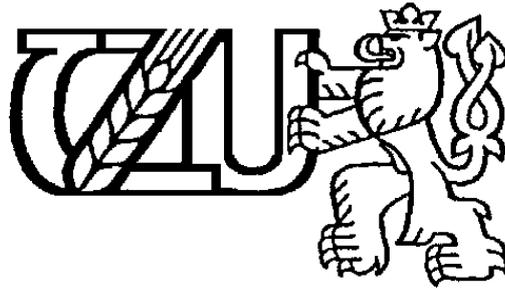


Česká zemědělská univerzita

Provozně ekonomická fakulta

Katedra informačních technologií



Diplomová práce

**Metody komprese obrazu v IP
kamerových systémech**

**Methods of image compression in IP camera
systems**

Autor diplomové práce: Bc. Tomáš Helmich

Vedoucí práce: Ing. Jiří Vaněk, Ph.D.

© 2011

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačních technologií

Akademický rok 2009/2010

ZADÁNÍ DIPLOMOVÉ PRÁCE

Tomáš Helmich

obor Informatika

Vedoucí katedry Vám ve smyslu Studijního a zkušebního řádu ČZU v Praze čl. 17 odst. 2 určuje tuto diplomovou práci.

Název práce: **Metody komprese obrazu v IP kamerových systémech**

Osnova diplomové práce:

1. Úvod
2. Cíl práce a metodika
3. Přehled řešené problematiky
4. Kompresní metody
5. Streamování videa
6. Případová studie
7. Závěr
8. Seznam použitých zdrojů
9. Přílohy

Rozsah hlavní textové části: 60 - 80 stran

Doporučené zdroje:

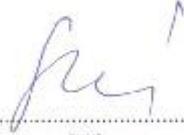
1. T. Lovčeck, P. Nagy. Bezpečnostné systémy - Kamerové bezpečnostné systémy. Žilina: EDIS, 2008. ISBN 978-80-8070-893-1.
2. Jan Čapek, Peter Fabian. Komprimace dat Principy a praxe. Brno: Computer Press, 2000. ISBN 8072262319
3. Sovič. MPEG-1,2 video. [online]
http://www.pakuj.host.sk/mpeg_video/video.html
4. Wikipedia. Real Time Streaming Protocol. [online]
<http://en.wikipedia.org/wiki/RTSP>
5. Kompresce a komprimační algoritmy. [online]
<http://gimli.mysteria.cz/kompresce/obsah.html>
6. Transform Methods and Image Compression. [online]
<http://www.linuxjournal.com/article/2567>

Vedoucí diplomové práce: **Ing. Jiří Vaněk, Ph.D.**

Termín odevzdání diplomové práce: duben 2011




.....
Vedoucí katedry


.....
Děkan

V Praze dne: 15. 1. 2010

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně, pouze za odborného dohledu vedoucího diplomové práce, Ing. Jiřího Vaňka, Ph.D.

Dále prohlašuji, že veškeré podklady, ze kterých jsem čerpal, jsou uvedeny v seznamu použitých zdrojů.

.....
Bc. Tomáš Helmich

V Lysé nad Labem dne

Poděkování

Rád bych tímto poděkoval panu Ing. Jiřímu Vaňkovi, Ph.D. za vedení a odborné rady při tvorbě mé diplomové práce Metody komprese obrazu v IP kamerových systémech.

Metody komprese obrazu v IP kamerových systémech

Souhrn

Diplomová práce se zabývá problematikou metod komprese obrazu v IP kamerových systémech. V úvodní části práce jsou stručně vysvětleny teoretické základy kamerových systémů, rozdíly mezi analogovým a digitálním kamerovým systémem, odlišnosti ve způsobu snímání obrazu a použitá rozlišení.

V druhé části se autor podrobněji zabývá jednotlivými kompresemi využívanými v IP kamerových systémech. Dále zde uvádí principy bezztrátových algoritmů, komprese statických obrázků a komprese videa. Konkrétně jsou analyzovány komprese MJPEG, MPEG-x a H.26x z hlediska vývoje a algoritmů, které používají ke komprimování videa.

Další kapitola pojednává o streamování videa, jsou zde uvedeny základy streamování, formáty pro streamování videa, přizpůsobování se zahlcené síti, logování a základy serverové architektury.

V praktické části je proveden test zatížení datové linky s ohledem na rozlišení dané IP kamery a s ohledem na využívanou kompresi. V další části kapitoly autor představuje možnosti vložení obrazu z IP kamer na webové stránky, včetně tvorby streamovacího serveru. Jsou zde uvedeny metody vyčítání obrazu z kamery, odesílání obrázku na FTP server, vložení obrazu pomocí komponenty Microsoft ActiveX a pluginu Apple QuickTime a také metoda streamování videa. U každé metody jsou provedeny testy náročnosti na hardware počítače a vytížení sítě, jsou zde uvedeny výhody, nevýhody a kompatibilita zobrazení v jednotlivých (v současnosti nejvíce využívaných) prohlížečích, konkrétně v Internet Exploreru, v Mozilla Firefox, Google Chrome, Opera a v Safari.

Klíčová slova

Komprese, MJPEG, MPEG-x, H.26x, IP kamerový systém, rozlišení, světové video standardy

Methods of image compression in IP camera systems

Summary

This diploma thesis deals with methods of image compression in IP camera systems. In the introduction are briefly explained theoretical basics of camera systems, differences between analog and digital camera system, different ways of imaging and used resolutions.

In the second part, author deals in detail with individual compressions used in IP camera systems. Furthermore he describes principles of lossless algorithms, compression of static images and video compression. Specifically, compressions MJPEG, MPEG-x and H.26x will be analyzed with aspect to development of algorithms which they use for video compression.

In the following chapter, the reader will be familiarized with video streaming, streaming basics, adaptation to flooded networks, logging and server architectures.

In the practical part, load test on data lines will be conducted with regard to resolution of the particular IP camera and used compression. In the next part of the chapter, author introduces possibility of inserting images from IP cameras to a website including creation of a streaming server. Methods of retrieving images from the camera, sending a picture to FTP server, inserting of image using the Microsoft ActiveX component and Apple Quick Time plugin and also method of video streaming will be described. Load tests on hardware and network will be done for every method, there will be stated positives and negatives and compatibility of display in common browsers, specifically in IE, Mozilla Firefox, Google Chrome, Opera and Safari.

Keywords

Compression, MJPEG, MPEG-x, H.26x, IP camera systém, resolution, world video standards

Obsah

1	Úvod.....	10
2	Cíl práce a metodika.....	11
2.1	Cíl práce.....	11
2.2	Metodika.....	11
3	Přehled řešené problematiky	12
3.1	Konvergence.....	14
3.2	Digitální video	15
3.3	Digitální proti analogovému zobrazení obrazu	16
3.4	Světové video standardy.....	16
3.5	Prokládané řádkování	17
3.6	Progresivní skenování.....	18
3.7	Digitální barevná hloubka	19
3.8	Rozlišení	19
3.8.1	Rozlišení videa	20
3.9	Barevná prezentace.....	22
3.9.1	RGB.....	22
3.9.2	YCrCb	22
4	Kompresní metody	24
4.1	Bezztrátové kompresní algoritmy.....	25
4.1.1	Huffmanovo kódování.....	25
4.1.2	RLE (Run-length Encoding)	27
4.2	Kompresce statických obrázků.....	28
4.2.1	JPEG.....	28
4.3	Kompresce videa	37
4.3.1	MPEG.....	38
4.3.2	H.26x.....	52

5	Streamování videa	61
5.1	Streamování	62
5.1.1	Formáty souborů pro streamování	63
5.1.2	Pomocné stopy	63
5.1.3	Přizpůsobování se zahlcené síti.....	64
5.1.4	RealNetworks	64
5.1.5	Nahrávání obsahu	65
5.1.6	Živé streamování	65
5.1.7	Oznamování obsahu	65
5.1.8	Webové odkazy	65
5.1.9	SMIL	66
5.2	Webcasting	66
5.2.1	Dělení a střídání	66
5.2.2	Multicasting.....	66
5.3	Playlisty	67
5.4	Logování a statistiky.....	67
5.5	Serverové architektury.....	68
5.5.1	Windows Media Services Series 9	68
5.5.2	RealNetworks Helix Universal Server	69
5.5.3	Apple QuickTime	69
5.6	Umístění serveru.....	69
5.6.1	Hosting	70
5.7	Souhrn.....	71
6	Případová studie	72
6.1	Výkon standardů pro kódování videa.....	72
6.2	Vložení obrazu do internetových prezentací	73
6.2.1	Vyčítání obrázku z kamery.....	74

6.2.2	Odesílání obrázků na FTP server	77
6.2.3	Microsoft ActiveX.....	78
6.2.4	Apple Quick Time	80
6.2.5	Streamování obrazu.....	81
6.3	Souhrn.....	87
7	Závěr.....	89
8	Seznam použitých zdrojů	90
9	Seznam použitých tabulek.....	92
10	Seznam použitých obrázků.....	92
11	Přílohy	94
11.1	Slovník cizích pojmů.....	94

1 Úvod

V současném digitálním světě představují bezpečnostní IP kamery účinnou zbraň či pomoc při střežení majetku nebo ochraně osob. IP kamery nepřetržitě sledují vše, co je třeba pozorovat a chránit, od soukromého majetku, veřejných prostranství, škol, nemocnic, bank, až po národní památky.

IP kamery v současnosti představují nejrychleji rostoucí segment trhu s výrobky pro sledování a ochranu majetku. [1] Důvodem, proč se stále více využívá IP kamer namísto starších CCTV technologií, je jejich nezávislost na rozlišení NTSC/PAL video standardů. IP kamery mají vyšší rozlišení obrazu, které se stále zvyšuje a neexistuje žádný limit nebo omezení, které by ho nějakým způsobem regulovalo. Další velkou výhodou oproti CCTV kamerám je možnost digitálního záznamu včetně zvuku, vnitřní inteligence (detekce pohybu, analýza chování), napájení po Ethernetu, progresivní snímání obrazu, zabezpečený přenos (HTTPS) a flexibilní výběr infrastruktury - IP síť je zpravidla již natažena ve většině lokalit.

Video získané IP kamerou je odesláno přes datové sítě záznamovým zařízením na vyhodnocení, či záznam. Odesílat video bez komprese je v praxi téměř nemožné z hlediska náročnosti na kapacitu média, ale zejména na přenosovou šířku pásma, jedna IP kamera by potřebovala vlastní gigabitovou síť. V IP kamerových systémech jsou využívány následující tři druhy kompresí: MJPEG, MPEG-4 a H.264. Každá z kompresí má určité výhody, ale i nevýhody. V této práci se autor zabývá porovnáváním jednotlivých způsobů komprese a jejich datovou náročností.

Umístění živého videa na webové stránky nebo zajištění přístupu na kameru více uživatelům lze provést několika způsoby - od vložení obnovujícího se obrázku, přes vložení živého videa pro několik uživatelů, až po streamování videa neomezenému počtu diváků. Streamování videa je nákladný proces náročný na hardwarový výkon a přenosovou šířku pásma, kdy server přijímá stream z IP kamery, zpracuje jej ve streamovacím softwaru a dále jej posílá do počítačové sítě. Autor zde vysvětluje nejen teoretický základ týkající se streamování, ale provádí i testování softwaru pro tyto účely využívaného.

2 Cíl práce a metodika

2.1 Cíl práce

Cílem této diplomové práce je analyzovat vliv použité komprese v IP kamerových systémech na zatížení datové linky a hardwaru počítače a uvést, jaké komprese se využívají, jejich vývoj a principy.

Výsledkem diplomové práce bude ukázka zatížení datových linek a následně také ukázka možností vložení obrazu z IP kamer na webové stránky.

Cíle práce jsou tedy především:

- Položení teoretického základu používaných kompresí a streamování
- Testy zatížení hardwaru a datových linek
- Možnosti vložení obrazu z IP kamer na webové stránky a tvorba streamovacího serveru
- Závěry a hodnocení

2.2 Metodika

V první části práce bude vysvětleno, co je to IP kamerový systém, jeho využití a nezbytné prvky k jeho realizaci.

Druhá kapitola se bude zabývat především jednotlivými kompresemi z hlediska jejich principu, funkčnosti a algoritmů.

Třetí kapitola bude obsahovat teoretický základ pro streamování videa. Bude zde vysvětlen princip pro tvorbu streamovaného videa.

Případová studie bude věnována zátěžovým testům hardwaru a datových linek v závislosti na použité kompresi a rozlišení IP kamery. Dále zde budou uvedeny možnosti vložení obrazu z IP kamer na vlastní webové stránky a tvorba streamovacího serveru.

3 Přehled řešené problematiky

Sledování pomocí kamer začali využívat dozorcí ve věznicích koncem 19. století pro odhalování metod používaných k útěkům. Pro ochranu majetku a lidí se sledování začalo využívat až od poloviny 20. století. Astronomické ceny prvních bezpečnostních kamerových systémů, postavených na tradičních fotografických kamerách a filmech, limitovaly jejich použití do vládních budov, bank a kasin. Pokud byla zaznamenána podezřelá aktivita, společnost, která měla monitorování na starosti, tento film vyvolala v temné komoře – laboratoři pro pozdější analýzu. Živý přenos byl příležitostně využíván během speciálních událostí pro monitorování davu lidí, ovšem policisté museli být v televizním studiu, aby mohli sledovat kamery.

Teorie, která je za video-sledováním, byla založena na stejných čtyřech klíčových faktorech, které přetrvávají dodnes:

- 1) zastrašování
- 2) efektivita
- 3) schopný ochránce
- 4) detekce

a) Zastrásování

Když potenciální pachatel ví o možnosti sledování a nahrávání, může se rozhodnout, že riziko odhalení je vícenásobně vyšší, než případný zisk. Sledovací systémy jsou jako zastrašovací prostředky používány u kasin, přes obchody, až po hromadnou dopravu. V zemích celého světa se sledovací systémy nejvíce zaměřují na hromadnou dopravu a vybraná veřejná prostranství. Množství zločinů, kterým se předešlo použitím sledovacích systémů, je ovlivněno prostředím a tím, jestli se jedná o systém pasivní, aktivní nebo o kombinaci obou. Pasivní systém využívá nahrávky po incidentu jako pomoc pro vyřešení zločinu. Aktivní systém je monitorován obsluhou, která reaguje v momentě upozornění. Historicky nejefektivnější sledovací systémy pro prevenci zločinů dokážou mnohem více, než pouze nahrát zločin na pozadí. Konkrétním příkladem je Střední škola Farragut v Chicagu, veřejná škola pověstná spoustou násilí, vykrádáním skříněk a vandalismem. Do jednoho roku od instalace CCTV sledovacího systému tyto negativní jevy téměř vymizely. Spousta amerických měst také zaznamenala snížení kriminality díky implementaci sledovacích systémů.

V aktuální UK Home Office Research studii ohledně efektivity sledovacích systémů jako zastrašovacích prostředků proběhlo 46 průzkumů na veřejných prostranstvích v Americe a v Anglii. Ze 46 studií poskytlo pouze 22 dostatečně kvalitní data, která se dala publikovat. Všechny 22 studií prokázalo značné snížení (až o 50 %) počtu vloupání, krádeží aut a násilných trestných činů. Je velice složité analyzovat data týkající se efektivity sledovacích systémů kvůli velkému počtu proměnných ve složitosti oblasti sledování. Například, monitorování sledovacím systémem může donutit kriminálníky přesunout se na jinou lokaci a tím přesunout i své trestné činy.

b) Efektivita

Přehrávání záběrů zaznamenaných kamerovým systémem současně se sledováním živého záběru z kamery přináší další informace o situaci. Současné prohlížení živého a uloženého záznamu může potvrdit nelegální aktivitu dříve, než proti podezřelému zasáhne bezpečnostní služba. V roce 2007 v Dallasu v Texasu policejní oddělení použilo záběry z 559 incidentů u 159 zatčení. Jejich zkušenosti ukázaly, že jediný policista, monitorující živé záběry a již uložené nahrávky, dokáže pokrýt mnohem větší oblast, než strážník v terénu.

c) Schopný ochránce

V článku „Social chase and crime rate trends: A routine activity approach“ od Lawrence Cohena a Marcuse Felsona (American sociological review) autoři naznačují, že prevence kriminality zahrnuje také přítomnost „schopného“ dohlížejícího ochránce. Tento ochránce nemusí být přímo na místě, stačí, když sleduje. Dnes ochránci nemusí ani sledovat, stačí ukládat pomocí moderních technologií. Současné sledovací systémy obsahují sofistikovaný, video-analyzující software s možností monitorování oblastí pro programovatelné situace (např. označení všech červených automobilů jako opuštěná auta, batohy, označování vozidel, nebo přímo konkrétních SPZ). Video-analytici mohou změnit původně pasivní systém na aktivní. Toto představuje schopného ochránce tím, že dáme pasivnímu systému „mozek“ a umožníme mu být citlivější vůči kriminální aktivitě.

d) Detekce

Detekce je faktor poskytující hmatatelný důkaz, že kamerové systémy fungují. Británie je proslulá svým kamerovým systémem, který umožňuje bezpečnostním složkám sledovat kohokoliv procházejícího Londýnem díky 200 000 kamerám (přes 4 miliony kamer v celé zemi).

Falešné kamery a náchylnost k „Falešnému pocitu bezpečí“

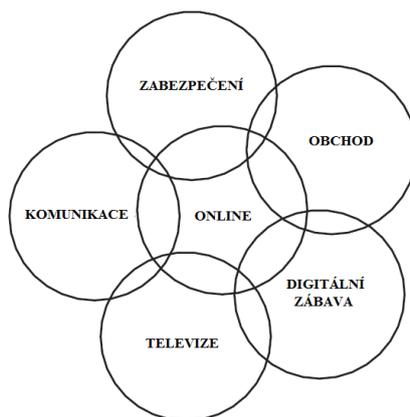
V kamerových systémech existuje mnoho způsobů sledování objektů, ale všechny mají společné to, že se snaží využít zastrašující faktor. Je mnoho společností, které prodávají falešné sledovací systémy. Ty mohou v krátkém časovém období pomoci zabránit kriminalitě. Lidé chodící po obchodech, náměstích a památkách mohou uvěřit, že je tam bezpečno, protože vidí kamery a myslí si, že někdo toto místo hlídá. V Americe, pokud dojde ke zločinu a kamery nejsou skutečné, falešný pocit bezpečí může být základem pro výhru v soudním procesu. I přesto, že pachatel porušil zákon nebo vstoupil na soukromý pozemek, soud může potrestat společnost za instalaci falešných kamer. Instalace falešných kamer může být považována za porušení kontraktu se zaměstnancem, společnost vědomě říká, že je zde bezpečno, i když to není pravda, a navozuje tak v zaměstnanci falešný pocit bezpečí. Nakonec mohou falešné kamery stát více než kamery skutečné, a to díky soudním výdajům a pokutám. Podle [www stránky „Video Surveillance Guide“ \(www.video-surveillance-guide.com\)](http://www.video-surveillance-guide.com), jakmile jsou falešné kamery odhaleny, dochází ke zvýšení kriminality, a to někdy s devastujícími následky. [2]

3.1 Konvergence

Digitální video zabezpečení (DVS) je možné používat díky rychlému růstu standardizovaných telekomunikačních technologií a kompresních video-formátů. Jakmile datové sítě dosáhly úrovně, kdy zvládají větší nápor dat spojený se zavedením lepší kvality komprese videa, algoritmů a výpočetního výkonu, bylo možné dodávat dobré video streamy z místa na místo bez standardizovaných televizních signálů, DVD přehrávačů nebo herních konzolí. DVS pracuje téměř stejně jako YouTube, který umožňuje komukoliv a kdekoliv přijímat a zobrazovat video-stream bez jakýchkoliv problémů s kompatibilitou. DVS není pouze o zábavě nebo zkoušení, ale o ochraně životů a majetku.

Univerzální přístup je umožněn díky všudypřítomným různým druhům datových telekomunikací, které zprostředkovávají kabelovou televizi, širokopásmové připojení k internetu a telefon v jedné lince (kabelu), zatímco před padesáti lety to bylo nereálné. Toto technologické propojení přináší „konvergence“. Konvergence je proces integrace samostatných technologií, které využíváme v běžném životě. Tato nová univerzální metoda umožňuje sledovat videa nebo poslouchat hudbu na televizi, DVD přehrávači, herní konzoli, počítači anebo na mobilním telefonu (obrázek č. 1).

Jak se svět digitalizuje, analogová televize na HDTV, klasické telefony na VoIP a mobily, analogové magnetické bezpečnostní karty na smart karty a VCR na DVR, není již složité integrovat tyto systémy. Jakmile je vše digitální, není potřeba mít jeden přístupový bod pro živé video nebo nahrané záznamy ze sledování. DVS totiž sdílí příslušné informace a stejně tak i odesílá upozornění managementu, ostraze nebo místnímu policejnímu oddělení pomocí informačních technologií z bezpečnostního přístupového managementu na modemy a mobilní telefony. [2]



Obrázek č. 1 - Konvergence digitálního světa
Zdroj: [2], upraveno

3.2 Digitální video

Rozdíl mezi digitálním a analogovým videem je v metodě přenášení dat. Digitální video následuje zákony digitálních technologií: rozbíjení videa (obrázků) na proud binárních dat, složený z jedniček a nul, vysílaných jako nízkonapěťové pulsy přes měděný drát, nebo světelné pulsy přes kabel optický. Cokoliv, co vyžaduje práci s bity (binárními číslicemi), používá mikroprocesor a data ukládá v binárních hodnotách 1 a 0.

Digitální video není zákonitě lepší než tradiční analogový signál, ale je mnohem více konzistentní při přenášení a zabírá mnohem menší pásmo při správné digitální kompresi. Analogové video je křivka, která může být poškozena nebo změněna prostředím, atmosférickým nebo technologickým rušením. Ve skutečnosti, jelikož jakákoliv křivka je dostupná a nekonzistentní, může být složité odlišit šum a rušení od základu analogového signálu. Lidé, kteří stále používají staré antény, mohli v dobách analogové televize často toto rušení zaznamenat. Digitální video je však matematicky absolutní, používá efektivnější kompresní techniky pro přenos většího množství dat včetně parametrů pro redundanci, opakované pokusy a kvalitu. Digitální video je také poněkud univerzálnější díky rychlosti rozšiřování internetu. Narozdíl od analogového videa, které má několik standardů se dvěma

dominantními standardy (NTSC a PAL) a limitovaný frekvenční rozsah pro přenos, digitální video je limitováno pouze šířkou pásma (rychlost linky v kbps). [2]

3.3 Digitální proti analogovému zobrazení obrazu

Obraz na jakémkoliv monitoru nebo obrazovce, ať už na tradiční analogové CRT televizi nebo na počítačovém monitoru, je výsledkem akumulace obdélníkových bodů zvaných pixely. Všechny obrázky nebo video jsou měřeny v pixelech. V analogovém světě je každý z těchto pixelů složen ze tří teček různých barev – červená, zelená a modrá (RGB). Pouhým okem je vidět jen směs těchto tří barevných teček jako jedna barva na CRT obrazovce. Analogové televizní obrazovky mají barevnou hloubku 256 úrovní pro každou vrstvu ze tří uvedených barev, takže každý analogový pixel má rozsah cca 16,8 milionu barev (16 777 216). To samé platí pro LCD monitory, jen výsledku je dosaženo jinak. LCD monitor je tenký, plochý displej z tekutých krystalů, zmáčknutých mezi dvě vrstvy polarizovaného skla, kterému se říká substrát. Malá fluorescentní lampa ozáří první substrát, a když elektrický proud z tenkého filmu tranzistorů aktivuje krystaly, aby se nastavily na různé úrovně světla a barvy, projde skrz druhý substrát, čímž vytvoří to, co vidíme na obrazovce.

Zatímco CRT monitor zobrazuje RGB s využitím fosforu, LCD monitor stále potřebuje tři stupně barevných teček, aby vytvořil pixel. Ideálně by tyto tři tečky nebo vrstvy měly být na přesně stejném místě, ve skutečnosti jsou ale jen dostatečně blízko nebo překryté v různém poměru tak, aby oklamaly pouhé oko, které si myslí, že vidí 16,8 miliónů různých barev. Vytisknutá barevná stránka (pod zvětšením) také ukazuje podobnou metodologii využívající čtyřbarevný proces CMYK (Cyan, Magenta, Yellow, Black). Tyto čtyři primární barvy, použité v různém poměru, mohou blízce simulovat (dostatečně pro lidské oko) stejných 16,8 milionu barev. [2]

3.4 Světové video standardy

Existují tři hlavní celosvětově používané standardy: americký NTSC systém pro barevnou televizi, evropský PAL a francouzský SECAM.

Základním rozdílem mezi těmito třemi systémy jsou počty řádků a rychlost snímků za sekundu (viz tabulka č. 1). Zatímco NTSC používá 525 řádků, PAL i SECAM používají 625 řádků. Počet snímků za sekundu u NTSC je o něco méně než půlka 60 Hz frekvence, zatímco PAL a SECAM mají přesně polovinu 50 Hz frekvence (25 fps).

Standardní NTSC analogový signál se posílá frekvencí mezi 30 Hz až 6 MHz, původně (ale i dnes) přes vlny vzduchem do antén nebo přes koaxiální kabely. NTSC je standard o 525 řádcích po 720 pixelech, což je dohromady 378 000 pixelů na jeden obrázek (frame), a analogový signál přenáší celý obraz rychlostí 29,97 fps. PAL dodává 625 řádků po 720 pixelech rychlostí 25 obrázků za sekundu nebo 450 000 pixelů na obrázek. [2]

	Počet řádků	Počet aktivních řádků	Frekvence	Počet sn/s
NTSC	525	484	60	29,97
PAL	625	575	50	25
SECAM	625	575	50	25

Tabulka č. 1 – Světové video standardy

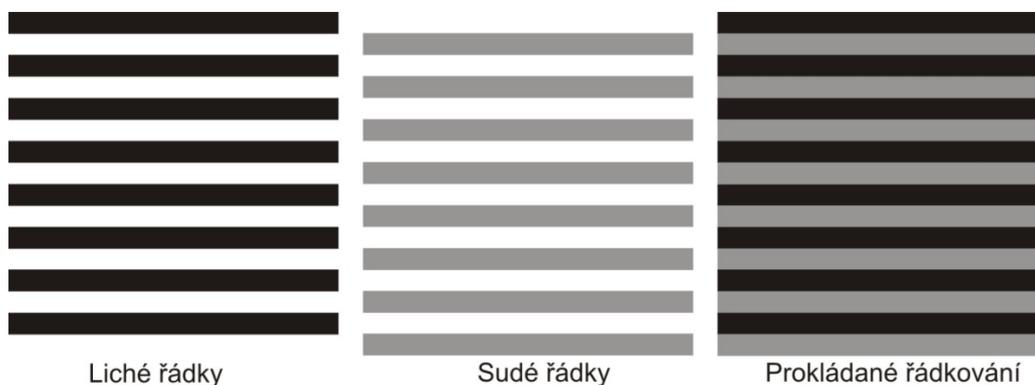
Zdroj: [2], upraveno

Proč 29,97 fps (frames per second – počet snímků za sekundu)?

Televize začínala v černobílém režimu s 525 řádky skenovanými v 1/30 sekundě. Řádky se vytvářely frekvencí 15 750 Hz (525×30). S vynalezením barevné televize bylo potřeba udělat prostor pro RGB signály, a proto se z 15 750 Hz stalo 17 734 Hz, což změnilo 30 snímků za vteřinu na 29,97 ($17\,734 \div 525$).

