dirac, WMV 9 (VC-1), VP8, VP9, H.264/MPEG-4 AVC and H.265/HEVC, which is processed in more detail. The practical part of the thesis compares compression efficiency of the H.264/MPEG-4 AVC, H.265/HEVC and VP9 standards, depending on the objective methods for measuring the video quality on selected video sequences.

Index term:

Video compression, Transform, Motion compensation, Intra prediction, Inter prediction, Motion vector

Obsah

Obsah	6
1 Úvod do video standardů	8
2 Video standardy	9
2.1 Apple ProRes	9
2.2 VC-3 (DNxHD)	11
2.3 Dirac & Dirac Pro (VC-2)	12
2.4 VC-1.....	13
2.5 MPEG-4 AVC.....	14
2.6 VP8	18
2.7 VP9	21
2.9 HEVC – High Efficiency Video Coding	23
2.9.1 Úvod.....	23
2.9.2 Bloková struktura.....	24
2.9.3 Řezy a dlaždice	25
2.9.4 Intra predikce	26
2.9.5 Inter predikce	27
2.9.6 Transformace	28
2.9.7 In-loop filtry.....	28
2.9.8 Entropie.....	29
3 Metody pro objektivní hodnocení kvality videa	30
3.1 PSNR (Peak Signal-to-Noise Ratio).....	30
3.2 SSIM (Structural Similarity).....	30
3.3 MS-SSIM (Multi-Scale Structural Similarity).....	31
3.4 VIF (Visual Information Fidelity)	31
4 Nastavení kodérů	32
4.1 Referenční software	32
4.2 Testovací video sekvence	32
4.3 Intra a inter srovnání	33
5 Výsledky porovnání video standardů H.264, HEVC a VP9.....	35
5.1 Intra srovnání	35
5.2 Inter srovnání	36
6 Závěrečné shrnutí.....	38
7 Použité zdroje	39
Zkratky.....	44
Přílohy.....	45
Příloha č. 1	46
Příloha č. 2	47
Příloha č. 3	48
Příloha č. 4.....	49
Příloha č. 5	50
Příloha č. 6	51
Příloha č. 7	52
Příloha č. 8	53
Příloha č. 9	54

Příloha č. 10	55
Příloha č. 11	56
Příloha č. 12	57
Příloha č. 13	58
Příloha č. 14	59
Příloha č. 15	60

1 Úvod do video standardů

Vývojem video standardů se zabývá mnoho vědeckých skupin a organizací. Mezi nejvýznamnější patří skupiny, které se podílí nebo podílely na standardech z rodiny MPEG, tedy skupina Moving Picture Experts Group (MPEG), ty se staly součástí běžného spotřebního průmyslu. Další skupinou je Video Coding Experts Group (VCEG). Spojení těchto skupin nese označení Joint Video Team (JVT).

V této práci se seznámíme s jednotlivými video standardy, které se řadí mezi ty novější a jejich použití stačí současným potřebám, jako je zpracování videa o vysokém rozlišení případně ultra vysokém rozlišení. Vybrané formáty by se daly rozdělit do dvou skupin podle použití, v jedné by byly standardy, které se použitím hodí do výrobní části, jako jsou studia, tedy studiové formáty a v druhé skupině už formáty s kterými se může setkat běžný uživatel, jak na internetu, optických disků, tak při digitálním vysílání televize. Budou uvedeny jejich základní parametry, jako je podporované rozlišení, barevný model, snímkový kmitočet, použité metody pro inter a intra predikci, typické přenosové rychlosti aj., kde formát H.265/HEVC bude zpracován podrobněji.

Další část práce se bude zabývat samotným srovnáním kompresní účinnosti vybraných standardů:

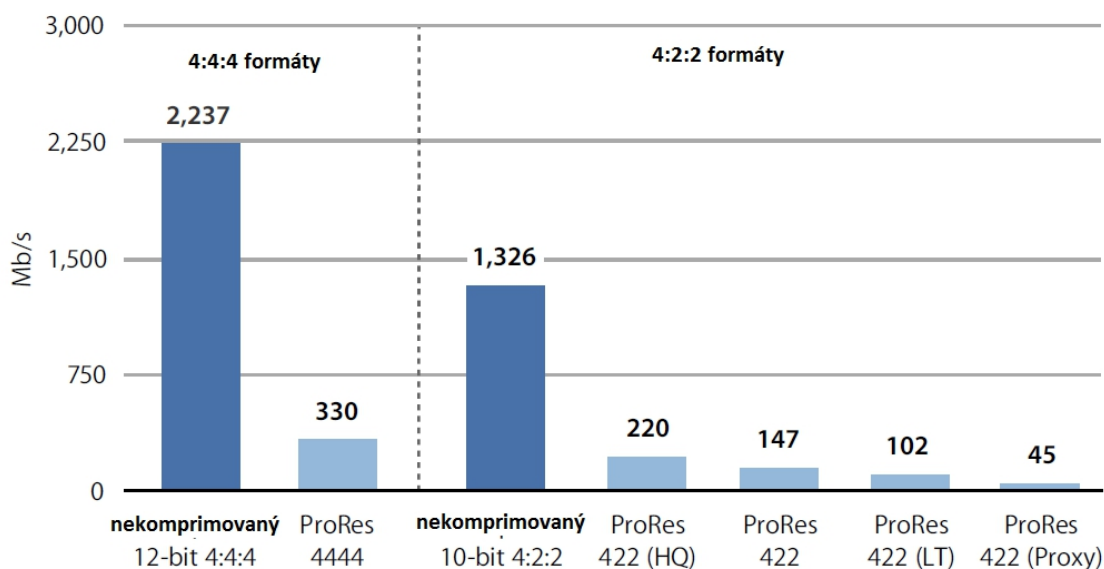
- H.264/MPEG-4 AVC
- H.265/HEVC
- VP9

Výše tři jmenované byly vybrány podle toho, že H.265 je nástupcem H.264 a chceme ověřit zlepšení kompresní účinnosti. VP9 je na druhou stranu prezentován, jako konkurenční formát k právě zmiňovanému HEVC formátu. Srovnání se provede na základě objektivních měřítek video kvality jako jsou PSNR, SSIM, MS-SSIM a VIF.

2 Video standardy

2.1 Apple ProRes

První standard používaný především v post produkci je vyvinut firmou Apple. Setkáme se s ním ve většině případech na počítačích s Mac OS. Existuje v pěti verzích, které se liší především podporovanou barevnou hloubkou a datovými toky, které se mohou pohybovat např. při rozlišení 1920 x 1080 pixelů a 25 fps od 38 Mbit/s u nejnižší verze až k 275 Mbit/s u verze ProRes 4444. Je podporováno téměř každé rozlišení (SD, HD, 2K, 4K a více). Jak už z použití může být jasné, tak se jednotlivé snímky kódují pouze v intra módu (I snímky), což je právě výhodné při editaci. Je podporováno prokládané i neprokládané video. Datový tok zde není pevně daný, ale je variabilní (VBR) podle dané scény. Je vždy pouze definován max. datový tok u jednotlivých verzí. Na obrázku 1 můžeme vidět typické datové toky pro video o rozlišení 1920 x 1080 a 29.97 fps. Na obrázku 2 vidíme srovnání obrazové kvality vyjádřené v PSNR pro StEM Mini-movie 1920 x 1080. Kvalitou a použitím je nejčastěji srovnáván s VC-3. [1][2]



obr.1 Srovnání datových toků jednotlivých verzí Apple ProRes [1]

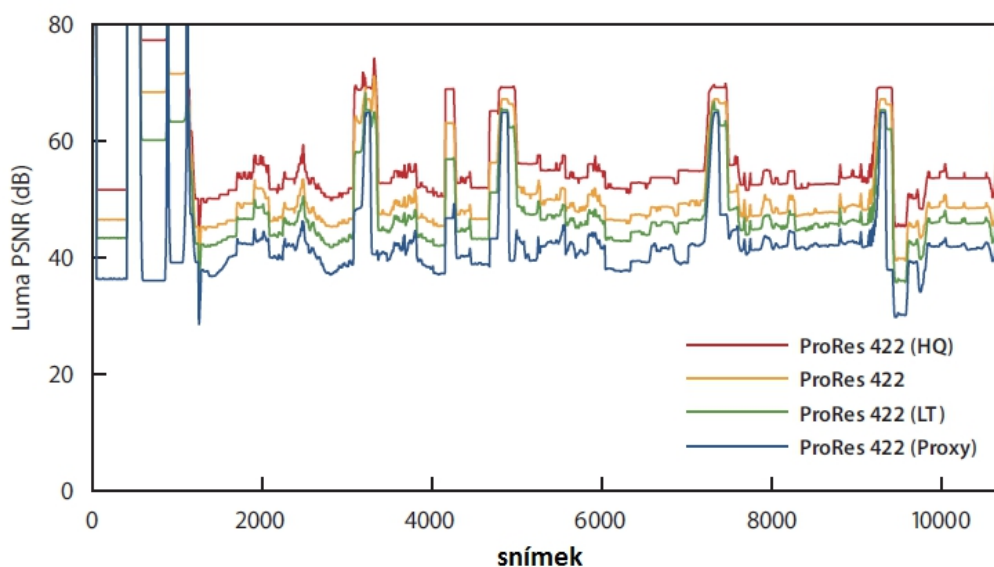
Apple ProRes 4444 je nejvyšší z dostupných kodeků, který nabízí nejlepší kvalitu. Podporuje 12 bitový 4:4:4 YUV neb 4:4:4 RGB obrazový formát a jen v této verzi je možná podpora až 16 bitového alpha kanálu. Kvalitou je srovnatelný s ProRes 422 (HQ), jen díky barevnému vzorkování umožňuje zobrazit více detailů. Datový tok pro HD video je kolem 330 Mbit/s. [1]

Apple ProRes 422 (HQ) je asi nejvíce používaná verze v post-produkci. V základu podporuje 10 bitovou hloubku při 4:2:2 nebo 4:2:0 YUV vzorkování, bez podpory alpha kanálu. Kvalita je vizuálně srovnatelná s ProRes 4444. Typický datový tok pro HD video je 220 Mbit/s. [1]

Apple ProRes 422 nabízí srovnatelnou kvalitu s ProRes 422 (HQ) i se stejnými podporovanými parametry, jako 4:2:2 nebo 4:2:0 YUV při 10 bitech a dosahuje lepších výkonů při real-time editaci. Jen datový tok je nižší, cca 145 Mbit/s u HD obsahu. [1]

Apple ProRes 422 (LT) podporuje video se stejnými parametry jako předchůdce, jen je opět použit nižší datový tok. Ten je maximálně 100 Mbit/s. Především tato verze je výhodná, kde musíme volit kompromis mezi kvalitou a přenosovou kapacitou, resp. úložištěm. [1]

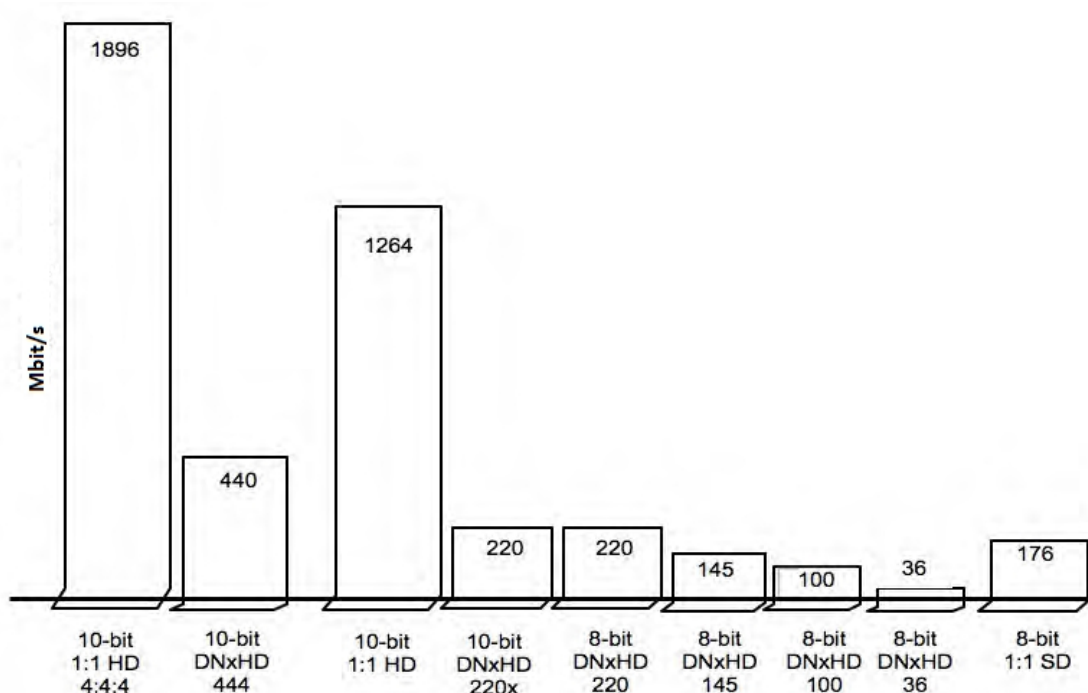
Apple ProRes 422 (Proxy) je nejnižší verze z rodiny standardů Apple ProRes. Podpora video formátu je stejná a datový tok je opět snížen na 40 Mbit/s. Tato verze je určena k pracovním náhledům, kde je dostačující nízký datový tok, ale je potřeba plné rozlišení. Také je vhodný pro offline editaci na počítačích, kde je kapacita disku omezená. [1]



obr. 2 Srovnání kvality jednotlivých verzí Apple ProRes [1]

2.2 VC-3 (DNxHD)

DNxHD se nazývá kodek vyvinutý společností Avid Technology, Inc, kdy se první verze objevila v roce 2004 [3]. Po dvouletém testování v roce 2008, byl jako první kodek vyhovující novému standardu SMPTE VC-3 [4]. Jak už je z názvu patrné, používá se pro videa ve vysokém rozlišení (720p až 1080p60) na úrovni post produkce. Jednotlivé snímky se kódují v intra módu. Každý snímek je zakódován zvlášť nezávisle na předchozích nebo následujících snímcích podobným způsobem jako JPEG obrázky. Podporuje 8/10 bitovou hloubku, vzorkování 4:2:2 nebo 4:4:4. Při kódování máme na výběr z několika možností, především vybíráme podle použití nebo podle zdroje videa, zda se jedná např. o HDCAM nebo když ukládáme video přímo z kamery. Na obrázku 3 vidíme příklady použití pro video o rozlišení 1920 x 1080, 29.97 fps včetně použitých datových toků v porovnání s nekomprimovaným videem [5]. Zdrojové kódy kodeku jsou pro nekomerční účely volně dostupné [6].



obr. 3 Přenosové rychlosti kodeku DNxHD pro video 1920x1080, 29.97 fps [5]

2.3 Dirac & Dirac Pro (VC-2)

Projekt, za kterým stojí výzkumná část britské televize BBC. Specifikace byla dokončena počátkem roku 2008 a v září téhož roku vznikla podmnožina Dirac Pro standardizována podle SMPTE jako VC-2. První veřejná ukázka ale proběhla už v roce 2005 u televizního vysílání ve vysokém rozlišení. Použití v základní verzi je široké, od videa v mobilních zařízeních, přes HDTV až k videím o rozlišení 8K používaných např. v digitálních kinech a zvládá snímkovací frekvenci od 23,97 až po 60 snímků za vteřinu. Komprimace může být buď bezztrátová, vizuálně bezztrátová nebo ztrátová, jejichž účinnost je srovnatelná s AVC/H264. Asi největší rozdíl oproti AVC/H264 je, že Dirac používá vlnkovou transformaci, díky které je možné dosáhnout lepších výsledků [7].

Verze Dirac Pro (VC-2) kóduje snímky pouze v intra módu, tedy máme jen I snímky, to je vhodné při výrobě, usnadňuje to práci. Další výhodou je snadná implementace do hardwarových zařízení. Cena za používání je nulová, jedná se totiž o otevřený kodek, takže je možné upravovat zdrojové kódy a opravovat případné chyby [7].

