

TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky a mezioborových inženýrských studií



BAKALÁŘSKÁ PRÁCE

Liberec 2007

Jakub Jelínek

TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky a mezioborových inženýrských studií

Studijní program: B2612 – Elektrotechnika a informatika
Studijní obor: 2612R011 – Elektronické informační a řídicí systémy

Přenos a komprimace digitálního videa

Digital video transfer and compression

Bakalářská práce

Autor: **Jakub Jelínek**
Vedoucí práce: Ing. Jiří Hnídek
Konzultant: Ing. Ondřej Raška, MITON CZ, s.r.o.

Rozsah práce: 60 stran textu
15 obrázků
8 tabulek
1 CD

V Liberci 18. 5. 2006

Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 o právu autorském, zejména § 60 (školní dílo).

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé bakalářské práce a prohlašuji, že s o u h l a s í m s případným užitím mé bakalářské práce (prodej, zapůjčení apod.).

Jsem si vědom toho, že užít své bakalářské práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem.

Datum: 18.5.2007

Podpis:

Jakub Jelínek

Chtěl bych poděkovat vedoucímu bakalářské práce Ing. Jiřímu Hnídkovi za poskytnutí cenných rad a poznatků při tvorbě této bakalářské práce

Liberec 2007

Autor Jakub Jelínek

Abstrakt

Tato bakalářská práce se zabývá metodami komprese a přenosem digitálního videa po internetu. Internet, stejně jako všechny ostatní obory a technologie úzce spjaté s výpočetní technikou, dosáhl za poslední desetiletí nebývalého rozmachu mezi širokou veřejností a obsahuje čím dál více multimediálních dat oproti počátku, kdy byl internet záležitostí čistě textovou. Objevují se nové trendy jako je Video over IP (IPTV), je snaha o vysílání a ukládání digitálního videa ve stále vyšším rozlišení (HDTV, HD DVD), avšak s tím také souvisí vzrůstající paměťové nároky a datové toky. Abychom tyto nároky udrželi v přijatelných mezích a mohli k těmto technologiím použít stávající přenosové trasy a digitální nosiče, je třeba využít efektivní komprese dat. Problematikou kompresí se zabývá především první část této práce. Pro potřeby digitálního videa se používá především ztrátová komprese, kde hlavní linie vývoje kompresních formátů začíná s ISO MPEG-1 a ITU H.261 aby nakonec vznikl společný standard H.264/AVC. Zdatně mu však konkurují i ostatní platformy a to především Microsoft se svým WMV9/10. Druhá část práce je věnována přenosu digitálního videa po internetu a to konkrétně streamování, nebo-li proudování, se zastoupením hlavní trojicí platform Windows Media, RealNetworks a QuickTime, jimž z celkového podílu na trhu stále více ubírá Flash. V poslední části této práce je uvedeno testování vybraných kodeků pomocí objektivních metod měření.

Abstract

This work deals with methods of compression and transmission of digital video through the internet. Internet, as well as all other branches and technologies adherent to computer technique, attained big boom between general during last ten years and there are much more multimedia data than before, when on internet were only text data. There are also some new trends like Video over IP alias IPTV, transmission and storage of digital video in better and better resolution (HDTV, HD DVD) and it relate to escalating memory requirement and data flow. If we want to maintain it in acceptable extent and could use existing transmission route and digital medium, it is necessary to use efficient data compression. About this problematic is the first part of this work. For digital video is mostly using loss compression The development guideline of compression format started with ISO MPEG-1 and ITU H.261, but finally they arose in one common standard H.264/AVC. It has big competition, the biggest from Microsoft with WMV9/10. The second part of this work is about transmission of digital video through the internet, concretely about streaming. There are three main platforms Windows Media, RealNetworks and QuickTime and their new competitor Flash. At the end there codecs testing via objective methods measurements.

Obsah

1 ÚVOD	7
2 ZÁKLADNÍ POJMY	8
2.1 VIDEO	8
2.2 ROZLIŠENÍ	8
2.3 BAREVNÁ HLOUBKA	10
2.4 PAMĚŤOVÉ NÁROKY	12
3 KOMPRIMACE	14
3.1 FORMÁT, KODEK	14
3.2 BEZETRÁTOVÉ KODEKY	15
3.2.1 HUFFYUV	15
3.2.2 LOGARITH	16
3.3 ZTRÁTOVÉ KODEKY	16
3.3.1 ZÁKLADNÍ PRINCIPY ZTRÁTOVÉ KOMPRIMACE	17
3.3.2 MJPEG	18
3.3.3 MPEG-1	19
3.3.4 MPEG-2	19
3.3.5 MPEG-4	20
3.3.6 MPEG-7	21
3.3.7 DIVX	22
3.3.8 XVID	23
3.3.9 VC-1	23
3.3.10 DV	23
4 PŘENOS VIDEO PO INTERNETU - STREAMOVÁNÍ	24
4.1 REŽIMY STREAMOVÁNÍ	25
4.1.1 PŘÍMÝ PŘENOS (ON-LINE)	25
4.1.2 ZÁZNÁMY NA VYŽÁDÁNÍ (VIDEO ON DEMAND)	25
4.1.3 UNICAS, MULTICAST	25
4.2 PŘENOSOVÉ PROTOKOLY	28
4.2.1 POUŽÍVANÉ TRANSPORTNÍ PROTOKOLY	28
4.2.2 POUŽÍVANÉ APLIKAČNÍ PROTOKOLY	29
4.3 NEJPOUŽÍVANĚJŠÍ STREAMOVACÍ PLATFORMY	31
4.3.1 WINDOWS MEDIA	32
4.3.2 REALNETWORKS	33
4.3.3 QUICKTIME	36
4.3.4 OGG THEORA	37
4.3.5 FLASH	38
5 KVALITA KODEKŮ	39
5.1 METODIKA MĚŘENÍ KVALITY KOMPRIMOVANÉHO VIDEO	39

5.2 VYBRANÉ OBJEKTIVNÍ METODY MĚŘENÍ	40
5.2.1 PSNR (PEAK SIGNAL-TO-NOISE RATIO)	40
5.2.2 BLURRING MEASURE (MÍRA ROZMAZÁNÍ)	41
5.2.3 VQM (VIDEO QUALITY EVALUATION)	41
5.3 VLASTNÍ MĚŘENÍ	42
5.4 VÝSLEDKY MĚŘENÍ	45
5.4.1 MRAVENCÍ	45
5.4.2 BÍLÝ ŠUM	48
5.4.3 HLAVY	50
5.4.4 GLADIÁTOR	54
5.4.5 KRTEK	54
6 STREAMOVÁNÍ V PRAXI	58
7 ZÁVĚR	58

Seznam tabulek

Tabulka 1: Parametry některých používaných počítačových formátů pro videoinformaci	9
Tabulka 2: Vybrané souborové systémy	13
Tabulka 3: Porovnání velikostí souborů komprimovaných pomocí kodeků MPEG	21
Tabulka 4: Nastavené a naměřené hodnoty při enkódování videosekvence <i>Mravenci</i>	45
Tabulka 5: Nastavené a naměřené hodnoty při enkódování videosekvence <i>Bílý šum</i>	48
Tabulka 6: Nastavené a naměřené hodnoty při enkódování videosekvence <i>Hlavy</i>	50
Tabulka 7: Nastavené a naměřené hodnoty při enkódování videosekvence <i>Gladiátor</i>	52
Tabulka 8: Nastavené a naměřené hodnoty při enkódování videosekvence <i>Krtek</i>	54

Seznam obrázků

Obrázek 1: Aditivní míchání základních barev a RGB model	10
Obrázek 2: Barevný model YCbCr	11
Obrázek 3: Huffmanovo kódování	16
Obrázek 4: GOP (Group Of Picture)	18
Obrázek 5: Unicast	26
Obrázek 6: Multicast	27
Obrázek 7: Metoda PSNR	40
Obrázek 8: Blurring measure	41
Obrázek 9: Video Quality Evaluation	41
Obrázek 10: MSU Video Quality Measurement Tool 1.5	43
Obrázek 11: Výsledné grafy pro videosekvenci <i>Mravenci</i>	47
Obrázek 12: Výsledné grafy pro videosekvenci <i>Bílý šum</i>	49
Obrázek 13: Výsledné grafy pro videosekvenci <i>Hlavy</i>	51
Obrázek 14: Výsledné grafy pro videosekvenci <i>Gladiátor</i>	53
Obrázek 15: Výsledné grafy pro videosekvenci <i>Krtek</i>	55

1 Úvod

Stěží by jsme v dnešní době hledali obor lidské činnosti ve kterém nenašel uplatnění počítač a výpočetní technika obecně. Počítače, mobilní telefony, DVD přehrávače, DVD recordery, digitální kamery a spousta dalších moderních elektrospotřebičů pracujících s digitálním videem se masově rozšířili do většiny domácností. Dnešní výpočetní technika umožňuje přehrávání, střih a další zpracování digitálního videa i naprostým laikům bez nutnosti vysokých investic. Navíc žijeme v době, kdy se z klasického analogového televizního vysílání přechází na digitální nebo kdy některé televize již nevyužívají dosud používaných přenosových tras, jako jsou terestrické či satelitní vysílání, ale s pomocí technologie nazývané IPTV se vrhly na internet.

Všechny tyto technologie a prostředky mají jeden společný obor zájmu a tím je problematika spojená s přenosem a komprimací digitálního videa, což je i tématem této bakalářské práce. Tato práce nemá za úkol detailně informovat o dané problematice, což vzhledem k obsáhlosti tohoto tématu není ani možné, ale má být jakým si přehledem používaných technologií a platforem, který může vzbudit zájem o další zkoumání a studium.

Účelem teoretické části této bakalářské práce, je obsáhnout problematiku spojenou s digitálním videem, počínaje základními pojmy, které jsou výchozí pro následné pochopení vlastních komprimačních metod využívaných v jednotlivých videokodecích (dále pouze kodek) a uvést přehled nejpoužívanějších kodeků. Dále osvětlit způsob přenosu digitálního videa po internetu, včetně používaných protokolů a platforem.

Cílem praktické části této práce, konkrétně kapitoly Kvalita kodeků, je využití teoretických poznatků a načerpání reálných zkušeností při zpracování digitálního videa vybranými kodeky a dále porovnat kvalitu těchto kodeků vzhledem k jejich výstupům. Posledním úkolem je vyzkoušet si práci se streamovaným obsahem.

2 Základní pojmy

2.1 Video

Pro začátek bychom si měli vysvětlit, co si pod pojmem video představit. V našem případě nepůjde o zařízení pro záznam či přehrávání televizních pořadů. Pojem video bude reprezentovat posloupnost obrázků (rámců, snímků), kde pro vjem spojitého pohybu je zapotřebí minimálně 15 takovýchto snímků jdoucích po sobě, nebo-li 15 fps (frame per sekund – rámců za sekundu). Ovšem v praxi se pro plynulé plně pohyblivé video (full-motion video) využívá snímkovací frekvence 24 fps a to u filmových pásů určených pro kina, v evropském televizním standardu PAL frekvence 25 fps a v americko-japonském standardu NTSC frekvence 29,97 fps. Nicméně nové technologie jako je například Showscan vytvářejí videosnímky o frekvenci 60 fps.

Dalšími důležitými parametry pro video, které dále přiblížím, kromě již uvedené snímkovací frekvence, jsou rozlišení a barevná hloubka a s tím související paměťové nároky.

2.2 Rozlišení

Dříve než-li se pustíme do popisu rozlišení, ujasněme si pojem **pixel**. Pixel z anglického picture element nebo-li obrazový prvek, je nejmenší zobrazitelný prvek v rastrové grafice, která je tvořena čtvercovou sítí a je tak možné (nutné) jednoznačně určit polohu každého pixelu. Například na monitoru je to nejmenší zobrazitelný bod a úzce tak souvisí s jeho rozlišením.

Rozlišení (rozlišovací schopnost) videa určuje jeho schopnost zobrazit jemné detaily. Vyšší rozlišení má požadavky na použití složitějších a obvykle dražších snímacích systémů a videosystémů.

Rozlišení digitálního videa v počítačových systémech je určeno počtem obrazových bodů (pixelů) daného snímku a udává se jako počet sloupců (horizontálně)

x počet řádků (vertikálně). Například formát VGA má rozlišení 640 sloupců na 480 řádků, tedy 640 x 480. V následující tabulce (Tabulka 1) jsou uvedeny nejznámější počítačové formáty z hlediska rozlišení obrazu.

Videoformát	Rozlišení (v pixlech)	Poměr stran
CGA – Color Graphics Adapter	320 x 200	16:10
EGA – Enhanced Graphic Adapter	640 x 350	5:3
VGA – Video Graphic Adapter	640 x 480	4:3
XGA- Extended Graphics Array a)	640 x 480	4:3
XGA- Extended Graphics Array b)	1024 x 768	4:3
SVGA – Super VGA	1024 x 768	4:3
WXGA - Wide XGA	1280x768	15:9
SXGA - Super XGA	1280x1024	5:4
SXGA+ - Super XGA+	1400x1050	4:3
WSXGA - Wide Super XGA	1600x1024	25:16
UXGA - Ultra XGA	1600x1200	4:3
WUXGA - Wide Ultra XGA	1920x1200	16:10

Tabulka 1: Parametry některých používaných počítačových formátů pro videoinformaci

V tabulce 1 je uveden také parametr poměr stran, který s rozlišením úzce souvisí, udává totiž poměr počtu sloupců (neboli počet pixelů horizontálně) ku počtu řádků (počet pixelů vertikálně). Pro formát VGA tedy 640/480 což je 4/3, někdy také vyjádřeno jako 1,33.

Pro úplnost uvedme, že v televizních systémech je rozlišení dáno obvykle počtem párů televizních řádků, které se zobrazí na obrazovce, vyjádřený počtem cyklů na výšku obrázku, resp. počtem cyklů na šířku obrázku. Například systém NTSC, charakterizovaný čísly 525/59,94, má přibližně 483 řádků. Systém televize s vysokou rozlišovací schopností HDTV používá přibližně dvojnásobný počet řádků ve srovnání s SDTV. Poměr stran u těchto systémů je pro SDTV 4/3 a pro HDTV 16/9.

SDTV (Standard-Definition TeleVision)

Současné standardní rozlišení televize (720 x 576 25 fps pro PAL, 720 x 480 30 fps pro NTSC).

HDTV (High-Definition TeleVision)

Nastupující rozšířené vyšší rozlišení televize (je definováno více typů, například 720p odpovídá 1280 x 720 nebo 1080p odpovídá 1920 x 1080)

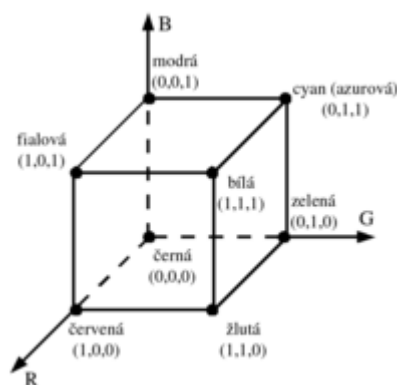
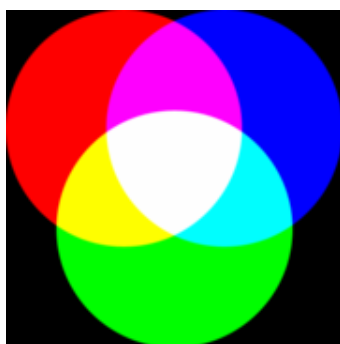
2.3 Barevná hloubka

Udává nám kolik bitů je potřeba k popisu barvy jednoho pixelu daného snímku videa nebo také jakou škálu barev máme k dispozici pro zobrazení našeho digitálního videa. Většinou se vyjadřuje jako počet bitů na pixel. Čím vyšší je tedy barevná hloubka tím více barev máme možnost zobrazit, musíme ovšem počítat i s vyššími paměťovými nároky.

Používané barevné hloubky:

- 1-bitová - $2^1 = 2$ barvy
- 4-bitová - $2^4 = 16$ barev
- 8-bitová - $2^8 = 256$ barev
- 16-bitová - $2^{16} = 65\,536$ barev (High Color)
- 24-bitová - $2^{24} = 16\,777\,216$ barev (True Color)

K popisu barev se využívá tzv. barevného modelu, který definuje základní barvy, ale také jejich míchání. V počítačích je nativním modelem RGB (Red – Green – Blue), kde jsou základní barvy červená – zelená – modrá. Ostatní barvy vznikají pomocí aditivního míchání těchto základních barev, jak je vidět na obrázku 1. RGB model si lze přestavit jako krychli, která má na osách x,y,z základní barvy o dané mohutnosti. Při použití nejběžnějšího barevného modelu RGB24, je tedy 24 bitů na jeden pixel, což znamená 8 bitů na každou ze základních barev. Jak je vidět na obrázku 1, tak například smíchání všech tří barevných složek o maximální mohutnosti nám dá barvu bílou. Můžeme se setkat také s barevným modelem RGBA, kde je navíc komponenta A, tzv. alfa kanál, který definuje průhlednost. V takovém případě se nejčastěji jedná o 32 bitovou barevnou hloubku.