3.5 Prokládané řádkování

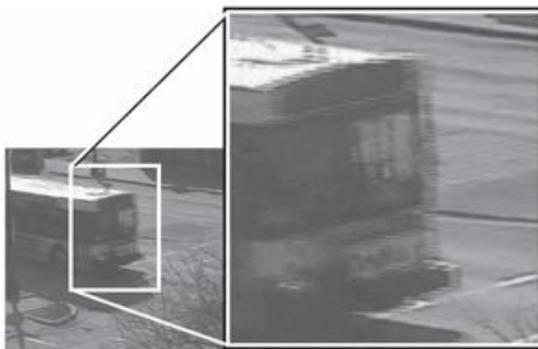
Analogové video včetně televize, VHS a analogových videokamer používá „prokládanou“ metodu skenování (viz obrázek č. 2). Prokládání pracuje na analogových TV a monitorech s nižším rozlišením, protože postrádá dostatečnou viditelnost detailů. Obrázky založené na prokládání byly vyvinuty pro CRT televizní monitory, kde se rozdělil počet viditelných řádků na standardní TV obrazovce na dvě separátní liché a sudé pole, která se obnovují v rychlosti 29,97 fps.



Obrázek č. 2 - Diagram prokládaného skenování

Zdroj: [2], upraveno

Jakékoliv drobné zpoždění v obnovovací frekvenci mezi těmito dvěma skupinami vytváří v obraze zubaté nebo šrafované rušení (viz obrázek č. 3). To je způsobeno tím, že první skupina se obnovuje v jiné frekvenci, než skupina druhá. [2]

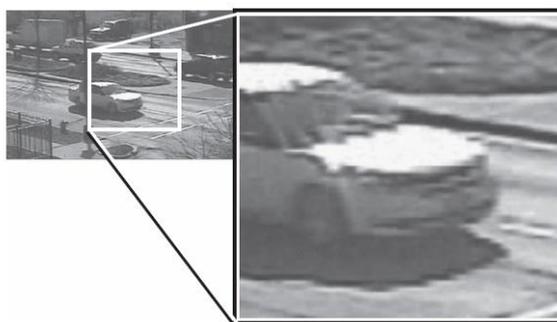


Obrázek č. 3 - Prokládání (interlacing) - pouze pohybující se objekty jsou ovlivněny
Zdroj: [2], upraveno

3.6 Progresivní skenování

Digitální kodéry poskytují filtr odstraňující prokládání, nebo volbu progresivního skenování, aby se předešlo zubatému obrazu prokládaného videa (což je návrat k původnímu analogovému videu). Odstraňování prokládání nebo progresivního skenování (viz obrázek č. 4) vyžaduje mnohem větší výpočetní výkon a je zvláště důležité při vysokých rozlišeních, jelikož v nich se dá rušení mnohem lépe rozpoznat. Veškeré kamery s pomalejší závěrkou mohou vytvářet rozmazané obrázky. Rozmazání je ještě navýšeno rušením v prokládaném obraze, protože kamera není dostatečně rychlá, aby zaznamenala pohyb.

Progresivní skenování (jako protiklad k prokládání) skenuje celý obrázek, postupně řádek po řádku za 1/16 vteřiny bez toho, aby došlo k rozdělení záběru na dvě políčka. Digitální technologie nepotřebuje prokládané skenování pro zobrazení videa. Digitální video eliminuje „blikající“ efekt videa za předpokladu, že monitor je také odpovídající kvality. Zároveň je důležité, aby digitální zařízení poskytovalo volbu deinterlacingu (odstranění prokládání) a volbu progresivního skenování. [2]



Obrázek č. 4 – Odstranění prokládání (deinterlacing) – odstraňuje rozostřené řádky
Zdroj: [2], upraveno

3.7 Digitální barevná hloubka

V digitálním světě absolutních matematických rovnic nejsou pixely měřeny v bodech, které se liší barvou světlo-vyzařujícího fosforu, ale v bitech a bytech. Stejných 256 barev z tradičních barevných TV zahrnuje 16 nebo 24 bitů v digitálním světě. Barevné rozlišení, také označované jako barevná hloubka, je počet bitů, které zaberou data, ve kterých jsou uloženy informace o barvě každého pixelu. Vyšší barevné rozlišení zvětšuje počet barev a detaily obrazu (zároveň zvětšuje i velikost souboru). Běžné barevné hloubky jsou 8, 16, 24 (true color) a 32 bitů.

Barevná hloubka také určuje kvalitu zobrazení viditelnou na obrazovce. Barevná hloubka je pouze číslo – počet bitů, které tvoří jeden pixel. Původní osmibitová verze dokázala zobrazit pouze 256 barev, ale jakmile došlo ke zdvojnásobení na šestnáct bitů, počet barev exponenciálně vzrostl na 65 536. Monitory eventuelně zobrazují 24 bitovou barevnou hloubku na pixel, čímž dosahují 16,8 milionů barev jako barevná televize, ale s mnohem lepšími detaily v dpi a s čtyřikrát více barevnými pixely.

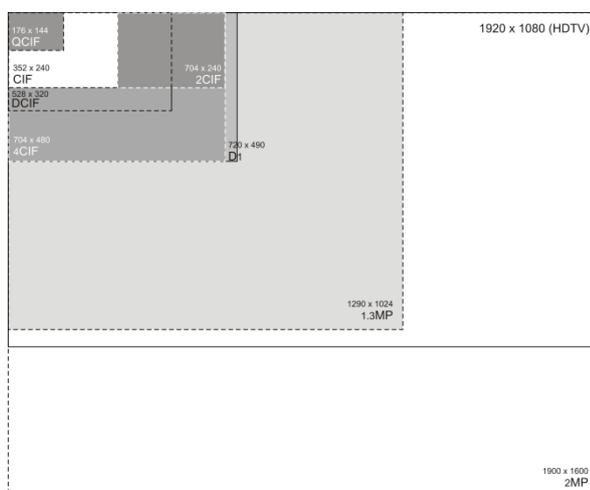
Většina dnešních monitorů umožňuje zobrazení 16 a 24 bitových barev. 32 bitové barvy jsou typicky 24 bitové s 8 bity „nebarevných“ metadat, která umožňují operačnímu systému a moderním grafickým kartám přídavnou funkcionalitu pro změnu rozlišení nebo použití dvou displejů. [2]

3.8 Rozlišení

Zatímco standardní analogový TV formát používá vertikální obdélníkové pixely, nový HDTV formát se skládá z menších čtvercových pixelů (podobě jako u PC monitorů). Na stejném místě, jako je jeden bod analogového NTSC pixelu, je čtyřiapůlkrát více digitálních pixelů (8 bit proti 32 bit), takže poskytuje čtyřiapůlkrát více detailů pro ostřejší obraz. Veškerá digitální rozlišení jsou měřena v těchto pixelech a všechny digitální obrázky, včetně tištěných materiálů, jsou založeny na pixelech. Ačkoliv tisk vyžaduje 300 dpi pro tisk v true color kvalitě, většina digitálních monitorů má buď 72 nebo 96 dpi (CRT nebo LCD), takže žádný digitální displej nevyžaduje stejné rozlišení jako tištěná stránka k dosažení stejné vizuální kvality. Na analogové CRT televizní obrazovce je rozlišení fixní, takže dpi se sníží podle velikosti obrazovky. Počet pixelů na šířku televizní obrazovky vydělený velikostí obrazovky určuje dpi. Například osm palců široká televize má rozlišení $560 \div 8 = 70$ dpi. Většina TV má 27“ úhlopříčku, tudíž rozlišení je kolem 26 dpi. [2]

3.8.1 Rozlišení videa

Tabulka č. 2 ukazuje množství dostupných rozlišení videa včetně více megapixelových kamer. Všechno digitální, co je spojeno se zobrazeným rozlišením a rozlišením displeje, se měří v pixelech. Například, jak ukazuje obrázek č. 5, 50“ LCD HDTV monitor je schopný zobrazit $1920 \times 1080p$, což je 1920×1080 pixelů v progresivním skenování. Typická bezpečnostní kamera s nízkým rozlišením generuje obraz 640×480 pixelů (VGA) a bezpečnostní kamera s vysokým rozlišením 2048×1539 (nebo také QXGA). Software pro sledování určuje, jak je video zobrazováno na HDTV monitoru, stejně jako DVD přehrávače automaticky přepínají obraz na celou video obrazovku. Avšak, protože rozlišení obrazovky je vyšší než rozlišení DVD, „zvětší“ ho z 480p tak, aby vyplnil obrazovku. Jedná se ale pouze o zvětšení velikosti pixelů. Pro simulaci HDTV (1080i) se dělá převzorkování. Převzorkování je interpolace mezi existujícími pixely pro získání odhadnuté hodnoty na novém umístění pixelu. Jinými slovy, DVD přehrávač, HDTV, projektor nebo jakékoliv digitální zobrazovací zařízení, které se použije, může mít převzorkovací schopnosti, které dokážou více než jen roztáhnout 720×480 DVD video na celou obrazovku. Převzorkování vlastně přepočítává mezery mezi pixely a vyplňuje prázdná místa tak, aby vytvořilo 1080i rozlišení.



Obrázek č. 5 - Digitální rozlišení a HDTV monitor

Zdroj: [2], upraveno

Existuje korelace mezi zobrazenými pixely a pixely, které se generují, aby byly zobrazeny. Ve filmech běžně strážci zákona zachytí portrét zločince na CCTV kameře a vytisknou ho jako $8'' \times 10''$ velkou lesklou fotografii, kterou si může každý prohlédnout. To, co ukazují, je ostrá fotografie, která se tváří jako by byla ve vysokém rozlišení. To není kouzlo moderní CCTV, ale kouzlo Hollywoodu. Pro vtištění obrázku zachyceného ve vysoké kvalitě ve velikosti, kterou vidíte na obrazovce, potřebujete více než třikrát tolik pixelů. Jinými slovy, 32 bitový, 2550×3300 pixelový obraz odpovídá $8,5'' \times 11''$ fotografii

(při 300dpi). Jakákoliv CCTV kamera, konvertovaná do digitálu přes enkodér, má stále maximální výstup 32 bitů, 704×480 pixelů a jen 96dpi. Aby tento obraz vypadal jako ve vysokém rozlišení, museli bychom konvertovat dpi z 96 na 300, ale jelikož máme pouze 337 920 pixelů zachycených v původním obrázku, jsme nuceni změnit proporce pro rozlišení tisku. Obrázek č. 6 zobrazuje příklad, jak vypadá tradiční obrázek 4CIF v NTSC v porovnání s megapixelovým obrazem. [2]

Název standardu	Rozlišení	Poměr stran	Počet pixelů na snímek
QCIF	176×144	4:3	25 344
VHS (NTSC)	320×480	4:3	115 200
CIF	352×240	4:3	76 800
DCIF	528×320	16:9	168 960
VGA	640×480	4:3	307 200
2CIF	704×240	4:3	168 960
4CIF	704×480	4:3	337 920
D1	720×480	16:9	345 600
SVGA	800×600	4:3	480 000
XVGA	1024×768	4:3	786 432
XVGA+	1280×1024	4:3	$1\,310\,720 = 1,3\text{Mpx}$
SVGA+	1600×1200	4:3	$1\,920\,000 = 1,9\text{Mpix}$
WUXGA	1920×1200	16:10	$2\,304\,000 = 2,3\text{Mpix}$
QXGA	2048×1536	4:3	$3\,145\,728 = 3\text{Mpix}$
HDTV 1080p	1920×1080	16:9	$2\,074\,000 = 2\text{Mpix}$
HDTV 720p	1280×720	16:9	922 000

Tabulka č. 2 – Přehled rozlišení

Zdroj: [2], upraveno



Obrázek č. 6 - (vlevo) vytisknutý obrázek z analogové bezpečnostní kamery v rozlišení 4CIF připojené k digitálnímu enkodéru (704×480). (vpravo) Vytisknutý obrázek stejné oblasti s použitím megapixelové kamery.

Zdroj: [2], upraveno

3.9 Barevná prezentace

Černobílý obraz (ve stupních šedi) může být zobrazen s použitím pouze jednoho čísla na prostorový vzorek. Toto číslo označuje jas nebo svítivost každé pozice vzorku, vyšší číslo značí světlejší vzorek. Jas je obvykle prezentován 8 bity na vzorek pro video aplikace. Vyšší „hloubky“ jasu (např. 12 bitů a více na určitý vzorek) je občas použito pro speciální aplikace (jako například pro digitalizaci rentgenových snímků).

Existuje několik alternativních systémů pro prezentaci barev. Dva nejběžnější způsoby barevné prezentace pro digitální obraz a videa jsou: RGB a YCrCb. [3]

3.9.1 RGB

V barevném prostoru RGB je každý pixel prezentován třemi čísly, indikujícími relativní hodnoty červené, modré a zelené. Toto jsou tři základní barvy světla: jakákoliv barva může být reprodukována kombinací různých poměrů červeného, zeleného a modrého světla. Jelikož tyto tři komponenty mají zhruba stejnou důležitost pro finální barvu, RGB systémy obvykle prezentují každou komponentu se stejnou přesností (a tudíž se stejným počtem bitů). [3]

3.9.2 YCrCb

RGB není nutně nejefektivnější zobrazení barev. Lidský vizuální systém je méně citlivý na barvy než na jas, nicméně, RGB barevné schéma neposkytuje jednoduchou cestu k využití tohoto poznatku. Jelikož všechny tři barevné složky jsou stejně důležité a jas je obsažen ve všech třech komponentech barvy, je efektivnější zobrazit barevný obraz oddělením jasu od informací o barvě.

Populární barevný prostor takového typu je Y:Cr:Cb. Y je komponent jasu, tj. černobílá verze barevného obrazu (Y je vážený průměr R, G a B),

$$Y = k_r R + k_g G + k_b B$$

kde „k“ jsou váhové faktory. Informace o barvě může být prezentována jako rozdíl barev nebo komponent, kde každá komponenta je rozdíl mezi R, G nebo B a jasnem Y:

$$C_r = R - Y \qquad C_b = B - Y \qquad C_g = G - Y$$

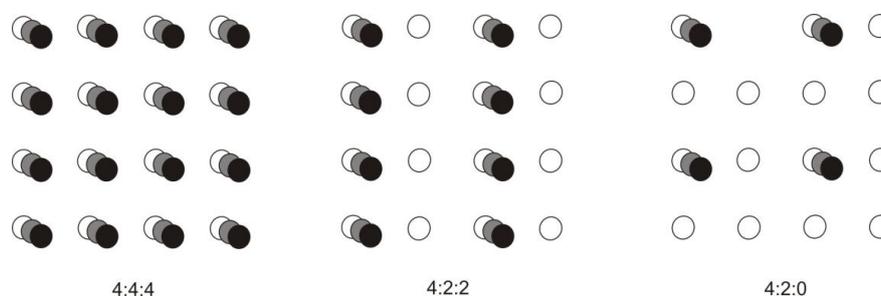
Úplný popis je dán Ypsilonem (jasová komponenta) a třemi barevnými rozdíly C_r , C_b a C_g , které prezentují „kolísání“ mezi intenzitou barvy a jasnem „na pozadí“ obrazu.

Tato prezentace má zjevně malý přínos, existují čtyři komponenty namísto tří. Nicméně se ukázalo, že hodnota $Cr + Cb + Cg$ je konstanta. To znamená, že je potřeba přenášet pouze dvě ze tří barevných komponent (třetí komponenta se vždy dá získat z prvních dvou). V $Y:Cr:Cb$ prostoru se přenáší pouze jas (Y) a červený a modrý barevný rozdíl (Cr a Cb).

Klíčovou výhodou $Y:Cr:Cb$ nad RGB je to, že komponenty Cr a Cb mohou být prezentovány nižším rozlišením než Y, protože oči jsou méně citlivé na barvy než na jas. Citlivost očí na jas redukuje množství dat potřebných pro barevné komponenty bez viditelného efektu na kvalitu obrazu: běžnému pozorovateli se zdá, že není zjevný rozdíl mezi RGB obrazem a $Y:Cr:Cb$ obrazem se sníženým rozlišením barevné komponenty.

Obrázek č. 10 ukazuje tři modely barevných rozdílů Cr a Cb. 4:4:4 znamená, že komponenty ($Y:Cr:Cb$) mají stejné rozlišení, a proto vzorek každého komponentu existuje na pozici každého pixelu. (Čísla indikují relativní vzorkovací poměr každého komponentu v horizontálním směru, tj. pro každé 4 jasové vzorky jsou 4 vzorky Cr a 4 vzorky Cb.) 4:4:4 vzorkování zachovává úplnou věrnost barevných komponent. Při 4:2:2 vzorkování mají barevné komponenty stejné vertikální rozlišení, ale pouze poloviční horizontální rozlišení (čísla udávají, že pro každé 4 jasové vzorky v horizontálním směru jsou 2 vzorky Cr a 2 vzorky Cb), pozice vzorků jsou znázorněny v obrázku. 4:2:2 video se používá pro vysoce kvalitní reprodukci barev.

4:2:0 znamená, že jak Cr, tak i Cb má poloviční horizontální i vertikální rozlišení jasu. 4:2:0 vzorkování je populární v aplikacích jako jsou video konference, digitální televize a DVD. Protože každý komponent rozdílu barvy obsahuje čtvrtinu vzorků Y komponentu, 4:2:0 video vyžaduje přesně polovinu vzorků než 4:4:4 (nebo R:G:B) video. [3]



- Y komponent
- Cr komponent
- Cb komponent

Obrázek č. 7 - Vzory barevných rozdílů
Zdroj: [3], upraveno

4 Kompresní metody

Pro záznam nebo přenos obrazu v digitální podobě je žádoucí zmenšená kapacita jeho dat. Na vysvětlení, jaký je vlastně objem dat přenášený nebo ukládaný z IP kamer, si připomeňme základní charakteristiku digitálního obrazu. Digitální obraz vzniká progresivním snímáním obrazu se snímkovací frekvencí 50Hz. Celý snímek je při rozlišení VGA tvořen maticí $640 \times 480 = 307\,200$ obrazových bodů (tabulka č. 3 zobrazuje počet obrazových bodů, datových toků apod. v jiných rozlišeních). Pokud bychom uvažovali o přenosu černobílého obrazu a pro každý obrazový bod maximálně o 256 úrovních jasu (od černé až po bílou), potřebovali bychom na zakódování jasové informace pro každý jasový bod 8 bitů. Jeden digitální snímek by v takovém případě zabral v paměti objem 2 457 600 bitů ($307\,200$ obrazových bodů vynásobeno 8 bity na zakódování jasu), to je přibližně 2,34 Megabitů (dále Mb). V případě, že by 8 bitové kódování úrovní jasu nebylo dostačující, je možné použít 10 bitové, které umožňuje zakódovat až 1024 úrovní jasu. V takovém případě by jeden digitální snímek měl objem přibližně 2,93 Mb.

Při zpracování barevného obrazu je objem dat třikrát větší než u černobílého. V případě, že bychom uvažovali o stejném počtu obrazových bodů (VGA), z nichž každý by byl složený ze třech základních barev (RGB) a jas každé barvy by byl kódovaný 8 bity, celkový objem dat jednoho snímku by byl přibližně 7 Mb. Při desetibitovém kódování jasu jednotlivých barev by datový tok byl 8,8 Mb. Toto jsou však pouze teoretické hodnoty, které jsou ve skutečnosti zpravidla menší.

Při přenášení barevného digitálního obrazu musí být přenosový systém schopný přenést 25 snímků za sekundu, to znamená při 10 bitovém kódování jasu přenos $25 \times 8,8 \text{ Mb} = 220 \text{ Mb}$. Této rychlosti dosahují pouze rychlé (tzv. gigabitové = 1000 Mb) počítačové sítě. Běžné sítě dosahují nižších přenosových rychlostí – lokální síť LAN využívá přenosovou rychlost 100 Mb/s, síť ADSL je schopna přenosové rychlosti maximálně 28 Mb/s, GPRS 80 kb/s, EDGE 200 kb/s .

V případě, že chceme zjistit potřebnou kapacitu záznamových médií můžeme provést podobné výpočty. Chceme-li ukládat plynulé barevné video (10 bitové kódování jasu a 25 snímků za sekundu), budeme potřebovat na jednu sekundu videa přibližně 27,5 Megabytů (dále MB). V současnosti disk s největší kapacitou (tedy 2 Terabyty - dále TB) by umožnil uchovat záznam o přibližné délce jedné hodiny. [4]

rozlišení	počet obrazových bodů (MPixelů)	Objem dat na snímek (RGB v Mb)	při 25 sn/s (Mb)	kapacita za 1 sekundu (MB)
VGA	640 × 480 = 0,31	8,79	219,73	27,47
PAL	704 × 576 = 0,41	11,60	290,04	36,25
SXGA	1280 × 1024 = 1,31	37,50	937,50	117,19
2Mpix	1600 × 1200 = 1,92	54,93	1 373,29	171,66
3Mpix	2048 × 1536 = 3,15	90,00	2 250,00	281,25
5Mpix	2560 × 1920 = 4,92	140,63	3 515,63	439,45

Tabulka č. 3 - Přehled nepoužívanějších rozlišení videa a výpočet jejich datových toků a požadavků na kapacitu pro desetibitové kódování jasu

Z výše uvedených úvah lze vyčíst, že na záznam či přenos digitálního obrazu je nutné použít vhodné kompresní metody, které zmenší celkový objem digitálních dat tak, aby na jejich záznam nebo přenos bylo možné použít v současnosti dostupné technologie.

Odstranění nadbytečných informací je základní myšlenkou komprese obrazu. Za nadbytečné informace lze ve statickém obraze považovat například snímek, kdy sousední obrazové body mají stejný jas i barvu (stejná informace se opakuje ve více obrazových bodech vedle sebe). To znamená, že při ukládání nebo přenosu obrazu je nutné přenášet informaci pouze o vybraných bodech, ne o všech. V případě pohyblivého obrazu patří mezi další nadbytečnost opakující se jednotlivé snímky v statické (nepohyblivé) scéně. Na přenos informace o nepohyblivé scéně je postačující přenést jeden úplný snímek a poté přenášet pouze informace o tom, že se nic nezměnilo, nebo informaci o změně. Tento způsob umožňuje výrazně redukovat objem přenášených a ukládaných dat tak, aby pozorovatel nezaznamenal jakoukoliv změnu kvality obrazu. [4]

4.1 Bezztrátové kompresní algoritmy

Bezztrátové kompresní algoritmy jsou určeny především na kompresi souborů, protože se předpokládá, že pro většinu souborů je nepřijatelná jakákoliv ztráta dat v důsledku komprese a dekomprese. V této kapitole jsou algoritmy uváděny proto, že jsou aplikovány jako součást ztrátových kompresních algoritmů používaných v dalších kapitolách.

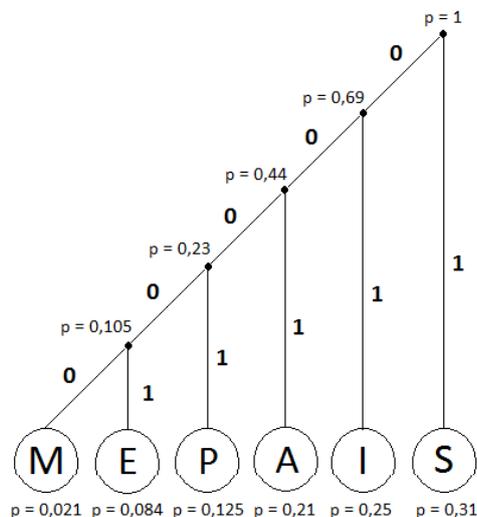
4.1.1 Huffmanovo kódování

Statické Huffmanovo kódování patří do skupiny algoritmů založených na četnosti znaků v kódovaných datech. Základní myšlenkou je, že znaky s velkou pravděpodobností výskytu mají nejkratší kód (malou bitovou složku) a že znaky s malou pravděpodobností

výskytu jsou kódovány s velkou kódovou složkou. Například je-li na vstupu datový soubor obsahující pouze znaky S, I, A, P, E, M s následujícími pravděpodobnostmi výskytu:

- Znak S se v souboru vyskytuje patnáctkrát, což je pravděpodobnost přibližně 31 %
- Znak I se v souboru vyskytuje dvanáctkrát, což je pravděpodobnost přibližně 25 %
- Znak A se v souboru vyskytuje desetkrát, což je pravděpodobnost přibližně 21 %
- Znak P se v souboru vyskytuje šestkrát, což je pravděpodobnost přibližně 12,5 %
- Znak E se v souboru vyskytuje čtyřikrát, což je pravděpodobnost přibližně 8,4 %
- Znak M se v souboru vyskytuje jednou, což je pravděpodobnost přibližně 2,1 %

Pro vytvoření binárního stromu je nutné seřadit znaky podle pravděpodobnosti výskytu jednotlivých znaků v textu. Binární strom se vytvoří tak, že znaky s nejmenší pravděpodobností výskytu se sečtou do skupin, jejichž pravděpodobnost je součtem jednotlivých pravděpodobností (viz obrázek č. 8). [5]



Obrázek č. 8 - Bitový strom Huffmanova kódování

Ve vytvořeném binárním stromě se hrany vystupující z jednotlivých uzlů stejným směrem označí logickou nulou a hrany vystupující opačným směrem se označí logickou jedničkou. Kódové kombinace pro příslušné znaky se provádějí od kořenového uzlu až po příslušný list binárního uzlu. V následující tabulce č. 4 jsou zobrazeny možné kódové kombinace pro uvedený příklad:

-Znak	Četnost	Pravděpodobnost	Kódové slovo
S	15	0,021	1
I	12	0,084	01
A	10	0,125	001
P	6	0,21	0001
E	4	0,25	00001
M	1	0,31	00000

Tabulka č. 4 - Kódové kombinace

Kompresce probíhá tak, že kompresní algoritmus čte postupně jednotlivé znaky vstupního datového souboru a na základě vytvořeného binárního stromu jim přiděluje jednotlivé posloupnosti bitů. Například slovo MISSISSIPPI by při standardním bitovém kódování vyžadovalo 3 bity na každý znak abecedy, což by znamenalo 11×3 , tedy 33 bitů na zakódování celého slova. Při použití Huffmanova kódování, za pomoci binárního stromu by bylo slovo MISSISSIPPI zakódováno následujícím způsobem: 00000 01 1 1 01 1 1 01 0001 0001 01. Pro zakódování tohoto slova Huffmanovým kódováním je dostačujících pouze 25 bitů oproti 33 bitům binárního kódování. Při použití 8 bitového kódování, které využívá například ASCII kód, by byl rozdíl v objemu dat ještě výrazně větší ve prospěch Huffmanova kódování (88 bitů proti 25 bitům). [6]

Pro dekompresi je důležité znát binární strom, který byl použitý při kódování. Algoritmus postupně čte komprimovaný soubor a postupuje po jednotlivých bitech od kořenového uzlu až k listu, představujícímu požadovaný znak. Po každém průchodu binárním stromem a po každém nalezení znaku pokračuje algoritmus opět od kořenového uzlu. Například: při posloupnosti bitů s hodnotou 0000001100001 bude dekomprimační algoritmus vyhledávat v binárním stromě následující: 00000 => M, 01 => I, 1 => S, 00001 => E, tedy slovo MISE.

Huffmanovo kódování je obvykle součástí jiných kompresních algoritmů, jako samostatná komprimační metoda se nepoužívá. [5]

4.1.2 RLE (Run-length Encoding)

Jedna z nejstarších metod komprese, která nalézá a redukuje posloupnosti opakujících se znaků v datovém toku, se nazývá metoda proudového kódování neboli RLE. Posloupnost znaků je znázorněna pouze speciálním znakem (indikátorem) a počtem opakování. Například posloupnost **httttyyyxxppppppp** lze uložit jako **h#t5#y4x#p7**, kde indikátor komprese představuje znak „#“. V uvedeném příkladě je řetězec osmnácti znaků nahrazen komprimovaným řetězcem o délce jedenácti znaků.