2.4 VC-1

VC 1 byl vyvinut společností SMPTE a jedná se v podstatě o implementaci Windows Media Video 9. VC 1 je navržen, jak pro nízké rozlišení, tak vysoké, až do 2048 x 1536 pixelů, které se používá například v digitálních kinech. Též datový tok se může pohybovat od pár desítek kilobitů až po maximálních 135 Mb za sekundu při nejvyšším profilu a levelu. V dnešní době se s ním můžeme setkat například na Blu-ray discích, nebo má hardwarovou podporu na minipočítačích Raspberry Pi [8] nebo především v aplikacích od Microsoftu. Též je dobré zmínit, že je možná i hardwarová akcelerace při přehrávání a kódování videa, kterou podporuje většina nových grafických procesorů [9].

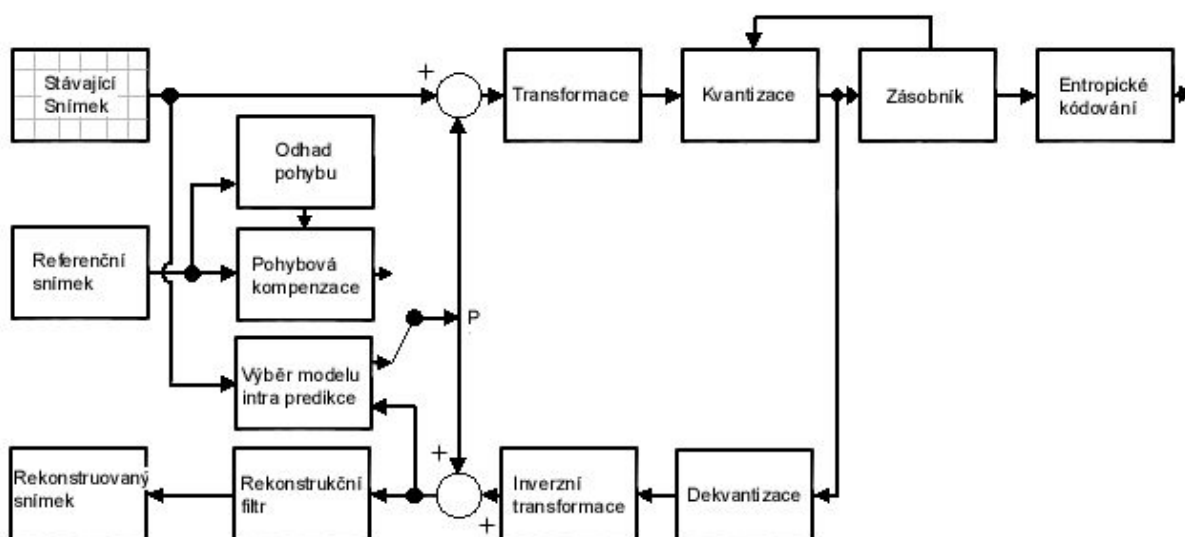
Podporovaný vstupní a výstupní formát je pouze YUV 4:2:0 a zdrojem též může být prokládané nebo neprokládané video. VC-1 používá celočíselnou transformaci podobné DCT, jako H.264, ale s rozdílnými velikostmi transformačních bloků. V intra módu je velikost bloku 8x8, v inter módu je velikost už proměnlivá a může být buď 8x8, 2 bloky 4x8, 2 bloky 8x4 nebo 4 bloky 4x4. Používá se zde 5 typů snímků – I, P, B, BI a skipped („přeskočený“) snímky. První tři jmenované jsou stejné jako v jiných MPEG standardech. Když je snímek podobný referenčnímu, tak není kódován a kóduje se jen odkaz na referenční místo celého snímku, tj. skipped. BI snímky používají jen makrobloky z I snímků. GOP struktura zde není pevně dána. Kompenzace pohybu je prováděna z makrobloků 16x16 a bloků 8x8, neboli predikce je tvořena z bloků této velikosti a může být tvořena až ze dvou referenčních snímků. Jsou k dispozici dva režimy použití pohybového vektoru: jednovektorový nebo čtyřvektorový režim. Jednovektorový režim používá bloky o velikosti 16x16, čtyřvektorový režim pracuje se 4 bloky o velikosti 8x8 pixelů. Přesnost pohybového vektoru je $\frac{1}{2}$ nebo $\frac{1}{4}$ pixelu. Též je zde loop filtr pro odstranění kostičkování. Entropické kódování je řešeno pomocí VLC, které má menší účinnost oproti CABAC v H.264, ale je i méně náročné. VC-1 obsahuje 3 profily: Simple (2 levely), Main (3 levely) a Advanced (5 levelů) [9][10].

2.5 MPEG-4 AVC

Jak tomu bylo i u JPEG, tak zkratka MPEG je odvozena od pracovní skupiny, která ho tvořila – “Motion Picture Experts Groups“. Formát MPEG-1 byl dokončen v roce 1991 a nyní se takřka nepoužívá. Uplatnění měl především u VCD (Video Compact Disc), které se již nevyrábí. Byl nahrazen MPEG-2, který našel uplatnění u DVD a digitálního televizního vysílání. S přechodem na vysoké rozlišení je ale postupně nahrazován MPEG-4 AVC, který je v dnešní době již standardem. Nyní se na něj podíváme podrobněji.

MPEG-4 AVC (MPEG-4 part 10, Advanced Video Coding) podle ISO/IEC nebo H.264 podle ITU-T označují tentýž standard, který byl prvně publikován v roce 2003 a nabízí efektivnější kompresi, větší flexibilitu v kódování, přenášení a ukládání videa v porovnání s MPEG-2 [11].

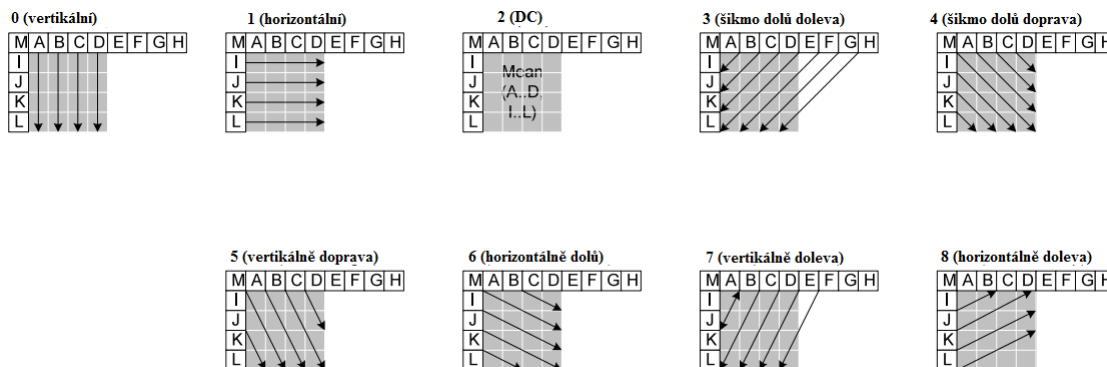
Kódování videa probíhá v těchto krocích – predikce, transformace a samotné entropické zakódování. Dále by se ještě dalo rozdělit na dvě cesty datového toku – „forward“ (dopřednou cestu) a rekonstrukční cestu. Na obrázku 4 je blokové schéma kodéru.



obr. 4 Blokové schéma kodéru H.264 [12]

V forward cestě to vypadá následovně. Tak jako u JPEG, tak i tady se jednotlivé snímky rozdělují na makrobloky o velikosti 16x16 zobrazených pixelů. Ty tvoří predikci makrobloku, která je založena na již dříve kódovaných datech, buď z aktuálního snímku (intra predikce) nebo z jiných snímků již dříve kódovaných (inter predikce). Poté se predikce odečte od současného makrobloku a vznikne zbytkový. [13]

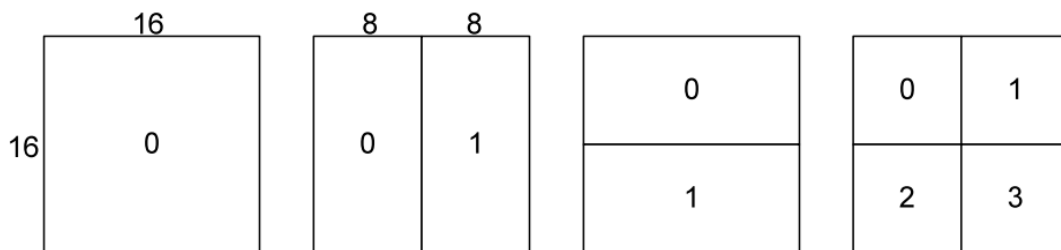
Intra predikce používá bloky o velikosti 16x16 a 4x4 pixelů. V tomto módu se predikce bloku tvoří na základě již zakódovaných a rekonstruovaných bloků (nefiltrovaných), pak jsou tyto bloky odečteny od aktuálního bloku před zakódováním. Jasové vzorky mohou být tvořeny pro každý 4x4 sub-blok s dalšími devíti volitelnými predikčními módy nebo pro 16x16 makrobloků se čtyřmi volitelnými predikčními módy a jeden mód, který je vždy použit u bloku s barevnou složkou. Na obr. 5 je zobrazeno devět predikčních módů pro jasové vzorky. Směr šipek určuje, jakým způsobem bude probíhat predikce. Písmena A-M značí vzorky, které byly již dříve kódovány. Kodér si vybírá model na základě absolutní odchylky od predikovaného bloku, volí kde je odchylka nejmenší. [15] [12]



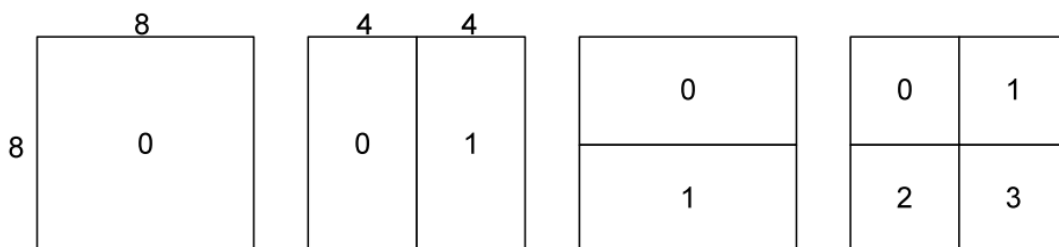
obr. 5 Intra predikční módy [15]

Inter predikce tvoří predikční model z jednoho nebo více již dříve zakódovaných snímků. Tento model je tvořen pomocí posunu vzorků v referenčních snímcích – pohybová kompenzace predikce. Pohybová kompenzace se používá již od MPEG-1, ale tady můžeme pracovat s bloky až do velikosti 4x4. U MPEG-4 AVC se používá stromová struktura, kde každou jasovou složku makrobloku mohou rozdělit 4 způsoby,

např. makroblok 16x16 na 16x16, 16x8, 8x16 a 8x8 (obrázek 6). Stejným principem můžu rozdělit 8x8 (obrázek 7). Touto kombinací lze získat mnoho způsobů uvnitř každého makrobloku. Volba těchto velikostí má velký vliv na komprimační výkon. Obecně platí, že pro oblasti s méně detaily se používá větší rozdělení a pro oblasti s hodně detaily, menší rozdělovací velikost. S touto volbou souvisí i počet bitů potřebných pro volbu pohybových vektorů a typů dělení, tj. pro větší oblasti stačí méně bitů a pro menší oblasti více bitů. Barevné složky (Cr a Cb) makrobloků odpovídají polovině jasových složek. Rozděleny jsou stejným způsobem, až na to, že mají vodorovně i svisle poloviční velikost (8x16 dělení v jasové oblasti, odpovídá 4x8 v barevné oblasti apod.). [16]



obr. 6 Dělení makrobloků na 16x16, 8x16, 16x8 a 8x8 [16]



obr. 7 Dělení makrobloků na 8x8, 4x8, 8x4 a 4x4 [16]

Mluvíme-li o pohybovém vektoru, myslíme tím posun mezi dvěma oblastmi rozdělení makrobloku, tj. mezi vypočteným a referenčním snímkem a má velikost $\frac{1}{4}$ pixelu (sub-pixel) pro jasovou složku. Zavedením sub-pixelu u pohybové kompenzace můžeme zajistit lepší komprimační vlastnosti, než u celočíselných pixelů, které se používají např. u MPEG-2. Pro ještě větší ušetření datového toku se u pohybových vektorů používá predikce, kdy se kóduje rozdíl mezi současným vektorem a vektorem, na který byla už použita predikce. [16]

Transformace nám nabízí hned několik možností podle zvoleného profilu, základní pracuje s velikosti bloků 4x4, Hadamardova transformace užívá velikosti bloků 4x4 pro jasové koeficienty a 2x2 pro barevné koeficienty, u high profilu se používají bloky o velikosti 8x8. Základní transformace používá celočíselnou transformaci, která je odvozena od DCT. To je jedna z mnoha dalších změn oproti MPEG-2. Poté se tyto koeficienty kvantizují podle kvantizační tabulky, vlastně až tady dochází ke kompresi a tedy ztrátě informace. Nyní přichází na řadu rekonstrukční cesta kódování, kde se snímky zpětně dekvantizují, použije se na ně inverzní transformace a rekonstrukční filtr pro potlačení blokové struktury. To vede ke snížení energie obsažené ve snímku a lepší pohybové kompenzaci. Filtr se používá jen na snímky zpracovávané v intra módu. Poté jdou snímky do zásobníku, kde jsou použity pro odhad pohybu a pohybové kompenzace. [11] [17]

Nakonec přijde entropické kódování a i tady přišlo vylepšení, které nám ušetří datový tok až o 15 %. Např. Huffmanovo kódování, které se používá už od dob JPEG, je zde vypuštěno. Jsou zde použity dva typy kódování: CABAC - adaptivní binární aritmetické kódování a CAVLC (Context-Adaptive Variable Length Coding) - kódování s proměnnou délkou slova. [11] [12]

2.6 VP8

Další otevřený standard, který je kvalitou srovnatelný s H.264, byl vytvořen v roce 2010 firmou On2 Technologies a poté ho odkoupila firma Google. Jeho hlavní záměr použití je pro videa na internetových stránkách s obrazovou kvalitou odpovídající PSNR v rozmezí „sledovatelné“ 30 dB až „vizuálně bezeztrátové“ 45 dB. Setkat se s ním můžeme třeba při videohovorech přes Skype [16], implicitně je podporován ve webovém prohlížeči Chrome nebo na video portálu youtube.com [17]. Video může mít teoreticky rozlišení až 16383 x 16383 pixelů, jsou podporovány jen neprokládané snímky (progressive scan) a v porovnání s H.264 je dekodování snímků značně rychlejší, v průměru okolo 30 %. [18]

Obrazová data se zpracovávají pouze v 8 bitovém YUV 4:2:0 video formátu. Obraz je rozdělen na makrobloky o velikost 16x16 pixelů pro jasovou složku a dva makrobloky 8x8 pro chrominanční složky. Dále se jasové a chrominanční bloky dělí na velikost 4x4 pro účely transformace a kvantizace. Tak jako v jiných standardech se při transformaci používá 2D DCT, ale v určité části se používá Walsh-Hadamardova transformace (WHT). Jedná se pouze o případ, kdy je použit „16x16 luma“ intra predikční mód. Stejnoseměrná složka každého 4x4 bloku, na který byla aplikována DCT jednoho makrobloku, je použita pro nový blok o velikost 4x4. Na tento blok je nyní použita Walsh-Hadamardova transformace. Tato transformace druhého řádu má využít korelaci mezi stejnosměrnými složkami. [19]

Oproti standardům z rodiny MPEG zde existují pouze dva typy snímků – intra (I snímky) a inter (P snímky), tedy nepoužívá se obousměrná predikce, čemuž odpovídaly B snímky. [19]

V intra predikčním módu pracujeme se třemi typy bloků:

4x4 luma

16x16 luma

8x8 chroma

Čtyři společné intra módy pro bloky:

H_PRED (horizontální predikce)

V_PRED (vertikální pred.)

DC_PRED (stejnoseměrná pred.)

TM_PRED („TrueMotion“ pred.)