Obrázek 1: Aditivní míchání základních barev a RGB model (zdroj: www.wikipedia.org)

V televizních standardech jako je PAL a NTCS se používá barevného modelu YUV, kde Y (luminance) je složkou jasovou, U a V (chrominance) jsou složky barvonosné. Někdy se také můžeme setkat s označením, že místo U a V je uvedeno B-Y a R-Y. Značení YUV se provádí pomocí tříčíselného zápisu, jako například u nejběžněji používaného způsobu převzorkování YUV 4:2:2, kde je uveden poměr zastoupení mezi barevnými složkami a jasovou složkou. Protože lidské oko je nejcitlivější na jas obrazu, má jasová složka oproti barevným ve většině případu minimálně dvojnásobnou hodnotu, jak je vidět i z předešlého zápisu. Digitálním představitelem YUV je barevný model YCbCr, zde je opět jasová složka Y a dvě barevné složky Cb a Cr, kde Cb představuje modrou a Cr červenou chrominační složku. Jak takový model vypadá v praxi, je vidět na obrázku 2, kde je vidět snímek jasové složky, barevných složek a složení všech snímků dohromady, tedy výsledný obraz.



Obrázek 2: Barevný model YCbCr - složky Y, Cb, Cr a výsledný obraz (zdroj: www.wikipedia.org)

Nespornou výhodou RGB modelu je jeho jednoduché zpracování, jako například sčítání, průměrkování či výpočet korekce barev, přechodových jevů, filtrace atd. Proto lze barevný model YCbCr převést na RGB podle následujícího vzorce, musíme mít však na paměti, že převod není zcela přesný a tudíž dochází k určitým ztrátám informace. Tyto nepřesnosti jsou způsobeny především tím, že dané barevné hodnoty nejsou mezi YUV (YCbCr) a RGB navzájem přesně převoditelné. Musí se tedy použít nejbližší hodnota daného barevného modelu, ale tím dochází k určitému barevnému posunu. Dalším problémem při převodu je rozdílná škála jasových hodnot, kde YUV může nabývat hodnot v rozmezí 16-235, kdežto RGB má rozsah 0-255. Barevný model YUV je zase vhodnější pro potřeby pozdější komprimace digitálního videa.

Vzorec pro převod barevného modelu YCbCr do RGB:

$$Y = 0.299R + 0.587G + 0.114B$$

$$Cb = -0.168736R - 0.331264G + 0.5B$$

$$Cr = 0.5R - 0.418688G - 0.081312B$$

Pro úplnost se ještě zmíním o další rozšířených barevných modelech. V tiskařské technice se používá CMY (azurová-purpurová-žlutá) či CMYK (azurová-purpurová-žlutá-černá), kde ostatní barvy vznikají subtraktivní mícháním barev. V počítačové grafice např. pro stínování reliéfů se využívá barevného modelu HSV s parametry Hue (barevný tón), Saturation (sytost) a Value (jas), někdy se tento model označuje jako HSB (Hue-Saturation-Brightness). Nedostatky HSV, jako jsou neplynulá změna barevného tónu a neplynulý přechod mezi bílou a černou (barevný tón se pohybuje po šestiúhelníku), odstraňuje barevný model HLS se složkami Hue (barevný tón), Lightness (světlost) a Saturation (sytost). Jedním z nejstarších barevných systémů je Munsellův systém, který vytvořil roku 1905 americký profesor Albert Henry Munsell.

2.4 Paměťové nároky

Jak je patrné z předchozích odstavců, tak rozlišení, snímkovací frekvence a barevná hloubka jsou parametry, které nám určují výsledné paměťové nároky na daný videosnímek.

Řekněme, že máme videozáznam ve videoformátu VGA, tedy o rozlišení 640 x 480 což nám dává 307200 obrazových bodů pro jeden rámeček. Obraz je snímán frekvencí 25 snímků za sekundu a je použit barevný model RGB24, což znamená, že pro každý pixel je třeba tři bajty. Tím se dostaneme k výpočtu 640 x 480 x 3 B, to se rovná necelý 1 MB dat pro jeden nezkomprimovaný snímek, vynásobíme-li to 25x, máme již 25 MB za jedinou sekundu videozáznamu. Hodina takového záznamu nám zabere už 3600 s x 25 MB, tedy 90 GB a to je i pro dnešní záznamová média stále dosti vysoká hodnota. Z tohoto výpočtu je tedy patrné, že komprimace digitálního videa je prakticky nevyhnutelná a bez ní by, zejména pro širokou veřejnost, nebylo zdaleka tak dostupné jako nyní.

Většina digitálních snímků obsahuje ve větší či menší míře nadbytečná data, alespoň z pohledu nedokonalosti lidského oka a jeho neschopnost rozpoznat některé informace. To znamená, že pokud použijeme efektivní metodu komprimace, můžeme dosáhnout značné redukce množství informací, potřebných k uložení a přenosu snímků. Přebytná data můžeme najít na úrovni jednotlivých obrazových prvků, řádků nebo rámců, pokud je scéna statická anebo se pohybují jen některé její části. K tomu se používají různé druhy kompresních algoritmů.

Za zmínku ještě stojí celková velikost videosouboru a tedy i souboru obecně. Tato velikost závisí na použitém diskovém souborovém systému, dnes jsou nejběžnější (pro uživatele MS Windows) dva a to FAT32 (File Allocation Table) a NTFS (New Technology File System). Pro FAT32 je omezení jednoho souboru do velikosti 4GB, což může být často problém, proto je pro práci s videem vhodnější využít souborového systému NTFS, který nám umožňuje vytvářet soubory až 16EB velké, tedy v dnešní době neomezené velikosti nebo přesněji jsme omezeni pouze kapacitou pevného disku. Nejčastěji používané souborové systémy v různých operačních systémech jsou uvedeny v následující tabulce 2.

Souborový Systém	Max. velikost souboru	Max. velikost disk. oddílu	Používán od roku	Původní OS
FAT16	2 Gib / 4GiB	16MiB až 4GiB	1983	MS-DOS
FAT32	2 Gib / 4GiB	2 TiB	1997	Windows 95
NTFS	16EB	16EB	1995	Windows NT
HFS+	8EB	8EB	1998	Mac OS
UFS2	512GB/32PB	1YB	2002	FreeBSD 5.0
ext3	16GB/2TB	2TB/32TB	1999	Linux
UDF	16EB	-	1995	-
ZFS	16EB	16EB	2004	Solaris 10

Tabulka 2: Vybrané souborové systémy

3 Komprimace

Nejdůležitějším parametrem pro komprimaci videa je bitový tok vzhledem k požadované vizuální kvalitě daného videa. Datový tok změříme velice snadno, velkým problémem je však “změření” vizuální kvality, neboť je to pojem notně subjektivní a je ovlivněn řadou činitelů, jako je plynulost videa, ostrost obrazu, viditelné komprimační bloky či věrnost barev. Proto se často musíme spolehnout na osobní názor konkrétního pozorovatele popřípadě skupinu pozorovatelů. Jediná objektivní obecně akceptovaná metoda je měření PSNR (Peak Signal-to-Noise Ratio) veličiny, kde je již z názvu zřejmé, že se zde zjišťuje úroveň percentuelně významného šumu původního obrazu videa vůči výstupnímu komprimovanému obrazu. Nevýhodou objektivních metod je především výpočetní a s tím související časová náročnost.

Před popisem vlastních metod komprese a používaných videokodeků, je vhodné si ujasnit rozdíl mezi pojmy obálkový formát (kontejner) a kodek, což často řada uživatelů chybně zaměňuje.

3.1 Formát, kodek

Pokud se člověk zabývá digitálním videem, ať už jako pouhý konzument hotových videosnímků nebo jako tvůrce snímků vlastních, tak se určitě setká se souborovými příponami AVI, MP4, WMV, MOV a další. Jsou to tzv. obálkové **formáty**, které definují metodu uložení video-dat na námi používané záznamové médium, ať už se jedná o pásy, optické či jiné disky a zapouzdřují tak jednotlivá multimediální data (včetně audio dat, titulků, animací, poznámek, tagů apod.) do jednoho souboru. Neurčují nám však jakým způsobem je dané video zkomprimováno, proto když máme například dva soubory s příponou AVI, tak se nám může stát, že jeden přehrajeme bez potíží a druhý přehrát nejde. AVI (Audio Video Interleave) je nejpoužívanějším kontejnerem, vyvinutým firmou Microsoft, jehož první verze byla uvedena roku 1992 a patří do skupiny Video for Windows.

Ke správnému přehrávání tedy nestačí pouze podpora obálkových formátů daným multimediálním přehrávačem, ale je třeba také mít na svém počítači nainstalovaný příslušný **kodek** (složení slov KOmpresor/DEKompresor nebo také

KOdér/DEKodér), který nám již zkomprimované video zpětně dekomprimuje abychom příslušné video mohli zhlédnout. Kodek je tedy nástroj, který obsahuje specifické algoritmy s jejichž pomocí provádíme vlastní komprimaci a dekomprimaci dat videa. Právě na použitém kodeku závisí míra komprese a kvalita uloženého videa. Některé kodeky, zejména mezinárodně standardizované, mají v řadě zařízení (kamery, stříhové karty, televizní karty) hardwarovou implementaci v podobě samostatných čipů, což má za následek nižší zatížení procesoru počítače při zpracování. Mezi nejpoužívanější kodeky patří MPEG-1 až 4, DivX, Xvid, DV a MotionJPEG (MJPEG).

Rozdělení kodeků

Kodeky lze rozdělit podle mnoha kritérií, mezi nejpodstatnější se řadí metoda kódování (ztrátové, bezztrátové), použití (offline, online, archivace, maximální komprese, profesionální použití atd.) a zabezpečení (ochrana autorských práv pomocí DRM ochrany a šifrování) kodeků.

3.2 Bezeztrátové kodeky

Komprimace pomocí bezztrátových kodeků zpravidla není tak efektivní (obvykle okolo 50 % původní velikosti) jako ztrátová komprimace dat, nicméně má tato metoda nespornou výhodu v možnosti zpětné rekonstrukce původních dat. Zástupcem bezztrátové komprimace pro obraz je především HuffYUV, dále pak jmenujme např. Lagarith. Bezeztrátové metody komprimace se jinak běžně využívá u archivačních formátů jako je RAR, ZIP, TAR apod.

3.2.1 HuffYUV

HuffYUV je bezztrátový kodek využívající ke komprimaci videa Huffmanovo kódování. Autorem je Ben Rudiak-Gould a je uvolněn pod GPL licenci. Pomocí tohoto kodeku je možné dosáhnout až 60% redukce původní velikosti souboru. Algoritmus se příliš neliší od bezztrátové komprese JPEG-LS v tom, že je odhadována hodnota následujícího pixelu a rozdíl (chyba) je zakódován pomocí Huffmanova kódování. Mezi jeho klady patří rychlost kódování, možnost komprimace v barevném modelu YUV i

RGB a v neposlední řadě jeho cena, neboť je zdarma. Záporům však je ukončení jeho vývoje roku 2002 a také jako u všech bezztrátových kodeků, nízká komprese.

Huffmanovo kódování podle svého objevitele David A. Hoffmana, pracuje na principu znalosti četnosti (popř. pravděpodobnosti) výskytu znaků v daném souboru. Znak s největší četností jsou kódovány nejkratším kódem a naopak nejméně časté pomocí nejdelšího kódu. Princip je názorně vidět na obrázku 3. Huffmanův kód se také často používá v kombinaci s dalšími kompresními algoritmy pro video, mezi něž patří např. MPEG.

$s_2(0,7)$		$s_2(0,7)$		$s_2(0,7)$	1	$s_1(0,08) \rightarrow 010$
$s_4(0,12)$		$s_{13}(0,18)$	1	$s_{134}(0,3)$	0	$s_2(0,7) \rightarrow 1$
$s_3(0,1)$	1	$s_4(0,12)$	0			$s_3(0,1) \rightarrow 011$
$s_1(0,08)$	0					$s_4(0,12) \rightarrow 00$

Obrázek 3: Huffmanovo kódování - s_1 až s_3 jsou 4 různé znaky, v závorce pak jejich četnost výskytu (zdroj: www.wikipedia.org)

3.2.2 Logarith

Bezeztrátový open source kodek, je pomalejší než-li HuffYUV, umožňuje vícevláknové zpracování na multiprocessorových systémech, podporuje mnoho barevných modelů a převod mezi nimi, každý snímek je klíčový což usnadňuje střih a editaci videa.

3.3 Ztrátové kodeky

Jak už je patrné z názvu, tak při ztrátové komprimaci dochází k nenávratné ztrátě některých dat, a tak je nelze jako při bezztrátové komprimaci zpětně rekonstruovat. Ztráta se týká především redundantních informací, ale dochází i k viditelným změnám obrazu, jako jsou rušivé kompresní artefakty, ztráta ostrosti obrazu, věrnosti barev atd. Tyto nežádoucí jevy jsou však kompenzovány velmi významnou redukcí výsledných souborů, poměr k původnímu nekomprimovanému souboru se pohybuje v rozmezí 1:4 až 1:100. Do této skupiny patří např. všechny verze MPEG a z nich vycházející DivX, XviD dále VC-1, RealVideo, Indeo® Video a další.

3.3.1 Základní principy ztrátové komprimace

Hlavním a společným principem pro všechny ztrátové kodeky je rozdělení rámce videa na **makrobloky**. Makroblokem rozumíme skupinu pixelů o konstantních rozměrech, např. 16 x 16 (pro MPEG-1 a MPEG-2), které se následně komprimují samostatně. Pro vyšší účinnost komprimace se u novějších kodků jako je např. MPEG-4, zavádí **adaptivní makrobloky**, které již nemají konstantní rozměry, ale jejich velikost se mění v závislosti na konkrétním snímku videa. S makrobloky souvisí pozdější problém při přehrávání, kde díky možnosti komprimace každého makrobloku jiným poměrem, dochází ke zobrazení rušivých viditelných bloků. Tomu se dá zabránit pomocí tzv. deblocking filtrů, které jsou obsaženy v dekodérech kodeků MPEG-4 a VC-1.

Dalším postupem při komprimaci je rozdělení videa (posloupnosti rámců) na **IBP rámce** (obrázek 4). Kde rámce I (Intraframes – klíčové snímky) jsou samostatně zobrazitelné, aniž bychom potřebovali informace z předchozích či následujících rámců. Podobně jako JPEG jsou kódovány pomocí DCT (Discrete Cosinus Transformation). Umožňují nám přímý přístup do sledu rámců videa a díky nejnížší kompresi tvoří největší podíl na výsledném souboru. Rámce P (Predicted – predikované) odhadují další vývoj rámce a konkrétní rámeček P je kódován vzhledem k jeho vztahu k předchozímu rámcu typu I nebo P. Různé kodeky se liší tím, že předcházející rámeček ke kterému se vztahujeme, nemusí být vždy fyzicky nejbližší předcházející. Kompresi P rámců je daleko vyšší než pro I rámce. Rámce B (Bidirectional, Between – interpolované snímky) jsou to obousměrné rámce, což znamená, že ke kódování dochází díky předchozímu a následujícímu rámcu a ty jsou buď typu I nebo P. Rámce typu B dosahují nejvyšší komprese ze všech typů rámců. U novějších kodeků se kódování IBP používá už na úrovni jednotlivých makrobloků, pak mluvíme o IBP blocích. Pro ještě účinnější komprimaci se využívá odhadu pohybu korespondujících makrobloků, úkolem je najít nejlépe vyhovující makroblok v dostupných referenčních rámcích a zakódovat je s pomocí vektorů pohybu. U nejnovějších metod komprimace se setkáváme ještě s tzv. řezy, které rámce rozdělí podle statických a dynamických zón v obraze a ve spojení s předchozími metodami dosahují dalšího zlepšení komprimace.



Obrázek 4: GOP (Group Of Picture) – Sekvence rámců od jednoho rámce I po další rámec I (zdroj: www.wikipedia.org)

Jak už bylo řečeno v předcházejícím odstavci, tak jednotlivé makrobloky se komprimují samostatně. K omezení prostorové redundance a tím tedy k vlastní komprimaci se využívá **DCT** (Discrete Cosinus Transformation – diskrétní kosinová transformace) či **DWT** (Discrete Wavelets Transformation – diskrétní vlnková transformace). Převědeme tedy makrobloky na koeficienty dané transformace, které se následně kvantizují. Výsledný proud koeficientů se ještě dále bezztrátově zkomprimuje a to buď pomocí Huffmannova kódování nebo přímo pro proud DCT koeficientů určeným CABAC (Context Adaptive Binary Arithmetic Coding).