Úspěšnost komprese lze vyjádřit parametrem kompresního poměru, který v tomto případě je možné vypočítat ze vztahu $18 \div 11$ a který je roven hodnotě přibližně 1,64. Komprese je tím úspěšnější (účinnější), čím je větší kompresní poměr. Dalším možným parametrem je převrácená hodnota kompresního poměru – faktor komprese, který lze v tomto případě vypočítat jako $11 \div 18 = 0,61$. Tento parametr udává, jakou část zabírají údaje po kompresi z původního prostoru, tedy 61 %.

Je důležité uvědomit si, že každá posloupnost znaků je zakódována pomocí tří znaků (indikátorem, opakujícím se znakem a počtem opakování). Na zakódování posloupnosti kratší než tři znaky je tedy potřeba více prostoru, než by zabrala v souboru nekomprimovaném. Při komprimování souborů o délce právě tří znaků nedojde k prodloužení ani zkrácení souboru, ale ztrácí se čas nutný k dekompresi textu. Proto se tato komprese využívá až od určité délky opakujících se znaků v řetězci, v tomto případě od délky čtyř znaků.

Jinou variantu představuje zakomponování do výstupu počet opakování -1 a opakující se znak. Poté již není nutné rezervovat místo pro žádný další speciální znak, začátek komprimovaného úseku je indikován výskytem celého čísla. Například řetězec **httttyyyyxppppppp** by se zakódoval 0h4t3y0x6p. Tento postup je opět nevhodný pro proud dat složený z jednoho znaku, výsledná délka komprimovaného souboru by byla dvojnásobná.

Metoda proudového kódování je symetrická, rychlá, bezztrátová, často má však nízký kompresní poměr. RLE je efektivní spíše pro komprese souborů, kde nedochází k velkým změnám, například pro kompresi souborů zachycujících řeč nebo obrázky s malou hloubkou barev (obrázky s jednobarevným pozadím). Není příliš vhodná ke kompresi běžných textových souborů, kde se posloupnosti opakujících znaků často neopakují. [7]

4.2 Komprese statických obrázků

V této kapitole je představen standard JPEG. Tento standard umožňuje ztrátové i bezztrátové kódování statických obrázků a podporuje čtyři různé režimy provozu: sekvenční mód založený na DCT, progresivní mód založený na DCT, bezztrátový mód a hierarchický mód.

4.2.1 JPEG

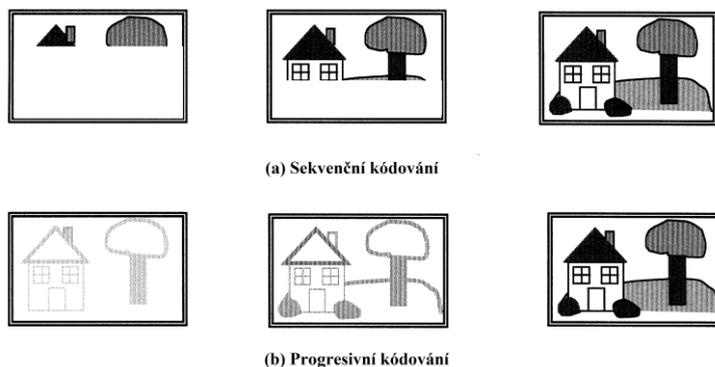
Důležitou funkcí datových kompresí je kódování statických obrázků. Když se digitalizuje analogový snímek nebo obraz, každý pixel je reprezentován pevným počtem bitů, které odpovídají určitému počtu úrovní šedi. V tomto nekomprimovaném formátu vyžaduje

digitalizovaný obraz velký počet bitů, aby mohl být uložen nebo přenášen. Kvůli limitované šířce přenosového pásma nebo kapacitě záznamového média se stala komprese nezbytnou. Od poloviny 80. let 20. století pracovaly ITU a ISO společně na vývoji mezinárodního standardu pro kompresi statických obrázků. JPEG se stal mezinárodním standardem v roce 1992. Standard JPEG umožňuje jak ztrátové, tak bezztrátové zakódování statických obrázků. Algoritmus pro ztrátové kódování je založen na DCT (Diskrétní kosinova transformace – viz [8]) kódovacím schématu. Ten je základem JPEGu a je to dostačující pro mnoho aplikací. Avšak pro vyhovění potřebám aplikací, které netolerují ztráty (například komprese lékařských snímků), je v něm bezztrátové kódovací schéma obsaženo také a je založeno na prediktivním kódovacím schématu. Z algoritmického pohledu obsahuje JPEG čtyři různé režimy provozu, jmenovitě: sekvenční mód založený na DCT, progresivní mód založený na DCT, bezztrátový mód a hierarchický mód.

V sekvenčním módu založeném na DCT je obraz nejprve rozdělen na bloky 8×8 pixelů. Bloky jsou zpracovány zleva doprava a shora dolů. Nakonec jsou vyčísleny DCT koeficienty, které jsou entropně zakódovány a na výstupu jsou součástí komprimovaných dat obrazu.

V progresivním módu založeném na DCT je proces rozdělování do bloků a dopředná DCT transformace stejná jako v sekvenčním DCT módu. Avšak v progresivním módu se kvantované DCT koeficienty nejdříve ukládají v zásobníku, než proběhne kódování. Uložené DCT koeficienty jsou poté kódovány několikanásobným procesem skenování. V každém průchodu jsou kvantované DCT koeficienty částečně zakódovány buď spektrální selekcí nebo postupnou aproximací. Při metodě spektrální selekce jsou kvantované DCT koeficienty rozděleny do více skupin podle cik-cak pořadí. Při každém průchodu je kódována specifická skupina. Při metodě postupné aproximace se nejdříve zakóduje specifický počet nejvýznamnějších bitů kvantovaných koeficientů a nejméně významné bity jsou poté kódovány v následujících průchodech.

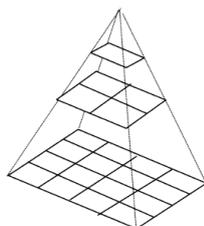
Rozdíl mezi sekvenčním kódováním a progresivním kódováním je zobrazen na obrázku č. 9. V sekvenčním kódování je obrázek zakódován po částech podle pořadí skenování, zatímco u progresivního kódování je obraz zakódován multiskenovacím procesem a při každém průběhu je převeden celý obraz na určitou úroveň kvality.



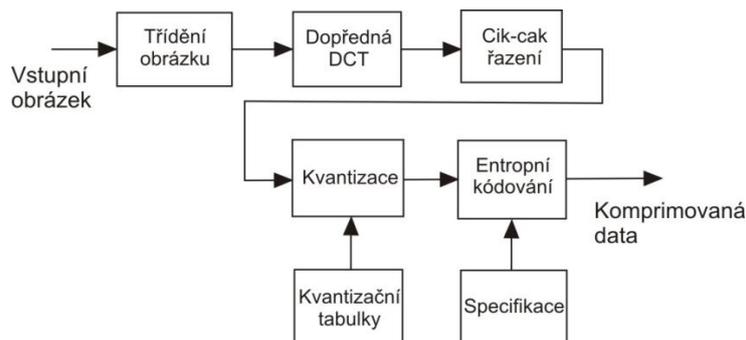
Obrázek č. 9 - Rozdíl mezi sekvenčním a progresivním kódováním
Zdroj: [9], upraveno

Jak již bylo uvedeno dříve, bezztrátového kódování je dosaženo pomocí prediktivního kódovacího schématu. V tomto schématu jsou použity tři sousední pixely k predikci pixelu, který bude kódován. Rozdíl předpovědi je entropně kódován s použitím Huffmanova nebo aritmetického kódování. Vzhledem k tomu, že predikce není kvantována, kódování je bezztrátové.

A konečně, v hierarchickém modelu je obraz nejdříve prostorově rozložen do několikavrstvé pyramidy, z čehož vznikne sekvence snímků (jak ukazuje obrázek č. 10). Tato sekvence je poté zakódována podle prediktivního schématu. Až na první snímek je prediktivní kódovací proces aplikován na rozdílové snímky, tj. rozdíly mezi snímkem, který má být kódován a snímkem prediktivně odpovídajícím. Je důležité poznamenat, že referenční snímek odpovídá předchozímu snímku, který by byl rekonstruován v dekodéru. Kódovací metoda pro rozdílový snímek může použít kódovací metodu založenou na DCT, bezztrátovou kódovací metodu, nebo procesy založené na DCT s finálním bezztrátovým procesem. Zmenšovací (downsampling) a zvětšovací (upsampling) filtry jsou použity v hierarchickém módu. Hierarchický kódovací mód poskytuje progresivní prezentaci podobnou progresivnímu módu založenému na DCT, ale je také použitelný v aplikacích, které mají požadavky na multirozklad. Hierarchický kódovací mód také poskytuje možnost progresivního kódování do finální bezztrátové fáze. [9]



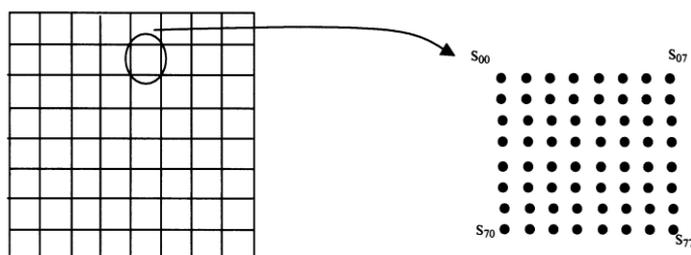
Obrázek č. 10 – Několikavrstvá pyramida v hierarchickém módu
Zdroj: [9], upraveno



Obrázek č. 11 - Blokový diagram sekvenčního kódovacího algoritmu založeného na DCT
Zdroj: [9], upraveno

4.2.1.1 Sekvenční kódovací algoritmus založený na DCT

Sekvenční kódovací algoritmus založený na DCT je základním algoritmem kódovacího standardu JPEG. Blokový diagram kódovacího procesu je zobrazen na obrázku č. 11. Jak ukazuje obrázek č. 12, digitalizovaná obrázková data jsou nejprve rozdělena do bloků 8×8 pixelů. Na každý blok je poté aplikována dvourozměrná dopředná DCT. [9]



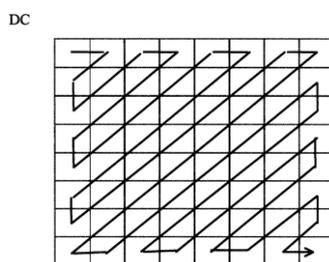
Obrázek č. 12 - Rozdělení do bloku 8×8 pixelů
Zdroj: [9], upraveno

SSSS	Hodnota DIFF	Dodatečné bity
0	0	-
1	-1, 1	0, 1
2	-3, -2, 2, 3	00, 01, 10, 11
3	-7, ..., -4, 4, ..., 7	000, ..., 011, 100, ..., 111
4	-15, ..., -8, 8, ..., 15	0000, ..., 0111, 1000, ..., 1111
5	-31, ..., -16, 16, ..., 31	00000, ..., 01111, 10000, ..., 11111
6	-63, ..., -32, 32, ..., 63
7	-127, ..., -64, 64, ..., 127
8	-255, ..., -128, 128, ..., 255
9	-511, ..., -256, 256, ..., 511
10	-1023, ..., -512, 512, ..., 1023
11	-2047, ..., -1024, 1024, ..., 2047

Tabulka č. 5 - Huffmanovo kódování DC koeficientů
Zdroj: [10], upraveno

Po aplikaci dvourozměrné DCT jsou každé nenulové kategorii přidány další bity do kódového slova pro identifikaci jeho unikátního rozdílu, který v rámci kategorie nastal. Počet dodatečných bitů je definován pomocí „SSSS“ a dodatečné bity jsou připojeny k nejméně významnému bitu Huffmanova kódu (nejdůležitější bit je jako první) podle následujícího pravidla (viz tabulka č. 5): pokud je hodnota difference kladná, přidají se „SSSS“ bity s nízkou prioritou z DIFF (rozdíl mezi kvantovaným DC koeficientem aktuálního bloku a bloku předchozího stejného komponentu), pokud je difference záporná, přidají se bity z DIFF-1.

Narozdíl od kódování DC koeficientů, kvantované AC koeficienty jsou seřazeny do cik-cak pořadí dříve, než jsou entropně kódovány. Pořadí tohoto průběhu zobrazuje obrázek č. 13. [9]



Obrázek č. 13 – Cik-cak pořadí DCT koeficientů
Zdroj: [9], upraveno

Podle cik-cak pořadí jsou kvantované koeficienty reprezentovány takto:

$$ZZ(0) = S_{q00}, ZZ(1) = S_{q01}, ZZ(2) = S_{q10}, \dots, ZZ(63) = S_{q77}.$$

Protože mnoho kvantovaných AC koeficientů se rovná nule, mohou být velmi efektivně kódovány s využitím proudu nul. Počet nul je identifikován nenulovými koeficienty. Osmi-bitový kód „RRRRSSSS“ je použit pro interpretaci nenulových koeficientů. Čtyři nejméně významné bity „SSSS“ definují kategorii pro hodnotu dalšího nenulového koeficientu v cik-cak sekvenci, která ukončuje tento proud nul. Čtyři nejvýznamnější bity „RRRR“ definují délku proudu nul v cik-cak sekvenci nebo pozici nenulového koeficientu v cik-cak sekvenci. Složená hodnota „RRRRSSSS“ je znázorněna na obrázku č. 14. Hodnota „RRRRSSSS“ = „11110000“ je definovaná jako ZRL, „RRRR“ = „1111“ představuje skupinu šestnácti nul a „SSSS“ = „0000“ představuje nulovou amplitudu. Proto je ZRL používán jako zástupce šestnácti nulových koeficientů, následovaných koeficientem nulové amplitudy, není to zkratka. V případě, že nulových koeficientů je více než patnáct, použije se více symbolů. Speciální hodnota „RRRRSSSS = „00000000“ se používá pro kódování konce bloku. Ke konci bloku dojde, pokud jsou zbylé koeficienty v bloku rovny nule. Vstupy označené „N/A“ nejsou definovány. Složená hodnota, RRRRSSSS, je poté kódována Huffmanovým

algoritmem. SSSS je vlastně číslo indikující „kategorii“ v Huffmanově kódovací tabulce. Hodnoty koeficientů pro každou kategorii jsou uvedeny v tabulce č. 6. [10]

		SSSS			
		0	1	2	9 10
RRRR	0	EOB	Složené hodnoty		
	.	N/A			
	.	N/A			
	15	ZRL			

Obrázek č. 14 - Dvoudimenzní pole hodnot pro Huffmanovo kódování
Zdroj: [9], upraveno

SSSS	Hodnota DIFF
1	-1, 1
2	-3, -2, 2, 3
3	-7, ..., -4, 4, ..., 7
4	-15, ..., -8, 8, ..., 15
5	-31, ..., -16, 16, ..., 31
6	-63, ..., -32, 32, ..., 63
7	-127, ..., -64, 64, ..., 127
8	-255, ..., -128, 128, ..., 255
9	-511, ..., -256, 256, ..., 511
10	-1023, ..., -512, 512, ..., 1023
11	-2047, ..., -1024, 1024, ..., 2047

Tabulka č. 6 - Huffmanova tabulka pro AC koeficienty
Zdroj: [10], upraveno

Každý Huffmanův kód je následován dalšími bity, které určují označení a přesný rozsah koeficientů. Stejně jako u tabulek DC kódu byly tabulky AC kódu také vyvinuty z průměrných statistik velké skupiny obrázků s osmibitovou přesností. Každá složená hodnota je reprezentována Huffmanovým kódem v tabulce AC kódu. Formát přidávaných bitů je stejný jako při kódování DC koeficientů. Hodnota SSSS udává počet přidávaných bitů potřebných pro specifikaci označení a přesný rozsah koeficientů. Přidávané bity jsou buď nižší SSSS bity ze $ZZ(k)$, když $ZZ(k)$ je kladné, nebo nižší SSSS bity ze $ZZ(k)-1$, když $ZZ(k)$ je záporné.

Jak bylo popsáno výše, Huffmanovo kódování je používáno jako způsob entropního kódování. Nicméně, může se použít i adaptivní aritmetická kódovací procedura. Stejně jako Huffmanova kódovací technika, také binární aritmetická kódovací technika je bezztrátová. Je možné převádět mezi oběma systémy bez FDCT nebo IDCT procesů. Protože toto

překódování je bezztrátový proces, neovlivňuje kvalitu rekonstruovaného snímku. Aritmetický kodér kóduje sérii binárních symbolů, jedniček a nul, kde každý symbol reprezentuje možný výsledek binárního rozhodování. Binární rozhodování zahrnuje volbu mezi pozitivními a negativními znaky, důležitosti nulové nebo nenulové, nebo jestli je konkrétní bit v sekvenci binárních čísel 0 nebo 1. [10]

4.2.1.2 Progresivní kódovací algoritmus založený na DCT

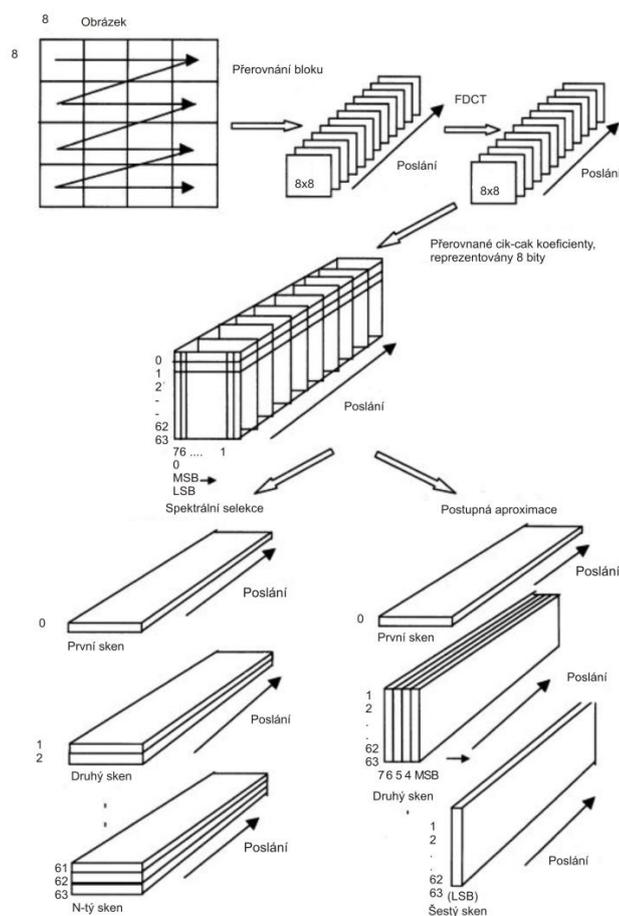
Při progresivním kódování založeném na DCT se vstupní obraz nejprve rozdělí do bloků 8×8 pixelů. Na každý blok se potom použije dvourozměrná 8×8 DCT. Transformovaná data DCT koeficientů jsou poté kódována vícenásobnými průchody. Při každém průchodu je kódována část transformovaných DCT koeficientových dat. Tato částečně kódovaná data mohou být rekonstruována pro získání celého obrázku se zhoršenou kvalitou obrazu. Kódovaná data každého dalšího průběhu zlepšují kvalitu obrazu rekonstruovaného obrázku do maximální kvality, které je dosaženo po zkompletování všech průběhů. Pro provádění progresivního kódování založeného na DCT se používají dvě metody. Jedná se o spektrální selekci a postupnou aproximaci.

Při metodě spektrální selekce se transformované DCT koeficienty nejprve přerovnají do cik-cak sekvence a poté se rozdělí do několika skupin. Frekvenční skupina je definována v hlavičce průběhu definováním počátečních a ukončujících indexů v cik-cak sekvenci. Skupina obsahující DC koeficienty se kóduje při prvním průběhu. V následujícím průběhu již není nutné, aby kódovací procedura postupovala podle cik-cak řazení.

Při metodě postupné aproximace je nejdříve DCT koeficientům snížena přesnost pomocí bodové transformace. Bodová transformace DCT koeficientů je aritmetický posun doprava o specifický počet bitů. Při kódování s využitím postupné aproximace jsou při prvním průběhu kódovány významné bity DCT koeficientů a každý následující průběh progresivně zlepšuje přesnost koeficientu o jeden bit. Tímto způsobem se pokračuje až k dosažení maximální přesnosti.

Principy spektrální selekce a postupné aproximace jsou zobrazeny na obrázku č. 15. Pro obě metody jsou kvantizované koeficienty kódovány při každém průchodu buď Huffmanovým, nebo aritmetickým kódem. Při spektrální selekci a při prvním průchodu postupné aproximace u obrázku je kódovací model AC koeficientu podobný tomu, který se používá při sekvenčním kódovacím módu založeném na DCT. Nicméně, Huffmanovy kódovací tabulky jsou rozšířeny tak, aby obsahovaly kódování konce skupiny (EOB). Pro

rozlišení end-of-band (konec skupiny) a end-of-block (konec bloku) se na konec skupiny (EOBn) přidává číslo „n“, které označuje délku skupiny. EOBn kódovací sekvence je poté definována následovně: každý EOBn je následován rozšiřujícím políčkem, které má minimální počet bitů potřebný pro specifikaci délky průběhu. Struktura konce skupiny umožňuje efektivně kódovat bloky, které mají pouze nulové koeficienty. Například, EOB průběh o délce pět značí, že aktuální blok a další čtyři bloky mají konec skupiny bez obsahu nenulových koeficientů. Struktura Huffmanova kódování následujících průběhů při postupné aproximaci daného obrázku je podobná jako kódovací struktura prvního průběhu tohoto obrázku. Každý nenulový kvantizovaný koeficient je popsán složenou bitovou hodnotou ve formě RRRSSSSS. Čtyři nejvýznamnější bity RRRR označují počet nulových koeficientů mezi současným koeficientem a posledním kódovaným koeficientem. Čtyři nejméně významné bity SSSS udávají velikost kategorie nulových koeficientů. Složená hodnota je poté kódována pomocí Huffmanova kódování. [9]

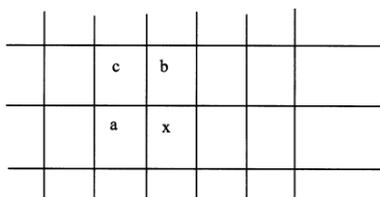


Obrázek č. 15 - Progresivní kódování pomocí spektrální selekce a postupné aproximace
Zdroj: [9], upraveno

Každý Huffmanův kód je poté následován dalšími bity, jeden bit je použit pro označení nenulových koeficientů a další bit je používán pro kódování oprav, kde nula znamená žádná oprava a jednička značí přidání jedničky pro dekódování velikosti koeficientu. Přestože předchozí techniky byly popsány s použitím Huffmanova kódování, pokládáme za důležité poznamenat, že se místo něj může použít i aritmetické kódování. [9]

4.2.1.3 Bezztrátový kódovací model

Při bezztrátovém kódovacím modelu je kódovací metoda založena na prostorovém kódování namísto kódování založeného na DCT. Nicméně, kódovací metoda je odvozena z metody pro kódování DCT koeficientů v sekvenčním kódovacím módu založeném na DCT. Každý pixel je kódován prediktivní kódovací metodou, kde se předvídaná hodnota získává z jednoho ze tří jednorozměrných nebo z jednoho ze čtyř dvourozměrných predikátorů, které jsou znázorněny na obrázku č. 16. [11]



Obrázek č. 16 – Prostorová vazba mezi pixelem, který bude kódován, a třemi dekódovanými sousedy
Zdroj: [11]

Na obrázku č. 16 je pixel, který se bude kódovat, označen jako „x“ a tři sousedi jsou označeni jako a, b, c. Predikovaná hodnota x (P_x) je získána ze tří sousedů a, b, c jedním ze sedmi způsobů, popsaných v tabulce č. 7.

Způsob	Predikce
0	Bez predikce (hierarchický mód)
1	$P_x = a$
2	$P_x = b$
3	$P_x = c$
4	$P_x = a+b-c$
5	$P_x = a+b((b-c)/2)$
6	$P_x = b+((a-c)/2)$
7	$P_x = (a+b)/2$

Tabulka č. 7 - Predikátory pro bezztrátové kódování
Zdroj: [11], upraveno

V tabulce č. 7 se hodnota 0 používá pouze pro diferenční kódování v hierarchickém kódovacím módu. Volby 1, 2 a 3 jsou jednorozměrné predikce a 4, 5, 6 a 7 jsou dvourozměrné predikce. Každá predikce je provedena s přesností na celé číslo a bez

překročení meze pro podtečení nebo přetečení vstupních limitů. Pro dosažení bezztrátového kódování jsou difference predikcí kódovány buď Huffmanovým nebo aritmetickým kódováním. Hodnoty rozdílů predikcí mohou být od 0 do 2^{16} pro osmibitové pixely. Pro kódování diferencí predikcí jsou použity Huffmanovy tabulky vyvinuté pro kódování DC koeficientů v sekvenčním módu založeném na DCT společně s jedním dalším vstupem. [11]

4.2.1.4 Hierarchický kódovací mód

Hierarchický kódovací mód poskytuje progresivní kódování podobné progresivnímu kódovacímu módu založeném na DCT, ale má větší funkcionalitu. Tato funkcionalita je směřována na aplikace s požadavky na vyšší rozlišení. V hierarchickém kódovacím módu je nejprve vstupní obraz rozdělen do sekvence obrázků (framů), podobně jako pyramida zobrazená na obrázku č. 10. Každý frame se získává při převzorkovacím procesu. První frame (nejnižší rozlišení) je kódován jako bezrozdílný frame. Následující framy se kódují jako rozdílné, kde rozdíl bere ohled na předchozí kódovaný frame. První frame může být kódován pomocí metod sekvenčního kódování založeného na DCT, spektrální selekcí, metodou progresivního kódování nebo bezztrátovým kódováním, ať s Huffmanovým, nebo aritmetickým kódem. Nicméně, v rámci obrazu jsou rozdílné framy kódovány buď metodou založenou na DCT, bezztrátovou metodou nebo procesem založeným na DCT s finálním bezztrátovým kódováním. Všechny framy v obraze musí používat stejné entropní kódování, buď Huffmanovo, nebo aritmetické. [12]

4.2.1.5 Souhrn

V této kapitole byl představen standard pro kódování statických obrázků – JPEG. JPEG standard obsahuje čtyři kódovací módy: sekvenční mód založený na DCT, progresivní mód založený na DCT, bezztrátový mód a hierarchický mód. Kódovací metoda založená na DCT je bezpochyby ta, která patří mezi nejpoužívanější metody. Nicméně, bezztrátové kódovací módy JPEG standardu, které využívají prostorové prediktivní kódovací procesy, mají také mnoho zajímavých okruhů uplatnění. Pro každé kódování může být implementováno entropní kódování buď s Huffmanovým kódováním, nebo aritmetickým kódováním. JPEG byl široce adoptován pro spoustu aplikací.

4.3 Komprese videa

Existuje mnoho kompresí videa, avšak z hlediska IP kamerových systémů jsou podstatné jen dvě a to komprese MPEG-x a H.26x.