TrueMotion predikční mód je právě použit pouze v VP8, za kterým ještě stojí vývojáři v On2. [18]

C	A ₀	A ₁	A ₂	A ₃
L ₀	X ₀₀	X ₀₁	X ₀₂	X ₀₃
L ₁	X ₁₀	X ₁₁	X ₁₂	X ₁₃
L ₂	X ₂₀	X ₂₁	X ₂₂	X ₂₃
L ₃	X ₃₀	X ₃₁	X ₃₂	X ₃₃

obr. 8 Příklad použití TM_PRED pro blok 4x4 [18]

Na obrázku 8 můžeme vidět, jak tento mód pracuje: C, A a L představují rekonstruované body z již dříve kódovaných bloků. X₀₀ až X₃₃ představuje předpokládané hodnoty pro aktuální blok. Pro bloky 8x8 a 16x16 je princip podobný. [18]

Ve formátu VP8 se používají tři typy referenčních snímků pro inter predikci:

- poslední snímek
- zlatý snímek
- alternativní klíčový snímek

Tento model je daleko méně náročnější na paměť kodéru a dekodéru, než při používání mnohem více referenčních snímků. Predikce je tedy možná buď z předešlého (posledního) snímku nebo ze zlatého a nebo alternativního referenčního snímku. Dále ještě platí, že každý referenční snímek je automaticky zlatý a alternativní, a také jakýkoliv intra snímek může nahradit zlatý nebo alternativní referenční snímek. [18][19]

Zlatý snímek v podstatě představuje vyrovnávací paměť, kde je uložen snímek z libovolné doby v minulosti. Např. to je výhodné pro udržení kopie pozadí obrazu, kde se před ním pohybuje objekt. Ten se pak vzdálí a obraz se efektivněji zrekonstruuje. Při chybě ve video konferencích je obraz rychleji obnoven za použití zlatého snímku. [18]

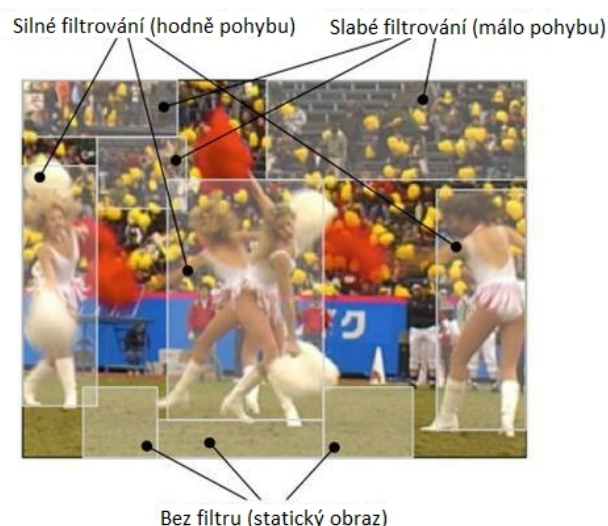
Alternativní klíčový snímek na rozdíl od jiných klíčových snímků, které jsou vidět v dekodéru, může, ale nemusí být vidět uživatelem při dekódování. Tento snímek může být jen odkaz, který slouží pro lepší kompresi, např. jeden takový snímek lze vytvořit z více zdrojových snímků, nebo z různých makrobloků různých snímků z minulosti nebo budoucnosti. Způsobů, jak se alternativní klíčový snímek vytvoří, je tedy mnoho, ale vždy to vede k efektivnímu využití. [18]

V inter režimu máme též několik módů pro predikci:

- blízký („NEAR“)
- nejbližší („NEAREST“)
- SPLITMV

Typicky je v intra režimu blok tvořen pohybovým vektorem z bloku z jednoho ze tří referenčních snímků. VP8 efektivně kóduje pohybové vektory za znovu použití vektorů z okolních makrobloků. Predikční módy NEAR a NEAREST používají poslední a předposlední nenulové pohybové vektory z okolních makrobloků. Další z módů, SPLITMV, flexibilně rozděluje makrobloky na další podbloky, kde každý podblok má svůj pohybový vektor. Tento mód je užitečný u objektů s makrobloky s rozdílnými pohybovými charakteristikami. Pro jasové body pohybová kompenzace používá $\frac{1}{4}$ pixelovou přesnost pohybového vektoru. Pro chrominanční body se pohybový vektor počítá z jasových protějšků průměrováním pohybového vektoru v makrobloku s přesností až osminy pixelu. [18]

Formát obsahuje velmi adaptivní filtr pracující ve smyčce pro odstranění kostičkování obrazu, které vzniká při kódování. V některých situacích se data ani nezpracovávají ve smyčce, což urychlí kódování. Použití filtru koresponduje i s použitými predikčními režimy nebo referenčními snímky, takže se vždy volí síla použití filtru v závislosti na scéně. Pro různé části snímku je možná různá síla filtru, viz obrázek 9.



obr. 9 Použití in-loop filtru [18]

Entropické kódování je založeno na podmíněné rozdělení pravděpodobnosti informací např. pro kódovací modely makrobloků, pohybové vektory, kvantované transformované koeficienty a to vše se vždy zpracovává na úrovni jednoho snímku. V porovnání s CABAC u H.264, tak u VP8 implementace do kodéru není tak složitá a též i obnova po chybě je lepší. [18]

2.7 VP9

Video formát se začal vyvíjet v roce 2011 firmou Google pod hlavičkou WebM.net. Jedná se o nástupce VP8. Tak jako VP8 je „konkurencí“ H.264, tak VP9 má být srovnatelný s HEVC. V době raného vývoje nesl též označení „Next Gen Open Video“ (NGOV) nebo VP-Next. Jako důležité požadavky si vývojáři kladli snížení datového toku o 50 % při stejné obrazové kvalitě, složitost dekódování by neměla být o než 40 % větší a spoustu dalších ve srovnání s předchůdcem [20]. Standard je stále vyvíjen, takže finální parametry se mohou lišit. Vývoj a přehled úprav ve zdrojovém kódu lze sledovat zde [21].

Jednou ze změn je nyní podpora dvou profilů: 0 a 1. Profil 0 podporuje obrazový formát YUV 4:2:0 při 8 bitech, jako VP8, profil 1 nově podporuje 4:2:2, 4:4:4 a alpha kanál [22]. Ve vývoji je i podpora 10 bitového formátu, to by se hlavně využilo pro videa o rozlišení 4K [23].

Snímek se nyní dělí do superbloků (SB) a do makrobloků (MB). SB představuje velikost 64x64 pixelů až do 4x4, vše se dělí stromovou strukturou, až na pár výjimek. Bloky o velikosti SB64, SB32 a MB mohou jako celek sloužit pro intra a inter predikci s použitím dvou pohybových vektorů a odpovídajícím referenčním snímkům. Intra a inter módy také prošly značným vylepšením. Deset režimů pro bloky o velikosti 4x4 a deset pro bloky 8x8 až 64x64 v intra módu. U inter predikce se uvažuje předpověď až ze 4 referenčních snímků, oproti 3 současným. Zcela novým predikčním režimem je inter/intra mód. Jedna se o kombinaci obou režimů. Maximální přesnost pohybového vektoru je nyní 1/8 pixelu. [24]

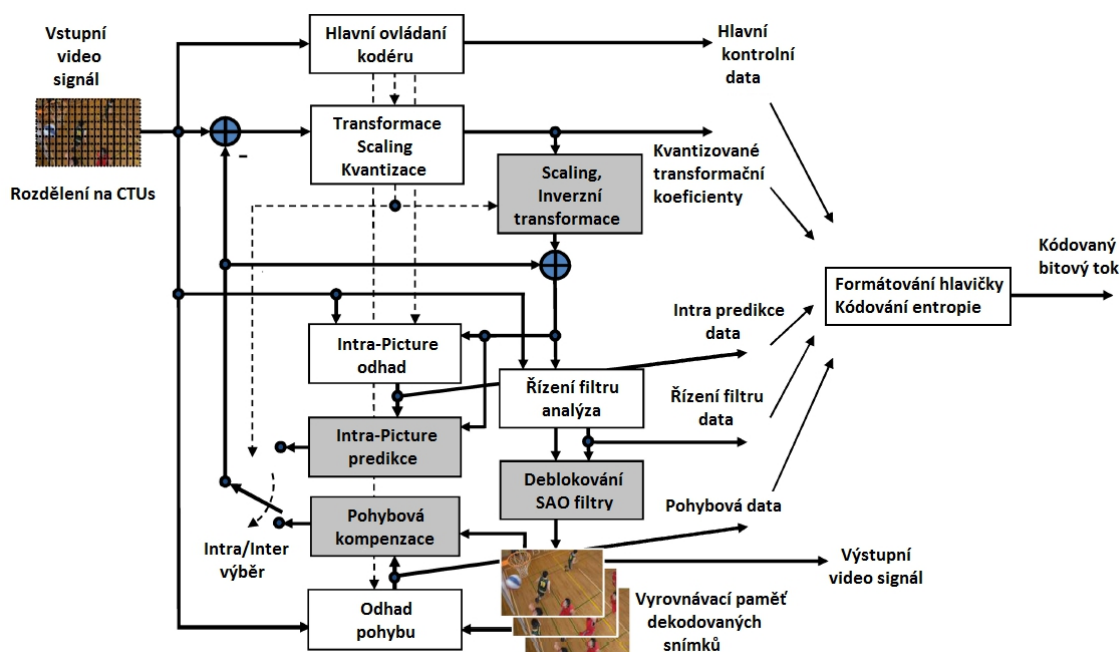
Diskrétní kosinová transformace (DCT) je nyní použita pro bloky o velikost 4x4, 8x8, 16x16 a 32x32. Transformace druhého řádu, která byla ve VP8, se nyní nepoužívá. Byla ale přidána asymetrická diskretní sinová transformace (ADST), která se používá s některými intra módy, kde přináší větší účinnost. [25]

2.9 HEVC – High Efficiency Video Coding

2.9.1 Úvod

Standard byl dokončen v lednu 2013. Stojí za ním spojení ITU-T a ISO/IEC MPEG ve společném vývoji s JCT-VC. Hlavní požadavek je snížení bitové rychlosti na polovinu při stejné obrazové kvalitě ve srovnání s H.264/MPEG-4 AVC, ale na úkor větší výpočetní složitosti. Mezi další požadavky patří podpora vyššího rozlišení videa a zlepšení paralelního zpracování metod.

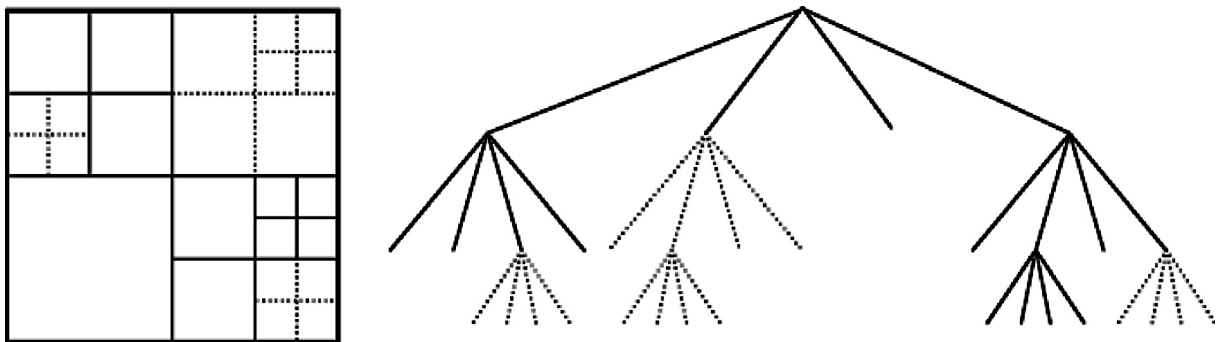
Kódování videa probíhá stejně jako v jiných standardech – intra/inter predikce snímku a 2D transformace. První obrázek je rozdělen na jednotlivé bloky a zpracovává se v intra predikci, další snímky jsou zpracovány v inter predikci, to je, že je predikční informace použita z jiných snímků. Až proběhnou predikční metody, jdou snímky do loop filtru a jsou uloženy do dekódovací vyrovnávací paměti, kde je možné je opět použít na predikci pro další snímky. Zbytkové signály po intra a inter predikci, což je rozdíl mezi původním blokem a predikčním blokem, jsou transformovány lineárními metodami (DCT a DST), pak proběhne scaling, kvantování a nakonec entropické kódování. Video může mít rozlišení až 8K. Typické blokové schéma kódovací části standardu HEVC na obr. 10. [26]



obr. 10 Blokové schéma standardu H.265/HEVC [26]

2.9.2 Bloková struktura

První změna oproti H.264 je, že se už nepoužívají makrobloky (16x16 pixelů), ale nové kódovací schéma používá větší blokovou strukturu až do velikosti 64x64 pixelů a rozdělené části do různých velikostí. Na začátku je obraz rozdělen do stromových částí (CTUs), kde je pak každá jasová a chrominanční složka rozdělena do stromových bloků (kořenů) (CTBs). Velikost stromového bloku může být 64x64, 32x32 nebo 16x16 pro jas, pro chrominanční složky je velikost poloviční při standardním vzorkování YUV 4:2:0. Větší bloky většinou zajistí efektivnější kódování. Stromové bloky jsou pak následně rozděleny na kódovací bloky (CBs). Jasové a chrominanční kódovací bloky se dohromady včetně syntaxe nazývají kódovací jednotky (CUs). Velikost kódovacího bloku je minimálně 8x8 až do 64x64 vzorků. Uspořádání kódovacích bloků uvnitř stromových bloků se říká tzv. kvadrantový strom, tj. rozdělením vzniknou další čtyři části (obr. 11). Kódovací jednotky tvoří základní velikost pro další dělení na predikční jednotky (PUs) nebo transformační jednotky (TUs).



obr. 11 Příklad rozdělení stromového bloku do kódovacích bloků (plná čára)
a transformačních bloků (přerušovaná čára) [26]

Jednotlivé jasové a chrominanční CBs se dále dělí na predikční bloky (PBs) o velikosti 4x4 až 64x64. To jestli predikční jednotka bude použita pro intra nebo inter predikci se určuje už na úrovni CU. Pro jednotlivé režimy nejsou vždy povoleny všechny velikosti bloků. Rozdělení na transformační bloky (TBs) má opět počátek na úrovni CB a dělení může být identické s CB nebo se může dále dělit na menší bloky, viz obr. 11. To platí pro jasové i chrominanční TBs. [26] [27]

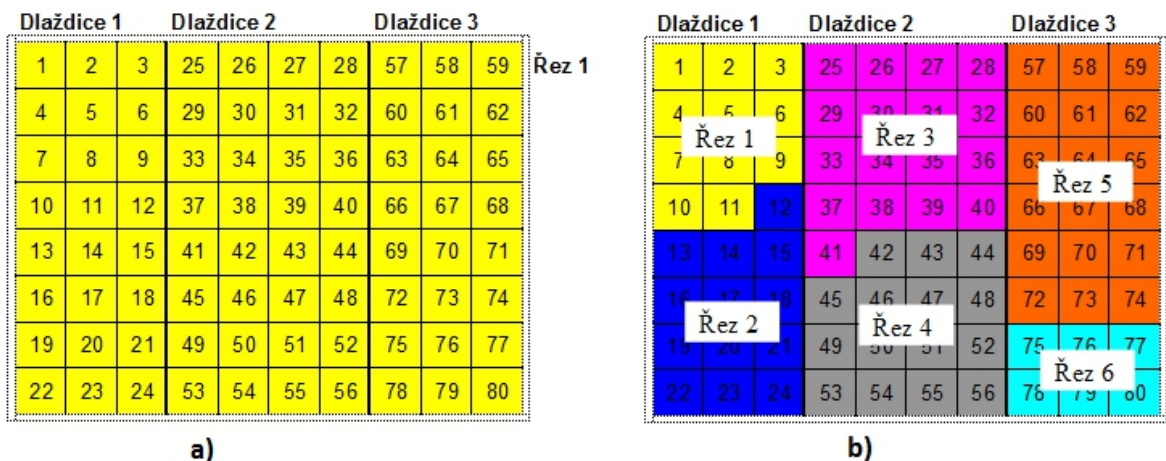
2.9.3 Řezy a dlaždice

Obraz je také možné rozdělit na „řezy“ (Slices) a „dlaždice“ (Tiles). Snímek mohu rozložit buď na jeden nebo více řezů, to se opět provádí na úrovni CTUs. Toto rozdělení se používá v pořadí rastrového snímání. Jednotlivé řezy jsou soběstačné, protože obsahují všechny potřebné hodnoty nutné ke správnému dekódování, takže nepoužívají data z jiných řezů (jen v in-loop filtru jsou blízké/okolní řezy potřebné). Jedná se o data ve smyslu kódování entropie, predikční signál a rozdílný zbytkový signál. Hlavní výhoda řezů je v synchronizaci po ztrátě dat při přenosu, např. streamovaného videa. [26][28]

Řezy dále rozděluje ma:

- I řezy:** Řez ve kterém jsou všechny obsahující CUs zpracovány pouze intra predikcí.
- P řezy:** Navíc mohou být některé CUs zpracovány v inter predikci s použitím nejvýše jednoho kompenzačního pohybového predikčního signálu na jeden PB. Reference je možná pouze z picture list 0.
- B řezy:** Navíc mohou být některé CUs zpracovány v inter predikci s použitím nejvýše dvou kompenzačních pohybových predikčních signálů na jeden PB. Reference je možná z picture list 0 a list 1.