Z výčtu těchto postupů je použito v různých obměnách v drtivé většině kodeků využívající ztrátové komprese, nicméně má každý kodek své specifické vlastnosti a další metody, které dále zvyšují kompresní poměr a vizuální kvalitu obrazu.

3.3.2 MJPEG

Motion JPEG, jak už je z názvu patrné, funguje na principu komprimace jednotlivých snímků metodou použitou pro JPEG obrázky. Jde vlastně o sled JPEG obrázků a díky tomu máme možnost nastavit kompresní poměr v rozmezí od 6:1 do 16:1. Má relativně dobrý poměr kvalita obrazu vs. velikost souboru a je také velice často implementován do hardwaru (střihové karty, webkamery, digitální fotoaparáty atd...). Nespornou výhodou tohoto kodeku je, že každý snímek je klíčový, což je velice vhodné pro případný střih takového videosnímku. Problémem občas bývá přehrávání a zpracování videa využívající MJPEG z důvodu rozdílných obměn hardwarové implementace tohoto kodeku u různých výrobců.

3.3.3 MPEG-1

Všechny verze MPEG jsou vyvíjeny v organizaci stejného názvu tedy MPEG (Motion Picture Expert Group), která zaštiťuje skupinu expertů z vědecké i komerční komunity a ty vyvíjejí průmyslové standardy pro zpracování videa.

MPEG-1 byl uvolněn roku 1993 a byl vyvinut s ohledem na plně pohyblivé video. Jeho cílem bylo vytvořit kodek, který by kvalitou odpovídal standardu VHS. Je orientován na komprimaci videa s rozlišením 320 x 240 pixelů s datovým tokem 1 až 1,5 Mbit/s, se snímkovací frekvencí 25 fps (pro PAL) a 30 fps (pro NTSC). Obsahuje i algoritmy pro komprimaci zvukových dat s faktorem komprimace 5:1 až 10:1. Tento standard je také zahrnut do tzv. White Book, čímž se stal normou pro VideoCD. Díky mezinárodní standardizaci organizací ISO se stal široce podporovaným a implementovaným v řadě hardwaru i softwaru. Přes všechny jeho klady je však už dosti zastaralý a nesplňuje požadavky dnešní doby, ať už pro použití na internetu a jiných datových sítích nebo pro televizní účely. Navíc také není příliš vhodný pro střih videa a podporuje pouze konstantní datový tok.

Ve stejné době jako MPEG-1 vznikl také kvalitativně podobný standard ITU H.261, který byl určen především pro telekomunikační potřeby. ITU (International Telecommunication Union) je mezinárodní sdružení, zabývající se vývojem a standardizací telekomunikačních přenosových systémů.

3.3.4 MPEG-2

Roku 1994 spatřil světlo světa MPEG-2, který vznik pro potřeby digitálního vysílání. Prosadil se nejen jako standard pro pozemní a satelitní digitální vysílání DVB (Digital Video Broadcast), ale také pro SVCD, DVD. Později se ukázalo, že jeho koncept vyhovuje i v tu dobu teprve plánovanému HDTV.

Jde tedy o kvalitativně vyšší koncepci než-li MPEG-1, ale zachovává si s ním zpětnou kompatibilitu. Oproti MPEG-1 má řadu vylepšení, za všechny jmenujme proměnlivý datový tok (VBR), který umožňuje při dynamických scénách datový tok

zvýšit a při statických naopak snížit, čehož lze docílit pomocí dvouprůchodového kódování. Při prvním průchodu dochází k analýze daného videa a k vytvoření souhrnu informací, podle kterých se při druhém průchodu určuje datový tok výsledného zkomprimovaného videa. Dalším vylepšením je podpora prokládaného snímkování tzv. půlsnímky, to je výhoda nejen pro samotné digitální vysílání, ale například pro kompenzaci pohybu makrobloků. MPEG-2 také podporuje rozdělení digitálního videa do více datových toků a tak je možné poskytovat video v různých kvalitativních proudech.

Pokud porovnáme MPEG-2 a MPEG-1 při vyšších rozlišeních a stejných datových tocích, tak MPEG-2 vykazuje vyšší kvalitu obrazu. V nízkých rozlišeních není tento rozdíl již tak patrný. MPEG-2 je také hardwarově náročnější oproti svému předchůdci. Stejně jako MPEG-1 není příliš vhodný pro střih digitálního videa. Dnes je MPEG-2 jedním z nekompatibilnějších formátů a hojně zastoupen jak v hardwarové implementaci, tak v podpoře softwaru.

Ani ITU nezaspala a odpovědí na MPEG-2 je přenosový standard H.262, který je posléze zahrnut do MPEG-2 co by transportní formát. Pro úplnost je třeba zmínit logického nástupce MPEG-2, tedy MPEG-3, který měl být vyvinut výhradně pro potřeby HDTV. Jak už bylo ale zmíněno, tak MPEG-2 pro tento účel postačoval, tudíž se MPEG-3 nerealizoval a vzniká přímo MPEG-4.

3.3.5 MPEG-4

S rozvojem internetu a mobilních sítí, bylo nutné řešit problematiku přenosu digitálního videa po takto úzkých přenosových trasách. Pro tyto účely vznikl MPEG-4, jeho síla se však ukázala i při komprimaci při vyšších datových tocích. MPEG-4 a kodeky z něj vycházející jsou dnes nejpoužívanějšími kodeky v domácnostech, ale ani přesto nedosáhl takového komerčního úspěchu jako jeho předchůdci.

S terminologií ohledně MPEG-4 je to trochu složitější. Po vypuštění první verze se zjistilo, že je příliš náročný, obsahuje řadu chyb a nedořešených otázek. Byla tedy nutná revize, což vedlo k vydání MPEG-4 Part 2, ale i tak se projevila jeho přílišná

komplikovanost. Nakonec je tedy uvedena finální verze MPEG-4 Part 10, označována také jako H.264/AVC, která vznikla ve spolupráci s ITU a jejich H.264 (H.26L). MPEG-4 je samozřejmě dál vyvíjen, v současné době je již MPEG-4 Part 17, tyto verze už nejsou ale tak zásadní.

Došlo zde opět k dalšímu kvalitativnímu posunu v kompresi oproti MPEG-2, kde mezi nejvýznamnější vylepšení patří použití adaptivních makrobloků, jejichž velikost může nabývat rozměrů mezi 16 x 16 a 4 x 4 pixely. Dále zavedení řezů a využití principu IBP kódování i pro makrobloky v rámci jednoho snímku. Pro zakódování videa do IBP rámců je zde možné využít reference až 5 snímků a ne pouze předešlého či předešlého a následujícího jako je tomu u jeho předchůdců. Komprimace proudu koeficientů DCT je provedena pomocí CABAC, oproti Huffmanovu kódování v MPEG-2.

Své uplatnění si MPEG-4 našel především v archivaci videa na offline nosiče (CD, DVD), zálohování filmů z DVD, ale i pro šíření videa po internetu a mobilních sítích, nikoliv však v reálném čase. Největší slabinou MPEG-4 je jeho hardwarová náročnost, která je cenou za dosahované vysoké kompresní poměry digitálního videa. Pro názornost v tabulce 3 uvádím porovnání dosahovaných výsledku komprese kodeků z rodiny MPEG.

MPEG-1	MPEG-2	MPEG-4 Part 10
100%	90%	35%

Tabulka 3: Orientační porovnání velikostí souborů komprimovaných pomocí kodeků MPEG. Referenčním souborem je soubor komprimovaný pomocí MPEG-1.

3.3.6 MPEG-7

Už z označení je jasné, že nejde o následníka MPEG-4. Standard MPEG-7 se už nezabývá dalším zvýšením kompresního poměru, ale způsobem jak rychle a efektivně vyhledávat v dnešní záplavě digitálního videa na sítích či offline nosičích.

3.3.7 DivX

Počátky tohoto kodeku jsou sice nelegální, ale přesto nebo možná právě proto se rozšířil mezi laickou veřejnost do domácností, jako se to nepovedlo žádnému kodeku před ním. Jeho první verze označovaná jako DivX;-) 3.0, je dílem Jerome Rota, který nelegálně využil zdrojových kódů kodeku MPEG-4 V3 od Microsoftu. Svojí oblibu si získal především kvalitní komprimací a to zcela zdarma. Díky hrozícím soudním sporům, vzniká v projektu Mayo kodek DivX 4.0, který je již zcela legální, ale nedosahuje takových výsledků jako verze předchozí. To vedlo k vývoji další verze DivX 5, ta odstraňovala nedostatky kodeku DivX 4. DivX 5 je rychlejší, kompatibilní s předchozími kodeky, dosahuje lepších komprimačních poměrů oproti DivX 4, ale na druhou stranu plnohodnotná verze již není zcela zdarma a od verze DivX 5 se stává kodekem uzavřeným. Nejnovější verzí je dnes DivX 6.

Jak už bylo řečeno, tak DivX vychází z MPEG-4 a proto jeho algoritmus funguje na velice podobném principu. Využívá také IBP rámce, kde vzdálenost mezi rámci I je možné volit prakticky libovolně s ohledem na výslednou kvalitu výstupu kodéru. I zde se pracuje s makrobloky o rozměrech 16 x 16 či 8 x 8 pixelů a výsledný proud koeficientů diskrétní kosinové transformace se dále komprimuje Huffmanovým kódem. Pro optimalizaci výsledného datového toku vzhledem k vizuální kvalitě, máme na výběr jednorůchodovou, dvouřůchodovou nebo víceřůchodovou komprimaci, kde při více než jednom průchodu dosáhneme lepších výsledků, musíme však počítat s větší časovou náročností. Při přehrávání dochází mimo dekodování také k post-processingu, který se dá popsat jako aplikování filtrů v jasové a barevné části signálu pro zvýšení výsledné vizuální kvality.

Tento kodek se uchytil především jako nástroj pro archivaci filmů z DVD a oblibu si našel i pro ukládání domácího videa. Horší už to je s uplatněním DivX v průmyslu, kde byl především díky svému původu dlouhou dobu výrobcí opomíjen. Jeho klady a zápory jsou prakticky totožné jako u MPEG-4, tedy vysoká vizuální kvalita videa při nízkém datovém toku na úkor vysokých hardwarových nároků.

3.3.8 XviD

V momentě kdy se stal původně otevřený DivX, kodekem uzavřeným a zpoplatněným, vzala skupina programátorů pracujících na DivX zdrojové kódy ještě otevřeného DivXu, aby posléze vytvořila kodek vlastní a nazvala ho XviD. Jak je vidět tak i název vznikl z DivX a to otočením jeho názvu. Jedná se tedy o další open source verzi kodeku vycházejícího z MPEG-4. Z počátku byl XviD v porovnání s DivX složitý a vyžadoval poměrně dobré znalosti v této problematice, při použití jeho kodéru. V dnešní době je však už na srovnatelné úrovni, dosahuje podobných výsledků jako DivX a je také jeho největším konkurentem. Z toho také vyplývá, že uplatnění XviD je obdobné jako je tomu u DivX.

3.3.9 VC-1

I Microsoft si uvědomil významu multimédií a představil jakousi alternativu k MPEG-4 a jeho odnožím. Odborníci, kteří na VC-1 pracovali se z části rekrutovali z týmu pracujícího pro ITU a tak se není co divit, že je principiálně velmi podobný MPEG-4 Part 10. Pravděpodobně nejpodstatnějším rozdílem oproti MPEG-4 je použití DWT (diskrétní vlnková transformace), která dosahuje lepší vizuální kvality při vyšším kompresním poměru a je známa už z JPEG 2000. Dalšími zajímavými odlišnostmi oproti jeho konkurentovi jsou například použití výhradně celočíselných operací, což se pozitivně projevuje na zatížení procesoru, má schopnost extrapolace snímků a obsahuje DRM (systém pro ochranu autorských práv). Celkově jde tedy o perspektivní kodek s velmi kvalitní kompresí a nižším zatížením hardwaru oproti MPEG-4. Určitým negativem je jeho uzavřenost a provázanost s produkty Microsoft. Microsoft se však netají tím, že má ctižádost proniknout do všech segmentů multimediálního průmyslu.

3.3.10 DV

Velice rozšířený kodek ve spotřební elektronice, jehož komprese je postavena především na diskrétní kosinové transformaci. Má pevně stanoven kompresní poměr na 5:1 a datový tok na 25 Mbps. Díky tomu, že výstupní video s použitím tohoto kodeku je kvalitativně velice podobné bezztrátovým kompresím a každý snímek je klíčový, tak se používá pro domácí střih a editaci digitálního videa.

4 Přenos videa po internetu - streamování

V dnešní době již dostatečně výkonných počítačů pro zpracování videa, i pro běžné domácnosti a dostupnost vysokorychlostního internetu pro většinu populace vyspělých států, včetně České republiky, jsme na internetu zaplavováni čím dál větším množstvím audio a video signálů. Ať už se jedná o komerční placené přenosy různých televizních stanic, neplacené televizní programy, servery kde vystavují svoje „video-dílka“ amatérští filmaři či domácí videa pro pobavení, je třeba využít efektivní technologii, jak takové množství dat přenášet.

V předchozích kapitolách jsme se bavili o efektivních komprimacích digitálního videa, které dosahují výborných kompresních poměrů, ale i přesto jsou tato data stále dosti objemná pro přenos po dnešních přenosových trasách, přece jen se bavíme u hodinového záznamu o datech několika stovek MB. Jen malé procento obyvatel má možnost připojení v řádu desítek či stovek Mb/s, kde je možné celé takovéto video přenést za pár minut. K takovému účelu je pak nejjednodušší využití některého způsobu pro přenos běžných dat, jako je FTP nebo různé P2P sítě. Pro dnes „běžné“ vysokorychlostní připojení v řádu stovek kb/s nebo jednotek Mb/s je ale takový přenos dosti zdlouhavý a nepohodlný. Navíc si dnes čím dál větší okruh zájmu získává tzv. IPTV neboli Video over IP, kde ani z principu není možné stažení celého videa najednou například u přímých přenosů nebo přenosů různých přenášek a konferencí. Pro tyto účely je nejvhodnější využití tzv. proudování nebo-li streamování.

Streamování

Streamováním se rozumí přenos dat videa směrem k uživateli, který má o data zájem a to tím způsobem, že se proud dat načítá po částech, kde se nejprve načte malá část videa do mezipaměti (bufferu), po naplnění mezipaměti již dochází k přehrání příslušného videa. Avšak i při přehrávání se neustále stahují další segmenty daného videosnímku, a tak dochází k plynulému přehrávání celého videa. Mohou se však vyskytnout komplikace v připojení, což by vedlo k postupnému vyprazdňování bufferu a pokud by nedošlo k nápravě, tak by se přehrávání přerušilo. Nutnou podmínkou je tedy požadavek na relativně stabilní propustnost sítě. Nicméně je tento způsob přenosu velice populární, zejména díky téměř okamžitému přehrávání, bez nutnosti mít celý

soubor uložený na disku a možnosti realizace přímých přenosů at' už sportovních, kulturních či různých konferencí.

Oproti běžnému uložení videa přímo u klienta, si musí streamovací technologie poradit s různou datovou propustností k jednotlivým klientům, měnícími se síťovými podmínkami a minimalizovat rozptyl zpoždění dat tzv. jitter.