4.3.1 MPEG

Skupina odborníků MPEG (Moving Picture Experts Group - Skupina expertů pro pohyblivý obraz) byla zřízena za účelem vytvoření standardu digitálního videa. Přestože metoda komprese je velmi podobná JPEG kompresi, metoda přenosu je odlišná, což snížilo šířku pásma a skladovací prostor na čtvrtinu oproti MJPEGu:

- MPEG-1 (ISO/IEC 11172) byl jako první vydán v roce 1993 a stále se používá v mnoha multimediálních aplikacích.
- MPEG-2 (ISO/IEC 13818) je standard používaný pro filmy na DVD, přenášející data vysokou rychlostí 6 MB za sekundu, což ho dělá velice neskladným pro sledovací systémy.
- MPEG-3 (nebo lépe známý jako MP3) se používá jako audio kodek.
- MPEG-4 (ISO/IEC 14496) se stal standardem pro streamování digitálního videa, protože poskytuje flexibilitu a bere v potaz šířku pásma a požadavky na skladování. MPEG-4 dokáže přenášet 6 MB za sekundu stejně jako MPEG-2, ale dokáže také poskytnout kvalitní video při 512kbps. [13]

4.3.1.1 MPEG-1

První standard, vyvinutý skupinou expertů pro pohyblivý obraz, známý pod jménem MPEG-1, byl navržen tak, aby poskytoval kompresi videa a zvuku pro ukládání a přehrávání z CD-ROMů. CD-ROM přehrávaný „jednoduchou rychlostí 1×“ měl přenosovou rychlost 1,4 Mbps. MPEG-1 měl za cíl komprimovat video a zvuk do přenosové rychlosti 1,4 Mbps v kvalitě srovnatelné s VHS kazetami. Cílovým produktem bylo „videoCD“ a standardní CD obsahující až 70 minut videa a zvuku. Video CD nikdy nemělo komerční úspěch, zlepšení kvality videa nebylo tak zjevné, aby donutilo spotřebitele vyměnit jejich VHS rekordéry, a maximální délka 70 minut vytvářela obtěžující pauzu v dlouhých filmech. Nicméně, MPEG-1 je důležitý ze dvou důvodů: získal si velikou oblibu u dalších aplikací pro ukládání a přenos videa (včetně ukládání na CD-ROM, součást interaktivních aplikací nebo přehrávání videa přes internet) a jeho funkce jsou použity v populárním standardu MPEG-2. [14]

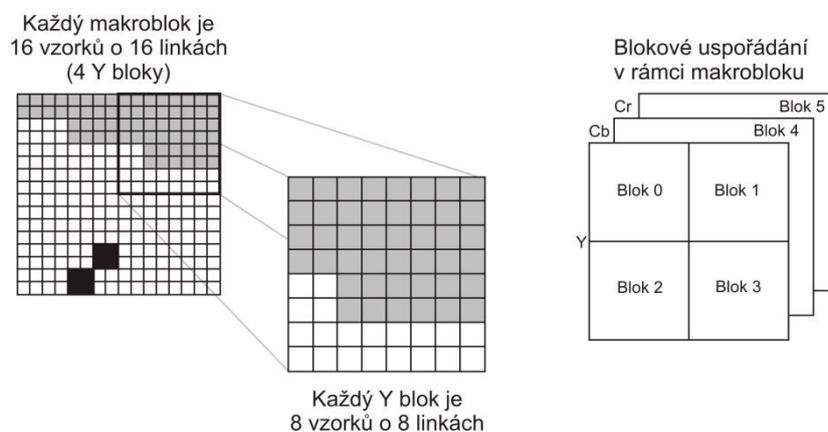
4.3.1.1.1 Funkce MPEG-1

Video-vstup do MPEG-1 kodéru je ve formátu: 4:2:0 Y:Cr:Cb (viz kapitola 3.10.2) s typickým prostorovým rozlišením 352×288 nebo 352×240 pixelů. Každý snímek videa je zpracován v makroblocích, odpovídajících oblasti 16×16 pixelů v zobrazovaném snímku.

Tato oblast je tvořena 16×16 vzorky jasu, 8×8 Cr vzorky a 8×8 Cb vzorky (protože Cr a Cb mají poloviční vertikální a horizontální rozlišení jasových komponentů). Makroblok se skládá ze šesti 8×8 bloků: čtyř jasových (Y) bloků, jednoho Cr bloku a jednoho Cb bloku (viz obrázek č. 17).

Každý snímek videa je kódován pro vytvoření kódovaného obrazu. Existují tři základní typy: I-obrazy, P-obrazy a B-obrazy. (Standard specifikuje čtvrtý typ obrazu, D-obraz, ale ten je v aplikacích využíván velice zřídka.)

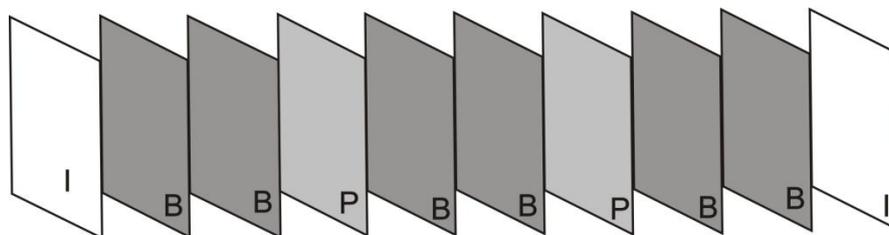
I-obrazy jsou bez jakékoliv pohybově kompenzované predikce (podobným způsobem jako výchozí JPEG). I-obraz se používá jako reference pro další predikované obrazy (P-obrazy a B-obrazy popsané níže). [13]



Každý Y blok je 8 vzorků o 8 linkách
Obrázek č. 17 - Struktura makrobloku
 Zdroj: [13], upraveno

P-obrazy jsou kódovány s použitím pohybově kompenzované predikce z referenčního obrazu (P-obraz nebo I-obraz předcházející aktuálnímu P-obrazu). P-obraz může být sám použit jako reference pro další predikované obrazy (P a B-obrazy).

B-obrazy jsou kódovány s použitím pohybově kompenzované predikce ze dvou referenčních obrazů, P- nebo I-obrazů před a za aktuálním B-obrazem. Dva pohybové vektory jsou vygenerovány pro každý makroblok v B-obrazu: jeden odkazující na odpovídající oblast v předchozím referenčním snímku a další odkazující na odpovídající oblast na budoucím referenčním obrázku. Makroblok pohybově kompenzované predikce může být zformován třemi způsoby: dopředná (budoucí) predikce s použitím dopředného vektoru, zpětná predikce s použitím zpětného vektoru nebo obousměrná predikce (kde reference pro predikci je tvořena průměrováním referencí dopředné a zpětné predikce). B-obrazy samotné nejsou používány jako predikční reference pro jakékoliv další predikované snímky. [13]

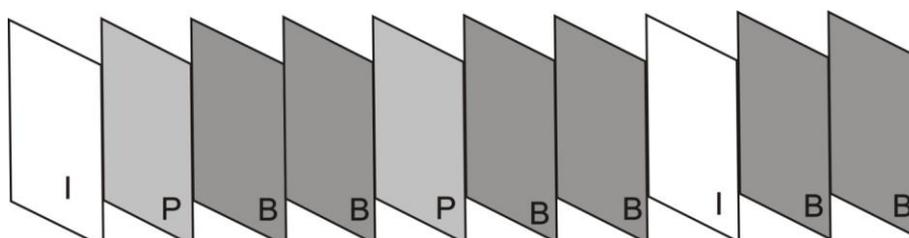


Obrázek č. 18 - MPEG-1 skupina obrázků (GOP - group of pictures)

Zdroj: [10], upraveno

Obrázek č. 18 znázorňuje typickou řadu I-, B- a P-obrazů. Pro kódování B-obrazu musí být dva sousední I- nebo P-obrazy zpracovány a uloženy v predikční paměti. Předtím, než snímek B z obrázku č. 18 může být kódován, jeho dva „kotvící“ snímky I a P musí být zpracovány a uloženy, tj. snímky I a P musí být zpracovány dříve, než snímky B a B mohou být kódovány. V tomto případě bude zpoždění minimálně tři snímky během kódování (snímky I, B1 a P musí být uloženy, než se zpracuje snímek B2) a toto zpoždění bude větší, pokud se použije více B-obrazů.

Aby se zmenšilo zpoždění u dekodéru, kódované obrázky jsou před přenosem seřazeny tak, že všechny „kotvící“ obrázky potřebné k dekódování B-obrazu jsou umístěny před B-obrazem. Obrázek č. 19 ukazuje stejný sled snímků seřazených před přenosem. P snímek je nyní umístěn před B a B snímky. P je dekódován ihned po I a je uložen dekodérem. Nyní mohou být dekódovány a zobrazeny B a B (protože jejich predikční reference, I a P, jsou obě k dispozici), poté se zobrazí P. Je zde registrováno maximálně jednosnímkové zpoždění mezi dekódováním a zobrazením, dekodér může ukládat pouze dva dekódované snímky. Toto je jeden příklad asymetrie mezi kódérem a dekodérem: zpoždění a ukládání v dekodéru jsou značně nižší než v kódéru. [13]



Obrázek č. 19 - MPEG-1 skupina obrázků - seřazení kotvících obrázků

Zdroj: [13], upraveno

I-obrazy představují vhodné obnovovací body v kódovaném bitovém streamu. Protože jsou kódovány bez predikce, I-obraz může být dekódován nezávisle na ostatních kódovaných obrazech. Kódování bez predikce podporuje náhodný přístup dekodéru (dekodér může začít dekódovat bitový stream na jakékoliv pozici I-obrazu) a odolnost proti chybám. Nicméně, I-obraz má nízkou efektivitu komprese, protože se u něj nepoužívá žádná dočasná predikce.

P-obrazy poskytují lepší efektivitu komprese díky pohybově kompenzované predikci a můžou být použity jako predikční reference. B-obrazy mají nejvyšší efektivitu komprese ze všech tří typů. [13]

4.3.1.1.2 Syntaxe MPEG-1

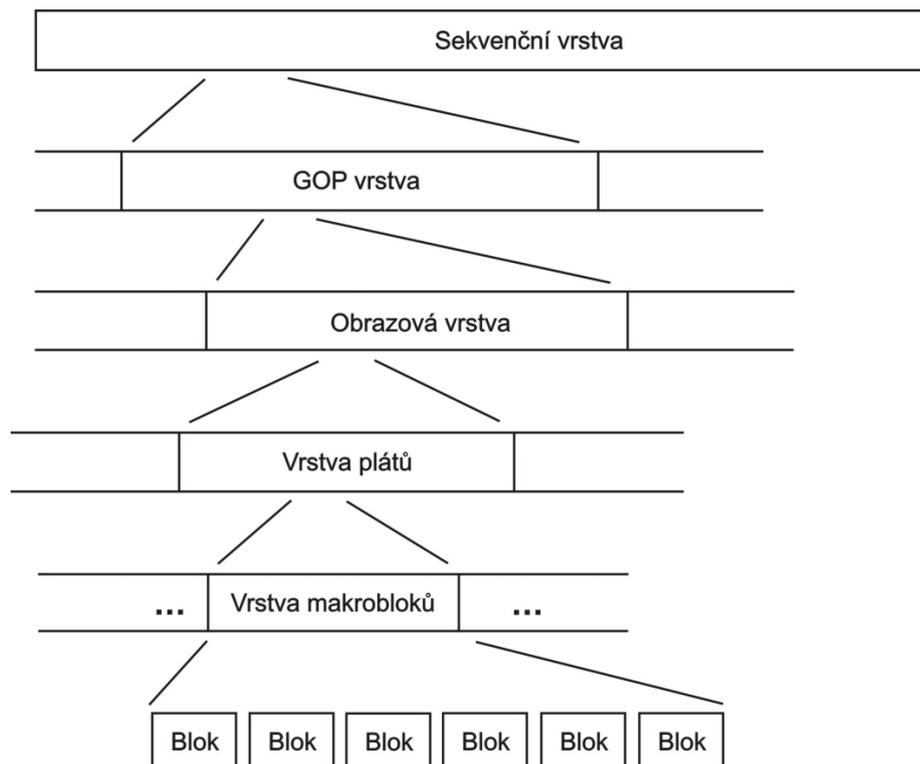
Syntaxe MPEG-1 kódované video-sekvence tvoří hierarchii zobrazenou na obrázku č. 20. Úrovně vrstev hierarchie jsou následující:

Sekvenční vrstva - Může odpovídat kompletnímu kódovanému videu. Sekvence začíná hlavičkou sekvence, která popisuje určité klíčové informace o kódované sekvenci včetně rozlišení obrazu a snímkovací frekvenci. Sekvence se skládá ze série skupin obrázků (GOP), neboli další vrstvy hierarchie.

GOP vrstva - GOP je jeden I-obraz, následovaný sérií P- a B-obrazů (např. obrázek č. 18). V obrázku č. 18 obsahuje GOP 9 obrazů (jeden I, dva P a šest B obrazů), je ale možných více druhů GOP struktur, například:

- a) Každý GOP obsahují pouze jeden I-obraz, tj. není použita žádná pohybově kompenzovaná predikce (podobně jako u MJPEGu).
- b) GOP obsahuje pouze I- a P-obrazy, tj. není použita dvousměrná predikce, efektivita komprese je relativně slabá, ale složitost je nízká (jelikož B-obrazy se generují složitěji).
- c) Velký GOP - počet I-obrazů v kódovaném streamu je malý, a proto získáváme velice efektivní kompresi. Nicméně, máme málo synchronizačních bodů. Není tedy ideální pro náhodný přístup a odolnost proti chybám.
- d) Malý GOP - obsahuje vysoký poměr I-obrazů, a proto je efektivita komprese nízká, avšak je zde možnost pro obnovení synchronizace. [13]

Obrazová vrstva – obraz definuje jeden kódovaný snímek. Hlavička obrazu definuje typ kódovaného obrazu (I, P, B) a dočasnou referenci, která definuje, kdy bude obraz zobrazen ve vztahu k ostatním obrazům v sekvenci.

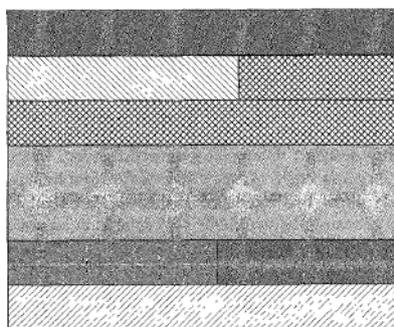


Obrázek č. 20 - MPEG-1 hierarchie

Zdroj: [13], upraveno

Vrstva plátů (slices) – obraz je tvořen několika pláty a každý z nich obsahuje celý počet makrobloků. U MPEG-1 není žádné omezení ohledně velikosti nebo uspořádání plátů v obraze vyjma toho, že pláty by měly pokrýt obraz v rastrovém uspořádání. Obrázek č. 21 ukazuje možné seskupení: každý vyšrafovaný region je jeden plát.

Plát začíná hlavičkou, která definuje jeho pozici. Každý plát může být dekodován nezávisle na ostatních plátech v rámci jednoho obrazu, což napomáhá dekodéru objevit chyby přenosu. Když dojde k chybě v rámci plátu, dekodér může začít dekodovat znovu od hlavičky dalšího plátu. [13]



Obrázek č. 21 - Příklad MPEG-1 plátů

Zdroj: [10], upraveno

Vrstva makrobloků – plát je tvořen určitým počtem makrobloků a každý makroblok se skládá ze šesti bloků – obrázek č. 17. Hlavička makrobloku popisuje typ makrobloku,

pohybové vektory a definuje, které 8×8 bloky vlastně obsahují kódovaná transformovaná data.

Vrstva bloků – blok obsahuje různě dlouhý kód, reprezentující kvantované transformované koeficienty v 8×8 bloku. Každý DC koeficient (DCT koeficient [0,0]) je kódován diferencially od DC koeficientu předchozího kódovaného bloku tak, aby se využilo toho, že sousední bloky mají tendenci mít velmi podobné DC hodnoty. AC koeficienty (všechny ostatní koeficienty) jsou kódovány jako páry, kde „proud“ indikuje počet předchozích nulových koeficientů a „úroveň“ označuje hodnotu nenulových koeficientů. [15]

4.3.1.2 MPEG-2

Dalším důležitým uplatněním kódovaného videa (po ukládání na CD-ROM) byla digitální televize. Za účelem poskytnutí vylepšené alternativy k analogové televizi bylo zapotřebí několik klíčových funkcí v kódovacím algoritmu videa, algoritmus musel efektivně podporovat větší rozlišení snímků (běžně 720×576 nebo 720×480 pixelů) a kódování prokládaného videa. MPEG-1 byl primárně navržen tak, aby podporoval progresivní video, kde je každý snímek jako samostatná jednotka v rastrovém uspořádání. U rozlišení televizní kvality poskytuje prokládané video (prokládané snímání je popsáno v kapitole 3.6) plynulejší obraz. Jelikož tato dvě pole jsou zachycena v různém okamžiku (běžně 1/50 nebo 1/60 vteřiny), může být dosaženo lepšího výkonu kódováním těchto polí odděleně.

MPEG-2 se skládá ze tří základních sekcí: Video (viz. níže), Audio (založeno na kódování audia u MPEG-1) a Systémů (definující detailněji než systémy MPEG-1, multiplexování a přenos kódovaného audio/video streamu). MPEG-2 video je (téměř) nadřazené MPEG-1 videu, tj. většina MPEG-1 video sekvencí by měla být dekódovatelná MPEG-2 dekodérem. Hlavní vylepšení přidaná MPEG-2 standardem jsou:

a) Efektivní kódování videa v televizní kvalitě

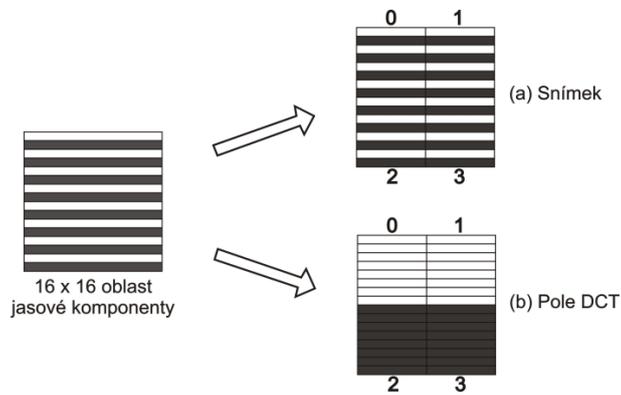
Nejdůležitější využití MPEG-2 je přenos digitální televize. Hlavní funkce MPEG-2 jsou optimalizované pro efektivní kódování televizního rozlišení při datovém toku okolo 3 - 5 Mbps.

b) Podpora pro kódování prokládaného videa

MPEG-2 video má několik funkcí, které podporují flexibilní kódování prokládaného videa. Dvě políčka, která tvoří celý prokládaný snímek, mohou být kódována jako samostatné obrázky (obrazová pole). P- a B- obrazová pole mohou být predikována z polí v jiném snímku

nebo z jiného pole aktuálního snímku, eventuálně může být s těmito dvěma poli zacházeno jako s jedním obrázkem se vzorky jasu v každém makrobloku. DCT kódování snímků je podobné struktuře MPEG-1, kde každý ze čtyř jasových bloků obsahuje alternativní řádky z obou polí. U DCT kódování polí obsahují horní dva jasové bloky pouze vzorky z horního pole a spodní dva jasové bloky obsahují vzorky ze spodního pole. Obrázek č. 22 ukazuje tyto dvě kódovací struktury.

V DCT kódování polí, horní a dolní 16×8 oblasti vzorků makrobloku mohou být pohybově kompenzovány nezávisle: z toho důvodu má každá z těchto oblastí svůj vektor (nebo dva vektory v případě B-obrazu). Tento 16×8 kompenzační režim může zlepšit výkon, protože obraz polí má pouze poloviční vertikální rozlišení obrazu snímku, a tudíž tu můžeme zaznamenat znatelnější rozdíly v pohybu mezi horní a dolní polovinou každého makrobloku.



Obrázek č. 22 - DCT kódování snímků (a) a polí (b)
Zdroj: [13], upraveno

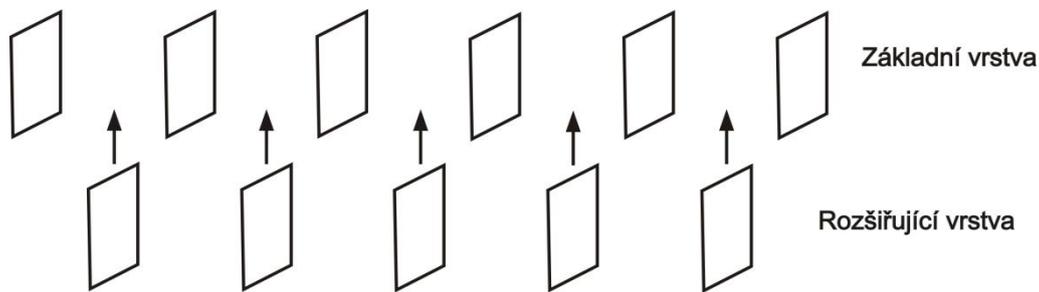
c) Škálovatelnost

Progresivní módy JPEGu (popsané dříve) jsou formy škálovatelného kódování. Škálovatelné kódování streamu se skládá z několika vrstev: základní vrstvy a jedné nebo více rozšiřujících vrstev. [13]

Prostorová škálovatelnost – Je analogická k hierarchickému kódování JPEG standardu. Základní vrstva je kódována při nízkém prostorovém rozlišení a každá rozšiřující vrstva, když se přidá k základní vrstvě, zvyšuje progresivně prostorové rozlišení.

Dočasná škálovatelnost - Základní vrstva je kódována při nízkém dočasném rozlišení (datovém toku) a rozšiřující vrstvy jsou kódovány při zvýšení datového toku (obrázek č. 23). Jedním využitím tohoto módu je stereoskopické kódování videa: základní vrstva poskytuje monoskopický „pohled“ a rozšiřující vrstva poskytuje stereoskopický kompenzační pohled. Zkombinováním těchto dvou vrstev se dekóduje plně stereoskopický obraz.

SNR škálovatelnost – Podobným způsobem jako u módu postupné aproximace u JPEGu je základní vrstva snímku kódována při „obyčejné“ vizuální kvalitě (s vysokou kompresí). Každá rozšiřující vrstva (po přidání k základní vrstvě) zvyšuje kvalitu videa. [13]



Obrázek č. 23 - Dočasná škálovatelnost
Zdroj: [13], upraveno

Dělení dat – Kódovaná sekvence je rozdělena do dvou vrstev. Základní vrstva obsahuje nejdůležitější komponenty kódované sekvence, jako například informace o hlavičce, pohybových vektorech a (volitelně) nízkofrekvenčních transformačních koeficientech. Rozšiřující vrstva obsahuje zbylá kódovaná data (obvykle méně důležitá pro úspěšné dekódování).

Tyto škálovatelné módy mohou být použity několika způsoby. Dekodér může dekódovat aktuální program při běžném rozlišení (720×576 , 25 nebo 30 snímků za sekundu) dekódováním pouze základní vrstvy, zatímco dekodér s vysokým rozlišením může dekódovat jednu nebo více rozšiřujících vrstev pro zvýšení dočasného nebo prostorového rozlišení. Vícenásobné vrstvy mohou podporovat současné dekódování základními a pokročilými dekodéry. Přenos základních a rozšiřujících vrstev je obvykle efektivnější než kódování a posílání samostatných streamů při nižších a vyšších rozlišeních.

Základní vrstva je nejdůležitější pro poskytnutí vizuálně akceptovatelného dekódovaného obrazu. Přenosové chyby v základní vrstvě mohou mít katastrofální účinky na kvalitu obrazu, zatímco chyby v rozšiřujících vrstvách mají relativně malý dopad na kvalitu. Chráněním základní vrstvy (například použitím odděleného přenosového kanálu s nízkou frekvencí chyb nebo přidáním korekčního kódování) může být vysoká kvalita obrazu udržena i v případě, že při přenosu dojde k chybě. [13]

d) Profily a úrovně

Většina aplikací potřebuje jen limitovanou podmnožinu ze široké škály funkcí podporovaných MPEG-2. S cílem podpořit součinnost pro jisté „klíčové“ aplikace (jako je digitální TV) obsahuje standard sadu doporučených profilů a úrovní, které definují

podmnožinu funkcionalit MPEG-2. Každý profil definuje sadu schopností, ty důležité jsou následující:

- Jednoduché: 4:2:0 vzorkování, jen I- a P-obrazy jsou povoleny, složitost je nízká za cenu špatné komprese.
- Hlavní: obsahuje všechny klíčové schopnosti MPEG-2, včetně B-obrazů a podpory prokládaného videa, používá se 4:2:0 vzorkování.
- SNR: jako „hlavní“ profil s výjimkou toho, že je přidána rozšiřující vrstva pro zlepšení kvality obrazu.
- Prostorový: jako „SNR“ profil s výjimkou toho, že prostorová škálovatelnost může být použita tak, aby poskytla rozšiřující vrstvy lepší kvality.
- Vysoký: jako „prostorový“ profil s přidáním podpory pro 4:2:2 vzorkování.

Každá úroveň definuje prostorové a dočasné rozlišení:

- Nízká: až do 355×288 rozlišení snímku a maximálně 30 snímků za sekundu.
- Hlavní: až do 720×576 rozlišení snímku a maximálně 30 snímků za sekundu.
- Vysoká: až do 1920×1152 rozlišení snímku a maximálně 60 snímků za sekundu.

Skupina MPEG původně plánovala vydat další standard, MPEG-3, pro podporu HDTV aplikací. Nicméně, jakmile bylo jisté, že MPEG-2 syntaxe si s touto aplikací dokáže adekvátně poradit, práce na tomto standardu byly ukončeny a tudíž není žádný MPEG-3 standard.

MPEG-2 oproti MPEG-1 standardu nabízí, to že pláty v MPEG-2 obrazu jsou omezeny tak, že se nemůžou překrývat z jedné řady makrobloků do další (narozdíl od MPEG-1, kde mohl plát zabrat i více řad makrobloků). D-obrazy v MPEG-1 byly považovány za nepřínosné a nejsou v MPEG-2 podporovány. [13]

4.3.1.3 MPEG-4

MPEG-1 a MPEG-2 standardy se zabývají celými snímky videa, kde je každý snímek kódovaný jako jedna jednotka. MPEG-4 standard byl vyvinut s cílem rozšířit schopnosti dřívějších standardů několika cestami.

Podpora aplikací s nízkým datovým tokem - MPEG-1 a MPEG-2 jsou rozumně efektivní pro aplikace nad 1 Mbps. Avšak, mnoho vznikajících aplikací (zejména aplikace založené na internetu) vyžadují mnohem nižší přenosový poměr a MPEG-1 a MPEG-2 nepodporují tak efektivní kompresi při nízkých přenosových poměrech (desítky kbps nebo méně).

Podpora pro objektově zaměřené kódování - Možná nejzásadnější posun u MPEG-4 standardu byl k objektově a obsahově založenému kódování, kde video může být zpracováno spíše jako sada objektů v popředí a v pozadí, než jako sada obdélníkových snímků. Tento typ kódování otevírá širokou škálu možností, jako je nezávislé kódování různých objektů na scéně, znovupoužití komponentů scény, kompozice (video objekty z několika zdrojů jsou zkombinovány do jedné scény) a vysoký stupeň interaktivity. [13]



Obrázek č. 24 - Video scéna ukazující více video objektů
Zdroj: [13], upraveno

Video scéna (VS - sekvence video snímků) je tvořena několika video objekty (VO). Například VS na obrázku č. 24 se skládá z VO pozadí a dvou předních VO. MPEG-4 poskytuje nástroje, které umožňují, aby každé VO bylo kódováno nezávisle. Celá scéna může být kódována jako jedna obdélníková VOP (video object plane), což představuje ekvivalent obrázku v MPEG-1 a MPEG-2 terminologii.

Kódování založené na nástrojích – MPEG-1 má velmi limitovaný stupeň flexibility, MPEG-2 představil koncept profilů a úrovní, které mohou být kombinovány různými způsoby pro různorodé využití. MPEG-4 toto rozšiřuje směrem k vysoce flexibilní sadě kódovacích nástrojů, což poskytuje řadu využití stejně tak jako standardizovaný systém, který umožňuje přidání nových nástrojů do balíku nástrojů.