Rozdělení do dlaždic má především uplatnění při paralelním zpracování jak v kódovací, tak dekódovací části. Jednotlivé dlaždice jsou soběstačné a nezávislé na jiných dlaždicích. Tvar je obdélníkový. Na obrázku 2 je vidět příklad možného rozdělení obrazu na jednotlivé řezy a dlaždice – čísla 1-80 představují jednotlivé stromové části (CTUs) a jsou zobrazeny i dva možné způsoby rozdělení, tj. jeden řez pokrývá všechny dlaždice (obr. 12a) nebo řezy pokrývající oblast v jednotlivých dlaždicích (obr. 12b). [26][28]



obr. 12 Příklad rozdělení na řezy a dlaždice

2.9.4 Intra predikce

Intra predikce je založená na stejném principu jako u H.264, tedy na prostorové extrapolaci z již kódovaných obrazových bloků téhož snímku a následné transformaci pomocí DCT. Samozřejmě s několika vylepšeními, které mají více zefektivnit účinnost v kódování textových a strukturních informací v obrazu.

Pro intra predikci máme k dispozici tři základní módy:

- Planar (rovinný)
- DC (stejnoseměrný)
- Angular N (úhlový) (N=2..34)

Číselné označení jednotlivým módům odpovídá: Planar – 0, DC – 1, Angular – 2-34. Ve všech módech se pro predikci užívají hodnoty ze základní referenční sady vzorků zleva a z nad blokem, pro který děláme predikci.

Intra Planar mód používá průměrnou hodnotu ze dvou lineárních predikcí (horizontální a vertikální) a podporuje všechny velikosti bloků.

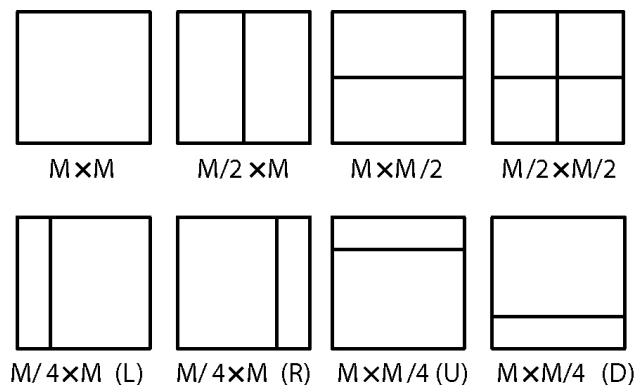
Intra DC mód používá průměrné hodnoty z referenčních vzorků určené pro predikci. Oba tyto módy jsou pouze alternativní k hlavnímu Angular módu.

Intra Angular (N) mód, neboli úhlový, pracuje ve 33 směrech, kde směr značí písmeno N, které nabývá hodnot od 2 do 34. V H.264 je takových směrů pouze 8.

Velikost predikčních bloků je od 4x4 až do 32x32. Na obrázku 13 je vidět podporované dělení do PBs (dělení M/4 není u intra predikce podporováno). [26][29]

2.9.5 Inter predikce

Inter predikce, neboli časová, se provádí stejně jako u H.264 z I, P a B snímků. Reference je možná až ze 6 snímků, v některém případě až ze 16 snímků při nižším rozlišení videa. Nejmenší velikost pro jasové bloky je 4x8 nebo 8x4 až do 32x32. K dispozici máme i asymetrické dělení (AMP – Asymmetric Motion Partitioning), tzn. že CB o velikosti MxM je možný rozdělit na jeden PB o velikosti M x M/4 nebo M/4 x M a druhý PB tedy na velikost 3M/4 x M nebo M x 3M/4. Např. blok o velikosti 32x32 bude rozdělen na dva bloky o velikosti 8x32 a 24x32. Na obrázku 13 je vidět typické rozdělení do PBs pro inter predikci.



obr. 13 Dělení CBs do PBs pro inter a intra predikci, intra predikce nepodporuje M/4 dělení [26]

Každá predikční jednotka, kromě jednotlivých jasových a chrominančních bloků, obsahuje jeden nebo dva pohybové vektory. Pohybové kompenzace se také dostalo změn, zvětšila se přesnost až na 1/4 vzorku pro pohybový vektor. Váhová predikce je zde také podporována. [26] [30]

Pohybový vektor, horizontální i vertikální, definuje 16 bitový rozsah. Máme k dispozici tři módy pro inter predikci – inter (neoficiálně nazývaný Advanced Motion Vector Prediction (AMVP)), merge (sloučený) a skipped (vynechaný) [31]. Všechny módy používají MVC (Motion Vector Competition) schéma, pro zvýšení účinnosti kódování, který vybírá jednoho pohybového kandidáta ze seznamu kandidátů obsahující

prostorové a časové pohybové kandidáty. Intra mód používá data z referenčního snímku nebo může použít data ze sousedních predikčních bloků. Merge mód umožňuje dědit informace ze sousedních bloků. To je podobné “skipped” a “direct” pohybového interferenčního módu u H.264 jen s dvěma vylepšeními. První, že je použito indexové informace k vybrání jednoho z několika kandidátů. Druhé vylepšení je, že je použita informace z referenčního obrazového listu nebo referenčního obrazového indexu. [26]

2.9.6 Transformace

Rozdílový signál po predikci (predikční chybový signál) je pak kódován za použití transformačních jednotek (TUs), které obsahují koeficienty pro prostorové bloky transformace a kvantizace. Velikost transformačních bloků je v rozmezí od 4x4 do 32x32 vzorků (u H.264 je nejmenší velikost bloku 8x8). Dělení na menší bloky má též stromovou strukturu a nazývá se zbytkový kvadrantový strom (Residual quadtree - RQT). Používá se celočíselná transformace odvozená od DCT pro všechny velikosti bloků. U intra predikce pro jasové transformační bloky 4x4 je použita celočíselná transformace odvozená od diskretní sinové transformace, která lépe odpovídá statickému zbytkovému signálu a dokáže snížit datový tok až o 1 %. Pro kvantování se používá stejné schéma jako v H.264 řízené kvantizačním parametrem QP. Tento parametr je definován v rozmezí od 0 do 51. [26] [32]

2.9.7 In-loop filtry

Ve standardu H.265/HEVC máme k dispozici dva in-loop filtry [26]:

Deblocking filter DBF (filtr odstraňující kostičkování)

Sample adaptive offset filter SAO

DBF se nachází už v H.264, ale zde je navržen s cílem lepší subjektivní kvality a zároveň nižší složitostí. Další z cílů bylo zlepšení podpory paralelního zpracování, které je výhodné pro rychlejší zpracování na více jádrových procesorech pro video ve vysokém rozlišení. Filtr se používá pouze na hraničních vzorcích PU nebo TU a jeho síla má až 5 úrovní, ale většinou se používají jen 3. Nově se oproti H.264 používá mřížka o velikosti 8x8 vzorků, místo 4x4. [26]

SAO je použit až po DBF a má za úkol lepší rekonstrukci z původní amplitudy pomocí posunu z „look up“ tabulky. Filtr byl navržen pro zlepšení kvality obrazu a snížení artefaktů. Nejprve jsou klasifikovány regiony vzorků do kategorií s vybraným třídičem, pro každou kategorii získám posun. Získaný posun následně přidám ke vzorkům v regionech. Index třídiče a posun se kóduje zvlášť ve výsledném bitstreamu. [26] [33]

2.9.8 Entropie

Poslední fází kódovacího (první dekodování) procesu je entropické kódování. V H.265/HEVC se používá CABAC, což je v podstatě stejné jako u H.264, jen tady to je jediná metoda, která se aplikuje. Samozřejmě i zde došlo ke změnám ke zvýšení výkonu. Práce CABAC by se dala rozdělit do tří hlavních kroků:

- binarizace
- kontextové modelování
- aritmetické kódování

Binarizace se stará o mapování jednotlivých syntaxí do binárních dat. Syntaxe jsou prvky, které vzniknou během kódování a popisují, jak se bude video rekonstruovat v dekodéru. Kontextové modelování vytváří přesný odhad pravděpodobnosti pro tato binární data, který je velmi důležitý pro vysokou účinnost a aritmetické kódování je komprimuje do bitů na základě pravděpodobnosti. [26][34]

3 Metody pro objektivní hodnocení kvality videa

V této práci pro porovnání video standardů H.264, H.265 a VP9 jsem zvolil čtyři metody pro objektivní hodnocení kvality videa. Jedná se o PSNR, SSIM, MS-SSIM a VIFp. Pro analýzu použiji software „Video Quality Measurement Tool“ VQMT [35].

3.1 PSNR (Peak Signal-to-Noise Ratio)

Nejpoužívanější metodou při základním posuzování kvality jednotlivých video standardů je PSNR (2), neboli špičkový odstup výkonu signálu k šumu. Při výpočtu je nutné definovat MSE (Mean Square Error) (1), tedy střední kvadratickou odchylku, která by při porovnání stejných obrazů byla nulová a PSNR bylo nekonečné. Hodnoty PSNR se udávají v dB a to v rozmezí od nuly do nekonečna, nejčastěji se pohybují mezi 20-60 dB. [36]

$$MSE = \frac{1}{M * N} \sum_{i=1}^M \sum_{j=1}^N (x_{(i,j)} - y_{(i,j)})^2 \quad (1)$$

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \quad (2)$$

Kde M a N představuje rozlišení obrazu, $x_{(i,j)}$ původní obraz, $y_{(i,j)}$ rekonstruovaný obraz [37].

3.2 SSIM (Structural Similarity)

SSIM index je navržen tak, aby respektoval vnímání lidského oka. Při porovnání dvou obrazů se berou v potaz tři faktory a to ztráta korelace, jasové zkreslení a kontrastní zkreslení. Výsledný SSIM index je kombinací těchto třech faktorů a kladné hodnoty jsou v rozmezí od 0 do 1, kdy 1 nastává pro shodné obrazy. [36]

3.3 MS-SSIM (Multi-Scale Structural Similarity)

Metoda je založená na SSIM, ale ještě se tady navíc opakovaně převzorkuje obraz na nižší rozlišení a výsledná hodnota je kombinací těchto hodnot. Jedná se o „přesnější“ metodu, než SSIM, která lépe odpovídá vnímání lidského oka. [38]

3.4 VIF (Visual Information Fidelity)

Metoda, která vychází ze statického modelu pro přírodní scény, zkraslení obrazu a z vnímání lidského oka. Hodnoty VIF se obvykle pohybují mezi 0 a 1, kdy 1 je pro porovnání stejných obrazů. Může ale nastat situace, kdy hodnota bude větší než 1, což představuje vizuálně lepší kvalitu měřeného snímku, než má referenční snímek. Toho se dá například docílit zvýšením kontrastu u měřeného snímku, kdy lidské oko dokáže rozpoznat více objektů ve scéně [38]. Program VQMT obsahuje „pixel domain“ verzi, která není tak složitá [39].

4 Nastavení kodérů

4.1 Referenční software

Pro převod testovacích video sekvencí jsem použil referenční softwary standardů H.264, H.265/HEVC a VP9. Software je dostupný jako zdrojový kód v programovacím jazyku C++. Vývojáři ho uvolňují, jak pro testování a vylad'ování samotného standardu, tak i pro porovnání jednotlivých účinností mezi standardy. Přehled standardů včetně použitých verzí softwaru v této práci je popsáno v tabulce 1.

standard	referenční software
H.264	JM 18.43
H.265	HM 9.0rc1
VP9	VPX v1.2.0-4742-g7fd2561

tab. 1 Video standardy a jejich referenční softwary

Všechny tři výše jmenované softwary se dají volně stáhnout na internetu ze stránek vývojových týmů [40][41][42]. Po stažení stačí zdrojový kód přeložit. Během zpracování této práce byly uvolněny již novější verze, takže především u HEVC jsem používal pracovní verzi.

Obsluha je pro všechny tři programy podobná. Vždy je nutné si vytvořit konfigurační soubor, kde si definujeme parametry vstupního a výstupního videa. Parametry pro vstupní video jsou jaksi dané již samotným vybraným videem, kdežto parametry pro výstupní již kódované video v určitém standardu si definujeme na základě našich požadavků.

4.2 Testovací video sekvence

Pro účely porovnání kompresní účinnosti vybraných video standardů jsem zvolil 4 videa v YUV 4:2:0 a 8 bitovém formátu o různých rozlišení. Videa jsem stáhnul z internetu zde [43]. Přehled názvů videí včetně rozlišení je uveden v tabulce 2.

video	Rozlišení
BasketballDrill_832x480_50.yuv	832 x 480
Johnny_1280x720_60.yuv	1280 x 720
Cactus_1920x1080_50.yuv	1920 x 1080
Traffic_2560x1600_30_crop.yuv	2560 x 1600

tab. 2 Přehled použitých testovacích video sekvencí

Snažil jsem si vybrat, aby každé video bylo obsahem trochu jiné. BasketballDril (obr. 15a) představuje sport, Johnny (obr. 15b) moderátora, Cactus (obr. 15c) různorodý pohyb na scéně a Traffic (obr. 15d) pohyb aut na silnici.



obr. 15 Použitá videa: a) BasketballDrill, b) Johnny, c) Cactus, d) Traffic

4.3 Intra a inter srovnání

Pro porovnání jsem vybral dva základní způsoby, v jakém režimu se bude celé video zpracovávat. Jedná se tedy o intra mód, tedy že všechny snímky jsou kódovány jen v intra predikci (I snímky). Druhý pohled je již typičtější pro použití, zde je první snímek zakódován v intra módu a další se už kódují v inter módu (B, P snímky).

Kvůli výpočetní náročnosti jsem do inter srovnání nezařadil video Traffic_2560x1600_30_crop.yuv.

Referenční softwary pro HEVC a H.264 se zrovna nemohou pyšnit rychlým kódováním, proto v intra srovnání jsem kódoval jen deset snímků a v inter jen jednu periodu snímkovacího kmitočtu, tedy 30, 50 nebo 60 snímků. I přes toto oseknutí, doba kódování jednoho snímku se pohybovala kolem půl hodiny až hodiny, takže jen samotné kódování zabralo nemálo času.

Dále bylo nutné zajistit podobného nastavení jednotlivých kodérů, tedy aby se snímky kódovaly v intra nebo v inter módu. V tabulce 3 jsou uvedeny klíčové parametry, které jsem použil. IntraPeriod říká, jestli se I snímky budou opakovat, či se tak bude kódovat jen první snímek. VP9 používá jinou snímkovou strukturu, takže nastavení je trochu jiné. Parametrem kf-max-dist si definuji jaký má být interval mezi jednotlivými klíčovými snímky.