4.1 Režimy streamování

4.1.1 Přímý přenos (on-line)

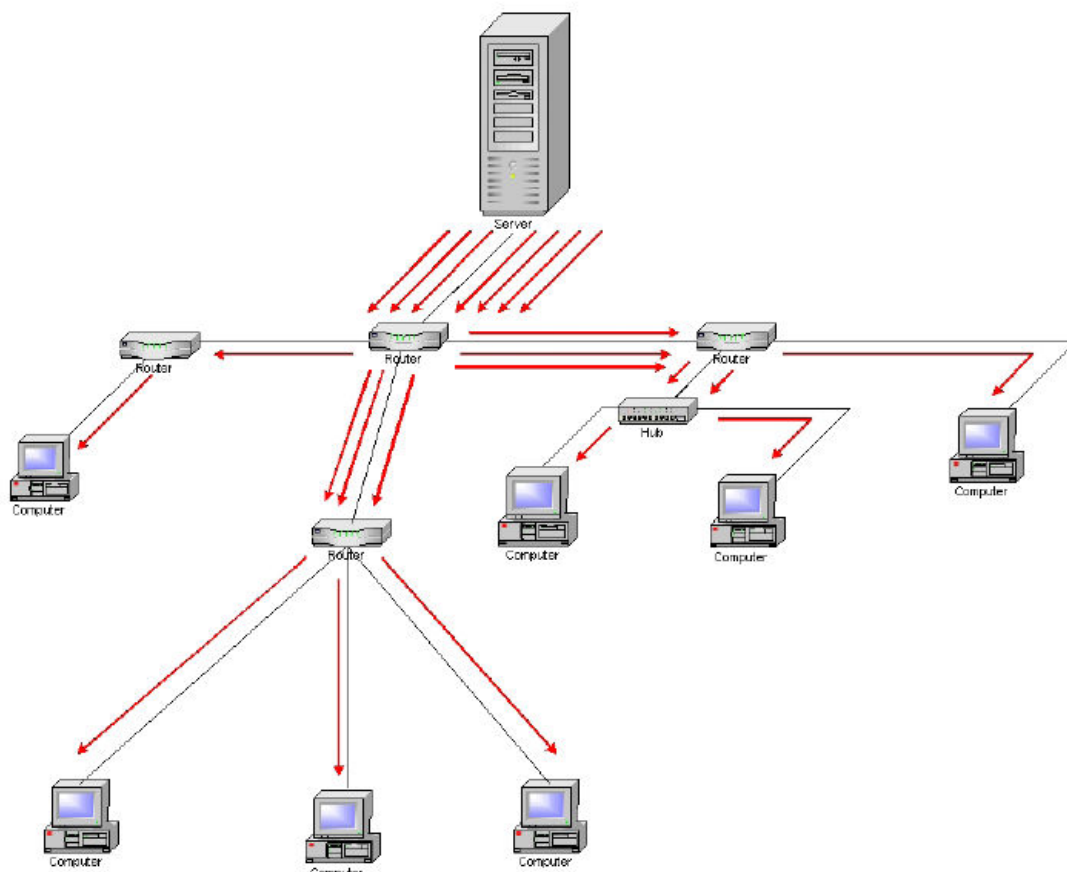
Přímý přenos digitálního videa znamená, že je video v reálném čase vytvářeno a pomocí vysílacího pracoviště (serveru) okamžitě přenášeno k jednotlivým klientům. Všichni účastníci takového přenosu, pak dostávají obsahově shodný signál, který se může lišit audiovizuální kvalitou, závislou na datových tocích. S ohledem na použité technologie dochází k určitému zpoždění které je obvykle v rozmezí 1 sekunda až půl minuty a vzhledem k datové náročnosti takového přenosu je třeba myslet na dostatečnou konektivitu vysílacího pracoviště.

4.1.2 Záznamy, na vyžádání (VOD – Video on Demand)

Při tomto způsobu proudování je na klientovi co a kdy si přeje přehrát. Videosnímek, videosnímky jsou uloženy na vysílacím serveru z kterých si může daný uživatel vybrat. Pokud bychom přirovnali on-line streaming ke klasickému televiznímu vysílání, pak VOD by se dalo přirovnat k videopůjčovně. Ve většině případů je také možné v takovém videu přeskakovat a volně se v něm pohybovat.

4.1.3 Unicas, Multicast

Unicast – přenos postaven na způsobu komunikace klient/server, dochází zde k přenosu dat vždy pouze mezi dvěma hosty. Velkou nevýhodou tohoto způsobu přenosu je vysoká zátěž vysílacího uzlu a přenosové soustavy při přenosu typu jeden zdroj – mnoho příjemců (Obrázek 5). Přenos paketu od zdroje k cíli, je zde iniciován zdrojem.

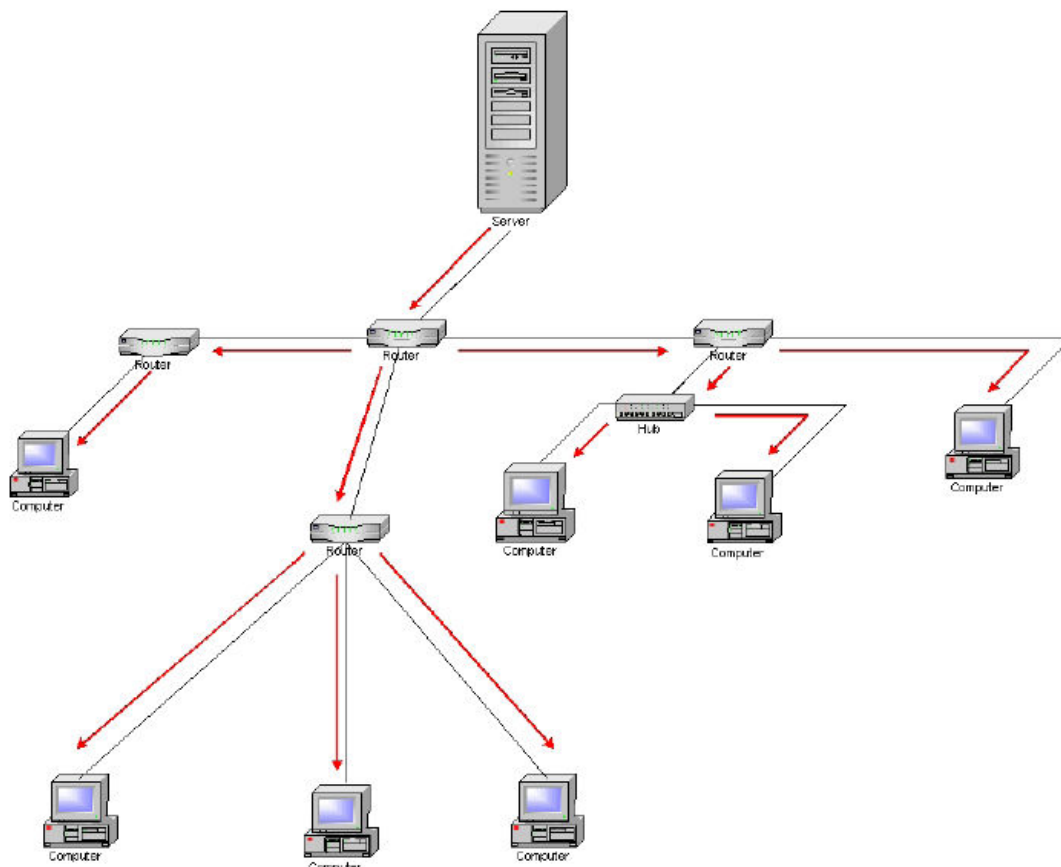


Obrázek 5: Unicast (zdroj: www.muni.cz)

Multicast - Data adresovaná skupinovou cílovou adresou jsou dopravena všem členům skupiny. Zdrojový host vysílá pouze v jednom datovém proudu a datový tok se dělí až u skupiny příjemců, za pomoci routerů (směrovačů). Routery vytvářejí optimální strom cest po kterých se šíří multimediální data, jak je vidět na obrázku 6. Směrovače zodpovídají za to, že pokud je požadavek od klienta o příjem signálu, aby mu byly data doručena a to nejvýše jedenkrát po každém spoji. Při skupinovém vysílání je zde tok paketů určován příjemci. Nejpodstatnějším argumentem pro využití této technologie pro přenos multimediálních dat je značné snížení zátěže celé přenosové soustavy a to v případě přenosu typu jeden zdroj – mnoho příjemců. Je postavený na protokolu UDP, nad TCP nemá smysl, protože TCP vytváří spojení mezi dvěma konkrétními uzly.

Tam kde není možné použití multicastu, když například nepodporují multicast všechny směrovače v dané síti, tak je možné využití přenosu za pomoci tzv. **zrcadla**. Zrcadlem se většinou rozumí software, který přijímá multimediální streamy od

jednotlivých klientů a přeposílá je ostatním připojeným klientům. Vytváří překryvovou síť, která emuluje multicast v síti, kde se multicast nešíří. Neřeší problém redundance multimediálních streamů na jednotlivých linkách.



Obrázek 6: Multicast (zdroj: www.muni.cz)

Porovnání Unicast a Multicast přenosu

Unicast	Multicast
jeden stream k jednomu uživateli	jeden stream do určitých „větví“ sítě
VOD, VTR funkce	není VOD služba, bez VTR funkcí
použití běžných směrovačů	vyžaduje sofistikované směrovače
vyšší zatížení přenosové soustavy	vyšší efektivita využití přenosového pásma

4.2 Přenosové protokoly

Pro streamování se používá řada protokolů, mezi zástupce v přenosové vrstvě jmenujme TCP a UDP, v praxi se však více využívá protokolu UDP. V aplikační vrstvě se používají protokoly RTP a z něho vycházející a doplňující protokoly RTCP a RTSP, dále pak ještě proprietární MMS.

4.2.1 Používané transportní protokoly

TCP (Transmission Control Protocol)

- Stavový protokol na transportní vrstvě ISO/OSI modelu

Výhody pro multimediální přenosy

- Bezchybný přenos
- Kontrola zahlcení linky
- Pakety vždy dorazí ve správném pořadí
- Férový protokol
- Retransmise ztracených paketů

Nevýhody pro multimediální přenosy

- Férovost nedovoluje dostatečnou šířku pásma na vytížených linkách
- Bezchybnost přenosu je na úkor nízké latence

UDP (User Datagram Protocol)

- Bezstavový protokol na transportní vrstvě ISO/OSI modelu
- Multimediální aplikace využívají v drtivé většině případů protokol UDP pro přenos dat (až na speciální případy)

Výhody pro multimediální přenosy

- V porovnání s TCP minimalistický, efektivnější a rychlejší

- Odpadá režie s ověřováním, že každý paket dorazil v pořádku (není nutné posílat ACK - potvrzovací znak při přenosu dat.)
- UDP prakticky nezvyšuje latenci při přenosu multimediálních dat

Nevýhody pro multimediální přenosy

- Nespolehlivý protokol
- Pakety se mohou ztratit bez jakéhokoliv upozornění
- Pakety mohou přicházet mimo původní pořadí

Vzájemné porovnání přenosových protokolů TCP a UDP

TCP	UDP
nutné zavedení spojení před přenosem dat	bez nutnosti navázání spojení
obousměrná komunikace	jednosměrná komunikace, bez interaktivity
robustní protokol – retransmise	nerobustní protokol
protichybové zabezpečení FEC	detekce chyb
řízení datového toku	bez řízení datového toku

4.2.2 Používané aplikační protokoly

RTP (Real-Time Transport Protocol)

Vyvíjen pro potřeby šíření digitálního audia a videa na internetu, původně měl být pouze jako multicast, ale s ohledem na internet je implementován do řady unicast aplikací.

- Postavený nad protokolem UDP
- Sekvenční číslování paketu
- Identifikace obsahu
- Časové značky pro jednotlivé pakety
- Protokol sám od sebe nezaručuje kvalitu přenosu, pouze poskytuje prostředky pro zaručení kvality aplikacím

RTCP (RTP Control Protocol)

- Real Time Control Protocol doplňuje protokol RTP
- Poskytuje out-of-band informace pro řízení proudu dat přenášeného pomocí RTP
- RTCP poskytuje aplikaci zpětnou vazbu na kvalitu přenosu pomocí protokolu RTP

RTSP (Real-time Streaming Protocol)

Další protokol pro streamování médií vycházející z RTP. Uživatel má naprostou kontrolu nad daným médiem, což mu umožňuje přehrávání přetáčet, pozastavit apod. Je implementován ve většině multimediálních přehrávačů nebo například do Real Server a Helix Server

- Stavový protokol založený na HTTP požadavcích (GET apod.)
- Ovládání streamovacího serveru (VCR příkazy jako Play, Pause a Stop) a přístup k souborům podle času
- Pro přenos dat se používá protokol RTP + RTCP případně jeho proprietární obdoba RDT
- RTSP je protokol řídící průběh streamování
- Neurčuje ani formát audia nebo videa, ani způsob jeho přenosu a zpracování
- Protokol typu out-of-band - udržuje vlastní dvojici portů

MMS (Microsoft Media Services)

Protokol pro streamování médií vyvinutý Microsoftem, který je samozřejmě implementován v jeho produktech, jako je například Microsoft Windows Media Server. Je to unicast protokol s plnou kontrolou médií, který je ve spojení s formátem Windows Media hojně využíván.

- Proprietární protokol pro streamování
- Pro přenos dat se používají protokoly UDP nebo i TCP pokud se nezdaří vyjednat spojení na protokolu UDP

- Jako poslední z možností je „streamování pomocí upraveného protokolu HTTP (tedy opět nad protokolem TCP)

4.3 Nejpoužívanější streamovací platformy

Jednotlivé platformy pro streamování videa se obvykle skládají ze třech komponent jako je enkodér, server a klient.

Enkodér je vždy na začátku přenosového řetězce, kde má za úkol převést videosignál do komprimované podoby vhodné pro další přenos po síti. Vstupním videosignálem může být analogový signál, který musíme ještě digitalizovat či digitální data. Výstupem enkodéru je pak komprimované video, které je při on-line režimu přímo posíláno serveru a při režimu Video on Demand je ukládáno na disk. Po technické stránce může mít enkodér buď hardwarovou podobu a to jako samostatné zařízení či karta do PC nebo může být realizován softwarově.

Server přesněji videosever, při on-line režimu přijímá data od enkodéru, při Video on Demand si načte již předem připravená data z disku a ty pak dále zprostředkovává jednotlivým klientům. Přenos dat z videoseveru se od klasického stažení souboru liší tím, že je tento přenos řízen s ohledem na kvalitu sítě a potřeby klienta. Takovým případem je multibitrate vysílání (vysílání na více rychlostech) zodpovídá tak za vybrání patřičného proudu pro danou šířku pásma. Ve většině případů je funkce serveru a enkodéru oddělena a to z důvodu vysokých nároků na konektivitu serveru. Z toho důvodu se videosever zpravidla umísťuje na páteřní síť. V případě menších lokálních sítí nebo při multicastu je někdy možné funkce serveru a enkodéru spojit. Videosevery jsou realizovány za pomoci specializovaného softwaru.

Klientem (přehrávač) se ve většině případu rozumí software, který na straně uživatele (diváka) přijímá data od videoseveru a zodpovídá za jejich správné zobrazení na výstupu (obrazovce monitoru, televize). Stejně jako u enkodéru však může být i implementace hardwarová.

Ze zástupců streamingových platforem jsou v současnosti nejrozšířenější: Windows Media, RealNetworks, QuickTime, za open source jmenujme například Ogg Theora.

4.3.1 Windows Media

Přestože je Windows Media poměrně mladou platformou, tak je v našich končinách jednou z nejrozšířenějších. Vděčí za to především tomu, že klient je prakticky na každém počítači s Windows. Windows Media je součástí libovolného microsoftího serverového operačního systému počínaje Windows NT a konče Windows Server 2003, management vysílacího serveru je plně integrován do správy Windows. Pakliže tedy máme některý z uvedených systémů, tak můžeme tuto platformu využívat zdarma. Pokud bychom však chtěli enkodér a server platformy Windows Media využívat na jiném OS, tak máme bohužel smůlu. Klienta je možné provozovat samozřejmě na platformách Windows, ale i na Mac, Solaris, s jistými omezeními i na Linuxu například pomocí mplayeru.

Platforma Windows Media obsahuje:

- Windows Media Encoder - enkodér pro vytváření obsahu
- Windows Media Services - server pro distribuci (streaming) multimediálního obsahu
- Windows Media Player - přehrávač multimedií
- Windows Media Audio and Video - kodeky pro zpracování zvuku a videa
- Windows Media DRM - pro ochranu elektronického obsahu proti kopírování
- Windows Media Software Development Kit - vývojový nástroj pro softwarové vývojáře

Tato platforma bere zdroj dat ze subsystému DirectShow, což znamená, že jakékoliv zařízení, které disponuje kompatibilními ovladači, může sloužit za zdroj signálu. Jako vstupní video formáty je možné využít MPEG1, AVI, ASF, WMV. Kontejner se pak používá buďto WMV pro soubory obsahující zvuk i video komprimované pomocí kodeků Windows Media Audio a Windows Media Video nebo ASF pro obsah komprimovaný pomocí jiných kodeků. Windows Media také obsahuje systém správy

digitálních práv (DRM) nabízí tak poskytovatelům obsahu a prodejcům flexibilní formát pro zabezpečenou distribuci obsahu digitálních médií, dále tato platforma podporuje multibitrate vysílání, kdy je možné posílat na server v jednom balíku více videostreamů v různé kvalitě, přičemž klient a server dohodnou kvalitu přenášeného videa v závislosti na dostupné přenosové kapacitě. Server Windows Media Services 9 používá pro vysílání jak proprietární protokol MMS, tak otevřený protokol RTSP což jsou protokoly nad UDP, je však možné použít i HTTP nad TCP. Je třeba mít na paměti, že na straně klienta Windows Media Playeru je podpora RTSP až ve verzi 9 a vyšší. Ovládání serveru je možné provádět přes <http://www> rozhraní.