Standard MPEG-4 je organizován tak, aby nové kódovací nástroje a funkce mohly být přidávány, když jsou vyvíjeny nové verze standardu, a tak se seznam nástrojů stále rozrůstá. Nicméně, hlavní nástroje pro kódování videa mohou být shrnuty takto:

Vstupní formát - očekává se, že videodata budou předzpracována a konvertována do jednoho z rozlišení uvedených v tabulce č. 8, při třiceti snímcích za sekundu a v 4:2:0 Y:Cr:Cb formátu. [13]

Formát	Rozlišení obrázku
SubQCIF	128 × 96
QCIF	176 × 144
CIF	352 × 288
4CIF	704 × 576
16CIF	1408 × 1152

Tabulka č. 8 - MPEG-4 rozlišení obrázků

Zdroj: [13], upraveno

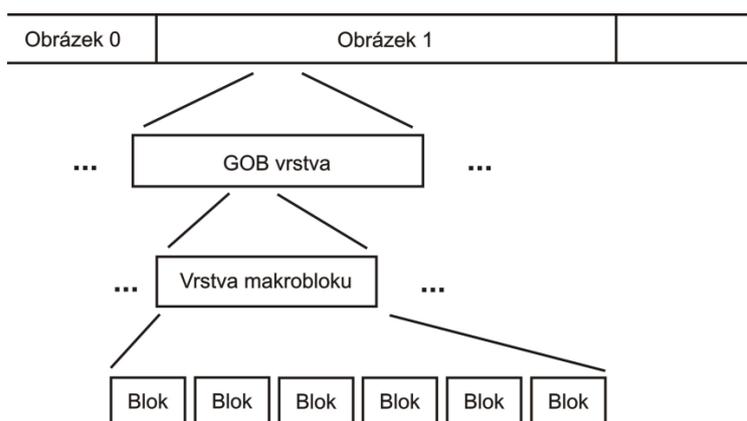
Typy obrazů – každý snímek je kódován jako I- nebo P-snímek.

Odhad pohybu a kompenzace – provádí se na 16 × 16 makroblocích nebo (volitelně) 8 × 8 makroblocích. Pohybové vektory mohou mít rozlišení poloviny pixelů.

a) Syntaxe

Syntaxe MPEG-4 je zobrazena na obrázku č. 25.

Obrazová vrstva – nejvyšší vrstva syntaxe obsahuje kompletní kódovaný obraz. Hlavička obrazu udává rozlišení obrazu, typ kódovaného obrazu a zahrnuje pole dočasných referencí. Tyto faktory indikují správný čas zobrazení pro dekodér (relativně k dalším kódovaným obrazům) a mohou pomoci zajistit, že obraz se nezobrazí brzo, ani pozdě. [13]



Obrázek č. 25 - MPEG-4 složení vrstev

Zdroj: [13], upraveno

GOB vrstva (skupina bloků) – skupina bloků (GOB) se skládá z kompletní řady makrobloků ve SQCIF, QCIF a CIF obrazech (dvě řady v 4CIF obraze a 4 řady ve 16CIF obraze). GOBy jsou podobné plátům u MPEG-1 a MPEG-2 v tom, že pokud je vložena

volitelná GOB hlavička, dekodér se může sesynchronizovat na start dalšího GOBu, pokud dojde k chybě. Nicméně, velikost a rozložení každého GOBu je fixní dle standardu (na rozdíl od plátů).

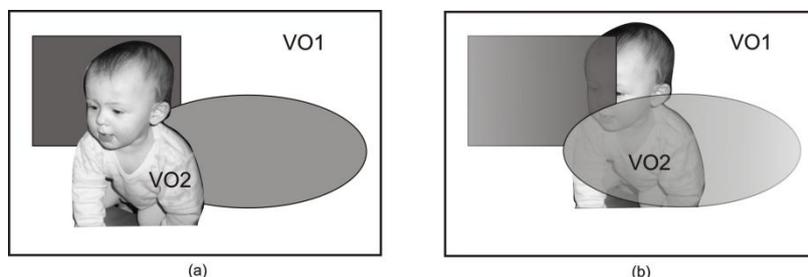
Vrstva makrobloku – makroblok se skládá ze čtyř jasových bloků a dvou barevných bloků. Hlavička makrobloku obsahuje informace o typu makrobloku, kódovací šabloně (indikující, který ze šesti bloků vlastně obsahuje transformované koeficienty) a kódované horizontální a vertikální pohybové vektory.

Vrstva bloků – blok se skládá z koeficientů odpovídajících 8×8 bloku vzorků. Použití 8×8 bloku pohybové kompenzace a návrh tabulek kódování s variabilní délkou dělají MPEG-4 efektivnějším než MPEG-1 nebo MPEG-2. [13]

b) Další kódovací nástroje

Funkce, které dělají MPEG-4 unikátním mezi kódovacími standardy, jsou následující:

Kódování tvarů – Kódování tvarů je potřebné pro specifikaci hranic každého neobdélníkového VOP na scéně. Informace o tvaru mohou být binární (tj. identifikují pixely, které jsou vnitřní k VOPu, popsané jako „neprůhledné“, nebo externí k VOP, popsané jako „průhledné“), nebo ve stupních šedé (kde je každému pixelu v rámci VOP přidělena 8-bitová hodnota ze stupnice šedi, která identifikuje průhlednost pixelu). Informace o stupních šedé jsou komplexnější a vyžadují více bitů ke kódování. Binární informace je jednodušší na kódování, protože každý pixel má pouze dva možné stavy - neprůhledný nebo transparentní. Na obrázku č. 26 je zobrazen koncept neprůhledných a transparentních VOPů: na obrázku a) VOP2 (vpředu) je neprůhledný a kompletně zakrývá VOP1 (pozadí), zatímco na obrázku b) je VOP2 částečně průhledný. [13]



Obrázek č. 26 – Neprůhledný (a) a průhledný (b) VOP

Zdroj: [13], upraveno

Binární informace o tvarech jsou kódovány v blocích 16×16 . Existují tři možnosti pro každý blok:

1. Všechny pixely jsou průhledné, tj. blok je „mimo“ VOP. Není kódován žádný tvar (nebo textura).
2. Všechny pixely jsou neprůhledné, tj. blok je celý „uvnitř“ VOPu. Není kódována žádná informace o tvaru.
3. Některé pixely jsou neprůhledné a některé jsou transparentní, tj. blok překračuje hranice VOPu.

Informace o tvarech ve stupních šedé jsou hodnoty od 0 (průhledné) do 255 (neprůhledné) a jsou komprimovány s použitím DCT na bázi bloků a pohybové kompenzace.

Pohybová kompenzace – existují podobné možnosti jako u I-, P- a B-obrazů u MPEG-1 a MPEG-2:

1. I-VOP - VOP je kódován bez jakékoliv pohybové kompenzace.
2. P-VOP - VOP je predikován pomocí pohybově kompenzované predikce z předchozího I- nebo P-VOPu.
3. B-VOP - VOP je predikován pomocí pohybově kompenzované predikce z předchozího nebo následujícího I- nebo P-obrazu (s budoucí nebo zpětnou dvousměrnou predikcí).

Kódování textur – pixely v rámci VOPu jsou kódovány jako „textura“. Základní nástroje jsou podobné jako u MPEG-1 a MPEG-2: transformace pomocí DCT, kvantizace DCT koeficientů následována seřazením a kódováním s různou délkou. Pro další zlepšení efektivity komprese mohou být DCT koeficienty predikovány z předchozích přenášených bloků (podobně jako diferenční predikce DC koeficientů použitá u JPEGu, MPEG-1 a MPEG-2).

Makroblok, který pokrývá hranice VOPu, bude obsahovat jak neprůhledné, tak transparentní pixely. Aby se aplikovalo regulérní 8×8 DCT, je nezbytné použít doplnění pro vyplnění pozic transparentních pixelů.

Odolnost vůči chybám – MPEG-4 zahrnuje množství mechanismů, které poskytují zvýšený výkon při výskytu chyb přenosu (jako jsou bitové chyby nebo ztracené pakety). Hlavní nástroje jsou:

1. Synchronizační ukazatel - podobně jako pláty u MPEG-1 a MPEG-2 s výjimkou toho, že tyto mohou být volitelně umístěny tak, že každý desynchronizační interval obsahuje přibližně stejný počet kódovaných bitů. To znamená, že chyby budou nejspíše pravidelně distribuovány mezi rozsynchronizované intervaly.
2. Dělení dat - podobně jako dělení dat u MPEG-2.
3. Rozšíření hlavičky - redundantní kopie informací o hlavičce jsou vloženy v intervalech do bitového streamu, takže pokud se ztratí důležitá hlavička (např. hlavička obrázku) kvůli chybě, mohou být tyto redundantní hlavičky použity pro částečné obnovení kódované scény.

Škálovatelnost - MPEG-4 podporuje prostorovou a časovou škálovatelnost. Prostorová škálovatelnost se používá na obdélníkové VOPy podobně jako u MPEG-2: základní vrstva poskytuje nízké prostorové rozlišení a rozšiřující vrstva může být dekódována společně se základní vrstvou pro získání většího rozlišení. Časová škálovatelnost je rozšířena (narozdíl od MPEG-2) tak, že může být aplikována na jednotlivé VOPy. Například, VOP na pozadí může být kódován bez škálování, zatímco VOP v popředí může být kódován s několika vrstvami časové škálovatelnosti.

Statická textura – Alternativní sada nástrojů k DCT může být použita ke kódování „statických“ textur, tj. texturových dat, která se nemění rychle. [13]

4.3.1.4 Souhrn

ISO vydalo mnoho standardů o kódování obrazu a videa, které silně ovlivnily vývoj technologie a trh pro video kódující aplikace. Původní standard JPEG komprese statického obrazu je nyní všudypřítomnou metodou pro ukládání a přenos statických obrázků a získal si velkou popularitu jako jednoduchý a robustní algoritmus pro kompresi videa.

První MPEG standard, MPEG-1, nikdy neměl úspěch na trhu u svých cílových aplikací (video CD). Avšak je široce používán pro PC a internetové aplikace a utvořil základ pro MPEG-2. Pomocí MPEG-2 byl umožněn celosvětový posun k digitální televizi a je zřejmě nejúspěšnějším standardem pro kódování videa, co se týká průniku na trh. MPEG-4 nabízí mnoho nástrojů pro kódování videa, které mohou v budoucnosti umožnit další využití této komprese. [13]

4.3.2 H.26x

ISO MPEG standardy se zaměřují na ukládání a distribuci videa pro zábavu a snažily se splnit požadavky poskytovatelů a zákazníků v „mediálním průmyslu“. ITU se (historicky) více zajímalo o telekomunikační průmysl, jeho standardy pro kódování videa (H.261, H.263, H.26L) byly tedy zaměřeny na komunikace z bodu do bodu nebo vícebodové komunikace v reálném čase.

První ITU-T standard kódování videa, který měl významný dopad (H.261), byl vyvinut na konci osmdesátých let dvacátého století se speciální aplikací a přenosovým kanálem. Zmiňovanou aplikací byla videokonference (dvousměrná komunikace přes video) a kanál byl N-ISDN. ISDN poskytuje konstantní datový tok $p \times 64$ kbps, kde p je celé číslo mezi 1 a 30. Tehdy se předpokládalo, že ISDN bude prostředkem pro video komunikace díky garantované šířce pásma a malému zpoždění. Modemové kanály byly považovány za příliš pomalé pro vizuální komunikaci a vysílání založené na paketech nebylo považováno za dostatečně spolehlivé.

H.261 byl vcelku úspěšný a stále se používá v mnoha aplikacích pro video konference. Zlepšení výkonu procesorů, technik kódování videa, vznik analogových modemů a internetových (IP) sítí vedlo v polovině devadesátých let dvacátého století k vývoji jeho následníka (H.263). Provedením mnoha vylepšení H.261 poskytoval H.263 významně kvalitnější výkon při kompresi, stejně tak jako vyšší flexibilitu. Původní standard H.263 (verze č. 1) měl čtyři volitelné módy, mezi kterými se dalo přepínat pro zlepšení výkonu. Tyto módy byly považovány za užitečné a verze č. 2 (H.263+) přidala dalších 12 volitelných módů. Nejnovější (a zřejmě i poslední) verze č. 3 obsahuje celkem 19 módů, kde každý mód přispěl ke zlepšení výkonu kódování a k odolnosti proti chybám.

Verze č. 3 standardu H.263 se stala poněkud nepraktickým standardem z hlediska velkého počtu nastavení. Nejnovější iniciativa ITU-T skupiny expertů VCEG je H.26L standard (kde L je Long term – dlouhodobý). H.26L je nový standard, který využívá některé z nejlepších funkcí H.263 a má za cíl zlepšit výkon komprese o cca 50% při nižších datových tocích. [13]

4.3.2.1 H.261

Typické provozní datové toky pro H.261 aplikace jsou mezi 64 a 384 kbps. V době vývoje se nečekalo, že paketově založený přenos dat přes internet bude důležitým

požadavkem a limitovaný výkon komprese videa (v té době dosažitelný) nebyl považován za vhodný pro podporu datových toků pod 64kbps.

Typický H.261 kodek je velice podobný pohybově kompenzovanému DCT-založenému kodeku. Video je zpracováváno ve formátu 4:2:0 Y:Cr:Cb. Základní jednotkou je makroblok, obsahující 4 jasové bloky a 2 bloky barev (viz obrázek č. 17). Na vstupu kodéru mohou být makrobloky pohybově kompenzovány s použitím celočíselných pohybových vektorů. Pohybově kompenzovaná zbytková data jsou kódována pomocí DCT s následnou kvantizací a cik-cak přerovnáním.

Ve standardu je definován chyby opravující kód, který by se měl přidávat do přenášeného streamu. V praxi je tento kód často vynecháván z H.261: výskyt chyb ISDN kanálu je dostatečně nízký, tudíž opravování chyb není obvykle potřeba a kód specifikovaný ve standardu není vhodný pro další kanály (jako například rušený bezdrátový kanál nebo paketově založený přenos).

Každý makroblok může být kódován bez pohybově kompenzované predikce, nebo s pohybově kompenzovanou predikcí. Podporována jsou pouze dvě rozlišení: CIF (352 × 288 pixelů) a QCIF (176 × 144 pixelů).

H.261 byl vyvinut v době, kdy zpracovávací výkon hardwaru a softwaru byl limitován, a proto má výhodu „nízké složitosti“. Mezi jeho nevýhody patří nízký výkon komprese (s nízkou kvalitou videa při datových tocích pod 100 kbps) a nedostatek flexibility. H.261 byl nahrazen kodekem H.263, který má větší kompresní efektivitu a lepší flexibilitu, ale je i nadále široce využíván v instalovaných video konferenčních systémech. [17]

4.3.2.2 H.263

Při vývoji standardu H.263 se VCEG zaměřilo na zlepšení H.261 v několika oblastech. Využitím výhod vývoje v algoritmech pro kódování videa a pokrocích ve výkonu zpracování poskytuje lepší kompresi. H.263 poskytuje větší flexibilitu než H.261, například je podporován větší rozsah rozlišení obrazu (seznam v tabulce č. 8). První verze H.263 představila 4 volitelné módy, každý popsany v doložce standardu, a další volitelné módy byly představeny ve verzi č. 2 standardu (H.263+). H.263 podporuje i video komunikaci s datovým tokem pod 20 kbps (s velice limitovanou kvalitou videa) a je nyní široce využíván jak v „zavedených“ aplikacích (jako je video telefonování a video konference), tak ve zvyšujícím se počtu nových aplikací (jako je video na internetu). [16]

a) Funkce

Základní H.263 kodek je funkčně stejný jako MPEG-4 kodek popsany v kapitole 4.3.1.3. Vstupní snímky jsou ve 4:2:0 formátu, pohybově kompenzované, transformované s 8×8 DCT, kvantizované, přeseřazené a entropně kódované. Hlavní faktory, které přispěly ke zlepšení výkonu kódování oproti H.261, jsou použití „půl-pixelových“ pohybových vektorů (které poskytují lepší pohybovou kompenzaci) a přepracované tabulky kódu variabilní délky (VLC). Funkce jako I- a P-obraz, větší velikost snímků a volitelné módy kódování dávají designérovi větší flexibilitu k tomu, aby si poradil s různými požadavky aplikací a přenosovým scénářem.

H.263 volitelné módy / H.263+

Původní standard H.263 (verze č. 1) obsahoval 4 volitelné módy kódování (přílohy D, E, F a G). Verze č. 2 standardu přidala dalších 12 módů (přílohy I až T) a nová verze počítá s ještě více módy (přílohy U, V a W). Kodeky, které implementují některé z volitelných módů, jsou někdy popisovány jako H.263+ nebo H.263++ v závislosti na tom, jaké módy mají implementovány.

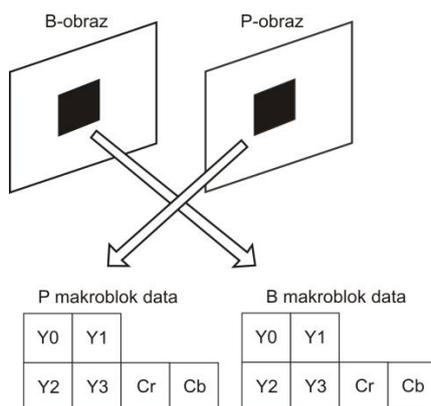
Každý mód zvyšuje nebo mění funkcionalitu kodeku H.263, obvykle za cenu zvětšení složitosti. Volitelné módy mají několik potenciálních výhod: některé zvyšují výkon komprese, jiné zlepšují odolnost proti chybám nebo poskytují nástroje, které jsou užitečné pro určitá přenosová prostředí (jako je například paketově založený přenos).

Příloha D, neomezené pohybové vektory. Volitelný mód popsany v příloze D standardu H.263 umožňuje pohybovým vektorům odkazovat mimo hranice obrazu. To představuje zvýšení kódovacího výkonu, zejména pokud jsou objekty v pohybu „do“ nebo „z“ obrazu.

Příloha E, aritmetické kódování založené na syntaxi. Aritmetické kódování se používá namísto kódování s variabilní délkou. Každý z VLC definovaný ve standardu je nahrazen pravděpodobnou hodnotou, která je použita aritmetickým kódérem.

Příloha F, pokročilá predikce. Efektivita odhadu pohybu a kompenzace je zvýšena povolením použití čtyř vektorů na jeden makroblok. Pohybová kompenzace překrývajících se bloků je použita pro zlepšení pohybové kompenzace v dekódovaném obraze. Příloha F vyžaduje, aby kodek podporoval neomezené pohybové vektory (příloha D).

Příloha G, PB-snímky. PB-snímek je pár snímků kódovaných jako kombinovaný celek. První snímek je kódován jako B-obraz a druhý snímek jako P-obraz. P-obraz je dopředně predikován z předchozího I- nebo P-obrazu a B-obraz je dvousměrně predikován z předchozích a aktuálních I- nebo P-obrazů. Na rozdíl od MPEG-1 (kde B-obraz je dekódován jako samostatná jednotka) obsahuje každý makroblok PB-snímku data jak z P-, tak i z B-obrazu (obrázek č. 27). PB-snímky mohou poskytovat zlepšení v efektivitě komprese. [13]



Obrázek č. 27 - Makroblok PB-snímku

Zdroj: [13], upraveno

Příloha I, pokročilé kódování. Tento mód využívá korelaci mezi DCT koeficienty v sousedních kódovaných blocích obrazu. DC koeficient a první řada nebo sloupec AC koeficientů mohou být predikovány z koeficientů sousedních bloků.

Příloha J, odblokovací filtr. Okraje každého 8×8 bloku jsou „vyhlazeny“ použitím prostorového filtru. To redukuje „kostičkování“ v dekódovaném obraze a také zlepšuje výkon pohybové kompenzace.

Příloha K, plátově strukturovaný režim. Tento mód poskytuje podporu pro obnovení synchronizace intervalů, které jsou podobné MPEG-1 „plátům“. Plát je řada kódovaných makrobloků, začínajících hlavičkou plátu. Pláty mohou obsahovat makrobloky v rastrovém pořadí, nebo v jakékoliv obdélníkové oblasti obrázku. Pláty mohou být volitelně posílány v náhodném pořadí. Každý plát může být dekódován nezávisle na jakémkoliv dalším plátu z obrazu, takže pláty jsou vhodné pro odolnost proti chybám, jelikož chyba v jednom plátu neovlivní dekódování dalších plátů.

Příloha L, doplňkové rozšiřující informace. Tato příloha obsahuje několik doplňujících kódů, které mohou být poslány kóděrem dekodéru. Tyto kódy indikují informace, týkající se údajů o video sekvenci (jako strnutí obrazu a informace o časování).

Příloha M, vylepšené PB-snímky. Jak název napovídá, je to vylepšená verze původního módu PB-snímků (příloha G). Příloha M přidává možnost dopředné nebo zpětné predikce pro B-snímkovou část každého makrobloku (stejně tak jako dvousměrná predikce definovaná v příloze G), což má za výsledek vyšší efektivitu komprese.

Příloha N, výběr referenčního obrázku. Tento mód umožňuje, aby si kodér vybral snímek z předtím kódovaných obrázků pro predikci aktuálního obrázku.

Příloha O, škálovatelnost. Časová, prostorová a SNR škálovatelnost jsou podporovány tímto volitelným módem. Podobným způsobem jako volitelné módy škálovatelnosti u MPEG-2 zvětšují prostorové rozlišení snímku, SNR zvyšuje kvalitu obrazu a časová škálovatelnost zvyšuje počet snímků za sekundu (fps). V každém případě, „základní vrstva“ poskytuje základní výkon a zvýšený výkon je získán dekodováním základní vrstvy společně s „rozšiřující vrstvou“. Časová škálovatelnost je užitečná zejména proto, že podporuje B-obrazy: ty jsou podobné „skutečným“ B-obrazům v MPEG standardech (kde B-obraz je samostatně kódovaná jednotka) a jsou více flexibilní než kombinované PB-snímky, popsané v přílohách G a M.

Příloha P, převzorkování referenčních snímků. Predikční referenční snímek použitý kodérem a dekodérem může být převzorkován před pohybovou kompenzací.

Příloha Q, obnovení sníženého rozlišení. Kodér se může rozhodnout aktualizovat vybrané makrobloky s nižším rozlišením, než je normální prostorové rozlišení snímku. To může být vhodné například pro možnost kodeku obnovovat pohybující se části snímku při nižším rozlišení s použitím malého počtu kódovaných bitů, zatímco statické části snímku kodér udrží v původním vyšším rozlišení.

Příloha R, dekodování nezávislých segmentů. Tato příloha rozšiřuje koncept nezávisle dekodovatelných plátů nebo GOBů (příloha K). Segmenty obrázku (kde segment je jeden plát nebo celočíselný počet GOBů) mohou být dekodovány zcela nezávisle na jakémkoliv jiném segmentu.

Příloha S, alternativní VLC. Kodér může použít alternativní tabulku variabilně dlouhého kódu pro transformaci koeficientů v kódovaných blocích. Alternativní VLC (vlastně stejná VLC, použitá pro kódované bloky v příloze I) může poskytovat lepší efektivitu kódování, pokud je zde velký počet vysoce hodnocených, kvantovaných DCT koeficientů (např. když je kódovaná rychlost přenosu vysoká nebo je mnoho změn ve scéně).

Příloha T, modifikovaná kvantizace. Tento mód představuje některé změny ve způsobu, jakým jsou prováděny operace kvantizace a změny rozměru.

Příloha U, rozšířený výběr referenčního obrazu. Příloha U upravuje mód selekce referenčního obrazu z přílohy N pro zlepšení odolnosti proti chybám a efektivitu kódování. Je tu několik změn, včetně změny mechanismu pro snížení požadavků na paměť pro ukládání předtím kódovaných obrázků a možnosti vybrat referenční obrázek pro pohybovou kompenzaci na bázi makroblok-makroblok.

Příloha V, datově rozdělený plát. Tento mód zlepšuje odolnost strukturovaných dat plátu proti chybám při přenosu (upravením přílohy K). V každém plátu jsou data makrobloku přeskládána tak, že všechny hlavičky makrobloků jsou přenášeny jako první, následovány všemi pohybovými vektory a konečně všemi daty transformovaných koeficientů. Když dojde k chybě v datech hlavičky nebo v datech pohybového vektoru, má to obvykle větší vliv na dekódovaný obraz než chyba v datech transformovaných koeficientů: přerovnáním dat tímto způsobem by chyba v části plátu měla ovlivnit pouze méně citlivá data transformovaných koeficientů.

Příloha W, další doplňující informace o vylepšení. Jsou definovány dvě rozšiřující informační položky (k těm, které už jsou v příloze L). Funkce „IDCT s pevným bodem“ indikuje, že se raději může použít přibližný inverzní DCT (IDCT) raději, než „přesná“ definice IDCT udaná standardem. Tento způsob může být vhodný pro nenáročnou implementaci standardu s pevným bodem. Funkce „obrazová zpráva“ umožňuje uživateli vložit definovanou zprávu do kódovaného streamu. [13]

b) H.263 Profily

Je velice nepravděpodobné, že pro jednu aplikaci bude potřeba všech 19 módů. Místo toho mohou být určité kombinace módů vhodné pro konkrétní přenosové scénáře. Shodně jako MPEG-2 a MPEG-4 definuje H.263 sadu doporučených profilů (kde profil je podmnožina volitelných nástrojů) a úrovní (kde úroveň určuje maximální hodnotu konkrétních kódovacích parametrů, jako je rozlišení snímku, počet snímků za vteřinu a datový tok). Profily a úrovně jsou definovány v poslední příloze H.263, v příloze X. Celkem je 9 profilů – viz níže:

Profil č. 0, základní. Jednoduše řečeno, jedná se o základní funkčnost H.263 bez jakýchkoliv volitelných módů.

Profil č. 1, efektivita kódování (verze č. 2). Tento profil poskytuje efektivní kódování s použitím pouze nástrojů z verze č. 1 a č. 2 (tj. do přílohy T). Vybrané volitelné módy jsou příloha I, J, L a příloha T. Přílohy I, J a T umožňují zlepšení efektivity kódování v porovnání se základním módem. Příloha J zahrnuje funkce první verze standardu, čtyři pohybové vektory na makroblok a neomezené pohybové vektory.

Profil č. 2, efektivita kódování (verze č. 1). V tomto profilu jsou použity pouze nástroje dostupné ve verzi č. 1 a ve skutečnosti je zahrnuta pouze příloha F. U dalších třech příloh D, E a G z původního standardu nebylo zjištěno, že poskytují dostatečné zvýšení výkonu při kódování pro jejich zahrnutí do nového standardu.

Profily č. 3 a č. 4, interaktivní a bezdrátové streamování. Tyto profily zahrnují efektivní kódovací nástroje (přílohy I, J a T) společně s plátově strukturovaným módem (příloha K) a v případě profilu č. 4 i datově rozdělený plát (příloha V). Tyto plátové módy mohou podpořit odolnost proti chybám, která je důležitá pro „rušené“ prostředí bezdrátového přenosu.

Profily č. 5, č. 6 a č. 7, konverzační. Tyto tři profily podporují malé zpoždění a vysokou kompresi „konverzační“ aplikace (jako například video telefonie). Profil č. 5 zahrnuje nástroje, které poskytují efektivní kódování. Profil č. 6 přidává plátový strukturovaný mód (příloha K) pro internetové konference. Profil č. 7 přidává podporu pro prokládané kamerové zdroje (část přílohy W).

Profil č. 8, vysoká odezva. Pro aplikace, které tolerují vysoké odezvy (zpoždění), jako je streamované video, přidává profil č. 8 další efektivní kódovací nástroje, jako jsou B-obrazy (příloha O) a převzorkování referenčních obrazů (příloha P). B-obrazy zvyšují efektivitu kódování za cenu větších zpoždění.