	INTRA	INTER
H.264	IntraPeriod = 0	IntraPeriod = 1
HEVC	IntraPeriod : 1	IntraPeriod : -1
VP9	--kf-max-dist=1	--kf-max-dist=360

tab. 3 Základní parametry konfiguračních souborů pro kódování

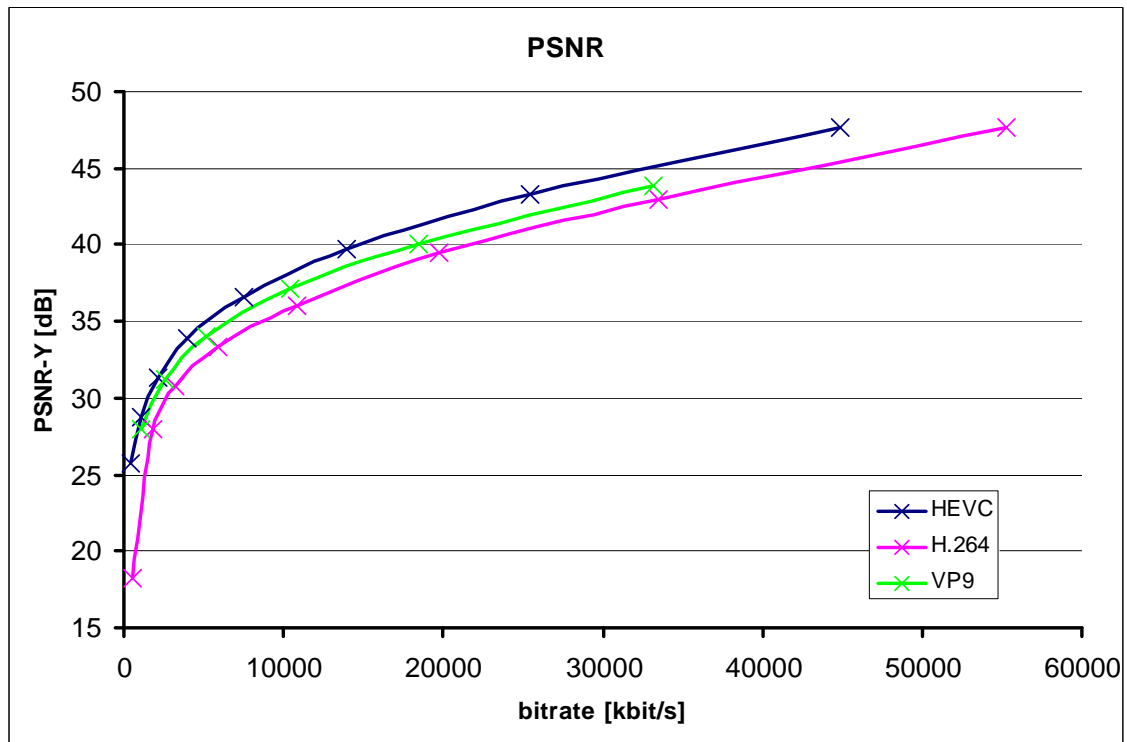
Kvalita se v HEVC a H.264 nastavuje pomocí kvantizačního parametru QP, který je v rozmezí od 0 do 51. Hodnota 0 představuje nejlepší kvalitu, 51 nejhorší. VP9 má podobný parametr, jen se u něj udává minimální a maximální hodnota, takže bylo vždy nutné oběma parametrům nastavit stejnou hodnotu, která se pohybovala od 0 do 63. Nenastavoval jsem tedy přímo přenosovou rychlost, ale snažil jsem se pokrýt pomocí kvantizačního parametru celou škálu možné obrazové kvality. Jednotlivé konfigurační soubory se nachází na přiloženém médiu v příslušných složkách odpovídající názvů standardů.

5 Výsledky porovnání video standardů H.264, HEVC a VP9

5.1 Intra srovnání

V první části porovnávání nejlépe dopadl, téměř pro každé video, standard HEVC metodou PSNR. Dle srovnání jiných objektivních metodách (SSIM, MS-SSIM a VIFp) též nejlépe dopadl HEVC. Standard VP9 se hodnotami u všech videí a objektivních metod velmi přibližoval HEVC, nebo se nacházel mezi HEVC a H.264. Jen v případě videa Traffic_2560x1600_30_crop.yuv dopadl nejlépe VP9, resp. jeho kompresní účinnost je větší. U tohoto videa má kupodivu lepší výsledky i H.264, což ale byl jeho jediný případ, jinak byl vždy horší. Na grafu 1 uvádím pro příklad část výsledků jasové složky PSNR pro video BasketballDrill_832x480_50.yuv, kde můžeme například porovnat přenosovou rychlost pro HEVC a H.264 při stejné kvalitě. Pro PSNR-Y 47,6 dB dosahuje HEVC 44,6 Mbit/s a H.264 55,3 Mbit/s, což je ušetření skoro 11 Mbit/s.

Grafy, včetně tabulek hodnot přenosových rychlostí a výsledků objektivních metod pro ostatní videa, jsou uvedeny v přílohách. Převedená videa jsou k nahlédnutí na přiloženém médiu.

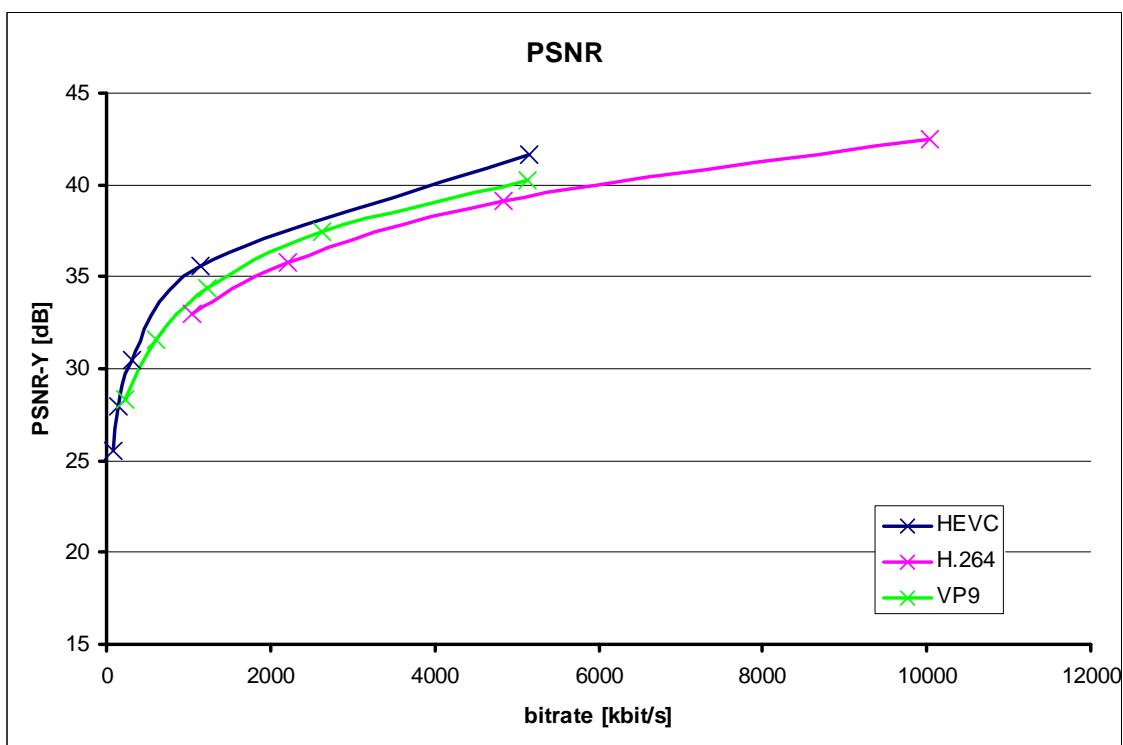


graf 1 Výsledky PSNR pro BasketballDrill_832x480_50.yuv v intra režimu

5.2 Inter srovnání

V druhé části vzájemného srovnávání jsem se zaměřil na standardní využití, kdy se používá inter predikce pro jednotlivé snímky, tedy snímky jsou už nyní predikovány z již dříve zakódovaných snímků. Toto je stěžejní část většiny video standardů, které inter predikci používají.

Jak už v intra predikci, tak i zde podle objektivních měřítek kvality nejlépe dopadl HEVC. VP9 se u všech videí pohyboval podle objektivních měřítek v druhé polovině mezi H.264 a HEVC. H.264 dopadl nejhůře. Opět na videu BasketballDril si ukážeme srovnání pro hodnotu jasové složky PSNR, viz graf 2.



graf 2 Výsledky PSNR pro BasketballDrill_832x480_50.yuv v inter režimu

Na grafu je vidět, že v určitém místě je kompresní účinnost HEVC oproti H.264 až skoro 50 %, což byl jeden z požadavků při vývoji. Pro PSNR-Y 35,57 dB je přenosová rychlost u HEVC 1157 kbit/s a u H.264 odpovídá 35,79 dB 2212 kbit/s. Nabízí se i srovnání v tomto bodě pomocí SSIM, MS-SSIM a VIFp, ale výsledky jsou stejné, takže při kompresi nevznikl šum, který by se neprojevil v PSNR, ale mohl by se objevit u těchto třech metod. Hodnoty pro zbývající videa jsou opět uvedeny v přílohách.

6 Závěrečné shrnutí

V této práci jsem se zaměřil na teoretický přehled současných video standardů, kde jsem se zaměřil na nejnovější H.265/HEVC. Praktická část se zabývá porovnáním kompresní účinnosti vybraných video standardů, jako H.264, HEVC a VP9, podle objektivních metod pro měření kvalita videa. Jedná se o metody PSNR, SSIM, MS-SSIM a VIFp. Metoda PSNR pracuje čistě s energií obrazu, kdežto zbývající metody berou i v potaz psycho-vizuální vnímání člověka. K porovnání jsem si vybral čtyři videa o rozlišení 832 x 480 až 2560 x 1600, viz tab. 2. Pro porovnání jsem si vybral dva pohledy, jakým způsobem se budou zpracovávat jednotlivé snímky. Jednak, že všechny snímky budou kódovány v intra módu, tj. každý snímek je zakódován jako celý, bez predikce z jiného. V druhém případě jsem již použil inter kódování, kdy první snímek je v intra predikci a následující již v inter predikci.

V obou případech vyšel HEVC nejlépe a to podle všech použitých objektivních metod pro měření kvality videa. Jen v případě videa o nejvyšším rozlišení Traffic dopadl hůře než H.264 a VP9 v intra režimu. Standard VP9 je prezentován, jako konkurenční formát k HEVC, ale dle mého měření zatím takových kvalit nedosahuje. Jeho kompresní účinnost je účinnější než H.264. Jedna z podmínek při vývoji HEVC byla, že má být dosaženo 50% snížení přenosové rychlosti při stejné obrazové kvalitě oproti předešlému standardu, tedy H.264. Tuto podmínku se vývojářům v určitém rozmezí obrazové kvality podařilo splnit, viz graf 2. Jedná se hodnoty, které se nacházejí v obloukové části jednotlivých křivek. V jiných částech již nebyly dosaženy tak velké rozdíly.

7 Použité zdroje

- [1] *Apple ProRes: White Paper October 2012* [pdf]. 2012 [cit. 2013]. Dostupné z: http://www.apple.com/final-cut-pro/docs/Apple_ProRes_White_Paper_October_2012.pdf
- [2] About the Apple ProRes Codecs. *Help library* [online]. 2013 [cit. 2013]. Dostupné z: <http://documentation.apple.com/en/finalcutpro/professionalformatsandworkflows/index.html#chapter=10%26section=1>
- [3] Avid DS. In: *Wikipedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013]. Dostupné z: http://en.wikipedia.org/wiki/Avid_DS_Nitris
- [4] Avid DNxHD is First Codec Compliant with SMPTE VC-3. *Internet Archive: Wayback Machine* [online]. 2008 [cit. 2013]. Dostupné z: http://web.archive.org/web/20080215145411/http://www.avid.com/company/releases/2008/080213_dnxhd_avid.html
- [5] *Avid DNxHD Technology: Beauty without the bandwidth. Revolutionary Avid DNxHD® encoding.* [PDF]. 2012 [cit. 2013]. Dostupné z: <http://www.avid.com/static/resources/US/documents/DNxHD.pdf>
- [6] Avid | DNxHD Codec End User Source Code Download Page. *Avid* [online]. © 2013 [cit. 2013]. Dostupné z: <http://www.avid.com/us/request-info/dnxhd>
- [7] *Dirac: The video compression family using open technology* [PDF]. 2007 [cit. 2013]. Dostupné z: <http://www.bbc.co.uk/rd/projects/dirac/press-downloads/dirac-1080p.pdf>
- [8] VC-1 Technical Overview. *Microsoft* [online]. 2007 [cit. 2013]. Dostupné z: <http://www.microsoft.com/windows/windowsmedia/howto/articles/vc1techoverview.aspx>
- [9] New video features! MPEG-2 and VC-1 decode, H.264 encode, CEC support. *Raspberry Pi* [online]. 2012 [cit. 2013]. Dostupné z: <http://www.raspberrypi.org/archives/1839>
- [10] Kalva, H.; Lee, J.-B., "The VC-1 Video Coding Standard," *MultiMedia, IEEE*, vol.14, no.4, pp.88,91, Oct.-Dec. 2007 [cit. 2013]. Dostupné z: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4354161&isnumber=435414>

- [11] H.264: Jeden standard pro všechna videa. *Chip*. 2009, č. 4. Dostupné z: http://earchiv.chip.cz/cs/earchiv/vydani/r-2009/h-264/_files/04-09-054-h264.pdf
- [12] H.264 (MPEG-4 part 10). *Goliáš* [online]. 2004 [cit. 2013]. Dostupné z: <http://www.golias.cz/index.php?modul=audio&sub=clanky&page=clanek&id=1697>
- [13] RICHARDSON, Iain. *White Paper: A Technical Introduction to H.264 / AVC* [PDF]. 2011 [cit. 2013].
- [14] RICHARDSON, Iain. *White Paper: An Overview of H.264 Advanced Video Coding* [PDF]. 2011 [cit. 2013].
- [17] RICHARDSON, Iain. *White Paper: 4x4 Transform and Quantization in H.264/AVC* [PDF]. 2011 [cit. 2013].
- [16] RICHARDSON, Iain. *White Paper: H.264 / AVC Inter Prediction* [PDF]. 2011 [cit. 2013].
- [15] RICHARDSON, Iain. *White Paper: H.264 / AVC Intra Prediction* [PDF]. 2011 [cit. 2013].
- [16] Skype goes VP8, embraces open video codec. *Gigaom* [online]. 2011 [cit. 2013]. Dostupné z: <http://gigaom.com/2011/08/03/skype-vp8-video-conferencing/>
- [18] BANKOSKI, Jim, Paul WILKINS a Yaowu XU. GOOGLE INC. *TECHNICAL OVERVIEW OF VP8, AN OPEN SOURCE VIDEO CODEC FOR THE WEB* [PDF]. 2011 [cit. 2013]. Dostupné z: http://static.googleusercontent.com/external_content/untrusted_dlcp/research.google.com/en/us/pubs/archive/37073.pdf
- [19] BANKOSKI, Jim, J. KOLESZAR, L. QUILLIO, J. SALONEN, P. WILKINS a Y. XU. GOOGLE INC. *VP8 Data Format and Decoding Guide* [PDF]. 2011 [cit. 2013]. Dostupné z: <http://datatracker.ietf.org/doc/rfc6386/>
- [24] GRANGE, A. a H. ALVSTRAND. GOOGLE. *A VP9 Bitstream Overview: draft-grange-vp9-bitstream-00* [PDF]. 2013 [cit. 2013]. Dostupné z: <http://datatracker.ietf.org/doc/draft-grange-vp9-bitstream/>
- [17] Přehrávač videí YouTube HTML5. *YouTube* [online]. 2013 [cit. 2013]. Dostupné z: <http://www.youtube.com/html5>
- [25] GRANGE, A. GOOGLE. *Overview of VP-Next* [PDF]. 2012 [cit. 2013]. Dostupné z: <http://www.ietf.org/proceedings/85/slides/slides-85-videocodec-4.pdf>
- [21] Webm/libvpx. *Chromium Git repositories* [online]. 2013 [cit. 2013]. Dostupné z: <http://git.chromium.org/gitweb/?p=webm/libvpx.git>