Mezi zajímavé vlastnosti poslední verze Windows Media jmenujme funkci rychlého spuštění, jakmile se divák připojí k datovému proudu, jsou po dobu několika prvních sekund data odesílána s využitím maximální dostupné šířky pásma, aby přehrávání mohlo být zahájeno co nejdříve. Přehrávání během archivace, lze archivované soubory zpřístupnit pro požadavky na vyžádání nebo opakované vysílání ještě před dokončením archivace. Modifikátory adres URL pro selhání kodérů tzn. že Windows Media Services lze nakonfigurovat tak, aby v případě, že dojde k selhání nebo zastavení primárního kodéru, po uplynutí určité doby vyžadovala obsah od alternativního kodéru nebo jiného zdroje obsahu, a to pomocí modifikátorů adres URL v cestě k primárnímu kodéru. Služba Windows Media Services je k dispozici jako samostatná součást ve verzích operačního systému Microsoft Windows Server 2003 pro procesory x64 což má za následek zvýšení škálovatelnosti. Za zmínku také určitě stojí to, že je zde podpora pro klienty na platformě protokolu IPv6.

4.3.2 RealNetworks

RealNetworks a její platforma je historiky nejstarší a průkopnická v oblasti mediálního průmyslu v prostředí internetu. Vždyť už v roce 1995, kdy internet nabízel ve valné většině pouze textové informace, vytvořili v RealNetworks dá se říci revoluční řešení pro přenos obrazu a zvuku – nástroje RealPlayer a RealAudio. Dnes portfolio jejích produktů sahá od vlastního formátu a příslušného přehrávače, přes vysokokapacitní videoservery až po speciální aplikace jako DRM řešení a streaming v mobilních sítích 2,5. a 3. generace.

Platforma RealNetworks zahrnuje:

- RealProducer - enkodér pro vytváření obsahu
- Helix Server - server pro distribuci (streaming) multimediálního obsahu
- RealPlayer - přehrávač multimedií
- RealAudio, RealVideo - kodeky pro zpracování zvuku a videa
- Helix Security Manager - zabezpečení multimediálních souborů
- RealSystem Development Kit (SDK) - vývojový nástroj pro softwarové vývojáře

Enkodér tedy RealProducer je možné provozovat na platformách Windows a Linux, Helix Server navíc na UNIXu, konkrétně Solaris a RealPlayer i na Macu. Jak server, tak enkodér jsou sice k dispozici zdarma, ale ve velmi omezené verzi, plnou je nutné zakoupit. Real Networks poskytují i open-source verzi serveru a kodéru v projektu s názvem Helix Community.

Co se týče použitelných vstupních video formátů pro enkodér, tak je opravdu z čeho vybírat: AVI, MOV, MPEG1, MPEG2, MPEG4 a QT. Jako výstupní formát je pak použito RM což je proprietární formát dat od RealNetworks. Tato platforma zvládá stejně jako Windows Media multibitrate vysílání a navíc je zde možné simulovat libovolnou šířku pásma mezi klientem a serverem. Což nám umožňuje optimalizovat kvalitu obrazu a zvuku dostupné šířce pásma. Helix Server je jediný videosever s univerzální podporou pro on-line a VOD přenos hlavních formátů jako jsou RealAudio, RealVideo, Windows Media, QuickTime, MP3, AAC a AAC+. Pro přenos dat se zde ve většině případů využívá protokolu RTSP, ale aplikovat se dá i přenos nad TCP přes HTTP. I RealNetworks si uvědomil, že bude nutné přejít na IPv6 a tak tento protokol implementoval do nejnovější verze svojí platformy, zachoval však samozřejmě i podporu IPv4. Navíc je zde použit i protokol SNMP (Simple Network Management Protocol), který nám mimo autentizaci podává důležité informace o stavu přenosu a přenosové síti a tak tento přenos můžeme optimalizovat podle daných potřeb a možností sítě a klientů. Tato platforma používá jak metodu enkodérem vynuceného přenosu datového proudu na videosever, tak i naopak videosever se připojí na definovaný enkodér a začne z něj data stahovat. Tento způsob je užitečný jednak pro šetření objemu přenesených dat v momentech, kdy žádný divák stream nesleduje (pak

server automaticky do další žádosti diváka dočasně přeruší spojení), jednak je vhodný v podmínkách, kdy může docházet na přenosové trase mezi enkodérem a videoseverem k výpadkům spojení (pak se server automaticky po každém výpadku sám opět připojí k enkodéru a započne stahování dat pro následnou distribuci divákům). Užitečným prvkem je také možnost použití příkazové řádky pro dávkové zpracování úloh při nižším výkonovém zatížení procesu počítače.

Největší devízou systému RealNetworks je však spojení kvalitních audio a video kodeků s enkodérem, serverem a klientem, které jsou vskutku multiplatformní a tak je můžeme realizovat na prakticky libovolném operačním systému.

Porovnání Windows Media a RealNetworks

Tyto dvě platformy jsou v našich končinách nejhojněji využívány a tak zde uvedu malé srovnání jejich vlastností a kvalit. Hned na začátek musím uvést, že obě dávají při použití analogických nástrojů velmi podobné výsledky. Co se týče rozšířenosti, tak má určitou výhodu na své straně Windows Media právě z důvodu těsného spojení s operačním systémem Windows a také s toho vyplývající cena, tedy nulová (pokud tedy již máme OS Windows zakoupený) a v neposlední řadě lokalizace. Na druhou stranu je toto spojení dosti podstatnou nevýhodou pro Windows Media, z hlediska multiplatformního vede RealNetworks a to jak z pohledu OS, tak i z podpory vstupních formátů.

Jak Windows Media, tak RealNetworks v nejnovějších verzích mají velice podobné možnosti nastavení kódování a samotného proudování s ohledem na to, že každý využívá vlastní proprietární kodeky. Obě platformy mají například prostředky pro minimalizaci efektu zvaného jitter, umožňují mutibitrate vysílání a to s prakticky "libovolnými" datovými toky, obě mají své systémy ochrany autorských práv, podporují muticast a tak by jsme mohli pokračovat dál. Nicméně ve starších verzích bylo u WM například problémem když měl v cestě NAT nebo firewall, k přenosu dat nad UDP bylo možné využít pouze uzavřeného proprietárního protokolu MMS, ale tyto problémy byly vyřešeny s příchodem nejnovější verze Windows Media. Ovládání obou serverů je možné činit přes www rozhraní, RealNetworks má konfigurační soubory v textovém souboru, zatímco Microsoft je uchovává v registrech operačního systému.

4.3.3 QuickTime

S platformou QuickTime přišla společnost Apple, která tento produkt vyvíjí již zhruba 16 let a tak není divu, že se za ty roky dočkala řady změn. Přestože je v této oblasti velká konkurence, tak si QuickTime stále drží pozici jedné nejrozšířenějších platform na poli streamovaných multimedií. QuickTime je de facto průmyslovým standardem ve světě multimediálních technologií a tak asi nikoho nepřekvapí, že se jím inspirovala řada standardů, za všechny jmenujme H.264/AVC.

Platforma QuickTime zahrnuje:

- QuickTime Player – pro přehrávání zvukových nebo obrazových záznamů
- QuickTime Pro – pro všestrannou tvorbu médií
- Zásuvné moduly prohlížečů – pro zobrazování médií v rámci webovských prohlížečů
- Picture Viewer – pro práci s obrázky
- QuickTime Streaming Server – pro vysílání médií po Internetu v reálném čase
- Darwin Streaming Server – pro vysílání médií na platformách Linux, Solaris a Windows
- QuickTime Broadcaster – pro vysílání živých přenosů po Internetu
- Komponent MPEG-2 – slouží k přehrávání MPEG-2 obsahu

Mohlo by se zdát, že produkt od společnosti Apple nebude příliš multiplatformní, opak je ale pravdou. I když byl QuickTime Streaming Server vyvíjen pro Mac OS X, tak Apple poskytl zdrojové kódy a pod názvem Darwin Streaming Server je vyvíjena open source verze, kterou je možné provozovat na platformách Windows, Linux, a Solaris. Celou tuto platformu je až na enkodér a komponent pro přehrávání MPEG-2 možné využívat zcela zdarma, což je také potěšující zjištění.

QuickTime podporuje více jak 50 vstupních formátů a jako výstup pro streamování lze zvolit QuickTime Movie (.MOV), MPEG-4 (.MP4), 3GPP (.3GP). Jak je vidět, tak je možné pracovat i s mezinárodně standardizovaným formátem MPEG-4, což znamená, že obsah je schopný přehrávat libovolný jiný MPEG-4 přehrávač, a naopak, QuickTime dokáže přehrát MPEG-4 z libovolného jiného zdroje. QuickTime je také

patentovaný formát, který je logicky rozdělen na stopy, které mohou obsahovat nejen audio a video, ale i textové informace, interaktivní animace, grafiku, html odkazy atd. Tato platforma stejně jako předešlé dvě umožňuje multibitrate vysílání, unicast i multicast. Pro přenos dat je zde využito opět protokolu RTP a RTSP. Administraci serveru je mimo jiné možné realizovat i pomocí webového rozhraní. QuickTime má také prostředky zaručující “quality of service” a tím jsou Skip Protection a Instant-On. Kde Skip Protection je ochrana proti přeskokování či přerušování přehrávání streamovaných multimedií a to tím, že využívá nadbytečnou šířku pásma přenosové linky k přenosu vyššího množství dat než je potřeba pro přehrávání. A pokud dojde k přerušení nebo dočasné ztrátě spojení se serverem, tak datový tok přehrává z místní vyrovnávací paměti.

Chceme-li steamovat mezinárodní standard jako je MPEG-4, popřípadě MOV, tak je platforma QuickTime bezesporu zajímavá volba.

4.3.4 Ogg Theora

Theora vlastně není ani platformou, ale kompletně svobodným (open source) videokodekem, který byl vyvinut ve společném projektu ON2 Technologies z New Yorku a neziskovou organizací Xiph.org Foundation. Da se říci, že je protějškem pro video oproti zvukovému kodeku Ogg Vorbis, který je svobodným kodekem konkurujícím MP3. Tato technologie by se měla stát konkurencí pro MPEG-4 a řešením od RealNetworks či Microsoftu. Výsledky tohoto kodeku a to především jeho první verze však zatím nedosahují kvalitativních výsledků oproti výše uvedeným platformám a jejich kodekům. Theora je díky svému open-source původu dostupná pro většinu OS, není ale na nich předinstalována a je nutné ji doinstalovat. Důvod proč jsem tento kodek zařadil do výčtu streamingových platforem je ten, že se v současné době čím dál častěji pro streamování využívá, zejména na akademické půdě a v linuxové komunitě. Pro streamování Theora se nejčastěji využívá řešení pomocí Helixu nebo Fluemotion vydaný pod GPL licencí. Fluemotion je streamingový open source videosever vyvíjený ve spolupráci se společností Fluendo, která se zabývá problematikou multimedií na platformách Unix a Linux.

4.3.5 Flash

V současné době velice oblíbené řešení řady projektů, včetně nejznámějšího videoseveru YouTube. Původní Flash (formát SWF) od Macromedia podporoval video pouze do šestnácti tisíc rámců, což bylo zoufale málo, a tak byl vyvinut nový kodek VP6. Ten si ve spojení s Flash Media Server získává čím dál větší zastoupení na trhu. Výhodou této platformy je dostupnost prakticky pro jakýkoliv OS, kde jsou v řadě z nich flash přehrávače již předinstalovány, a tak jsou velice rozšířené. Kvalitativně se kodek VP6 blíží konkurentům jako jsou WMV9 či H.264. Specifikem flashe je, že přehrávače si musí každý distributor naprogramovat sám, popřípadě zakoupit již vytvořené od jiných distributorů či uživatelů. Odměnou za toto nestandardní řešení je, že si každý může vytvořit přehrávač dle vlastních představ. Komerční varianta od Adobe, která Macromedia odkoupila, stojí nemalé peníze, a tak je tu open source řešení Red5 založeného na Javě, které je samozřejmě zdarma.

5.1 Metodika měření kvality komprimovaného videa

Jak už jsem psal v kapitole o komprimaci videa, tak nejdůležitějším parametrem kvality výsledného komprimovaného digitálního videa je datový tok vzhledem k vizuální kvalitě obrazu. Nutno podotknout, že problematika týkající se porovnání kvality obrazu je značně složitá a stále se vyvíjí spolehlivější srovnávací metody. Vizuální kvalita obrazu je také velice subjektivním pojmem, a tak pro někoho kvalitní obraz může někomu jinému připadat méně kvalitní. Nicméně jsou určité parametry na které je každý člověk citlivý a jejich změnu vnímá více než-li jiné. Mezi takové parametry lze zařadit jas, barevné podání, ostrost obrazu, viditelný šum, viditelné komprimační bloky, plynulost pohybu a další. Jednotlivé kodeky byly a jsou vyvíjeny pro různé účely a tak některé podávají dobré výsledky už při nízkých datových tocích, zatím co jiné svoje přednosti ukazují až při tocích vyšších.

Pro srovnání kvality kodeků se tak používá dvou základních principů a to subjektivní a objektivní měření kvality videa. Při subjektivním měření se vychází ze subjektivního vnímání daného obrazu videa jednotlivými pozorovateli, kteří video hodnotí podle různých metodik. Pro toto měření je tedy zapotřebí dostatečného počtu pozorovatelů (doporučeno alespoň 15 osob), výběr vhodných sekvencí a vhodné metodiky pro posouzení. V principu jsou všechny subjektivní metody založeny na tom, že se jednotlivým pozorovatelům pouští referenční (výchozí) videosekvence a videosekvence komprimovaná daným kodekem a pozorovatel pak již podle dané metodiky hodnotí například zhoršení kvality, zlepšení kvality nebo uděluje body podle dané stupnice. Některé subjektivní metodiky jsou i standardizované v ITU-R BT.500-11, mezi ně patří DSIS (Double Stimulus Impairment Scale), DSCQS (Double Stimulus Continuous Quality Scale), SCACJ (Stimulus Comparison Adjectival Categorical Judgement), SAMVIQ (Subjective Assessment Method for Video Quality evaluation), MSUCQE (MSU Continuous Quality Evaluation).

Objektivní měření kvality obrazu se provádí za pomoci matematických modelů, kde se každá konkrétní metoda zaměřuje na jiné parametry porovnávaného videa.

Principiálně je tu opět zdrojové (výchozí) video, které se pomocí dané metody porovná s videem komprimovaným. Nejuznávanější a mezinárodně akceptovanou objektivní metodou je PSNR (Peak Signal-to-Noise Ratio). Mezi dalšími jmenujme například MSAD, Delta, Blurring measure, Blocking measure, SSIM INDEX, VQM, MSE. Největší nevýhodou objektivní metody měření je její časová a hardwarová náročnost.

5.2 Vybrané objektivní metody měření

Pro praktické porovnání kvality kodeků jsem si vybral objektivní měření kvality videa, a to konkrétně pomocí metody PSNR, Blurring measure a VQM, proto bych uvedl jejich stručný popis.

5.2.1 PSNR (Peak Signal-to-Noise Ratio)



Obrázek 7: Metoda PSNR

Tato metoda vychází z MSE (Mean Square Error), kde je určen šum jednoho oproti druhému monochromatickému snímku velikosti $m \times n$. Výpočet se provádí podle vzorce:

$$d(X, Y) = \frac{\sum_{i=1, j=1}^{m, n} (X_{i,j} - Y_{i,j})^2}{mn}$$

V PSNR se počítá odstup maximálního možného výkonu signálu od výkonu šumu v decibelech, za předpokladu, že jeden pixel může nabývat 255 hodnot při osmibitovém vzorkování. PSNR se pak počítá podle vzorce:

$$d(X, Y) = 10 \log_{10} \frac{(2^B - 1)^2}{MSE} \Rightarrow 10 \log_{10} \frac{255^2 nm}{\sum_{i=1, j=1}^{n, m} (X_{i,j} - Y_{i,j})^2}$$

Vyšší PSNR nemusí vždy značit, lepší subjektivní obrazovou kvalitu. Obvyklou hodnotou PSNR kvalitního komprimovaného videa je mezi 30 až 40 dB. Názorná ukázka jak funguje metoda PSNR je vidět na obrázku 7.

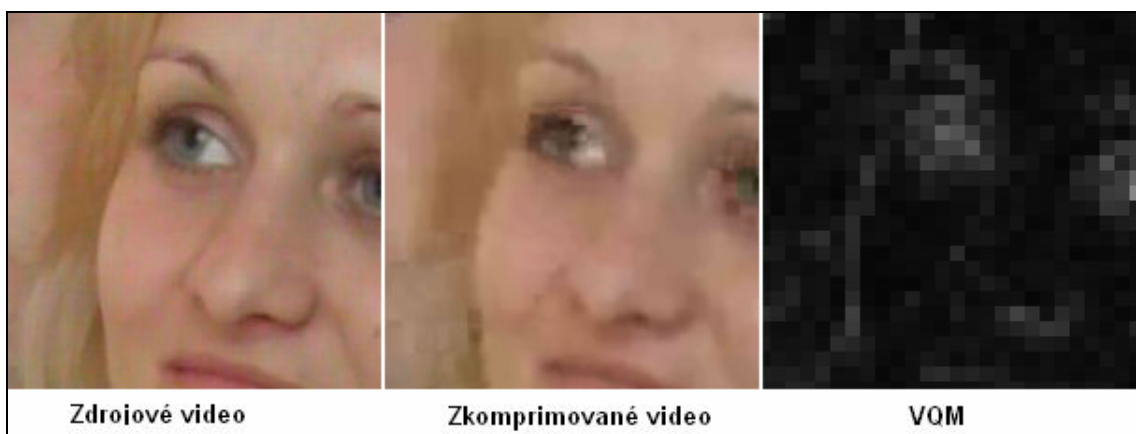
5.2.2 Blurring measure (míra rozmazání)



Obrázek 8: Blurring measure

Udává míru rozmazání jednoho snímku vůči druhému. Pokud je hodnota blurring prvního obrazu vyšší než hodnota druhého, znamená to, že je druhý obraz více rozmazán než první. Tato metoda použitá v praxi je názorně zobrazena na obrázku 8.