Zbývající nástroje v rámci devatenácti příloh nejsou zahrnuty v žádném profilu buď proto, že jsou považovány za příliš složité pro cokoliv jiného než aplikace pro speciální účely, nebo proto, že je nahradily efektivnější nástroje. [13]

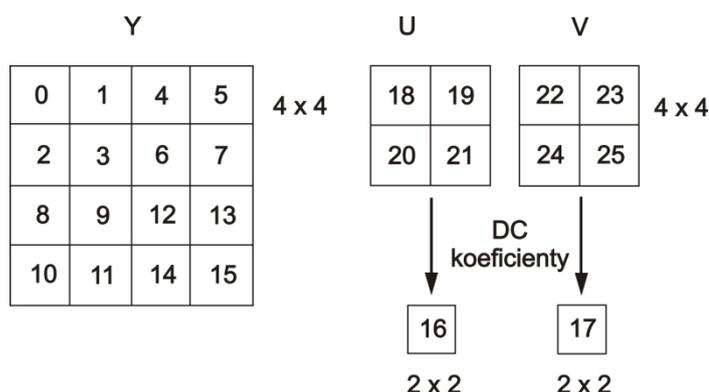
4.3.2.3 H.26L

Devatenáct volitelných módů z H.263 zlepšilo efektivitu kódování a přenosové schopnosti, nicméně, vývoj standardu H.263 je omezen požadavkem pokračování v podpoře „základní“ syntaxe. Posledním standardizačním úsilím VCEG je vyvinout novou syntaxi kódování, která nabízí značné výhody oproti starším standardům H.261 a H.263. Tento nový

standard, aktuálně popisovaný jako „H.26L“, kde L znamená „Long-term = dlouhodobý“ a odkazuje na fakt, že tento standard byl plánován jako dlouhodobé řešení mimo „krátkodobé“ přídatky k H.263 (verze 2 a 3).

Cílem H.26L je poskytnout řešení „nové generace“ pro video kódovací aplikace nabízející značně vylepšenou efektivitu kódování za snížení „zmatku“ ve spoustě volitelných módů v H.263. Nový standard má také za cíl vzít v úvahu měnící se povahu aplikací kódujících video. Hlavní funkce H.26L mohou být popsány takto:

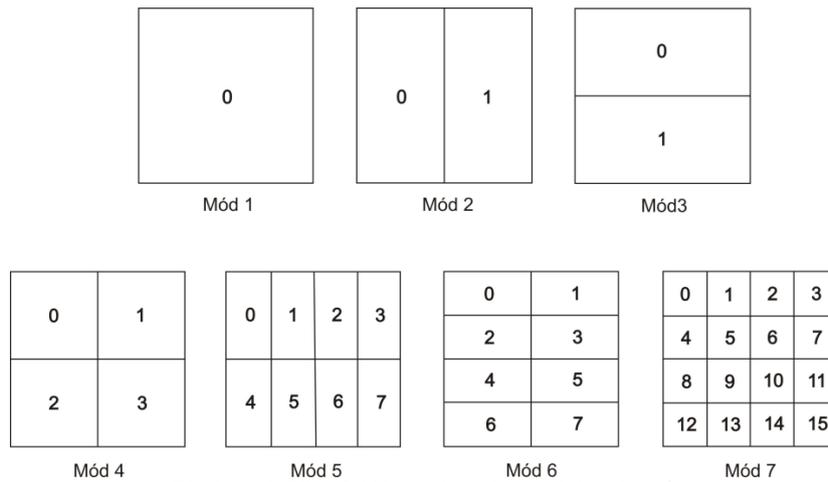
Zpracovávací jednotky. Základní jednotka je makroblok, jako u předchozích standardů. Nicméně podjednotka je nyní blok 4×4 . Makroblok obsahuje celkem 26 bloků (obrázek č. 28): 16 bloků pro jas (každý 4×4), čtyři 4×4 bloky pro komponenty barev a dva 2×2 subbloky, které uchovávají DC koeficienty každého z osmi barevných bloků. Je efektivnější kódovat tyto DC koeficienty společně, protože je velice pravděpodobné, že budou vysoce korelované. [13]



Obrázek č. 28 - H.26L blok v makrobloku
Zdroj: [13], upraveno

Predikce. Před kódováním 4×4 bloku v makrobloku je každý pixel v bloku predikován z předtím kódovaných pixelů. Tato predikce redukuje množství dat kódovaných v oblastech malých detailů v obraze.

Možnosti pohybových vektorů. H.26L nabízí sedm různých možností pro přiřazování pohybových vektorů v rámci makrobloku, tj. v rozsahu od jednoho vektoru na makroblok (mód 1 na obrázku č. 29) až pro individuální vektor pro každý ze šestnácti jasových bloků (mód 7 na obrázku č. 29). To umožňuje modelovat pohyb objektů nepravidelných tvarů s rozumnou přesností. [13]



Obrázek č. 29 - H.26L módy pohybových vektorů
Zdroj: [13], upraveno

Obsahově založené adaptivní binární aritmetické kódování. Tento alternativní entropní kodér používá aritmetické kódování pro získání vyšší efektivity komprese než variabilně dlouhé kódování.

B-obrazy. Jsou označovány jako velmi užitečný kódovací nástroj, konkrétně pro aplikace, které nejsou citlivé na přenosová zpoždění. H.26L podporuje B-obrazy podobným způsobem jako MPEG-1 a MPEG-2, tj. není tu žádné omezení ohledně počtu B-obrazů, které mohou být přenášeny mezi páry I- nebo P-obrazů. [13]

4.3.2.4 Souhrn

ITU-T skupina expertů vyvinula standard H.261 pro videokonferenční aplikace, které poskytovaly přiměřený výkon komprese s relativně nízkou složitostí. Tento standard byl nahrazen standardem H.263, který nabízel lepší výkon díky funkcím, jako je ½ pixelová pohybová kompenzace a zlepšení kódování s variabilní délkou. Další dvě vydané verze H.263 nabízely mnohé volitelné kódovací módy pro podporu lepší efektivity komprese a větší flexibility. Verze č. 3 obsahuje devatenáct volitelných módů, ale je omezena kvůli požadavku na podporu „základního“ H.263 kodeku. H.26L standard zahrnuje několik nových nástrojů pro kódování, jako je 4 × 4 bloková transformace a flexibilní možnosti pohybových vektorů. [13]

5 Streamování videa

Co se stane poté, když je úspěšně zakódován příslušný multimediální obsah? Podobně jako při publikování webových stránek je soubor nahrán na server. Zde se postupy začínají lišit. Obecný web-server jednoduše stahuje soubor. Streamovací server musí řídit přenosovou rychlost streamu pro poskytnutí přehrávání v reálném čase.

Když prohlížeč vyšle požadavek na webovou stránku, jsou soubory doručovány maximální rychlostí, kterou nabízí síťové připojení. TCP zajišťuje bezchybný přenos znovuposíláním ztracených paketů, čas stažení je závislý na dostupné šířce pásma. TCP zaručuje přesné doručení, ale doručení na čas zaručit nelze. Streamování médií má opačné požadavky: doručování musí být v reálném čase a je akceptovatelná přiměřená míra přenosových chyb.

Streamovací servery mohou být založeny na vlastní architektuře, nebo navrženy tak, aby zvládly formáty standardů jako je MPEG-4. Architektura systému se může lišit od jednoho serveru, sloužícího pro malou firemní stránku, až po velké farmy distribuovaných serverů, schopných poskytovat stovky tisíc streamů pro přímé přenosy, jako jsou aktuální zprávy, přehlídky a koncerty.

Stream může být dodáván jako proces push nebo pull. Push je používán pro streamování živého nebo nahraného obsahu jako webcastu – televizní model. Push streamování může být použito pro televizní kanály nebo živě přenášené události, případně si uživatel může stáhnout (pull) nahraný obsah na vyžádání. Pull je podobné používání CD-ROMu nebo používání webového prohlížeče.

Webcast může být kombinace živého a předehraného obsahu. Při událostech se server chová jako distribuční bod, pouze odráží stream na diváky. Pro předehraný obsah má server dvě funkce. První funkcí je nahrání obsahu z lokálních diskových polí a druhou funkcí je kontrolování rychlosti dodávání streamu.

V případě interaktivního obsahu si klient nebo přehrávač vyžádá soubory ze serveru. Při simulování živého přenosu přehrává server playlist, který streamuje soubory v předem stanovený čas nastavený v přehrávači.

5.1 Streamování

Nejpoužívanější streamovací server pro dodávání multimediálního obsahu je webserver, zprostředkovaný Apachem. Web servery používají HTTP přes TCP/IP pro přenos HTML stránek a jejich přidružených obrázků.

TCP/IP se používá jako transportní vrstva na internetu. Soubory jsou stahovány do vyrovnávací paměti prohlížeče tak rychle, jak jen to systém dovolí. TCP obsahuje řízení toku dat pro správu rychlosti stahování. Neexistuje předem stanovená rychlost pro přenos. TCP zvyšuje rychlost přenosu dat, dokud ztráta paketů nezačne indikovat, že síť je přetížená. V tomto bodě se rychlost sníží. Dalším omezením je přijímací zásobník. Přijímač zpracovává pakety okamžitě, jak dorazí. Pokud data přicházejí moc rychle, přijímací zásobník může přetéct. Přijímač vyšle zprávu vysílači, aby zpomalil, a tím zamezil plnění zásobníku. [19]

Při streamování obsahu v reálném čase jsou mediální pakety zpracovávány přehrávačem poté, co dorazí. Není zde žádné lokální „kešování“, čímž je vyřešen problém lokálního ukládání. Toto nemusí být problém pro uživatele PC, ale mnoho přehrávačů médií má velice limitovanou paměť, například mobilní telefony.

Stále tu existuje problém s kontrolou rychlosti. Pokud je stream kódován rychlostí 40 kbit/s, musí být pro uspokojivé zobrazení obrazu v této rychlosti také přenášen. Jedna z funkcí transportní vrstvy protokolu je regulovat rychlost streamu. Co se stane, pokud je síť přetížena například už při 30 kbit/s? Přehrávači dojdou data a přehrávání se zastaví, což je jedno z hlavních omezení při streamování.

Jednou z cest, jak toto obejít, je kódovat pod takovou hodnotou, kterou je síť schopna zvládnout. To však může být těžké odhadnout, takže existují sofistikovanější způsoby. Nejpoužívanější je kódování v několika rychlostech a poté automatické zvolení optimální rychlosti pro šíření. Toto přepínání mezi různými soubory s různou rychlostí je také úkol pro server.

Jednou z velkých výhod streamování je interaktivita. Uživatel se může v klipu libovolně pohybovat. Server jen musí najít a poskytnout správné části klipu s použitím indexu.

Z těchto příkladů je patrné, že streamovací server má navíc několik funkcí oproti standardnímu web serveru:

- Kontrola datového toku v reálném čase

- Inteligentní přepínání streamů
- Interaktivní navigace v klipu

HTTP nepodporuje ani jednu z těchto funkcionalit, takže pro streamovaná média byly vyvinuty nové protokoly. Pod záštitou IETF bylo vyvinuto několik nových protokolů pro multimediální výměnu souborů v reálném čase: RTSP, RTP a RTCP. Windows media původně využíval Microsoft Media Server (MMS) pro přenosový framework (ale nyní podporuje RTSP).

Real-Time Streaming Protocol (RTSP) je framework, který může být použit pro interaktivní ovládání přehrávání (Play, Pause, atd...). Používá se také pro získávání správného souboru z diskového pole. RTSP může být také použit pro oznamování dostupnosti dalších media streamů, např. při živém vysílání. Real-Time Protocol (RTP) se používá pro datové pakety. Real-Time Control Protocol (RTCP) poskytuje zpětnou vazbu z přehrávače k indikaci kvality streamu, může například hlásit ztráty paketů a pakety mimo provoz. Server může poté reagovat na přetíženou síť snížením počtu snímků videa nebo zařazením souborů kódovaných s nižším datovým tokem. Real-time media stream může být přenášen pomocí UDP nebo TCP, volba závisí na podmínkách šíření. Ovládací protokoly používají TCP/IP pro obousměrné server-klient spojení. [20]

5.1.1 Formáty souborů pro streamování

Pro streamování souborů v reálném čase musí být soubory uloženy v jednom z formátů pro streamování. Tyto formáty mají kontrolu časování, která může být použita serverem pro ovládání rychlosti toku dat. Pokud klient používá interaktivní ovládání, pomáhá mu souborový index při navigaci.

Hlavní formáty jsou MPEG-4 (mp4), Microsoft advanced format (wmv a wma), RealNetworks (rm a ra), a QuickTime (mov). [19]

5.1.2 Pomocné stopy

Tým MPEG-4 využil konceptu pomocných stop z QuickTime pro kontrolu přenosu streamu. Streamovaný soubor se nazývá film. Film se skládá ze stop, což může být video, audio, nebo jiná data klipu. Stopa se skládá z kontrolních informací, které odkazují na média (nebo objekty), které představují stopy. To znamená, že několik různých filmových souborů může odkazovat na stejný videomédia objekt, což může být velmi užitečné pro rozsáhlé mediální prezentace. Film není video a audio-mediální soubor, jsou to metadata nebo

instrukce pro specifickou prezentaci dat média. Tyto soubory jsou spojeny do jednoho souboru při kódování streamu.

Pomocná stopa dává softwaru na serveru ukazatele na RTP informaci, aby server poskytoval správné části média. Tato informace umožňuje serveru dodávat správný videomateriál v pořadí stanoveném v souboru se stopou a za správné rychlosti pro přehrávač. [20]

5.1.3 Přizpůsobování se zahlcené síti

Jak RealNetworks, tak Windows Media nabízejí možnost změnit datový tok streamu podle toho, jak se mění zahlcení sítě. Pro získání co nejlepšího zážitku ze zobrazení chceme streamovat nejvyšší možnou rychlostí. Když síť zpomalí, je lepší rychlost snížit, spíše než pokračovat s vysokou přenosovou rychlostí. Pokud se zahlcení sítě sníží, může se přenosová rychlost opět zvýšit. Touto cestou nedochází k zablokování streamu, pouze ke snížení kvality obrazu. Tyto technologie pracují však pouze při unicastu.

Real-Time protokol udržuje správnou rychlost přenosu přes UDP/IP (nebo TCP/IP, pokud to šířka pásma dovolí). RTSP framework podporuje interakci klienta se streamem, ovládání jako u videorekordéru – play, pause apod. Aplikace streamovacího serveru může použít RTCP zprávy z přehrávače pro měření zahlcení sítě a měnit rychlosti streamu pro vícerychlostní soubory. Přehrávač může hlásit ztracené a zpožděné pakety a příjem paketů mimo sekvenci. [19]

5.1.4 RealNetworks

Spolehlivý stream umožňuje kombinování různých kódovacích rychlostí do jednoho mediálního souboru. Nejnižší tok je pojistka a bude streamován, pokud je přetížení velice vysoké. Jelikož pouze Helix server umí extrahovat správný komponent, web server poskytne soubor úplně celý, včetně všech obsažených datových toků.

a) Windows Media

Obsahuje funkce umožňující kódování a zabalení několika konstantních datových streamů do jednoho souboru. Streamovací server poté streamuje nejlepší možný datový tok pro aktuální podmínky sítě. Windows media enkodér nabízí několik předdefinovaných profilů datových toků, tyto profily mohou být i upraveny tak, aby vyhovovaly zvláštním požadavkům uživatele. Při multicastování souboru, který byl kódován s několika různými datovými toky, bude přenášen pouze ten nejvyšší.

b) QuickTime

Jedná se o méně propracovanou metodu spočívající v nabízení různých datových toků. QuickTime umožňuje kódovat film jako několik oddělených filmů, kódovaných s různými datovými toky i s rozdílnými jazykovými zvukovými stopami.

c) MPEG-4

MPEG tým navrhl odlišný způsob, jak se vyrovnat s různou kapacitou sítě. Server vysílá základní stream s nízkým rozlišením. Další pomocné streamy mohou přenášet více detailů. Pokud je k dispozici větší šířka pásma, pak tyto streamy umožňují přehrávači sestavit lepší obraz. [21]

5.1.5 Nahrávání obsahu

Při používání řízené služby, nebo vlastního serveru, prvním krokem je dodání obsahu na streamovací servery. Pokud se nestreamuje pouze po lokální síti, servery by měly být blízko páteřní síti. Ve všech možnostech je ale enkodér a server na různých místech. Nejjednodušším způsobem doručení obsahu je pomocí přenosu souborů, použitím FTP. Některé enkódovací programy jsou schopné přenášet soubory automaticky, okamžitě po skončení enkódování. [19]

5.1.6 Živé streamování

Soubor může být samozřejmě poslán na CD-ROM, avšak pokud je obsahem živý přenos, pak není tato metoda použitelná a obsah musí být streamován. Mediální enkodér se obvykle připojuje k serveru pomocí TCP pro dvousměrné kontrolní spojení a pro jednosměrný stream média pomocí UDP. Je velmi důležité, aby okruh, použitý pro toto spojení, měl více než dostatečnou šířku přenosového pásma (malou ztrátovost paketů a zpoždění). Jakákoliv ztráta nebo poškození dat bude viditelná všemi diváky přenosu. [19]

5.1.7 Oznamování obsahu

Přehrávač obvykle lokalizuje streamované médium pomocí hyperlinku vloženého na www stránce. Tento odkaz obsahuje nejen URL pro obsah, ale také instrukce pro spuštění přehrávače. [19]

5.1.8 Webové odkazy

Obvyklou cestou, jak používat stream média soubory, je pomocí odkazu umístěného na www stránce. Odkaz ukazuje na malý soubor na webserveru. Windows media nazývá tento

soubor „přesměrovač“, Real používá RealAudio Metasoubor nebo Ram soubor. Jakmile prohlížeč obdrží tento soubor, pomocí MIME typu je metasoubor předán streamovacímu pluginu. Metasoubor obsahuje plnou adresu a jméno streamovaného obsahu. Media player poté vyšle žádost na specifický stream server pro média soubor. Metasoubory mohou obsahovat seznam několika souborů hraných v sekvenci. [19]

5.1.9 SMIL

Při streamování datově vydatných médií se musí u klienta synchronizovat několik různých klipů a obrázků pro správné přehrávání. Jedním způsobem, jak toto udělat, je použít Synchronized Multimedia Integration Language (SMIL) pro zápis ovládacího souboru. SMIL je podporován QuickTimem, Real a Windows Media architekturami. [19]

5.2 Webcasting

Webové vysílání může být živý obraz, nahraný obraz nebo kombinace obojího. Webcast může být nejjednodušeji pouze jediný klip. V případě přehrávání sekvence klipů je nutné nastavit playlist, který určuje serveru přehrávání relevantních klipů v určitou dobu. [19]

5.2.1 Dělení a střídání

Streamovací server může zvládnout několik stovek klientů současně. Pro poskytnutí streamu velkému počtu klientů je nutné použít více serverů. „Dělicí“ server rozděluje živý stream, takže to, co byl původně jediný živý stream z enkodéru, vypadá jako větve stromu.

Pokud by stream sledovalo jen pár klientů ze vzdálených míst, stream se bude doručovat přímo ze serveru v centrále, ale každý klient bude využívat šířku pásma na kmenových sítích. Při velkém počtu klientů by server nezvládl zátěž. Další možností je použít lokální servery na vzdálených místech pro obdržení jednoho streamu a jeho lokální distribuci, což šetří síť na velkou vzdálenost a redukuje zatížení původního serveru. [19]

5.2.2 Multicasting

Pokud provádíte on-line přenos pro velké publikum, obvyklý unicast vytvoří separátní síťové spojení ke každému klientovi. Toto spojení spotřebovává nemalé síťové prostředky i přesto, že každý klient sleduje naprosto stejný přenos. Multicasting nabízí řešení pomocí poskytování jednoho streamu, který je poté routován ke všem klientům. Síťové routery tedy zprostředkovávají distribuci namísto streamovacích serverů.

Nevýhoda multicastingu je ta, že potřebujete mít kontrolu nad routery. Problémy mohou nastat v případě, že chcete provádět multicast veřejně přes internet. Síť peer-to-peer

(klient - klient), která připojuje server ke klientovi, nemusí mít multicast povolen, nebo routery nemusí být nastaveny pro multicast napříč doménami. Při použití multicasu nelze použít automatické měnění datového toku. Server vysílá jeden stream a neví o klientech, takže nemůže vyjednávat o poskytování streamu v určitém datovém toku. [19]

5.3 Playlisty

Při streamování živého vysílání, je potřeba prostředků pro přehrávání klipů dle připraveného programu. Stejně jako u televizní stanice mohou být klipy vloženy do playlistů a streamovány ve stopě. To znamená, že firemní prezentace a dálková výuka může být naplánováno v předem určených časech. Opět může být použit SMIL pro naprogramování playlistu. [19]

5.4 Logování a statistiky

U jakéhokoliv systému je životně důležité monitorovat provoz ke klientům. To je nutné částečně kvůli monitorování toho, kdo co sleduje, a částečně pro zjištění využití serverů a síťových prvků. Logování umožňuje zaznamenat, kdo je divákem klipu, kolik lidí sleduje celé klipy a kolik jen prvních deset vteřin. Monitorování může být prováděno na straně klienta nebo na straně serveru.

Několik společností nabízí produkty, které dokáží sbírat logy ze serverů a poté zpracovat tato data do reportů. Například Arbitron se specializuje na měření počtu diváků s jejich službou MeasureCast. Statistiky ze strany klientů mohou poskytnout více demografických informací než surová data ze serveru. Obě možnosti jsou kombinovány tak, aby poskytovaly komplexní statistiky pro poskytovatele vysílání.

Log serveru může obsahovat základní informace o klientovi, jako například IP adresu, protokol pro připojení (RTSP, HTTP), verzi přehrávače, operační systém a verzi, nastavení jazyka a rychlost CPU. Logy také sbírají informace o relaci zahrnující čas, kdy byl požadavek logován a jak dlouho byl stream sledován. Z reportů lze poznat, jestli diváci viděli celý klip, nebo ho v půlce opustili, kdy došlo k interakcím od uživatele. Toto reportování je možné vypnout na přehrávačích uživatelů, kteří se obávají o soukromí. Logovací software posbírání informace od unicastů, při používání multicasu, server o klientech nebude vědět. [19]

Detailnější informace mohou být sbírány také o kvalitě přenosu, ztrátách paketů, počtu resedů a kolik paketů dorazilo mimo pořadí (dříve, nebo později). Některé ze statistik, které lze sledovat, jsou:

- Počet souběžných streamů
- Počet hitů
- Čas strávený sledováním obsahu
- Množství doručeného obsahu
- Počet unikátních diváků
- Typ použitého přehrávače
- Datový tok streamu

5.5 Serverové architektury

5.5.1 Windows Media Services Series 9

Streamovací server Microsoftu je svázaný s Windows Server 2003/2008 jako Windows Media Services. Windows Media Server podporuje několik různých způsobů vysílání proto, aby streamy mohly být přenášeny přes firemní firewally. S vydáním Windows Media 9 přidal Microsoft podporu pro RTSP protokol. Windows Media Services také podporují multicasting přes IP. [21]

RTSP a TCP

Tento mód streamování nabízí plnou podporu pro doručování v reálném čase, rychlou vyrovnávací paměť a interaktivitu klienta. Rychlá vyrovnávací paměť umožňuje přehrávači použít zásobník přehrávače jako datovou vyrovnávací paměť pro pomoc ke zprůměrování měnících se datových toků na síti (pouze s TCP).

RTSP a UDP

Jedná se o nouzový režim pro případ, že RTSP a TCP nejsou podporovány přehrávačem. Někteří administrátoři sítě nastavují firewall tak, aby blokoval provoz přes UDP, takže není možné tento protokol použít vždy.

HTTP a TCP

Alternativa vhodná pro firewally, které mají povolené regulérní HTTP webový provoz. Windows Media Services nabízejí možnost ovládání, jako například „rychle vpřed, zpět“ atd. Stream trpí kvůli TCP převodům a přizpůsobování se datovému toku.

5.5.2 RealNetworks Helix Universal Server

RealNetworks nabízí Helix universal server pro distribuci živého videa a audia. Helix server může být nasazen na platformách Windows a UNIX (AIX, FreeBSD, Linux, Solaris, HP-UX). Mobilní verze serveru přidává sadu funkcí pro dodávání 3GPP obsahu do bezdrátových přehrávačů. Real server podporuje SMIL pro prezentace médií. [21]

RTSP a TCP

Používá se pro ovládání spojení, poskytuje plnou klient-server interaktivitu.

RTP a UDP

Optimální volba pro streamování mediálního obsahu.

RTP a TCP

Druhá volba pro streamování mediálního obsahu, pokud firewall blokuje UDP data.

HTTP a TCP

Této volby lze využít pro stahování obsahu, pokud není jiná možnost doručení dat.

5.5.3 Apple QuickTime

QuickTime vznikl jako CD-ROM formát a byl vyvinut pro progresivní stahování. Opravdové streamování v reálném čase je podporováno od verze číslo 4. Apple přijal RTP pro streamování přes UDP. Pokud je toto stahování blokováno firemními firewally, je alternativou progresivní stahování přes HTTP. Apple má dva způsoby pro streamování: jedním je QuickTime 5 Streaming server (QTSS), který je součástí Apple OS X serveru, druhým je open source Darwin server. [21]

5.6 Umístění serveru

Streamovací server musí číst média z lokálního disku, paketovat je a doručit při správném datovém toku pro přehrávání v reálném čase. V druhém případě, v době vrcholu poptávky po zdrojích (serveru), je doručování souborů zpožděno. U videostreamu to znamená, že dojde k zastavení přehrávání.

Jak se s postupem času zvyšovala výkonnost komponent serveru, bylo jednodušší splnit požadavky pro streamování. Oblasti, na které je potřebné se zaměřit, zahrnují:

- Výkon disku

- Výkon síťové karty: může být výhodou použít více síťových karet pro streamování a ovládání
- Prostupnost systému a I/O sběrnice
- Symetrické multiprocessorové CPU
- Odstranění nepotřebných softwarových služeb, což ponechá více zdrojů pro nepřerušené doručování streamu
- Dostatečně velká systémová paměť pro zvládnutí několika vysokorychlostních streamovacích zásobníků

Jedním ze způsobů, jak dosáhnout dobrého výkonu, je využít více malých serverů, spíše než jeden multiprocessorový server. Tento způsob dělá systém odolnější proti chybám. [19]

5.6.1 Hosting

Pro vysoce výkonné streamování k velkému počtu veřejných klientů, je potřeba zajistit širokou linku k páteřní síti internetu. Nejjednodušší cestou je použít hosting nebo síť pro doručování obsahu. Poskytovatelé hostingu jsou obvykle umístěni u hubů internetu s širokou konektivitou k páteřní síti. Další možností je využít outsourcingu serverů. Outsourcing má výhody týkající se fyzické bezpečnosti. Poskytovatelé služeb se obvykle nacházejí v zabezpečených budovách, včetně záložních zdrojů energie, hasicích systémů a s několika přístupovými cestami k páteřní síti. [19]

5.6.1.1 Vysoká dostupnost

Pro videoservery není vůbec neobvyklé, že se na ně někdo obrátí s požadavkem na poskytování videoobsahu v jakékoliv době, ve dne nebo v noci, kdekoliv na světě. Pro dosažení vysoké úrovně dostupnosti systému je potřeba navrhnout systém odolný proti chybám. Tento systém může být ve formě redundantních několikanásobných serverových architektur. [19]

5.6.1.2 Bezpečnost

Bezpečnost streamování je určována několika faktory. Existují zde fyzické hrozby jako oheň, výpadek elektřiny nebo krádež. Redundance a zrcadlení serverové stránky jsou dobrou cestou, jak se vypořádat s těmito problémy. Další hrozby pochází od hackerů. Pokud mohou získat přístup k serverům, mohou způsobit katastrofu. [19]

Autentifikace a autorizace

Přístup k zdrojovému serveru by měl být omezen pro autorizované uživatele. Existují tři typy uživatelů: systémoví administrátoři, tvůrci obsahu uploadující obsah a diváci, kteří se na obsah dívají.