- [22] Merge "Add bits for colorspace, profile" into experimental. *Chromium Git repositories* [online]. 2013 [cit. 2013]. Dostupné z: <http://git.chromium.org/gitweb/?p=webm/libvpx.git;a=commit;h=229f9a5f0942a7902bceb0d6a4149e17c9293447>
- [23] VP9 Bitstream finalization update. *Google* [online]. 2013 [cit. 2013]. Dostupné z: <https://groups.google.com/a/webmproject.org/forum/?fromgroups=#!topic/webm-discuss/UzoX7owhwB0>
- [20] GOOGLE. *Next Gen Open Video (NGOV) Requirements* [PDF]. 2012 [cit. 2013]. Dostupné z: <http://downloads.webmproject.org/ngov2012/pdf/02-ngov-product-requirements.pdf>
- [26] Sullivan, G.J.; Ohm, J.; Woo-Jin Han; Wiegand, T., "Overview of the High Efficiency Video Coding (HEVC) Standard," *Circuits and Systems for Video Technology, IEEE Transactions on* , vol.22, no.12, pp.1649,1668, Dec. 2012
<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6316136&isnumber=6403920>
- [32] Ohm, J.; Sullivan, G.J.; Schwarz, H.; Thio Keng Tan; Wiegand, T., "Comparison of the Coding Efficiency of Video Coding Standards—Including High Efficiency Video Coding (HEVC)," *Circuits and Systems for Video Technology, IEEE Transactions on* , vol.22, no.12, pp.1669,1684, Dec. 2012
 Dostupné z: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6317156&isnumber=6403920>
- [28] Misra, K.; Segall, A.; Horowitz, M.; Shilin Xu; Fuldseth, A.; Minhua Zhou, "An Overview of Tiles in HEVC," *Selected Topics in Signal Processing, IEEE Journal of* , vol.7, no.6, pp.969,977, Dec. 2013
 URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6547985&isnumber=657464>
- [27] Bross, B.; Helle, P.; Oudin, S.; Tung Nguyen; Marpe, D.; Schwarz, H.; Wiegand, T., "Quadtree structures and improved techniques for motion representation and entropy coding in HEVC," *Consumer Electronics - Berlin (ICCE-Berlin), 2012 IEEE International Conference on* , vol., no., pp.26,30, 3-5 Sept. 2012
 URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6336460&isnumber=6336443>

- [30] Ohm, J.; Sullivan, G.J., "High efficiency video coding: the next frontier in video compression [Standards in a Nutshell]," *Signal Processing Magazine, IEEE* , vol.30, no.1, pp.152,158, Jan. 2013
URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6375943&isnumber=6375903>
- [29] Lainema, J.; Bossen, F.; Woo-Jin Han; Junghye Min; Ugur, K., "Intra Coding of the HEVC Standard," *Circuits and Systems for Video Technology, IEEE Transactions on* , vol.22, no.12, pp.1792,1801, Dec. 2012
URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6317153&isnumber=6403920>
- [31] Jian-Liang Lin; Yi-Wen Chen; Yu-Wen Huang; Shaw-Min Lei, "Motion Vector Coding in the HEVC Standard," *Selected Topics in Signal Processing, IEEE Journal of* , vol.7, no.6, pp.957,968, Dec. 2013
URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6553006&isnumber=6657464>
- [34] Sze, V.; Budagavi, M., "High Throughput CABAC Entropy Coding in HEVC," *Circuits and Systems for Video Technology, IEEE Transactions on* , vol.22, no.12, pp.1778,1791, Dec. 2012
URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6317157&isnumber=6403920>
- [33] Chih-Ming Fu; Alshina, E.; Alshin, A.; Yu-Wen Huang; Ching-Yeh Chen; Chia-Yang Tsai; Chih-Wei Hsu; Shaw-Min Lei; Jeong-Hoon Park; Woo-Jin Han, "Sample Adaptive Offset in the HEVC Standard," *Circuits and Systems for Video Technology, IEEE Transactions on* , vol.22, no.12, pp.1755,1764, Dec. 2012
URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6324411&isnumber=6403920>
- [35] VQMT: Video Quality Measurement Tool. *École polytechnique fédérale de Lausanne EPFL* [online]. 2013 [cit. 2013]. Dostupné z: <http://mmspg.epfl.ch/vqmt>
- [36] Hore, A.; Ziou, D., "Image Quality Metrics: PSNR vs. SSIM," *Pattern Recognition (ICPR), 2010 20th International Conference on* , vol., no., pp.2366,2369, 23-26 Aug. 2010
URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5596999&isnumber=5595735>

- [37] ČVUT. *Koprese obrazu podle standardu MPEG 2 (A2B37MMT Laboratorní cvičení)*. 2012. [cit. 2013].
- [38] Wang, Z.; Simoncelli, E.P.; Bovik, A.C., "Multiscale structural similarity for image quality assessment," *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Seventh Asilomar Conference on* , vol.2, no., pp.1398,1402 Vol.2, 9-12 Nov. 2003
URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1292216&isnumber=28784>
- [38] Image Information and Visual Quality. *Laboratory for Image & Video Engineering* [online]. 2012 [cit. 2013]. Dostupné z: <http://live.ece.utexas.edu/research/Quality/vif.htm>
- [39] Sheikh, H.R.; Bovik, A.C., "Image information and visual quality," *Image Processing, IEEE Transactions on* , vol.15, no.2, pp.430,444, Feb. 2006
URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1576816&isnumber=3331>
- [40] H.264/AVC Software Coordination. *Fraunhofer Heinrich Hertz Institute* [online]. 2013 [cit. 2013]. Dostupné z: <http://iphone.hhi.de/suehring/tml/>
- [41] High Efficiency Video Coding (HEVC). *Fraunhofer Heinrich Hertz Institute* [online]. 2013 [cit. 2013]. Dostupné z: <http://hevc.hhi.fraunhofer.de/>
- [42] Developer Overview. *The WebM Project* [online]. 2013 [cit. 2013]. Dostupné z: <http://www.webmproject.org/code/>
- [43] <ftp://hvc:US88Hula@ftp.tnt.uni-hannover.de/testsequences>
- [44] CASSIDY, Sean A. *An Analysis of VP8, a New Video Codec for the Web*. Rochester, NY, 2011. Dostupné z: <https://ritdml.rit.edu/handle/1850/14525>. Diplomová. Rochester Institute of Technology.
- [45] *HM Software Manual* [PDF].
- [46] *H.264/14496-10 AVC Reference Software Manual* [PDF].
- [47] VP8 Encode Parameter Guide. *The WebM Project* [online]. 2013 [cit. 2013]. Dostupné z: <http://www.webmproject.org/docs/encoder-parameters/>

Zkratky

AVC - Advanced Video Coding
CABAC - Context-adaptive binary arithmetic coding
CB - Code Block
CTB - Code Tree Block
CTU - Code Tree Unit
CU - Code Unit
DCT - Discrete Cosine Transform
DST - Discrete Sine Transform
GOP - Group of Pictures
HD - High Definition
HDTV - High Definition Television
HEVC - High Efficiency Video Coding
JCT-VC - Joint Collaborative Team on Video Coding
JPEG - Joint Photographic Expert Group
JVT - The Joint Video Team
MPEG - Moving Picture Experts Group
MS-SSIM - Multi-Scale Structural Similarity
PB - Prediction Block
PSNR - Peak Signal-to-Noise Ratio
PU - Prediction Unit
SD - Standard Definition
SMPTE - Society of Motion Picture and Television Engineers
SSIM - Structural Similarity
StEM - Standard Evaluation Material
TB – Transform Block
TU - Transform Unit
VBR - Variable bitrate
VCEG - The Video Coding Experts Group
VIF - Visual Information Fidelity
VLC - Variable-length Code

Přílohy

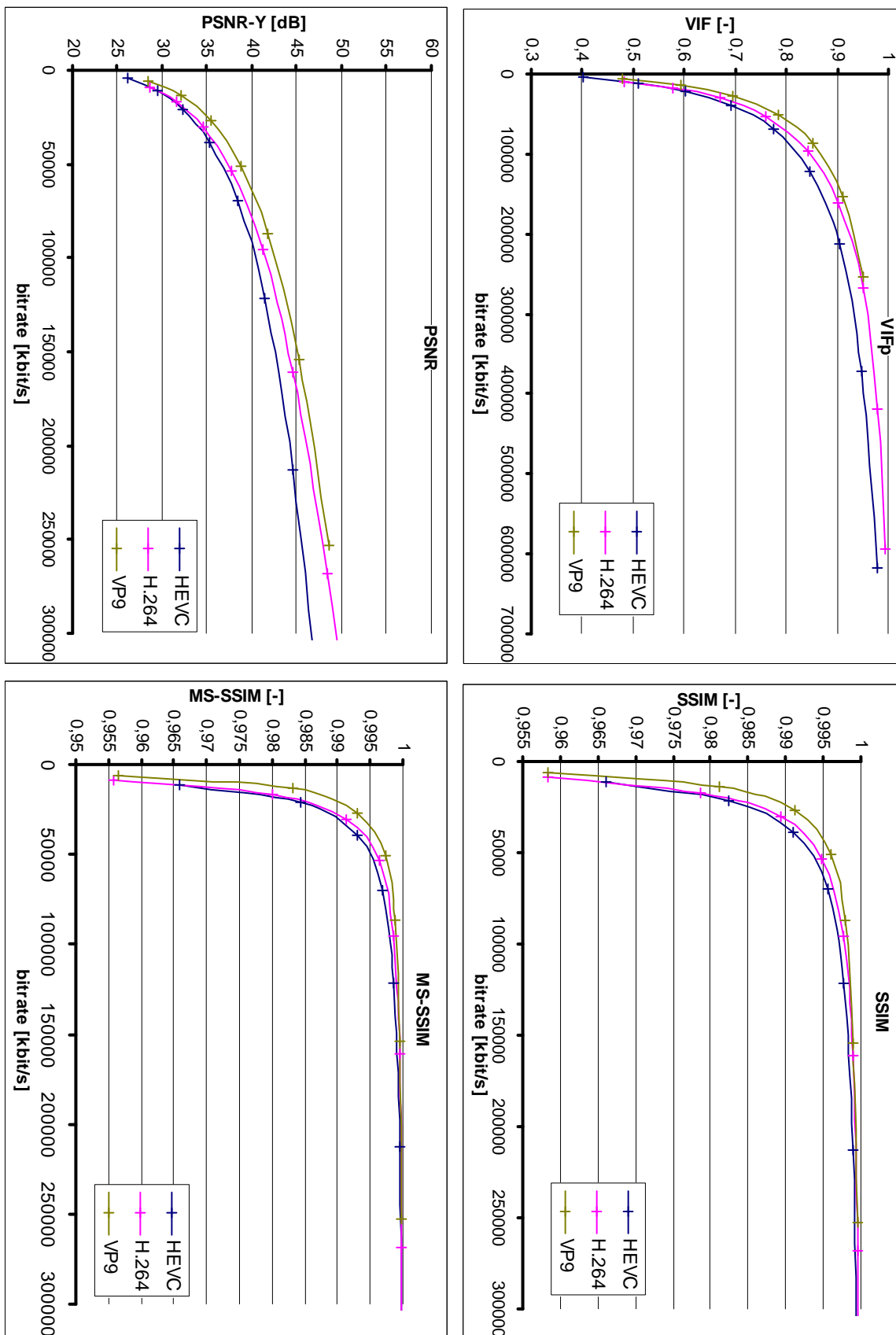
- Příloha č. 1 Tabulka intra – Traffic
- Příloha č. 2 Graf intra – Traffic
- Příloha č. 3 Tabulka intra – Johnny
- Příloha č. 4 Graf intra – Johnny
- Příloha č. 5 Tabulka intra – Basket
- Příloha č. 6 Graf intra – Basket
- Příloha č. 7 Tabulka intra – Cactus
- Příloha č. 8 Graf intra – Cactus
- Příloha č. 9 Tabulka inter – Johnny
- Příloha č. 10 Graf inter – Johnny
- Příloha č. 11 Tabulka inter – Basket
- Příloha č. 12 Graf inter – Basket
- Příloha č. 13 Tabulka inter – Cactus
- Příloha č. 14 Graf inter – Cactus
- Příloha č. 15 Popis obsahu přiloženého média

Příloha č. 1

HEVC	bitrate [kbit/s]	PSNR [dB]	SSIM [-]	MS-SSIM [-]	VIFp [-]
	617417,8	52,50	0,99983	0,999929	0,978332
	371140,6	48,20	0,999537	0,999789	0,948364
	212957,0	44,65	0,998939	0,999449	0,903412
	121212,3	41,42	0,997785	0,998676	0,844718
	69651,5	38,34	0,995536	0,996943	0,774638
	38926,4	35,28	0,991035	0,992935	0,692995
	21322,5	32,29	0,98232	0,984294	0,603429
	10992,8	29,39	0,966147	0,965798	0,509737
	4279,7	26,23	0,931535	0,918862	0,401695
H.264	bitrate [kbit/s]	PSNR [dB]	SSIM [-]	MS-SSIM [-]	VIFp [-]
	594168,9	59,14	0,999963	0,999985	0,994993
	419713,1	52,55	0,999832	0,99993	0,97857
	268301,8	48,35	0,999554	0,999799	0,949802
	161110,8	44,47	0,998905	0,999416	0,900647
	95502,8	41,26	0,997707	0,99864	0,841399
	53597,0	37,72	0,994842	0,996337	0,758534
	30274,9	34,52	0,989307	0,991298	0,671042
	16928,8	31,49	0,97863	0,980099	0,578656
	9012,7	28,55	0,958377	0,955781	0,482108
VP9	bitrate [kbit/s]	PSNR [dB]	SSIM [-]	MS-SSIM [-]	VIFp [-]
	252915,0	48,56	0,99957	0,999802	0,951608
	154179,1	45,20	0,999058	0,99951	0,911456
	86806,5	41,77	0,997934	0,998765	0,851528
	51085,1	38,80	0,995935	0,997241	0,785145
	26974,9	35,42	0,991227	0,993126	0,695969
	13502,6	32,03	0,981143	0,983097	0,594339
	5833,9	28,50	0,958442	0,956583	0,478806