5.2.3 VQM (Video Quality Evaluation)



Obrázek 9: Video Quality Evaluation

Porovnává originální video vůči komprimovanému pomocí DCT, která lépe odpovídá lidskému vnímání obrazu, a tak je spolehlivějším ukazatelem subjektivní vizuální kvality obrazu nežli například MSE. Čím více se hodnota VQM komprimovaného videa blíží nule, tím toto video více odpovídá originálu a je tudíž kvalitnější obraz. VQM při konkrétním použití je vidět na obrázku 9.

Tato metoda probíhá v pěti bodech. Nejprve se provede transformace barevného prostoru do YUV, následuje DCT transformace, převod DCT koeficientů na tzv. local contrast (LC) podle vzorce:

$$LC(i,j) = DCT(i,j) * \text{Power}(DC/1024, 0.65) / DC$$

DCT hodnota 1024 se používá pro 8-bitový obraz. Parametr 0,65 je nejlepším parametrem vhodným pro psychofyzikální data.

Následně se aplikují tzv. SCSF (spatial contrast sensitivity function) matice zvlášť na statické a dynamické rámce. Posledním krokem je zjištění středního a maximálního zkreslení a výsledné VQM dle vzorců:

$$\text{Mean_Dist} = 1000 * \text{mean}(\text{mean}(\text{abs}(\text{diff})))$$

$$\text{Max_dist} = 1000 * \text{maximum}(\text{maximum}(\text{abs}(\text{diff})))$$

$$\text{VQM} = (\text{Mean_dist} + 0.005 * \text{Max_dist})$$

Maximální zkreslení je zatíženo parametrem 0,005, zjištěného z jednoduchých psychofyzikálních experimentů. Parametr 1000 je standardizovaný poměr.

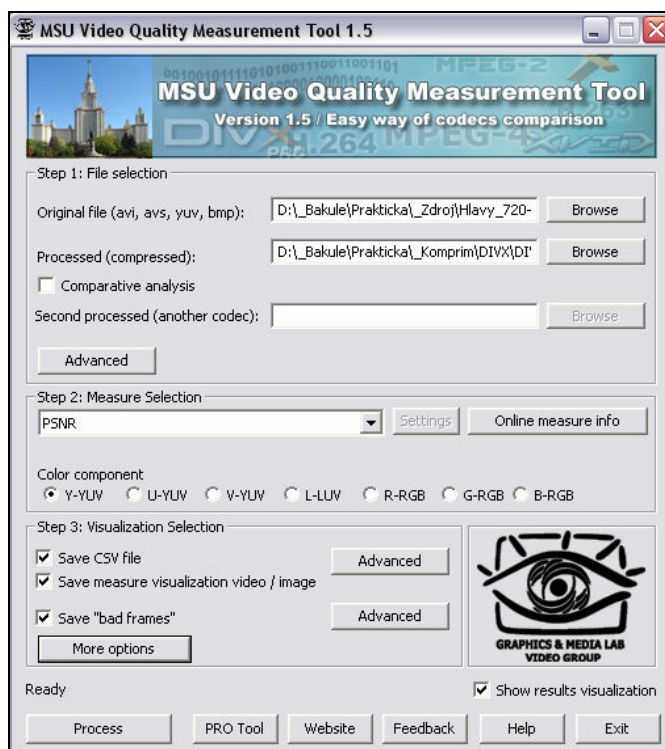
5.3 Vlastní měření

Metody měření: Objektivní měření PSNR, Blurring measure, VQM

Konfigurace PC: Mobile Intel Pentium M 740, 1733 MHz
512 MB DDR2 SDRAM
Samsung 60 GB, 5400 RPM, SATA
GeForce Go 6600, 128 DDR
OS Windows XP SP2

Měřicí software: MSU Video Quality Measurement Tool 1.5 [1] (Obrázek 10)

Je to software vyvinutý Moskevskou státní univerzitou, který umožňuje porovnání kvality obrazu pomocí většiny dosud známých objektivních metod.



Obrázek 10: MSU Video Quality Measurement Tool 1.5

Použité kodeky: DivX 6.6.1

XviD MPEG-4 Codec 1.1.3

Microsoft MPEG4 Video Codec V3

Použité zdrojové videosekvence:

Všechny použité digitální videosekvence obsahují pouze obrazové informace, jsou tedy bez audiostopy.

Mravenci

- rozlišení 720 x 576, snímkovací frekvence 25 fps, datový tok 28800kbps, délka 30 s
- natočeno digitální videokamerou Sony DCR TRV-255E, komprimace DV
- videosekvence se vyznačuje velkým počtem změn malých objektů, reprezentovaných pohybem mravenců v mraveništi. Kamera se v celém úseku této sekvence pomalu pohybuje nad celým mravenišťem.

Bílý šum

- rozlišení 720 x 576, snímkovací frekvence 15 fps, datový tok 149299kbps, délka 30 s
- nekomprimované AVI
- videosekvence vytvořená v Matlabu následujícím algoritmem[\[2\]](#):

```
clear all
mov = avifile('Bily_sum.avi')
mov.compression='none'; %zadna komprese
mov.quality=100;
for i=1:900

    I = ones(576,720,3); %velikost obrazu
    J(:, :, 1) = imnoise(I(:, :, 1), 'gaussian');
    J(:, :, 2) = imnoise(I(:, :, 2), 'gaussian');
    J(:, :, 3) = imnoise(I(:, :, 3), 'gaussian');
    F=im2frame(J);
    mov=addframe(mov,F);
end;
mov=close(mov);
```

Hlavy

- rozlišení 720 x 576, snímkovací frekvence 25 fps, datový tok 28800kbps, délka 30 s
- natočeno digitální videokamerou Sony DCR TRV-255E, komprimace DV
- scéna obsahuje detail hlav dvou osob s občasným mírným pohybem, jinak je zde statické pozadí a kamera je po celou délku videosekvence v klidu.

Gladiátor

- rozlišení 720 x 576, snímkovací frekvence 25 fps, datový tok 248832kbps, délka 30 s
- nekomprimované AVI
- ukázka z filmu Gladiátor, která se vyznačuje rychlými pohyby, častou změnou záběru a množstvím plamenů

Krtek

- rozlišení 720 x 576, snímkovací frekvence 25 fps, datový tok 248832kbps, délka 30 s
- nekomprimované AVI
- ukázka z animované pohádky Krtek a autíčko, tzn. větší jednobarevné plochy, méně barev a většinou statické pozadí

Použitá nastavení a pravidla:

- U všech z testovaných kodeků je nastavena hodnota klíčových snímků na jednotnou hodnotu každých 300 snímků, jednorůchodové kódování a příslušný datový tok. Jinak je ponecháno defaultní nastavení.
- každý kodek je testován pro čtyři datové toky pro všechny zdrojové videosekvece: 128 kbps, 512 kbps, 1024 kbps, 2048 kbps
- Kromě výsledků zjištěných použitými objektivními metodami, jsem zaznamenával i skutečný datový tok, dobu komprimace a velikost jednotlivých výsledných videosouborů.

5.4 Výsledky měření

5.4.1 Mravenci

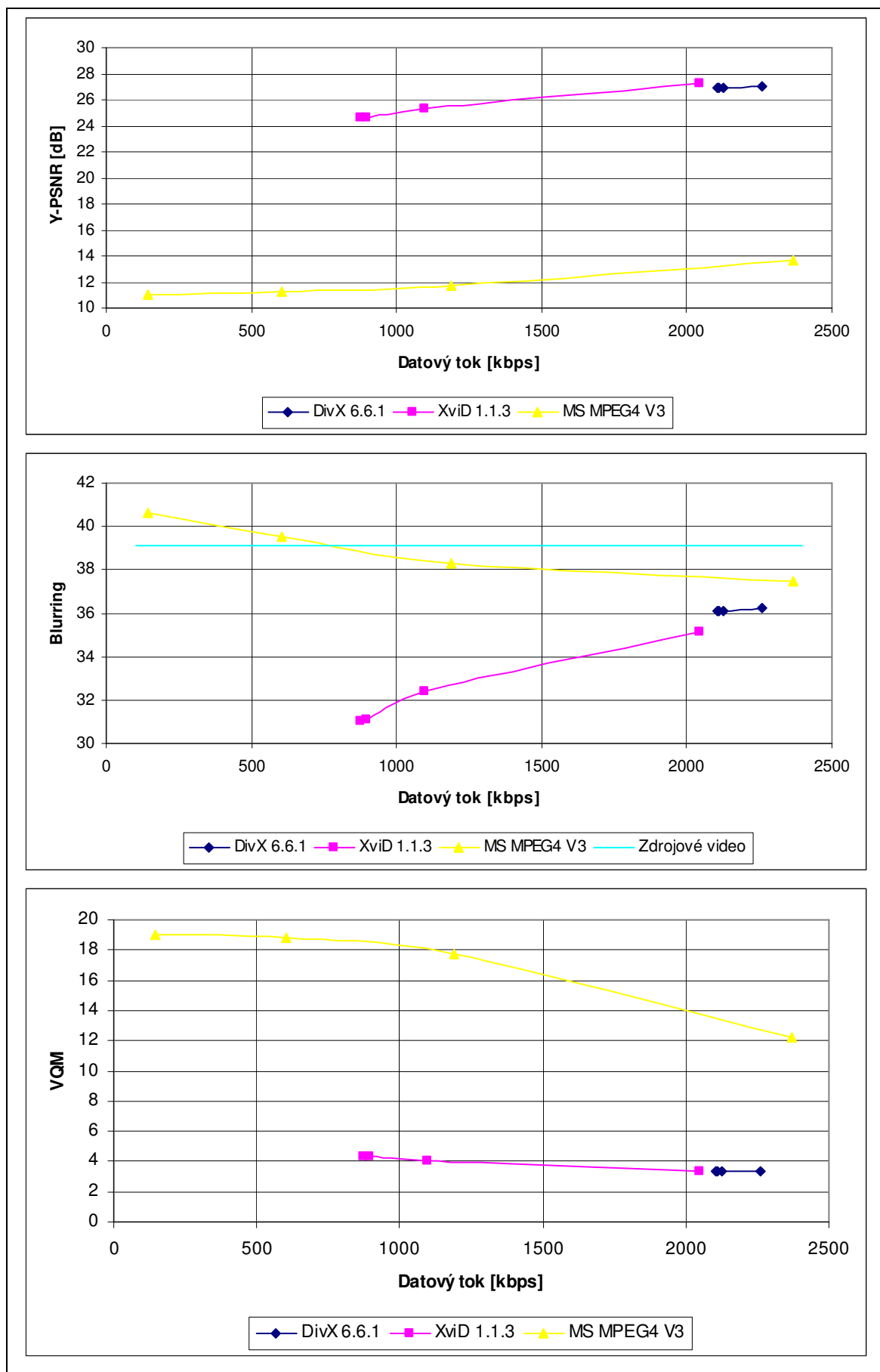
Zhodnocení:

Nastavený DT [kbps]	Skutečný DT [kbps]	Doba enkódování [s]	Velikost souboru [kB]	PSNR (AVG) [dB]	Blurring (zdroj: 39,13057)	VQM
DivX 6.6.1						
128	2105	55	7937	26,94229	36,12872	3,37295
512	2110	55	7954	26,94691	36,13217	3,36938
1024	2124	55	8009	26,95258	36,13588	3,36643
2048	2260	55	8519	27,05531	36,25756	3,32223
XviD MPEG4 Codec 1.1.3						
128	875	53	3306	24,66621	31,03748	4,34280
512	896	53	3384	24,67486	31,07772	4,33914
1024	1095	54	4145	25,36977	32,43039	4,02462
2048	2046	58	7713	27,20495	35,13425	3,33579
Microsoft MPEG4 Video Codec V3						
128	144	99	579	11,08261	40,59665	18,98396
512	604	33	2306	11,27304	39,52585	18,85260
1024	1188	27	4499	11,66640	38,28078	17,73097
2048	2368	24	8929	13,64377	37,48525	12,18582

Tabulka 4: Nastavené a naměřené hodnoty při enkódování

Z výsledků naměřených pro videoukázku *Mravenci* (Tabulka 4) je patrné, že s hodnotou nastaveného datového toku (dále pouze DT) 128 kbps si nejlépe poradil kodek MS MPEG4 Video Codec V3 (dále pouze MPEG4), kde hodnota skutečného DT činila 144 kbps, čehož tento kodek docílil vynecháním snímků, a tak je výsledné video dosti trhavé. MPEG4 také v průměru nejrychleji enkódoval zdrojové video, u ostatních dvou kodeků byla doba enkódování téměř shodná. Naopak kodek DivX 6.6.1 (dále pouze DivX) nebyl schopen skutečný DT videosekvence stlačit pod 2 Mbps, tím nám ovšem zaručuje plynulost přehrávání tohoto videa. Videoukázka komprimovaná kodekem XviD MPEG4 Codec 1.1.3 (dále pouze XviD) má sice nejnižší skutečný DT až 875 kbps, ale už při této hodnotě je výsledné video plynulé, a tak XviD dosáhl nejnižší hodnoty DT výsledného videa bez trhavého pohybu.

Při porovnání kodeků pomocí jednotlivých objektivních metod a při pohledu na výsledné grafy (Obrázek 11) pro videosekvenci *Mravenci* bych celkově nejlépe hodnotil výsledky dosažené pomocí komprimace kodekem XviD. Při nejvyšším DT má nevyšší PSNR, míra rozmazání není příliš velká v porovnání se zdrojovým videem a hodnota VQM je druhá nejlepší (tedy druhá nejnižší). Hodnoty Blurring a VQM jsou při 2Mbps DT pro kodeky DivX a XviD velice vyrovnané, ale negativem pro DivX oproti XviD je, že nebylo možné porovnání i pro DT nižší. Naopak nejhorších výsledků dosáhl kodek MPEG4, který se vyznačoval nízkým PSNR, vysokým VQM a sice nejnižším rozmazáním, pro nižší DT dokonce menším rozmazáním než zdrojové video. Tuto skutečnost bych, ale přičetl i několikavteřinovým zásekům výsledných videosekvencí komprimovaných tímto kodekem.