Ověřování identity je použito pro potvrzení identity klienta. Autorizace poté umožňuje ověřeným uživatelům přístup k důvěrným informacím. Databáze může uchovávat seznam ověřených uživatelů, kteří získávají přístup pomocí hesla. [19]

5.7 Souhrn

Streamovací server je typem serveru s obsahem, který používá speciální softwarovou aplikaci pro přenos streamovaných médií k přehrávačům v reálném čase. Od normálního webservru se liší tím, že má stálou kontrolu nad tokem dat.

Existují dva typy přenosů: živý a na vyžádání. Přenos na vyžádání poskytuje interaktivitu, která odlišuje aplikace od tradičních systémů pro přenášení videa. Ovládací kanál může také kontrolovat rychlost přenášení médií jako reakci na přetížení sítě.

Streamování je obvykle realizováno přes spojení jeden-na-jednoho mezi serverem a klientem, nazývané unicast. Alternativní volbou pro online vysílání je multicast. Server vysílá pouze jeden stream a jakýkoliv počet klientů se může připojit ke streamu. Toto vysílání potenciálně šetří využití sítě, ale ještě není univerzálně rozšířeno na internetu.

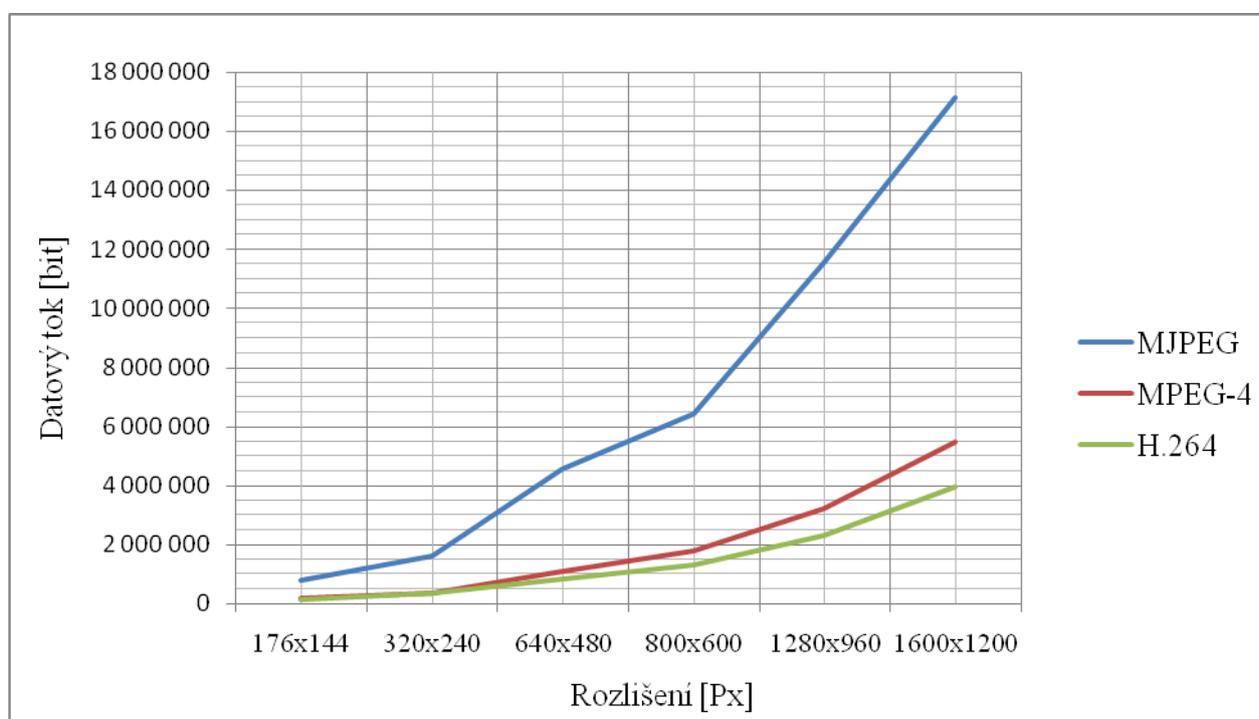
Jedním z nejdůležitějších aspektů řízení streamovacího systému je logování serveru. Logování poskytuje důležité informace pro administrátory sítě, aby mohli vyladit systémy a identifikovali možná problémová místa.

6 Případová studie

V praktické části je ukázán průběh několika testů: test zátěže standardů pro kódování videa a test možností vložení videa do webových stránek. Testy byly provedeny na kamerách společnosti Vivotek, konkrétně na modelu FD8161.

6.1 Výkon standardů pro kódování videa

Každý ze standardů pro kódování videa a obrazů (popsaný v kapitole č. 4) byl navržen z určitého důvodu a obsahuje odlišné funkce. Toto je znemožňuje přímo porovnávat. Obrázek č. 28 porovnává naměřené hodnoty (tabulka č. 9) datového toku při různých rozlišeních (vzor je kódován při statické scéně a 15 snímcích za sekundu). Výsledky zobrazené na obrázku by měly být interpretovány obezřetně, jelikož v závislosti na video sekvenci, datovém toku atd. může být změřen různý výkon. Nicméně trend ve výkonu kompresí je zřejmý. MJPEG pracuje neúspěšně (tj. zabírá relativně velké přenosové pásmo pro dosažení daného rozlišení), protože nepoužívá žádnou mezikomprimaci. MPEG-4 dosahuje podstatného nárůstu komprese oproti MJPEGu díky použití celočíselné pohybové kompenzace. Nejlepšího výsledku v testu dosáhl kodek H.264, který dosahuje ještě vyššího nárůstu výkonu pohybově kompenzovaných snímků.



Obrázek č. 28 - Porovnání datových toků při různých rozlišeních

Toto porovnání není úplně kompletní, protože velikost datového toku se mění se sledovanou scénou a každý výrobce aplikuje standard s odlišnou úrovní komprese. Tabulka

č. 9 porovnává standardy z hlediska datových toků (bps) při různém počtu snímků za sekundu (fps). Ve sloupečku rozdíl v procentech je znázorněn relativní rozdíl mezi kompresemi. V současnosti se pro zabezpečovací systémy stává nejpoužívanější variantou komprese H.264, alternativou je MPEG-4. Kompresce MJPEG byla vyspělou technologií, která zredukovala datový tok ze surových dat až na 10%. Velkou výhodou této komprese je, že každý snímek je klíčový a komprese není tedy tolik náchylná k výpadkům paketů v síti.

Rozlišení	Kompresce	Bps	Bps / 1 snímek (sn)	Bps / 15 sn	Bps / 25 sn.	Rozdíl v %
176 × 144	MJPEG	630 944	52 579	788 680	1 314 467	100
	MPEG-4	101 216	12 652	189 780	316 300	24,06
	H.264	66 592	8 324	124 860	208 100	15,83
320 × 240	MJPEG	853 384	106 673	1 600 095	2 666 825	100
	MPEG-4	201 656	25 207	378 105	630 175	23,63
	H.264	150 992	21 570	323 554	539 257	20,22
640 × 480	MJPEG	2 121 408	303 058	4 545 874	7 576 457	100
	MPEG-4	518 392	74 056	1 110 840	1 851 400	24,44
	H.264	391 120	55 874	838 114	1 396 857	18,44
800 × 600	MJPEG	2 991 112	427 302	6 409 526	10 682 543	100
	MPEG-4	829 376	118 482	1 777 234	2 962 057	27,73
	H.264	603 208	86 173	1 292 589	2 154 314	20,17
1280 × 960	MJPEG	6 163 360	770 420	11 556 300	19 260 500	100
	MPEG-4	1 724 496	215 562	3 233 430	5 389 050	27,98
	H.264	1 208 376	151 047	2 265 705	3 776 175	19,61
1600 × 1200	MJPEG	9 138 312	1 142 289	17 134 335	28 557 225	100
	MPEG-4	2 925 272	365 659	5 484 885	9 141 475	32,01
	H.264	2 095 760	261 970	3 929 550	6 549 250	22,93

Tabulka č. 9 - Porovnání kompresí v různých rozlišeních

Porovnání všech možných kódovacích standardů je složité, protože porovnání velikosti datového toku nebere v potaz některé faktory, jako jsou funkce a flexibilita. Zdá se jasné, že všechny standardy MJPEG, H.264 a MPEG-4 mají své výhody pro video-zabezpečující systémy. Standardy H.264 a MPEG-4 využívají běžných kódovacích technologií (jako odhad a kompenzace pohybu, blokovou transformaci a entropní kódování).

6.2 Vložení obrazu do internetových prezentací

Vzhledem k rozšíření internetu a bezpečnostních kamer chce stále více uživatelů zpestřit své webové stránky o živé video z těchto zařízení. Existuje několik možností, jak vložit obraz (případně zvuk) z kamer do www prezentací. Některé z možností vložení obrazu:

- Vyčítání obrázků z kamery a jeho obnovování na www stránce

- Odesílání obrázků na FTP server a jeho obnovování na www stránce
- Komponenta Microsoft Active X
- Apple Quick Time
- Streamování obrazu

6.2.1 Vychítání obrázku z kamery

Jedná se o nejjednodušší z metod vkládání obrázků do www stránek. Metoda spočívá ve volání URL adresy streamu z kamery v určitém časovém intervalu. Zařízení při každém volání vrací obrázek v kompresi JPEG. Pokud je www server v internetu a kamera ve vnitřní síti, je třeba veřejné IP adresy.

Vzhledem k tomu, že se jedná o klasický JPEG obrázek, je metoda nezávislá na operačním systému a používaném prohlížeči. Velkou výhodou je velmi snadná implementace a že není nutné nic nastavovat. Mezi nevýhody patří: vysoká zátěž kamery při více sledujících, potřeba veřejné IP adresy a sčítání datových toků (každý uživatel stahuje vlastní obrázek z kamery). Jelikož se jedná o JPEG obrázek, není možné přenášet zvuk a obnovovací frekvence obrázku je přibližně 10 sekund.

Zařízení, z kterého vychítáme obraz, nesmí být chráněno heslem, jelikož Internet Explorer neumožňuje ověřování uživatelů a hesel prostřednictvím URL (bylo by zapotřebí upravit registry).

Kód pro vložení obnovujícího se obrázku:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=utf-8">
  <meta name="generator" content="PSPad editor, www.pspad.com">
  <title>Obnovujici se obrazek z kamery Vivotek FD8161</title>
  <script language="JavaScript">
  <!--

  // Promenne obnovujiciho se obrazku

var interval = 5;          // cas obnoveni obrazku v sekundach
var sirka = 640;          // sirka obrazku
var vyska = 480;          // vyska obrazku
var stream = "http://10.0.0.139/cgi-bin/viewer/video.jpg"; //adresa obrazku
var alternativnitext = "Obnovujici se obrazek z kamery Vivotek FD8161.";
                          // alternativni text

  <!-->
```

```

</script>
</head>
<body>
<h1>Obnovující se obrázek z kamery Vivotek FD8161</h1>
<div class="img">
<SCRIPT language="JavaScript" type="text/javascript">
<!--
document.write('');
function Start() {
timestamp = new Date();
timestamp = "?timestamp="+timestamp.getTime()
document.images["obnoveni"].src = stream+timestamp
setTimeout("Start()", interval*1000)
}
Start();
// -->
</SCRIPT>
</div>
</body>
</html>

```

Skript pro vyčítání obrázků a jejich obnovování na www stránkách je možné mimo jazyka javascript naprogramovat například v jazyce AJAX. Tento jazyk je vhodný s ohledem na další možnosti využití tohoto skriptu, kdy se pomocí AJAXu dají z kamery vyčítat další parametry.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta http-equiv="Pragma" content="no-cache">
<meta http-equiv="Content-language" content="cs" />
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Obnovující se snímek z IP kamery</title>
<style type="text/css">
div#ReloadThis { background: black; padding: 5px;;
</style>
<script language="JavaScript">
<!--
// proměnné obnovujících se obrázků
var interval = 10; // čas obnovení obrázku v sekundách
var imgwidth = 640; // šířka obrázku
var imgheight = 480; // výška obrázku
var stream = "http://10.0.0.139/cgi-bin/viewer/video.jpg"; //URL adresa
obrázku
var divContent = '';

function Ajax(){
var xmlhttp;
try{
xmlhttp = new XMLHttpRequest();// Firefox, Opera 8.0+, Safari
}
catch (e){
try{
xmlhttp = new ActiveXObject("Msxml2.XMLHTTP"); // Internet Explorer
}

```

```

        catch (e){
            try{
                xmlhttp = new XMLHttpRequest("Microsoft.XMLHTTP");
            }
            catch (e){
                alert("No AJAX!?");
                return false;
            }
        }
    }
    xmlhttp.onreadystatechange = function(){
        if(xmlhttp.readyState == 4){
            var date = new Date();
            document.getElementById('streamImage').src = stream +
'?timestamp=' + date.getSeconds();
            setTimeout('Ajax()', interval * 1000);
        }
    }
    xmlhttp.open("GET", stream, true);
    xmlhttp.send(null);
}
window.onload=function(){
    setTimeout('Ajax()', interval * 1000);
}
-->
</script>
</head>
<body>
    <div id="ReloadThis">
        <script type="text/javascript">
document.write(divContent);
        </script>
    </div>
</body>
</html>

```

Testování

Oba výše uvedené skripty provádějí stejnou činnost, obnovují stream z kamery každých x sekund. Rozdíl mezi nimi je pouze v použitém jazyce, použitém pro nakódování skriptu. Při nastavení obnovování obrázku každou jednu sekundu kamera nestíhala a vracela obraz každé tři sekundy. Po určitém čase se kamera zahltila dotazy a bylo nutné ji restartovat. Při nastavení obnovování každých pět sekund kamera stíhala stream obnovovat, ale po určitém čase bylo nutné kameru restartovat kvůli zamrznutí. Ideálním řešením je nastavení obnovování obrázku každých 10 sekund. Kompatibilita zobrazení byla testována v prohlížečích Mozilla Firefox verze 3.6.15, Google Chrome verze 10.0.648.151, Opera 11.01 a Internet Explorer verze 5, 6, 7, 8.

Obnovující se obrázek z kamery Vivotek FD8161



Obrázek č. 29 - Test javascriptu pro obnovení obrázku v prohlížeči Mozilla Firefox

6.2.2 Odesílání obrázků na FTP server

Jedná se o principiálně shodnou metodu s metodou vyčítání obrázků z IP kamery s tím rozdílem, že kamera obrázek odešle na FTP server, z kterého je obrázek následně načítán do www stránek. Proti vyčítání obrázků přímo z kamery má tato metoda výhodu v nižším zatěžování kamery (obrázek se odešle pouze jedenkrát) a prakticky neomezený počet současně sledujících uživatelů. Za nevýhodu by se dala označit nutnost FTP serveru a minimální interval jedné minuty mezi obrázky.

Samotné nastavení kamery je jednoduché. V menu kamery se pouze nastaví periodické odesílání obrázků v požadovaném intervalu a přístupové údaje k FTP serveru.

Kód pro vložení obnovujícího se obrázku:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=utf-8">
  <meta name="generator" content="PSPad editor, www.pspad.com">
  <title>Obnovující se obrazek z kamery Vivotek FD8161</title>
  <script language="JavaScript">
    <!--

      // Promenne obnovujiciho se obrazku

var interval = 5;          // cas obnoveni obrazku v sekundach
var sirka = 640;          // sirka obrazku
var vyska = 480;          // vyska obrazku
var stream = "http://10.0.0.139/cgi-bin/viewer/video.jpg"; //adresa obrazku
var alternativnitext = "Obnovující se obrázek z kamery Vivotek FD8161.";
                          // alternativni text

    <!-->

</script>
</head>
<body>
  <h1>Obnovující se obrázek z kamery Vivotek FD8161</h1>
  <div class="img">
  <SCRIPT language="JavaScript" type="text/javascript">
    <!--
document.write(' name=obnoveni border=1>');
function Start() {
```

```

timestamp = new Date();
timestamp = "?timestamp="+timestamp.getTime()
document.images["obnoveni"].src = stream+timestamp
setTimeout("Start()", interval*1000)
}
Start();
// -->
</SCRIPT>
</div>
</body>
</html>

```

Testování

Skript provádí načítání obrázku z FTP serveru, kam ho zašle kamera každou minutu. Kompatibilita zobrazení byla testována v prohlížečích Mozilla Firefox verze 3.6.15, Google Chrome verze 10.0.648.151, Opera 11.01 a Internet Explorer verze 5, 6, 7, 8.

6.2.3 Microsoft ActiveX

Metoda vložení obrazu a zvuku z IP kamer do vlastní webové stránky pomocí komponenty ActiveX. Výhodou je relativní jednoduchost této metody a možnost plynulého videa a zvuku. Mezi nevýhody patří: omezení na prohlížeč Internet Explorer, sčítání datových toků při více uživatelích sledujících stream současně nebo nutnost vlastnit veřejnou IP adresu.

Pro využití komponenty ActiveX je nutné se zaregistrovat a stáhnout High Profile SDK (Software Development Kit) od výrobce, jehož obsahem je ActiveX komponenta VitaminCtrl.

Kód pro vložení komponenty do www stránek:

```

<html>
<head>
<title>VisualServer</title>
<SCRIPT LANGUAGE="VBScript">
<!--
Sub VACtrl_OnClick(V1, V2)
    val = "X = " + CStr(V1) + ", Y = " + CStr(V2)

    MsgBox val
End Sub
//-->
</SCRIPT>

</head>
<body>
<table width="800" border="0" cellpadding="0" cellspacing="0">
    <tr>
        <td align=center>
            <table width=800 border=0>
                <tr>
                    <td align=center>
                        <script language="JavaScript">
                            <!--

```

```

        if ((navigator.appName == "Microsoft Internet
Explorer") && (navigator.platform != "MacPPC"))
        {
document.write("<OBJECT ID=\"VitCtrl\" WIDTH=640 HEIGHT=480
name=\"VitCtrl\" ");
document.write(" CLASSID=CLSID:70EDCF63-CA7E-4812-8528-DA1EA2FD53B6");
document.write(" CODEBASE=\"VitaminCtrl_3_0_0_12.cab#version=3,0,0,12\">");
document.write("<PARAM NAME=\"VSize\" VALUE=\"SIF\">");
document.write("<PARAM NAME=\"Language\" VALUE=\"EN\">");
document.write("<PARAM NAME=\"ClickEventHandler\" VALUE=\"3\">");
document.write("<PARAM NAME=\"RemoteIPAddr\" VALUE=\"10.0.0.139\">");
document.write("<PARAM NAME=\"UserName\" VALUE=\"\">");
document.write("<PARAM NAME=\"Password\" VALUE=\"\">");
document.write("<PARAM NAME=\"ServerModelType\" VALUE=\"6\">");
document.write("</OBJECT>");
        }

        //-->
        </script>
        </td>
        </tr>
        </table>
    </td>
</tr>
</table>
</body>
</html>

```



Obrázek č. 30 - Zobrazení obrazu z kamery pomocí komponenty ActiveX

Testování

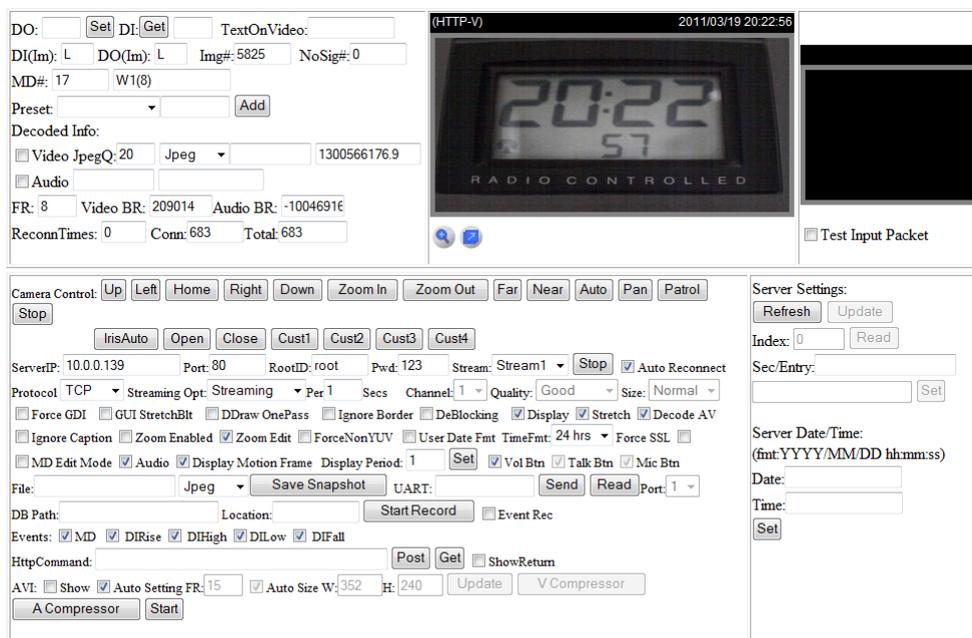
Výše je uveden skript, který umožňuje zobrazit stream z kamery, využívat funkce digitálního zoomu a zvětšit obraz na celou plochu displeje. Výhodou zobrazení pomocí ActiveX je plynulost videa a zvuku. Kompatibilita zobrazení byla testována v prohlížeči Internet Explorer verze 6, 7 a 8, v jiných prohlížečích nelze spustit ActiveX komponenta, což má za následek nezobrazení streamu z kamery.

Kamery byly testovány při rozlišení VGA (640 × 480 px), 15 snímcích za sekundu a kompresi MJPEG, maximální počet současně připojených uživatelů byl deset. Datový tok při deseti připojených uživatelích byl přibližně 22 Mbitů.

Pomocí SDK se nedá zobrazovat pouze obraz (provádí uvedený skript), ale obsahuje mnoho jiných parametrů, díky kterým se dá naprogramovat například vlastní záznamová

aplikace. Všechny dostupné parametry jsou uvedeny v dokumentaci, která je přiložena k programu.

Soubor .cab s ActiveX VitaminCtrl musí být uložen v adresáři společně s www stránkou.



Obrázek č. 31 - Zobrazení všech parametrů pomocí VitaminCtrl

6.2.4 Apple Quick Time

Metoda vložení obrazu nebo zvuku z IP kamery do vlastní www stránky pomocí přehrávače Apple QuickTime. Výhodou je relativní snadnost provedení této metody, možnost vložení plynulého videa včetně zvuku a nezávislost na operačním systému a prohlížeči.

Nevýhodou je jednoznačně sčítání datových toků při více uživateli, sledujících stream najednou. Mezi další nevýhody patří nutnost vlastnit veřejnou IP adresu a instalace QuickTime pluginu.

Nastavení kamery pro RTSP streamování je relativně jednoduché. Ve výchozím nastavení na routeru přesměrujeme porty pro http, RTSP, RTP a RTCP pro audio a video. Jedná se tedy o porty 80, 554, 5556, 5557, 5558 a 5559 ve výchozím nastavení. Dále autor doporučuje vypnout RTSP autentifikaci, není nutné vyžadovat heslo pro video v případě, že používáme kameru pro zpestření www stránek.

Pro správnou funkčnost je dále potřeba do adresáře společně s www stránkou vložit soubor „realqt.mov“, který musí obsahovat parametr:

```
Rtsptextrtsp://dummyurl/mpeg4/media.amp
```

Kód pro vložení pluginu QuickTime do www stránky:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8">
    <meta name="generator" content="PSPad editor, www.pspad.com">
    <title>Vložení obrazu z kamery Vivotek do www stránky</title>
  </head>
  <body>
    <h1>Obnovující se obrázek z kamery Vivotek FD8161</h1>
    <embed src="realqt.mov" width="640" height="480" autostart="true"
controller="true" border="0" scale="ToFit" type="video/quicktime"
qtsrc="rtsp://10.0.0.139/live.sdp" >
  </body>
</html>
```

Testování

Jediný, ale velmi podstatný rozdíl ve vložení obrazu z IP kamery do www stránky pomocí pluginu Apple QuickTime proti komponentě ActiveX je v podporovaných prohlížečích. Obraz je proti realitě zpožděn o tři sekundy, což je způsobeno bufferováním QuickTime. Kompatibilita zobrazení byla testována v prohlížečích Mozilla Firefox verze 3.6.15, Google Chrome verze 10.0.648.151, Opera 11.01 a Internet Explorer verze 5, 6, 7, 8.

Kamery byly testovány při rozlišení VGA (640 x 480 px), 15 snímcích za sekundu a kompresi MJPEG, maximální počet současně připojených uživatelů byl deset. Datový tok při deseti připojených uživatelích byl přibližně 22 Mbitů. Nevýhoda Apple QuickTime je vyšší náročnost na výpočetní výkon procesoru.

6.2.5 Streamování obrazu

Na realizaci nejsložitější metoda ze všech uvedených. Jedná se o podobnou metodu, jako byla metoda vložení pluginu Apple QuickTime. Nevýhodou QuickTime je přístupování přímo na stream kamery a tedy umožnění sledování obrazu maximálně deseti uživatelům. Při použití streamingu je streamovací server připojen pouze jedenkrát na RTSP stream kamery a dále se o rozesílání streamu stará tento server. Teoretický základ streamování byl položen v kapitole č. 5, ve zkratce se jedná o výkonný počítač, umístěný k internetové přípojce v serverhousu, na němž běží specializovaný software pro streamování, jako například VLC player, Windows media player, Windows media services, Helix, Darwin a další.

Výhodou tohoto způsobu zobrazení je malé vytěžování kamery, nezávislost na operačním systému a prohlížeči a teoreticky neomezený počet uživatelů. Nevýhodou je náročnost na hardware počítače a vysoké datové toky při více uživatelích, sledujících stream současně.

6.2.5.1 VLC media player

Jednou z nejdostupnějších metod streamování videa je streamování prostřednictvím free a opensourceového programu VLC media player. Ve VLC playeru se nastaví zdroj videa, protokol, port, kodek, ve kterém se má streamovat, a pustí se video proudem. Pro vkládání videa do internetových stránek a zobrazování prostřednictvím Windows Media playeru je nutné využít protokolu MMS a kodeku DIV3 + MP3 (ASF). Po tomto nastavení se již může na stream připojit teoreticky neomezený počet uživatelů. Prakticky je počet závislý na hardwaru streamovacího serveru a velikosti datové linky směrem upload.