Tabulka intra – Traffic

Příloha č. 2



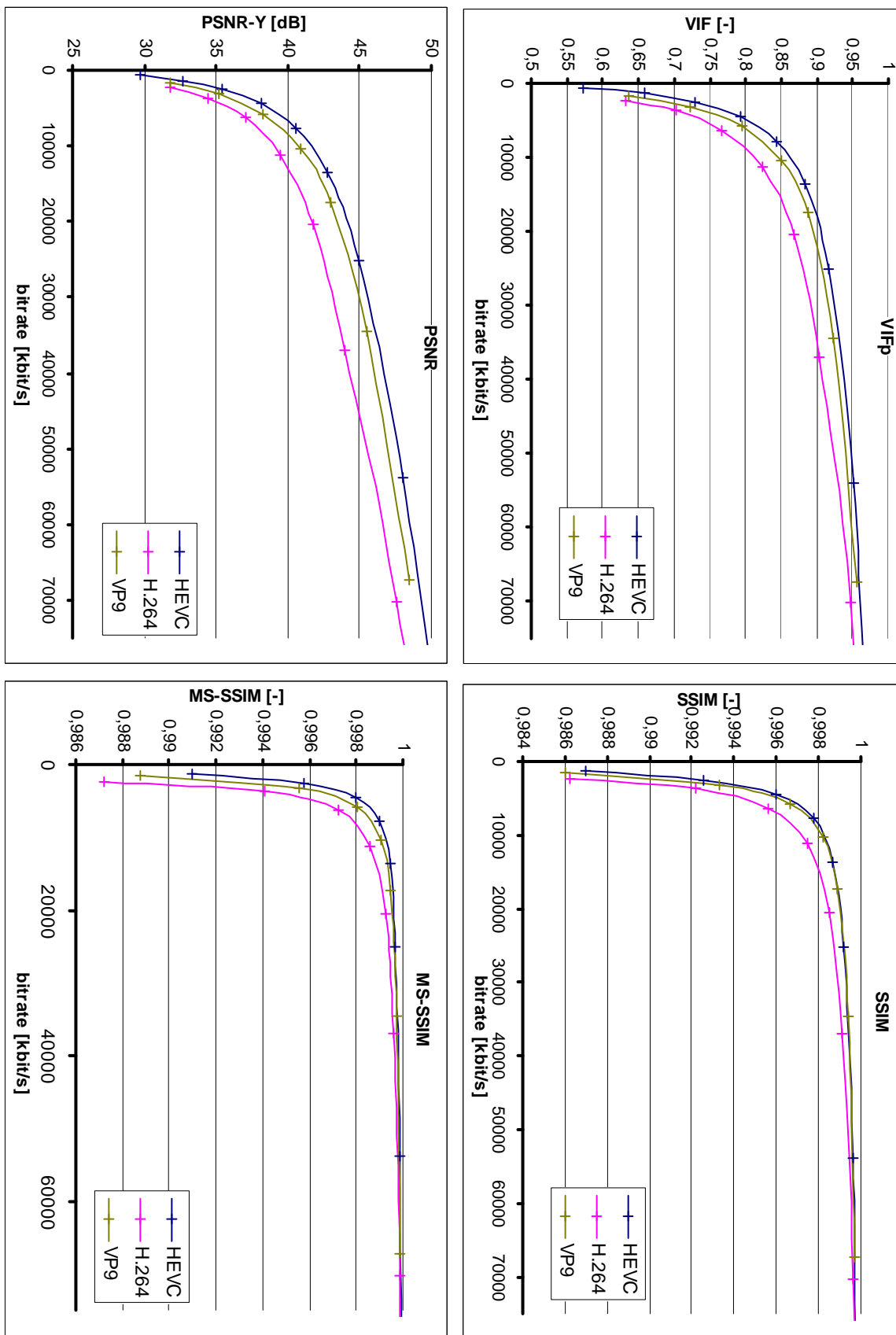
Graf intra – Traffic

Příloha č. 3

HEVC	bitrate [kbit/s]	PSNR [dB]	SSIM [-]	MS-SSIM [-]	VIFp [-]
	111918,1	52,26	0,999852	0,999949	0,979305
	53875,1	48,05	0,999609	0,999851	0,951712
	25111,4	44,91	0,999184	0,999675	0,915653
	13600,7	42,77	0,998633	0,999428	0,882972
	7769,0	40,58	0,997732	0,99896	0,84351
	4444,6	38,12	0,995997	0,997969	0,792532
	2486,3	35,39	0,992536	0,995752	0,728444
	1377,2	32,71	0,986994	0,990979	0,658248
	696,5	29,68	0,97543	0,977568	0,571924
H.264	bitrate [kbit/s]	PSNR [dB]	SSIM [-]	MS-SSIM [-]	VIFp [-]
	205235,4	58,27	0,999967	0,999987	0,994466
	128963,6	51,95	0,999861	0,999946	0,977961
	70213,7	47,56	0,999616	0,999836	0,94722
	37048,7	43,95	0,999099	0,99959	0,902144
	20442,9	41,81	0,998517	0,999274	0,867287
	11167,4	39,49	0,997458	0,99861	0,822755
	6321,3	37,01	0,995592	0,997207	0,767475
	3664,3	34,40	0,992162	0,994064	0,702498
	2292,4	31,79	0,986243	0,987228	0,632234
VP9	bitrate [kbit/s]	PSNR [dB]	SSIM [-]	MS-SSIM [-]	VIFp [-]
	67283,4	48,48	0,999691	0,99987	0,955677
	34523,1	45,50	0,999377	0,999723	0,923904
	17393,8	43,03	0,99889	0,999476	0,887662
	10348,8	40,92	0,998193	0,999062	0,850467
	5784,4	38,25	0,996654	0,998054	0,795561
	3156,4	35,17	0,993273	0,995572	0,72316
	1599,4	31,85	0,986038	0,988727	0,635874

Tabulka intra – Johnny

Příloha č. 4



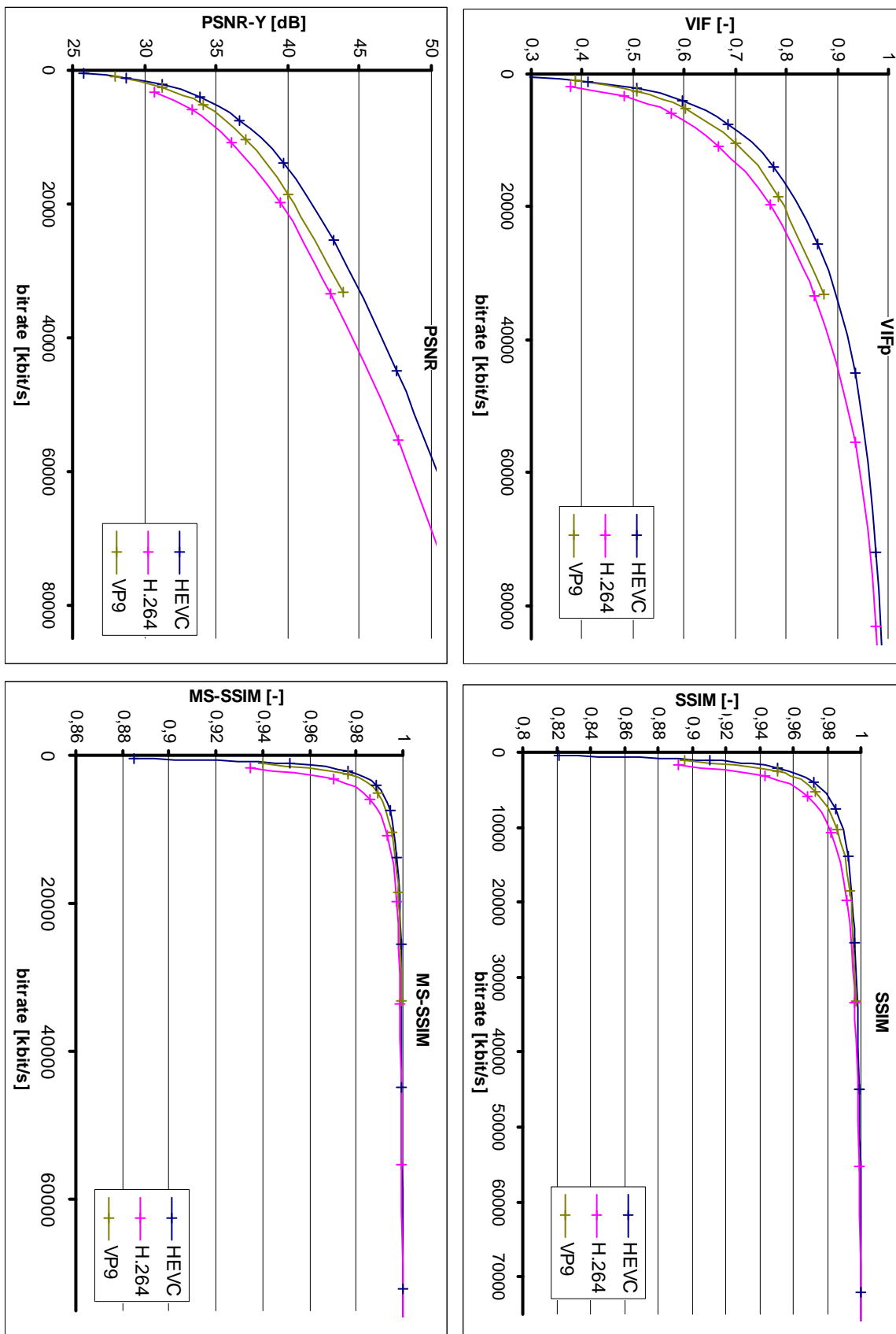
Graf intra – Johnny

Příloha č. 5

HEVC	bitrate [kbit/s]	PSNR [dB]	SSIM [-]	MS-SSIM [-]	VIFp [-]
	121629,2	74,78	0,999998	1	0,99983
	112901,2	70,14	0,999993	0,999998	0,999525
	105392,5	61,59	0,999952	0,999986	0,996717
	91813,8	56,35	0,999837	0,999951	0,989235
	72028,9	52,63	0,999614	0,999887	0,975724
	44906,0	47,64	0,998778	0,999649	0,933814
	25452,8	43,25	0,99663	0,99902	0,86075
	13915,5	39,68	0,992342	0,99759	0,774622
	7503,6	36,60	0,984669	0,994547	0,684852
	3940,7	33,87	0,972015	0,988566	0,596645
	2125,5	31,30	0,950602	0,976643	0,507
	1099,0	28,70	0,910758	0,951385	0,412232
	402,3	25,76	0,821034	0,88513	0,296311
H.264	bitrate [kbit/s]	PSNR [dB]	SSIM [-]	MS-SSIM [-]	VIFp [-]
	1776,1	27,95	0,891805	0,934826	0,376007
	3254,5	30,73	0,942979	0,970466	0,482307
	5874,4	33,31	0,968066	0,986008	0,575488
	10812,9	36,05	0,9826	0,993547	0,666622
	19715,0	39,47	0,991989	0,997477	0,768642
	33484,6	42,97	0,996418	0,998927	0,854879
	55323,5	47,66	0,998784	0,999648	0,933897
	83016,5	52,52	0,999604	0,999883	0,975133
	113810,3	59,42	0,99992	0,999976	0,994588
VP9	bitrate [kbit/s]	PSNR [dB]	SSIM [-]	MS-SSIM [-]	VIFp [-]
	33148,9	43,84	0,997074	0,99913	0,872776
	18504,0	40,07	0,993039	0,9978	0,784968
	10386,0	37,11	0,986374	0,995298	0,700552
	5179,4	34,07	0,973386	0,989345	0,603318
	2581,2	31,24	0,950274	0,97674	0,505914
	1032,8	27,98	0,895815	0,940391	0,385713

Tabulka intra – Basket

Příloha č. 6



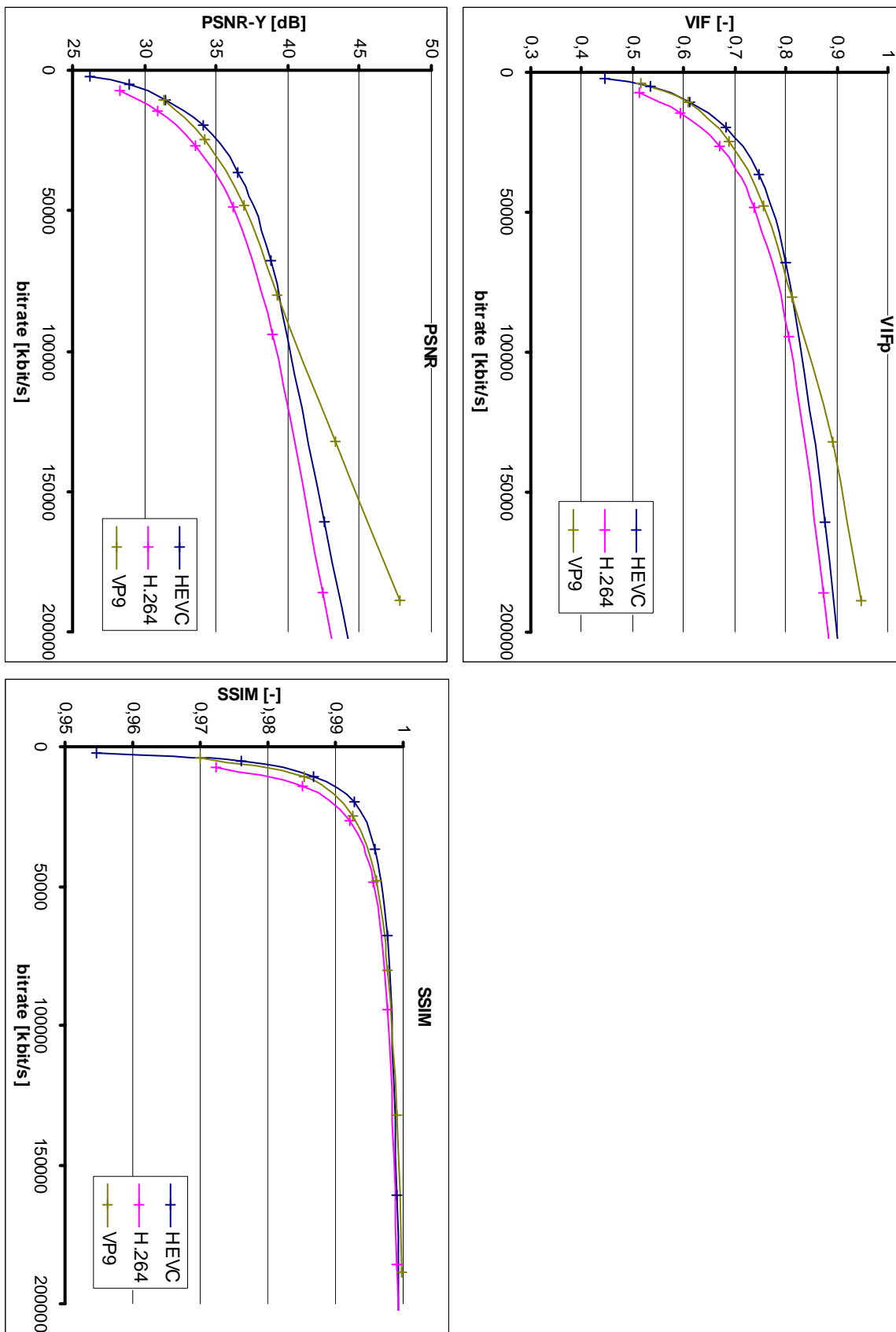
Graf intra – Basket

Příloha č. 7

HEVC	bitrate [kbit/s]	PSNR [dB]	SSIM [-]	VIFp [-]
	294167,1	47,83	0,999706	0,94925
	160748,7	42,54	0,999007	0,877249
	68056,8	38,77	0,997627	0,800699
	36559,2	36,51	0,995916	0,746635
	19726,7	34,07	0,992764	0,68312
	10500,3	31,50	0,986816	0,611147
	5254,0	28,90	0,975956	0,534093
	1995,1	26,16	0,954642	0,446447
H.264	bitrate [kbit/s]	PSNR [dB]	SSIM [-]	VIFp [-]
	639744,7	60,10	0,999983	0,996313
	472904,7	52,85	0,999908	0,981622
	321068,9	47,71	0,999699	0,948083
	186135,6	42,38	0,998979	0,874478
	94277,0	38,94	0,997718	0,804455
	48477,1	36,20	0,995625	0,738694
	26648,2	33,61	0,991979	0,669969
	14338,1	30,96	0,985076	0,593853
	7453,2	28,30	0,972391	0,513413
VP9	bitrate [kbit/s]	PSNR [dB]	SSIM [-]	VIFp [-]
	188736,3	47,82	0,99969	0,948951
	132230,1	43,36	0,999136	0,89109
	80167,0	39,29	0,99778	0,811621
	47925,4	36,94	0,996101	0,756579
	24576,5	34,26	0,992639	0,687596
	10756,0	31,37	0,985453	0,606998
	4196,0	28,29	0,970104	0,514643