Obrázek 11: Výsledné grafy pro videosekvenci *Mravenčí*

5.4.2 Bílý šum

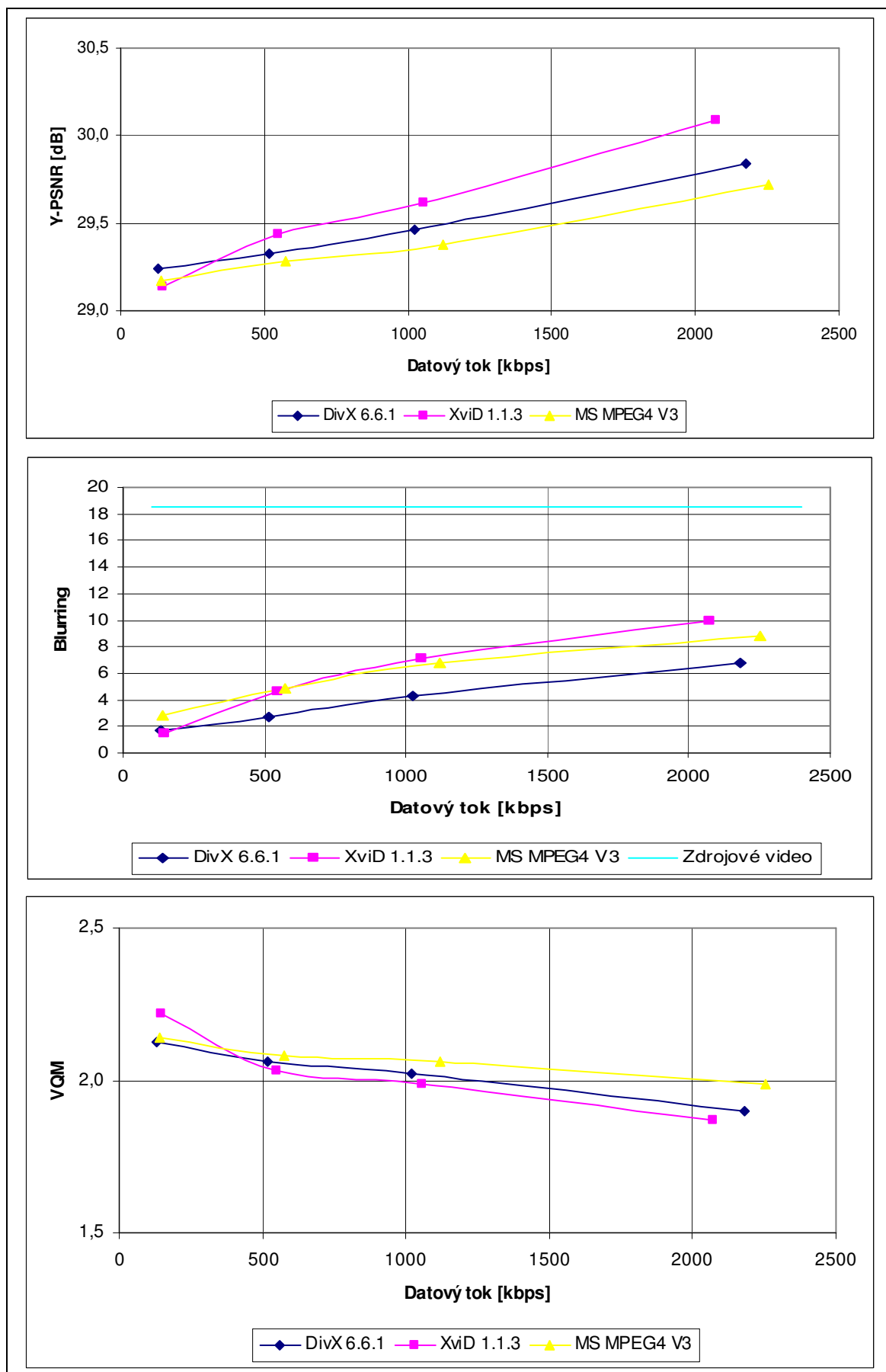
Nastavený DT [kbps]	Skutečný DT [kbps]	Doba enkódování [s]	Velikost souboru [kB]	PSNR (AVG) [dB]	Blurring (zdroj: 18,47849)	VQM
DivX 6.6.1						
128	132	30	528	29,23908	1,67261	2,12472
512	517	30	2000	29,32388	2,71419	2,06400
1024	1022	29	3933	29,45954	4,28854	2,02319
2048	2179	29	8357	29,83756	6,83165	1,89802
XviD MPEG4 Codec 1.1.3						
128	145	27	577	29,14123	1,42664	2,22096
512	549	34	2122	29,43711	4,58879	2,03318
1024	1055	38	4058	29,61754	7,09869	1,98607
2048	2073	49	7950	30,08547	9,94159	1,87097
Microsoft MPEG4 Video Codec V3						
128	139	27	565	29,17395	2,81760	2,13843
512	574	29	2229	29,28380	4,80730	2,08278
1024	1121	28	4324	29,37447	6,77922	2,06383
2048	2254	30	8655	29,71898	8,86659	1,98630

Tabulka 5: Nastavené a naměřené hodnoty při enkódování

Zhodnocení:

Oproti předchozí videosekvenci byly všechny tři kodeky schopné enkódovat zdrojové video, konkrétně *Bílý šum*, pro všechny DT. Z pohledu výsledných hodnot DT a doby enkódování jsou všechny testované kodeky velice vyrovnané a nedá se říci, že by některý výrazně zaostával či výrazně předčil ostatní.

Názorné grafy na obrázku 12 nám ukazují, že se ani při měření objektivními metodami neprojevila výrazná kvalitativní převaha některého z použitých kodeků. Nicméně až na nejnižší DT, kde byly lepší DivX i MPEG4, má ve všech ostatních DT nejlepší výsledky pro PSNR, Blurring i VQM kodek XviD. Při pohledu na míry rozmazání je vidět, že při použití všech kodeků došlo ke značnému rozmazání oproti zdrojovému videu, dá se tedy říci, že si v tomto případě, nevedl příliš dobře ani jeden kodek.



Obrázek 12: Výsledné grafy pro videosekvenci *Bílý šum*

5.4.3 Hlavy

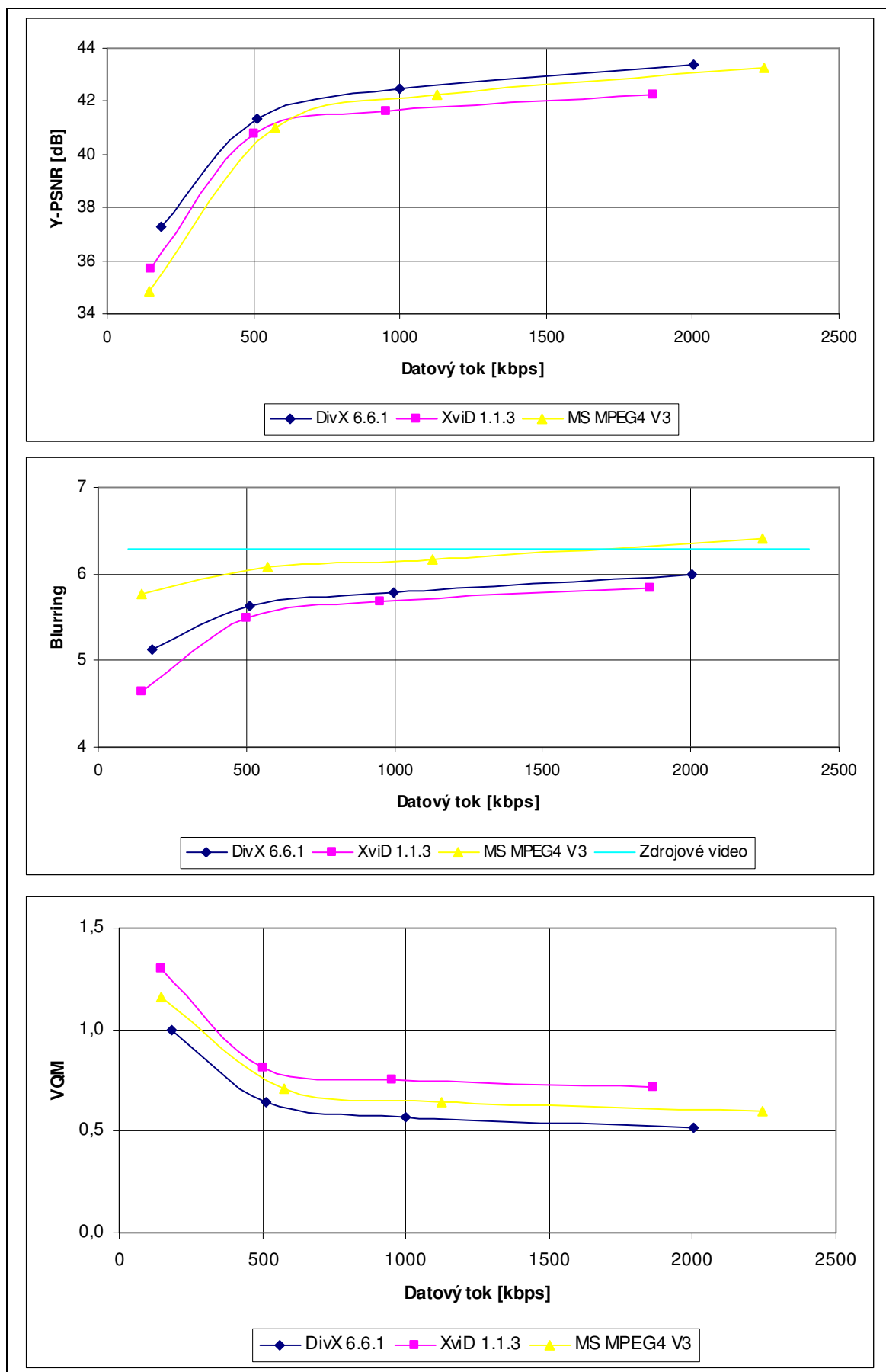
Nastavený DT [kbps]	Skutečný DT [kbps]	Doba enkódování [s]	Velikost souboru [kB]	PSNR (AVG) [dB]	Blurring (zdroj: 6,28494)	VQM
DivX 6.6.1						
128	183	44	718	37,27836	5,13559	0,99704
512	513	57	1957	41,36756	5,63080	0,64399
1024	997	64	3772	42,46054	5,79292	0,57266
2048	2003	69	7544	43,39839	5,98561	0,52037
XviD MPEG4 Codec 1.1.3						
128	149	34	591	35,72062	4,63322	1,30227
512	500	41	1905	40,80025	5,49555	0,81185
1024	951	44	3597	41,63840	5,68079	0,75642
2048	1864	45	7023	42,23255	5,83236	0,71741
Microsoft MPEG4 Video Codec V3						
128	146	18	586	34,87388	5,76202	1,16017
512	574	18	2191	41,01766	6,07327	0,71177
1024	1127	19	4264	42,23315	6,17381	0,64095
2048	2242	21	8444	43,27330	6,40541	0,59517

Tabulka 6: Nastavené a naměřené hodnoty při enkódování

Zhodnocení:

Až na DivX při nejnižším DT, nedošlo při enkódování videosekvence *Hlavy* k výraznějšímu nárůstu skutečného DT oproti nastavenému DT. V některých případech, především u kodeku XviD, byl dokonce výsledný DT nižší než nastavený. Doba enkódování vyšla nejpříznivěji pro kodek MPEG4, která byla většinou méně než-li poloviční vůči dalším testovaným kodekům.

Z měření objektivními metodami vyplývá, že s videoukázkou *Hlavy* si všechny použité kodeky poradili velice dobře. Výsledné hodnoty PSNR jsou pro všechny kodeky vysoké, nejlépe si však vedl od nejnižších až po nejvyšší DT kodek DivX, který je od vyšších DT těsně následovaný kodekem MPEG4. Míra rozmazání je nejpříznivější pro MPEG4, kde je dokonce pro nejvyšší DT lepší než-li zdrojové video. I hodnoty VQM jsou velice dobré, nejnižších a tudíž nejlepších výsledků dosáhl kodek DivX. Vzhledem ke kvalitě obrazu výsledného komprimovaného videa je pro tuto videosekvenci mírně lepší kodek DivX, následovaný kodekem MPEG4. Na druhou stranu DivX potřeboval nejvíce času pro enkódování videa. Výsledky pro tuto videoukázku tedy nemají jednoznačného vítěze.



Obrázek 13: Výsledné grafy pro videosekvenci *Hlavy*

5.4.4 Gladiátor

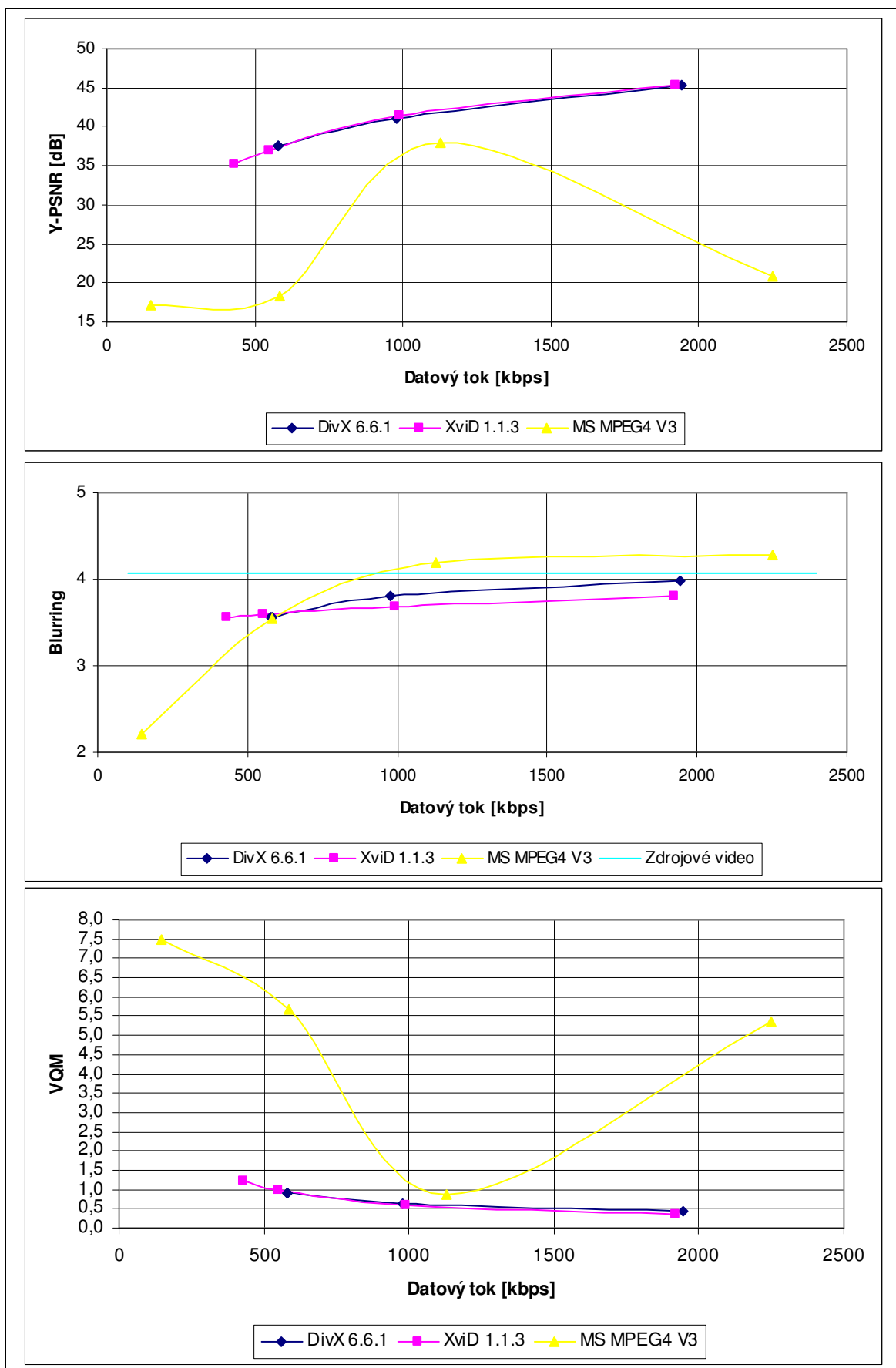
Nastavený DT [kbps]	Skutečný DT [kbps]	Doba enkódování [s]	Velikost souboru [kB]	PSNR (AVG) [dB]	Blurring (zdroj: 4,07031)	VQM
DivX 6.6.1						
128	578	45	2200	37,63257	3,55537	0,89213
512	580	46	2208	37,64154	3,56521	0,89388
1024	977	45	3697	41,06872	3,80536	0,63092
2048	1944	48	7324	45,30386	3,98791	0,42369
XviD MPEG4 Codec 1.1.3						
128	428	44	1637	35,26229	3,55569	1,20798
512	550	43	2095	37,00109	3,59538	0,98907
1024	990	44	3746	41,41056	3,68302	0,59003
2048	1922	51	7240	45,25734	3,81570	0,35019
Microsoft MPEG4 Video Codec V3						
128	148	46	592	17,08647	2,21511	7,46875
512	584	47	2228	18,38837	3,53628	5,67677
1024	1129	46	4273	38,01241	4,18608	0,88064
2048	2251	53	8479	20,92032	4,27827	5,36884

Tabulka 7: Nastavené a naměřené hodnoty při enkódování

Zhodnocení:

Výsledky měření videoukázky *Gladiátor* ukazují, že s dodržáním nejnižšího DT neměl větší problémy pouze kodek MPEG4, ale za cenu neplynulého videa. Jinak DivX i XviD nastavenou hodnotu DT přesáhly několikanásobně. Při vyšších DT však tyto problémy odpadly i u nich. Doba potřebná pro komprimaci je u všech testovaných kodeků prakticky shodná.

Data zjištěná měřením pomocí objektivních metod určují nejlepšími kodeky pro tuto videosekvenci DivX a XviD s tím, že nepatrně lepší je DivX. Oba dva kodeky dosahují velice dobrých výsledků ve všech třech metodách objektivního testování. Naopak kodek MPEG4 vykazuje dosti nízké PSNR, kromě hodnoty PSNR při DT 1129 kbps, kdy prudce vzroste, ale při 2Mbps PSNR opět rapidně klesá. Míra rozmazání je u tohoto kodeku dobrá, ale hodnoty VQM jsou, mimo již zmíněný DT 1129 kbps, velice špatné. Takovýto průběh je dosti neobvyklý, a tak jsem přistoupil k opakované komprimaci a měření, ale výsledky se opakovaly.



Obrázek 14: Výsledné grafy pro videosekvenci *Gladiátor*

5.4.5 Krtek

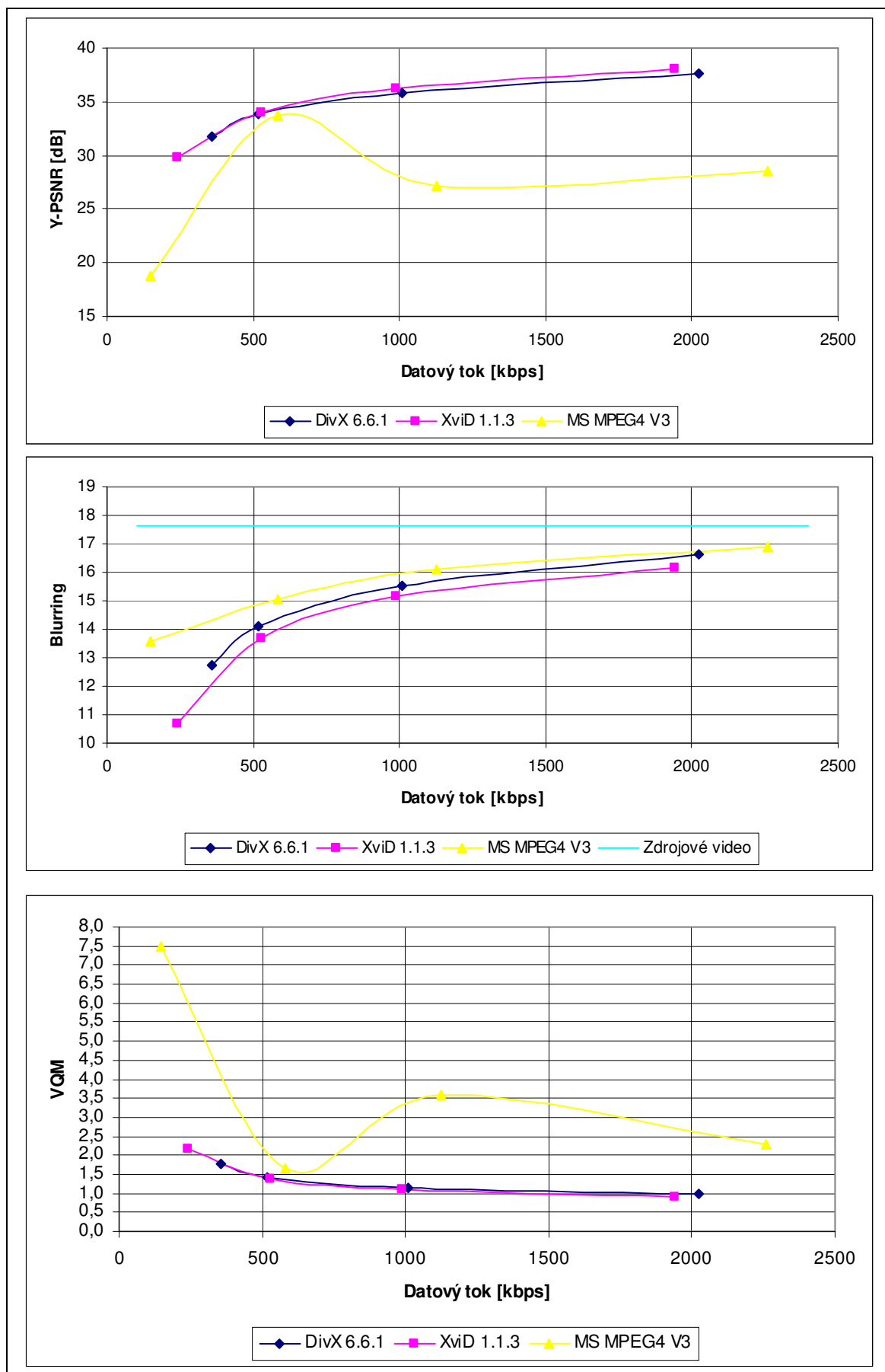
Nastavený DT [kbps]	Skutečný DT [kbps]	Doba enkódování [s]	Velikost souboru [kB]	PSNR (AVG) [dB]	Blurring (zdroj: 17,63276)	VQM
DivX 6.6.1						
128	358	52	1374	31,82058	12,72707	1,76708
512	517	52	1972	33,78999	14,08802	1,43210
1024	1008	54	3813	35,82776	15,53137	1,15850
2048	2024	59	7622	37,56238	16,61450	0,96558
XviD MPEG4 Codec 1.1.3						
128	243	44	941	29,74796	10,67614	2,17445
512	530	41	2018	34,06309	13,68404	1,38301
1024	990	41	3744	36,18861	15,13783	1,10654
2048	1940	44	7307	38,00372	16,15557	0,90945
Microsoft MPEG4 Video Codec V3						
128	147	41	591	18,80836	13,59354	7,50705
512	582	43	2220	33,72903	15,05866	1,66328
1024	1127	44	4265	27,16504	16,10990	3,58692
2048	2260	45	8514	28,59748	16,91304	2,27873

Tabulka 8: Nastavené a naměřené hodnoty při enkódování

Zhodnocení:

Poslední testovanou videosekvencí je videoukázka *Krtek*, kde došlo u kodeků DivX a XviD opět k výraznému překročení nastaveného nejnižšího DT. Nejpomaleji komprimoval zdrojové video kodek DivX, XviD a MPEG4 mají tuto dobu enkódování zhruba stejnou.

Z výsledných grafů (obrázek 15), vytvořených ze zjištěných hodnot objektivními metodami měření, je vidět vyrovnanost kodeků DivX a XviD. Oproti videosekvenci *Gladiátor* má však v tomto případě mírně navrch kodek XviD. Hodnoty pro oba dva kodeky dosažené při jednotlivých metodách měření jsou velmi dobré, zejména tedy pro nejvyšší DT. I zde je neobvyklý průběh hodnot pro kodek MPEG4, zlomovým bodem je zde však hodnota DT 582 kbps, kde tento kodek dosahuje nejlepších výsledků jak u PSNR, tak u VQM. Kromě tohoto DT, jsou hodnoty PSNR a VQM horší a dosti rozdílné oproti DivX a XviD. Míru rozmazání má MPEG4 nejlepší. I pro tuto videosekvenci jsem provedl opakovanou komprimaci a měření se shodnými výsledky.



Obrázek 15: Výsledné grafy pro videosekvenci *Krtek*

6 Streamování v praxi

Aby jsem streamovací platformy, konkrétně Windows Media a RealNetworks, nepopisoval pouze z manuálů a tištěných prezentací, rozhodl jsem se vyzkoušet si tyto platformy osobně a stručně popsat svoje postřehy při práci s nimi a při realizaci vlastního streamování.

Nainstaloval jsem si tedy Windows Media Encoder 9 Series a RealProducer 11. Po instalaci Windows Media se mi na počítači objevil jak samotný enkodér, tak i další tři utility pro editaci streamů a tvorbu profilů pro enkódování. I součástí RealProduceru je enkodér a utilita na editaci streamů, která nám umožňuje jednoduchý střih či doplnění informací o klipu. Pro samotné enkódování jsem si jako zdrojové video vybral již použité video pro testování kodeků a to videoukázku *Hlavy*.

Po prvním spuštění RealProduceru se člověk poměrně rychle zorientuje a zjistí, že má vcelku intuitivní ovládání. Levá část okna programu je určena pro zadání zdrojového videa, kde je možné aplikovat opravdu širokou škálu vstupních formátů. Já jsem použil již více zmíněnou videoukázku *Hlavy* uloženou v AVI (kodek DV) a na zkoušku i v MPG. Pravá část okna encoderu slouží pro nastavení výstupního videa, tedy vlastního streamu. Jsou zde tři hlavní tlačítka Audiences, Video Filters a Clip Information. Audiences nám umožňuje vybrat kodek, kvalitu audia a videa, rozlišení, ale především datový tok výsledného streamu. Datový tok může být VBR či CBR, může být použit pouze jeden datový tok či více datových toků (multibitrate, SureStream) pro jeden klip. Video Filters obsahuje filtry pro ořezání, deinterlaci a redukci šumu. Clip Information slouží ke vložení textových informací o daném streamu, jako je například titulek, autor, copyright, klíčová slova a další. Vyzkoušel jsem různá nastavení, ale pro další použití jsem vytvořil tři streamy o konstantních datových tocích 100 kbps, 350 kbps a 700 kbps, bez použití jakýchkoliv filtrů. Dvoupřechodové enkódování půlminutového klipu probíhalo mezi 2 až 3,5 minutami bez jakýchkoliv komplikací. Pomocí RealProduceru je také možné vytvořit šablony pro nastavení streamu a pro nastavení videoserveru.

Windows Media Encoder (WME) nám hned po spuštění nabízí průvodce, takže jednotlivé nastavení streamu není složité ani pro laika. Nejprve tedy vybereme umístění vstupního a výstupního videosnímku, poté způsob distribuce výsledného videa (např. Hardware devices, Windows Media server, Web server, Pocket PC a další), následují možnosti enkódování, kde jsou možnosti nastavení obdobné jako u RealProduceru. Pak je zde ještě okno pro vložení informací o klipu a nakonec přehled o použitých nastaveních. Po kliknutí na tlačítko <Dokončit> se spustí vlastní enkódování. Jako zdroj jsem opět použil video *Hlavy* v AVI (kodek DV), způsob distribuce Windows Media server a v okně pro možnosti enkódování jsem vybral z nabízených profilů (je samozřejmě možné vytvořit si profily vlastní) datové toky 130 kbps, 330 kbps a 750 kbps. Doba dvouprůchodového enkódování se pohybovala v rozmezí 40 s až 1 min, což je velice dobrá hodnota. I zde jsem odzkoušel zdrojové video i v MPG, ale WME při snaze enkódovat toto video opakovaně “padal“, přestože má MPG podporovat a RealProducer s ním neměl žádné problémy. WME také umožňuje opatřit stream DRM ochranou, tedy jakousi ochranou autorských práv digitálního videa.

Ted' když jsem měl připravené klipy pro streamování, tak bylo nutné vyřešit jakým způsobem je streamovat. Nakonec jsem se rozhodl využít služeb CESNETu a jejich videoseveru, který podporuje jak Real Video, tak Windows Media a je pro akademickou obec zdarma. K vytvoření účtu na CESNETu jsem si musel vytvořit tzv. elektronický podpis, pro ověření identity po e-mailu, kam mi byly posléze zaslány informace o mém novém účtu. Pro umíst'ování audiovizuálních souborů na videosever mi byl zřízen domovský adresář, ke kterému lze přistupovat přes protokol scp/ssh. Nainstaloval jsem si tedy program WinSCP, pomocí kterého jsem uložil mnou vytvořené videoukázky na videosever. Práce s WinSCP je velice podobná jako se souborovými manažery vycházejícími z Norton Commanderu.

Posledním krokem bylo vytvoření jednoduchých webových stránek, kde je možné streamovaná videa zhlédnout. Samozřejmě by pro přehrání streamu stačilo využít multimediální přehrávač, do kterého by se vložil příslušný odkaz, ale to je velice nepraktické. Vytvořil jsem tedy na WebZdarma.cz webové stránky www.streaming.wz.cz, na kterých je možné přepínat mezi Windows Media Playerem, Real Playerem a jednotlivými datovými toky. Ty jsou označeny jako Pomalé, Střední a Rychlé připojení.

7 Závěr

Cílem této práce bylo objasnění pojmů spojených s problematikou týkající se komprimace a přenos digitálního videa po internetu a tím se také zabývám v celé práci. Nejprve jsem vysvětlil základní pojmy, které se netýkají pouze videa, ale obrazu obecně, jako jsou například rozlišení, barevná hloubka a paměťové nároky. Dále jsem se popsal principy a dělení vlastní komprimace videa. Jsou zde uvedeny nejčastěji používané kodeky, za představitele bezztrátové komprimace jmenujme zejména HuffYUV. Mezi kodeky využívající ztrátové komprimace jsem uvedl především zástupce MPEG, jako je například MPEG4 a z něho vycházející DivX či XviD, ale i standard od Microsoftu VC-1. V části týkající se přenosu digitálního videa, jsem se zabýval technologií streamování (proudování), kde je uveden princip streamování, jeho režimy, jako jsou on-line a Video on Demand. Dále technologie přenosu dat Unicast, Multicast, které se samozřejmě nevyužívají pouze u přenosu digitálního videa. Je zde také uveden přehled používaných protokolů a platforem. Z platforem, které jsem měl možnost vyzkoušet a to ne pouze jako divák hotových streamovaných videí, ale i jako tvůrce vlastních streamů, bych z hlediska uživatelské přívětivosti, dostupnosti a podporovaných formátů nejkladněji hodnotil platformu od RealNetworks. Pro úplnost uvádím, že jsem pracoval ještě s platformami Windows Media a okrajově s QuickTime.

Při rozmyšlení jakým způsobem provedu porovnání kodeků, jsem narazil na internetové stránky Moskevské státní univerzity, která se touto problematikou zabývá a využil jsem tak i software vyvinutý na této instituci. S pomocí tohoto softwaru (MSU Video Quality Measurement Tool) jsem mnou zkomprimovaná videa objektivně porovnal se zdrojovými videi. Ostatně metodiky, použité videosekvence, použité kodeky, výsledky a jednotlivá zhodnocení jsou součástí poslední části této práce. Závěrem bych snad uvedl, že jsem testoval videosekvence komprimované kodeky založenými na MPEG4. A to konkrétně DivX 6.6.1, XviD MPEG-4 Codec 1.1.3 a Microsoft MPEG4 Video Codec V3. Jak je vidět, využil jsem pouze tří kodeků a to především z časových důvodů. Nicméně si myslím, že to pro potřeby této práce dostačuje, protože šlo především o získání praktických zkušeností při zpracování a porovnání digitálního videa. A tak bych při použití více kodeků dozajista dosáhl různorodějších výsledků, ale s opakující se metodikou testování. Pokud bych měl

zhodnotit práci s vybranými kodeky a výsledky měření pro jednotlivé videosekvence, tak bych to shrnul následovně. Kodeky DivX a Xvid skýtají spoustu možností nastavení, kdežto u MS MPEG4 V3 je to pouze vzdálenost klíčových snímků, datový tok a kvalita komprese, tedy základní možnosti nastavení. To se také projevilo na výsledcích, kde je vidět, že použité verze DivX i XviD jsou daleko novější než MS MPEG4 V3. Přestože měl ve třech testech z pěti mírně navrch kodek XviD 1.1.3 nad DivX 6.6.1, tak bych ho nefavorizoval a oba kodeky hodnotil jako velice vyrovnané. Naproti tomu se u kodeku MS MPEG4 V3 pravděpodobně projevila již jeho starší doba vzniku, konkrétně rok 1999, a tak jsou jeho výsledky z objektivního měření nevyrovnané a některých případech dosti zaostává za ostatními testovanými kodeky. Při použití vhodného nastavení, především dostatečného datového toku, jsou však všechny použité kodeky schopné dosáhnout výborné vizuální kvality a podstatné úspory velikosti výsledného videosouboru.

Doufám, že informace obsažené v bakalářské práci jsou srozumitelně podané a dodávám, že jsem tuto problematiku nemohl popsat podrobněji, protože je zkrátka příliš obsáhlá, o čemž jsem se opravdu přesvědčil až při shromažďování informací pro tuto práci. Nalézt některé informace nebylo příliš jednoduché a časově dosti náročné, ale díky tomu jsem si prohloubil vlastní znalosti v této problematice, která se v budoucnu dozajista dostane ještě více do popředí zájmu.

Použitá literatura

- [1] Jiří Žára, Bedřich Beneš, Jiří Sochor, Petr Felke, Moderní počítačová grafika, Brno, Computer Press, 2004, ISBN: 80-251-0454-0
- [2] Rita Pužmanová, Moderní komunikační sítě od A do Z, Computer Press, ISBN: 80-251-1278-0
- [3] Jan Čapek, Petr Fabian, Komprimace dat: principy a praxe, Praha, Computer Press, 2000, ISBN: 80-7226-231-9
- [4] <http://wikipedia.org/>
- [5] <http://www.cesnet.cz/doc/>
- [6] <http://www.ics.muni.cz/zpravodaj/>
- [7] <http://www.lupa.cz/>
- [8] <http://www.zive.cz/>
- [9] <http://www.tvfreak.cz/>
- [10] http://www.compression.ru/index_en.htm
- [11] <http://www.civ.cvut.cz/info/info.php?id=137>
- [12] <http://www.realnetworks.com/>
- [13] <http://www.microsoft.com/cs/cz/>
- [14] <http://www.apple.com/quicktime/>
- [15] <http://www.divx.com/>
- [16] <http://www.xvid.org/>
- [17] <http://www.xiph.org/>
- [18] <http://www.chiariglione.org/mpeg/>
- [19] <http://www.flumotion.com/en/>
- [20] <http://www.grafika.cz/>