Tabulka intra – Cactus

Příloha č. 8



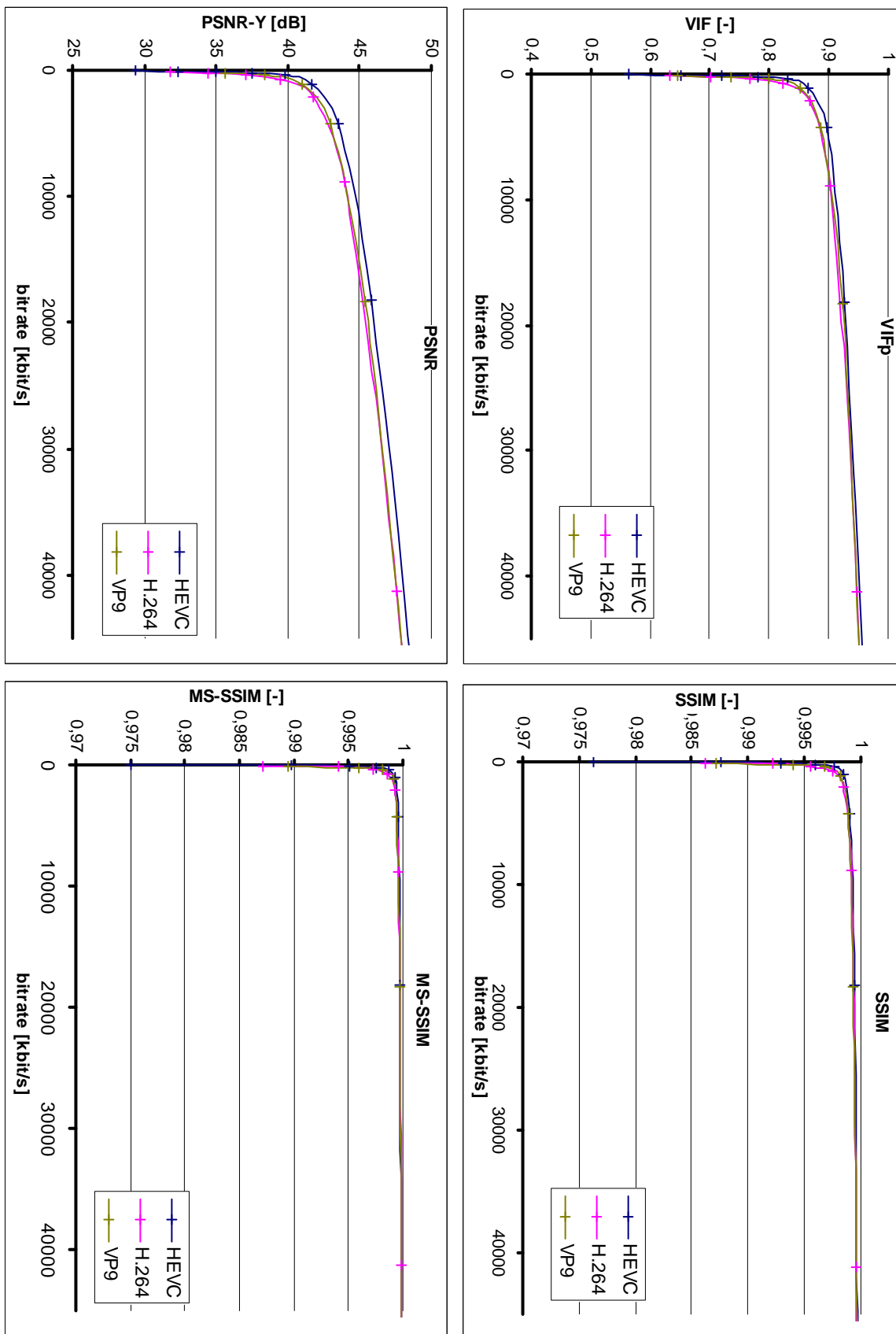
Graf intra – Cactus

Příloha č. 9

HEVC	bitrate [kbit/s]	PSNR [dB]	SSIM [-]	MS-SSIM [-]	VIFp [-]
	133311,4	54,82	0,99992	0,99997	0,986839
	57787,8	49,47	0,999746	0,999897	0,962593
	18165,2	45,79	0,999405	0,999737	0,926952
	4279,1	43,55	0,999003	0,999547	0,895902
	1070,6	41,71	0,998475	0,999259	0,865562
	431,4	39,81	0,997636	0,998724	0,829544
	216,3	37,53	0,996018	0,997565	0,780547
	117,4	34,95	0,9929	0,995025	0,718964
	63,9	32,33	0,987551	0,989694	0,649927
	32,8	29,36	0,976328	0,975106	0,563732
H.264	bitrate [kbit/s]	PSNR [dB]	SSIM [-]	MS-SSIM [-]	VIFp [-]
	178920,8	58,27	0,999967	0,999987	0,994466
	101713,6	51,95	0,999861	0,999946	0,977961
	41213,7	47,56	0,999616	0,999836	0,94722
	8874,6	43,95	0,999099	0,99959	0,902144
	2112,2	41,81	0,998517	0,999274	0,867287
	745,0	39,49	0,997458	0,99861	0,822755
	355,3	37,01	0,995592	0,997207	0,767475
	188,6	34,40	0,992162	0,994064	0,702498
	116,5	31,79	0,986243	0,987228	0,632234
VP9	bitrate [kbit/s]	PSNR [dB]	SSIM [-]	MS-SSIM [-]	VIFp [-]
	50981,2	48,41	0,999685	0,999868	0,955092
	18281,0	45,38	0,999357	0,999717	0,922409
	4270,9	42,95	0,998862	0,999471	0,886428
	1177,8	40,96	0,998201	0,99909	0,851535
	416,4	38,42	0,996766	0,998139	0,799433
	208,0	35,69	0,994028	0,99592	0,735771
	91,672	32,25	0,987107	0,989533	0,646919

Tabulka inter – Johnny

Příloha č. 10



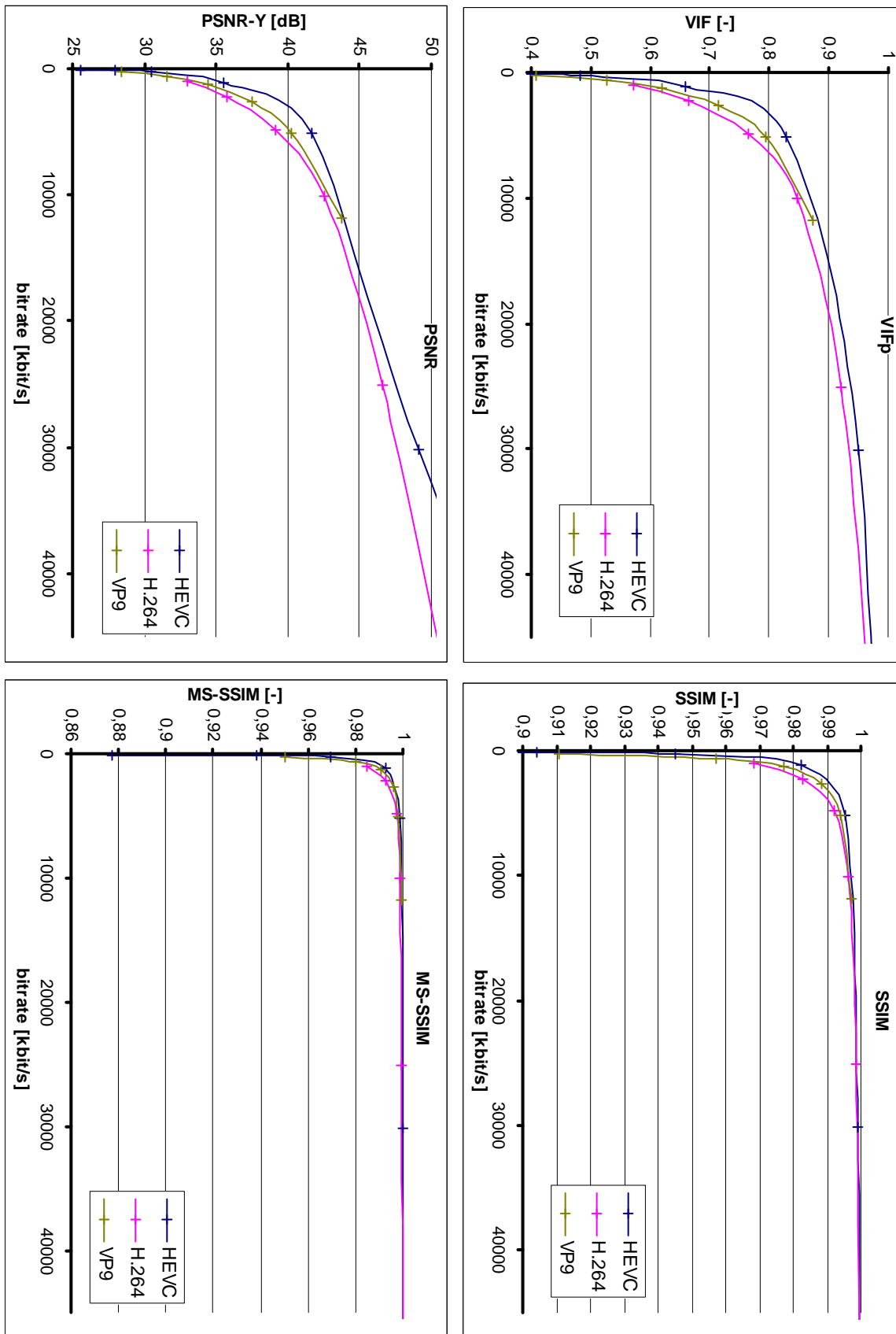
Graf inter – Johnny

Příloha č. 11

HEVC	bitrate [kbit/s]	PSNR [dB]	SSIM [-]	MS-SSIM [-]	VIFp [-]
	73,8	25,52	0,832624	0,877525	0,290904
	155,4	27,98	0,904091	0,938172	0,387956
	304,5	30,44	0,945335	0,969308	0,481744
	1156,9	35,57	0,982367	0,992638	0,65836
	5145,5	41,69	0,995532	0,998546	0,828469
	30106,9	49,15	0,999125	0,999744	0,948914
	84445,8	67,56	0,999981	0,999994	0,998612
H.264	bitrate [kbit/s]	PSNR [dB]	SSIM [-]	MS-SSIM [-]	VIFp [-]
	1037,0	32,99	0,968207	0,984621	0,571124
	2211,7	35,79	0,98293	0,992981	0,664296
	4841,8	39,14	0,991967	0,997164	0,763515
	10043,1	42,51	0,996294	0,998794	0,847455
	25097,8	46,65	0,998549	0,999567	0,921769
	52347,7	51,79	0,999558	0,999868	0,971412
	78954,6	57,48	0,99988	0,999964	0,991675
VP9	bitrate [kbit/s]	PSNR [dB]	SSIM [-]	MS-SSIM [-]	VIFp [-]
	11844,5	43,70	0,997163	0,999119	0,87212
	5125,3	40,27	0,993809	0,997921	0,793852
	2626,4	37,45	0,988296	0,995717	0,715505
	1237,4	34,40	0,977008	0,990432	0,620308
	598,1	31,61	0,957339	0,979797	0,525864
	239,2	28,35	0,910537	0,949946	0,408149

Tabulka inter – Basket

Příloha č. 12



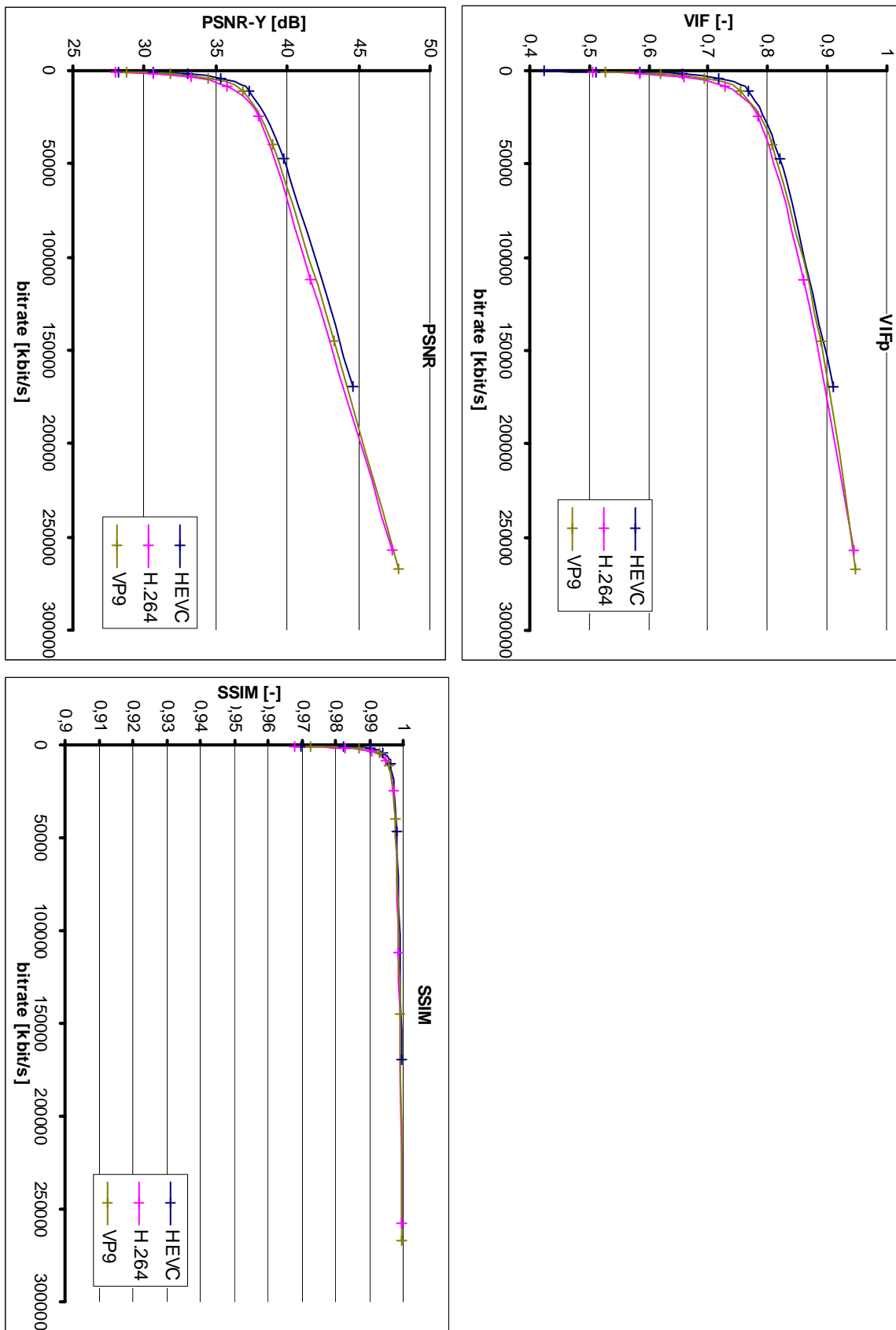
Graf Inter – Basket

Příloha č. 13

HEVC	bitrate [kbit/s]	PSNR [dB]	SSIM [-]	VIFp [-]
	169567,9	44,61	0,99932	0,9089
	46927,6	39,72	0,997933	0,820399
	10566,9	37,33	0,996389	0,766533
	4336,2	35,32	0,994122	0,716365
	2069,3	33,04	0,990007	0,655204
	1033,0	30,64	0,982525	0,585783
	494,1	28,21	0,969636	0,511442
	192,5	25,54	0,943794	0,424661
H.264	bitrate [kbit/s]	PSNR [dB]	SSIM [-]	VIFp [-]
	257237,6	47,31	0,999651	0,943979
	112022,4	41,64	0,998724	0,860627
	24278,2	38,02	0,996999	0,783035
	8084,9	35,77	0,994862	0,727459
	3704,4	33,26	0,990677	0,660115
	1785,6	30,64	0,982562	0,584028
	913,5	28,02	0,967976	0,504636
VP9	bitrate [kbit/s]	PSNR [dB]	SSIM [-]	VIFp [-]
	267135,5	47,75	0,999686	0,948381
	145131,0	43,27	0,999117	0,889561
	39641,9	39,03	0,997646	0,806104
	10810,7	36,88	0,996046	0,755335
	4390,8	34,44	0,992949	0,692792
	2014,1	31,80	0,986836	0,619365
	791,7	28,70	0,972665	0,526964

Tabulka inter – Cactus

Příloha č. 14



Graf inter – Cactus

Příloha č. 15

Na přiloženém datovém nosiči jsou k dispozici všechna zdrojová a převedená videa jak v příslušném bitstreamu tak i v YUV formátu. Dále se zde nacházejí jednotlivé konfigurační soubory pro referenční softwary. Struktura obsahu vypadá následovně:

```
\Videa\Zdroj  
\Videa\H264\Inter  
\Videa\H264\Intra  
\Videa\HEVC\Inter  
\Videa\HEVC\Intra  
\Videa\VP9\Inter  
\Videa\VP9\Intra  
\KonfigSoubory\H264  
\KonfigSoubory\HEVC  
\KonfigSoubory\VP9
```