Kód pro vložení obrazu do www stránky:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <meta http-equiv="content-type" content="text/html; charset=windows-1250">
    <metaname="generator" content="PSPad editor, www.pspad.com">
    <title></title>
  </head>
  <body>
    <OBJECT id='mediaPlayer1' width="640" height="480"
classid='CLSID:22d6f312-b0f6-11d0-94ab-0080c74c7e95'
codebase='http://activex.microsoft.com/activex/controls/mpplayer/en/nsmp2inf.cab#Version=5,1,52,701'
standby='Loading Microsoft Windows Media Player components...'
type='application/x-oleobject'>
  <param name='fileName' value="mms://10.0.0.140:8080">
  <param name='animationatStart' value='true'>
  <param name='transparentatStart' value='true'>
  <param name='autoStart' value="true">
  <param name='showControls' value="true">
  <param name="ShowAudioControls" value="true">
  <param name="ShowStatusBar" value="true">
  <param name='loop' value="true">
  <embed
src="mms://10.0.0.140:8080" width=640 height=480 autostart=1 loop=0
align="absmiddle" type="application/x-mpplayer2"
pluginspage="http://www.microsoft.com/Windows/MediaPlayer/download/default.asp"
showcontrols=1 showdisplay=0 showstatusbar=1 > </embed>
  </OBJECT>

  </body>
</html>
```

Testování

Streamování videa prostřednictvím VLC playeru a skriptu s vloženým přehrávačem Windows Media Player je velice dostupnou a relativně snadno realizovatelnou metodou. Testování bylo provedeno na VLC playeru verze 1.1.7, zdrojový stream byl v rozlišení VGA, 15 snímků za sekundu a rozlišení JPEG, datový tok byl přibližně 1 Mbit/s. VLC player

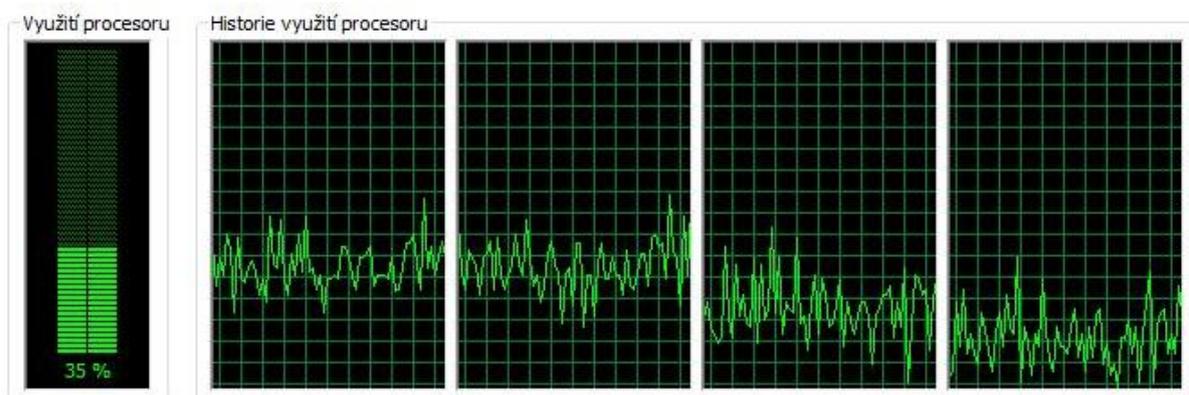
dekódoval stream na velikost 0,2 Mbit/s, který dále posílal do sítě. Počítač, který prováděl streamování, má následující parametry:

- Procesor: Intel Core i5 M 520 2,4GHz
- Paměť: GB (použitelná 3,23GB – 32bitový OS)
- Grafická karta: NVIDIA GeForce GT330M 1GB
- Operační systém: Windows 7 Professional 32bit

Test streamování proběhl v lokální síti z důvodu nízké šířky pásma připojení k internetu (ADSL rychlost odesílání 512kb/s). K serveru bylo připojeno 40 instancí, čímž se odstranily nedostatky předchozí metody zobrazení videa pomocí komponenty QuickTime. Kompatibilita zobrazení byla testována v prohlížečích Mozilla Firefox verze 3.6.15, Google Chrome verze 10.0.648.151, Opera 11.01 a Internet Explorer verze 5, 6, 7, 8. Výsledky testu jsou následující:

Titulek	Rychlost Stahování	Limit Stahov...	Fw ...	Rychlo...	Limit O...	F
vlc.exe	1,009	0,041		10,126	0,041	
Process 7424	1,009	0,041		10,126	0,041	
0.0.0.0:8080		0,041			0,041	
127.0.0.1:7180		0,041			0,041	
10.0.0.139:554		0,041			0,041	
10.0.0.139:59650	1,008	0,041			0,041	
10.0.0.139:59651	0,000	0,041		0,000	0,041	
10.0.0.144:8080		0,041		0,246	0,041	
10.0.0.144:8080		0,041		0,253	0,041	
10.0.0.144:8080		0,041		0,258	0,041	
10.0.0.144:8080		0,041		0,261	0,041	
10.0.0.144:8080		0,041		0,252	0,041	
10.0.0.144:8080		0,041		0,241	0,041	
10.0.0.144:8080		0,041		0,251	0,041	
10.0.0.144:8080		0,041		0,251	0,041	
10.0.0.144:8080		0,041		0,254	0,041	
10.0.0.144:8080		0,041		0,253	0,041	
10.0.0.142:8080		0,041		0,214	0,041	

Obrázek č. 32 – VLC player zatížení sítě, měřeno pomocí programu NetLimiter3



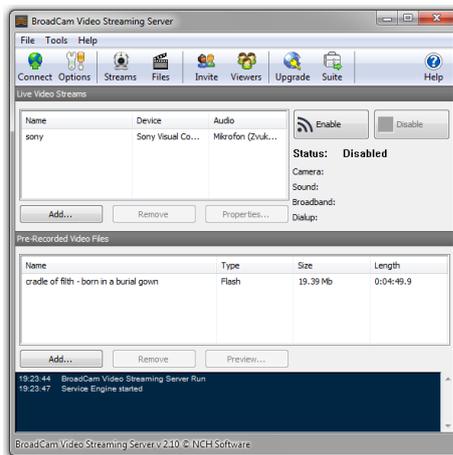
Obrázek č. 33 – VLC player zatížení procesoru

6.2.5.2 BroadCam Video Streaming Server

BroadCam Live Video Streaming software je placený streamovací software společnosti NCH Software. Tento streamovací server je určen k vysílání předem nahraných klipů a živého videa z webových kamer. Jednou ze základních předností BroadCamu je jeho jednoduchost a funkcionalita. Program lze spustit na většině operačních systémů společnosti Microsoft, od Windows XP až po Windows 7, přehrávaný stream není závislý na operačním systému, spouští se v internetovém prohlížeči moderních operačních systémů Windows, Mac nebo Linux.

Nastavení programu je relativně jednoduché, pro streamování předem nahraných klipů využijeme nabídky Pre-Recorded Video. Po nahrání videa, nastavení portů, nastavení veřejné adresy, kvality obrazu a spuštění streamu nabídne program dvě možnosti přehrání videa, pro pomalejší připojení nabídne posloupnost JPEG obrázků, pro rychlejší plynulé video. Obě možnosti si uživatel může spustit pomocí přehrávače Windows Media Player nebo pomocí Flash playeru.

Pro živé videostreamy z bezpečnostních kamer je nutné nainstalovat Audio Video Capture Filter, který je dostupný po podepsání SDK na stránkách výrobce (stejný postup jako v případě komponenty ActiveX z kapitoly č. 6.2.3). Tento zachytávací filtr umožní přenos dat z bezpečnostní kamery stejným způsobem, jako přenáší data klasická webkamera.



Obrázek č. 34 - Rozhraní programu BroadCam Video Streaming Server

Testování

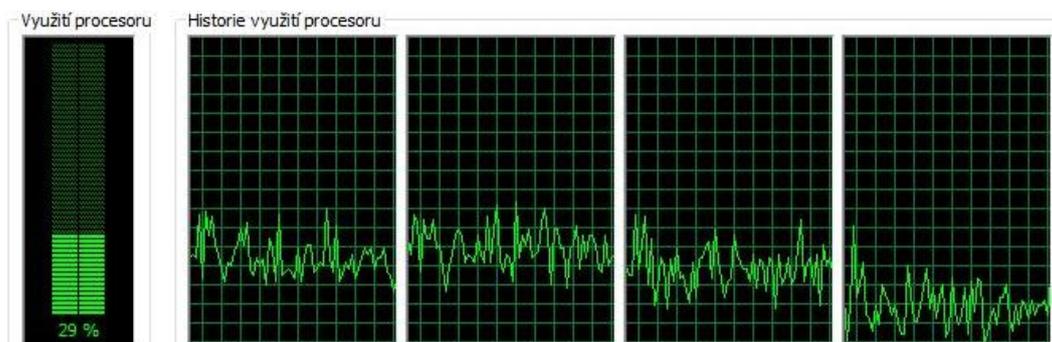
Realizace streamování videa pomocí BroadCam Video Streaming Serveru je jednodušší než u streamování prostřednictvím VLC playeru. Veškerá komunikace s programem BroadCam běží na jednom portu (defaultně 86), je tedy velmi jednoduché nastavit stream na

veřejnou adresu. Program byl nainstalován na stejný hardware jako v případě VLC playeru a Broadcam taktéž provádí dekódování jako VLC player.

Test streamování proběhl v lokální síti z důvodu nízké šířky pásma připojení k internetu (ADSL rychlost odesílání 512kb/s). K serveru bylo připojeno 40 instancí. Kompatibilita zobrazení byla testována v prohlížečích Mozilla Firefox verze 3.6.15, Google Chrome verze 10.0.648.151, Opera 11.01 a Internet Explorer verze 5, 6, 7, 8. Výsledky testu jsou následující:

Titulek	Rychlost Stahování	Limit Stahov...	Fw ...	Rychlo...	Limit O...
BroadCam Video Streaming Ser...	0.027	0.041	<input type="checkbox"/>	4.801	0.041
Process 6592		0.041	<input type="checkbox"/>		0.041
Process 3208	0.027	0.041	<input checked="" type="checkbox"/>	4.801	0.041
0.0.0.0:86		0.041	<input type="checkbox"/>		0.041
0.0.0.0:1935		0.041	<input type="checkbox"/>		0.041
10.0.0.144:86		0.041	<input type="checkbox"/>	0.047	0.041
10.0.0.144:86		0.041	<input type="checkbox"/>	0.160	0.041
10.0.0.144:86		0.041	<input type="checkbox"/>	0.124	0.041
10.0.0.144:86		0.041	<input type="checkbox"/>	0.087	0.041
10.0.0.144:86		0.041	<input type="checkbox"/>	0.134	0.041

Obrázek č. 35 - BroadCam zatížení síť, měřeno pomocí programu NetLimiter3

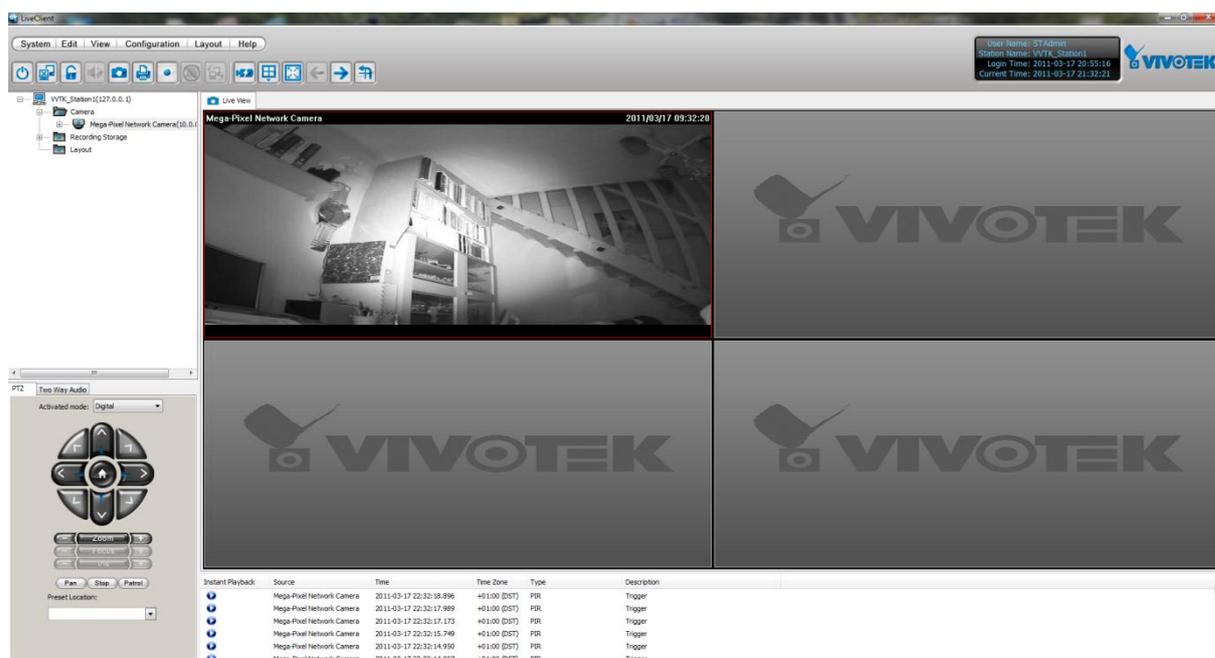


Obrázek č. 36 - BroadCam zatížení procesoru

6.2.5.3 Vivotek ST7501

ST7501 je zdarma dodávaná záznamová klient-server aplikace společnosti Vivotek. ST7501 je propracovanější následovník lokálního záznamového softwaru ST3402, umožňující nejen stálý záznam z IP kamer a videoservertů společnosti Vivotek, ale i sledování vstupů a výstupů z kamer, detekci pohybu a jejich následné vyhodnocení. Nastavení aplikace je uživatelsky přívětivé, po nainstalování na server a spuštění se v programu vyhledají automaticky všechna podporovaná zařízení na síti a uživatel si vybere, která zařízení chce do pohledů přidat a následně je v programu nastaví podle požadavků. Po zprovoznění aplikace a nastavení routeru na veřejnou IP adresu je možné se na server připojit vzdáleně z jakéhokoliv místa pomocí klientské části aplikace a vzdáleně obsluhovat software.

Mimo záznamu z kamer je možné aplikaci využít jako streamovací server až pro 32 kamer s relativně neomezeným počtem sledujících klientů (závislé na hardware serveru a velikosti datové linky). Pokud je kamera již přidána v aplikaci, je už nastavena i pro streamování a stačí pouze zjistit URL adresu streamu. Poté lze stream z kamery otevřít v jakémkoliv přehrávači, umožňujícím přehrávání z URL adresy (Apple QuickTime, Windows media server, VLC player, apod.).



Obrázek č. 37 - Rozhraní aplikace ST7501

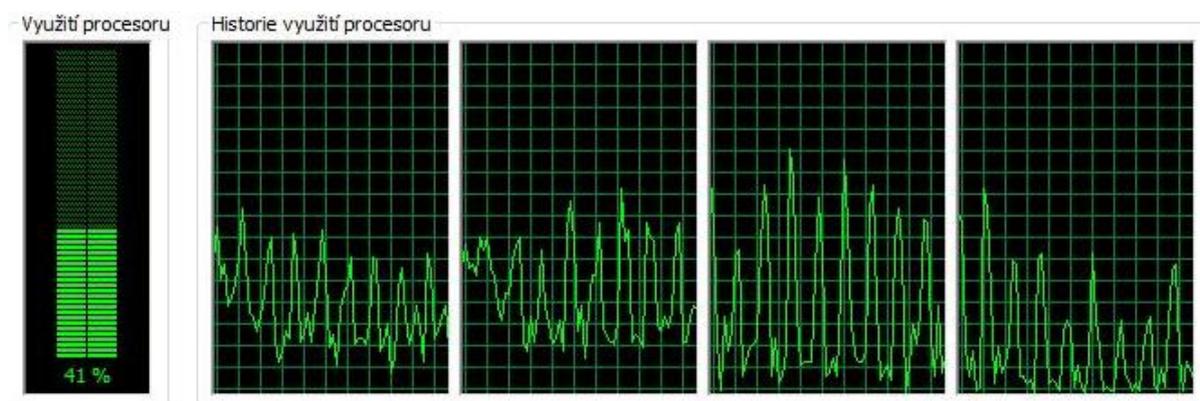
Testování

ST7501 je primárně záznamový software určený pro podporu produktů společnosti Vivotek, který umožňuje streamování videa. Jeho nevýhodou tedy je, že podporuje pouze produkty Vivotek. Realizace streamování je po přidání kamery snadná, jedním kliknutím zjistíme URL adresu streamu a vše je připraveno na streamování videa. Program ST7501 byl nainstalován na stejný hardware jako v předchozích případech.

Test streamování proběhl v lokální síti z důvodu nízké šířky pásma připojení k internetu (ADSL rychlost odesílání 512kb/s). K serveru bylo připojeno deset instancí. Program ST7501 neumí překódovat stream, proto měl každý stream 0,9 Mbitů. Kompatibilita zobrazení byla testována v prohlížečích Mozilla Firefox verze 3.6.15, Google Chrome verze 10.0.648.151, Opera 11.01 a Internet Explorer verze 5, 6, 7, 8. Výsledky testu jsou následující:

VAIO		Aplikace: 109	▼ Blokátor	Vytvořit filtr:	Jednotky:	Nastavení
UL rychlost: 8.046		Procesy: 63	▼ Limiter	Zóny	Mbits	O aplikaci
DL rychlost: 0.838		Spojení: 144	▼ Statistiky	Oprávnění		
Seznam spojení						
	Titulek	Rychlost Stahování	Limit Stahov...	Fw ...	Rychlo...	Limit O... F
Rozložení	vmstreamingserver.exe	0.835	0.041		8.043	0.041
Hlavní	Process 3112		0.041			0.041
Kompaktní	Process 3180	0.835	0.041		8.043	0.041
Plný	0.0.0.0:555		0.041			0.041
Vlastní	127.0.0.1:8777		0.041			0.041
	127.0.0.1:8816		0.041			0.041
O aplikaci	10.0.0.139:80	0.833	0.041			0.041
	10.0.0.144:50500		0.041		0.804	0.041
	10.0.0.144:50501	0.000	0.041		0.000	0.041
	10.0.0.144:50502		0.041		0.804	0.041
	10.0.0.144:50503	0.000	0.041		0.000	0.041
	10.0.0.144:51250		0.041		0.804	0.041

Obrázek č. 38 - ST7501 zatížení sítě, měřeno pomocí programu NetLimiter3



Obrázek č. 39 - ST7501 zatížení procesoru

Mezi další záznamové softwary, umožňující streamování videa, patří například Milestone nebo NUUO. Obě tyto aplikace jsou založeny na architektuře klient-server, pro streamování je nutná autentizace a nejsou proto vhodné na zobrazení videa na www prezentacích.

6.3 Souhrn

První část případové studie dokázala ověřit praktické předpoklady, které byly položeny v kapitole č. 4. Kompresce MJPEG, která nemá žádné pohybově kompenzované snímky a je pouze posloupností JPEG snímků, má největší datové toky. Její následovník (kompresce MPEG-4) již zavádí pojem pohybové kompenzace a její účinek je znatelný, datový tok je proti MJPEGu přibližně čtvrtinový. H.264 přináší zlepšení algoritmů pohybové kompenzace a datové toky jsou přibližně pětina oproti MJPEGu.

V druhé části případové studie jsou popsány některé z mnoha možných způsobů realizace zaintegrování obrazu z IP kamer do webových stránek. Jsou zde předloženy metody od vložení obnovujících se obrázků, až po vložení videa včetně zvuku.

Každá z možností má své výhody a nevýhody, které jsou uvedeny v následující tabulce č. 10.

Metoda	Požadavky	Výhody	Nevýhody
Vyčítání obrázků z kamery	- HTML stránka - veřejná IP adresa - JavaScript	- nezávislost na operačním systému a prohlížeči - jednoduchost	- velká zátěž kamery - sčítání datového toku - bez zvuku - malý počet uživatelů (desítky)
Odesílání obrázků na FTP server	- HTML stránka - FTP server - JavaScript	- neomezený počet uživatelů - nezávislost na operačním systému a prohlížeči - nízký datový tok - jednoduchost	- bez zvuku
Microsoft ActiveX	- HTML stránka - veřejná IP adresa - Active X komponenta	- plynulé video - možnost zvuku - jednoduchost	- sledování pouze z prohlížeče Internet Explorer - nutnost instalace ActiveX komponenty - velké datové toky - sčítání datového toku - velká zátěž kamery - malý počet uživatelů (10)
Apple Quick Time	- HTML stránka - veřejná IP adresa - QuickTime plugin na straně klienta	- nezávislost na operačním systému a prohlížeči - jednoduchost - plynulé video - možnost zvuku	- velká zátěž kamery - sčítání datového toku - malý počet uživatelů (10)
Streamování obrazu	- HTML stránka - veřejná IP adresa - streamovací server	- nezávislost na operačním systému a prohlížeči - jednoduchost - plynulé video - možnost zvuku - neomezený počet uživatelů	- velké nároky na hardware počítače - relativně obtížná realizace

Tabulka č. 10 - Možnosti vložení obrazu do www stránek

7 Závěr

Digitální kamery patří k nejrychleji rostoucímu segmentu trhu s výrobky pro sledování a ochranu majetku. Tato práce měla za cíl objasnit vlivy použité komprese v IP kamerových systémech na zatížení datové linky a na zatížení hardwaru počítače a uvést, jaké komprese se využívají, jejich vývoj a principy.

V rešeršní části práce bylo cílem položit teoretické základy kamerových systémů jako celku v porovnání s analogovým zobrazením obrazu. Autor zde uvedl rozdíly mezi způsoby snímání obrazu, světovými video standardy, rozlišením, digitální barevnou hloubkou a barevnou prezentací. Dále bylo cílem detailně analyzovat jednotlivé komprese využívané v IP kamerových systémech. Konkrétně byly analyzovány komprese MJPEG, MPEG-x a H.26x z hlediska vývoje a algoritmů, které používají ke komprimování videa. Další kapitola pojednávala o streamování videa, základech streamování, formátech pro streamování videa, přizpůsobování se zahlcené síti, logování a serverovým strukturám.

V praktické části byl proveden test zatížení datové linky s ohledem na rozlišení dané IP kamery a s ohledem na kompresi, využívanou ke komprimování videa. V další části kapitoly autor představil možnosti vložení obrazu z kamer na webové stránky, včetně tvorby streamovacího serveru. Byly zde uvedeny metody vyčítání obrazu z kamery, odesílání obrázků na FTP server, vložení obrazu pomocí komponenty Microsoft ActiveX a pluginu Apple QuickTime a streamování obrazu. U každé metody byly uvedeny výhody a nevýhody, náročnost dané metody na realizaci a hardware počítače a kompatibilita zobrazení v jednotlivých (v současnosti nejvíce využívaných) prohlížečích, konkrétně v Internet Exploreru, v Mozilla Firefox, Google Chrome, Opera a v Safari. Nejjednodušší a nejméně náročnou metodou zobrazení obrazu z IP kamery neomezenému počtu návštěvníků je metoda odesílání obrázku na FTP server a jeho následné obnovování. Nejvíce náročnou metodou, jak na hardware počítače, tak z hlediska realizace, je metoda streamování obrazu. Výhodou této metody je možnost zobrazení plynulého videa včetně zvuku.

8 Seznam použitých zdrojů

- [1] Joanna Lewadowska, IP Emerges the Buzzworld in Modern Video Surveillance Technology
URL: <http://www.frost.com/prod/servlet/press-release.pag?docid=203060324>
- [2] Anthony C. Caputo: Digital Video Surveillance and Security. Burlington: Butterworth-Heinemann, 2010. ISBN 978-1-85617-747-4.
- [3] Iain E. G. Richardson: H.264 and MPEG-4 video compression. UK: John Willey & Sons, 2003. ISBN 0-470-84837-5.
- [4] Tomáš Loveček, Petr Nagy: Bezpečnostné systémy – Kamerové bezpečnostné systémy. Žilina: Žilinská univerzita v Žilíně, 2008. ISBN 978-80-8070-893-1.
- [5] Dušan Sovič: Huffmanovo kódovanie,
URL: <http://pakuj.host.sk/single/huffman.html>
- [6] Huffmanovo kódování,
URL: http://janban.misto.cz/MAIL/_huffmancz.htm
- [7] Kompresie textu – Metoda RLE,
URL: <http://www.cs.vsb.cz/benes/vyuka/pte/texty/kompresie/ch02s01.html>
- [8] K. R. Rao, P. C. Yip: The transform and data compression handbook. USA: CRC Press, 2001. ISBN 0-8493-3692-9.
- [9] Yun Q. Shi, Huifang Sun: Image and video compression for multimedia engineering: fundamentals, algorithms, and standarts. USA: CRC Press, 2000. ISBN 0-8493-3491-8.
- [10] John Miano: Compressed image file formats – JPEG, PNG, GIF, XBM, BMP. Canada: ACM press, 1999. ISBN 0-201-60443-4.
- [11] David Salomon: Data compression: The complete reference. USA: Springer, 2004. ISBN 0-387-40697-2.
- [12] Tinku Acharya, Ping-Sing Tsai: JPEG2000 Standard for image compression – Concepts, algorithms and VLSI architectures. USA: John Wiley & Sons, 2005. ISBN 978-0-471-48422-6.
- [13] Iain E. G. Richardson: Video Codec Design: Developing image and video compression systems. UK: John Willey & Sons, 2002. ISBN 0-471-48553-5.

- [14] Joan L. Mitchell, William B. Pennebaker, Chad E. Fogg, and Didier J. LeGall: MPEG video compression standard. USA: Kluwer Academic Publishers, 2002. ISBN 0-306-46983-9.
- [15] Mohammed Ghanbari: Standard codecs: Image compression to advanced video coding. UK: The Institution of Electrical Engineers, 2003. ISBN 0-85296-710-1.
- [16] Khalid Sayood: Introduction to data compression. San Francisco: Morgan Kaufman Publishers, 2006. ISBN 978-0-12-620862-7.
- [17] Raymond Westwater, Borko Furht: Real-time video compression: Techniques and algorithms. USA: Kluwer Academic Publishers, 1996. ISBN 978-0-7923-9787-8.
- [18] Keith Jack: Video Demystified. USA: Elsevier Inv., 2005. ISBN 0-7506-7822-4.
- [19] David Austerberry: The technology of video & audio streaming. UK: Focal press, 2005. ISBN 0-240-80580-1.
- [20] Michael Topic: Streaming media demystified. USA: McGraw-Hill, 2002. ISBN 0-07-138877-X.
- [21] Joe Follansbee: Get streaming! USA: Focal Press, 2004. ISBN 0-240-80559-3.

9 Seznam použitých tabulek

Tabulka č. 1 – Světové video standardy.....	17
Tabulka č. 2 – Přehled rozlišení	21
Tabulka č. 3 - Přehled nejpoužívanějších rozlišení videa a výpočet jejich datových toků a požadavků na kapacitu pro desetibitové kódování jasu	25
Tabulka č. 4 - Kódové kombinace	27
Tabulka č. 5 - Huffmanovo kódování DC koeficientů.....	31
Tabulka č. 6 - Huffmanova tabulka pro AC koeficienty.....	33
Tabulka č. 7 - Predikátory pro bezztrátové kódování	36
Tabulka č. 8 - MPEG-4 rozlišení obrázků.....	48
Tabulka č. 9 - Porovnání kompresí v různých rozlišeních.....	73
Tabulka č. 10 - Možnosti vložení obrazu do www stránek.....	88

10 Seznam použitých obrázků

Obrázek č. 1 - Konvergence digitálního světa	15
Obrázek č. 2 - Diagram prokládaného skenování	17
Obrázek č. 3 - Prokládání (interlacing) - pouze pohybující se objekty jsou ovlivněny	18
Obrázek č. 4 – Odstranění prokládání (deinterlacing) – odstraňuje rozostřené řádky	18
Obrázek č. 5 - Digitální rozlišení a HDTV monitor.....	20
Obrázek č. 6 - (vlevo) vytisknutý obrázek z analogové bezpečnostní kamery v rozlišení 4CIF připojené k digitálnímu enkodéru (704 × 480). (vpravo) Vytisknutý obrázek stejné oblasti s použitím megapixelové kamery.	21
Obrázek č. 7 - Vzory barevných rozdílů	23
Obrázek č. 8 - Bitový strom Huffmanova kódování	26
Obrázek č. 9 - Rozdíl mezi sekvenčním a progresivním kódováním.....	30
Obrázek č. 10 – Několikvrstvá pyramida v hierarchickém módu	30
Obrázek č. 11 - Blokový diagram sekvenčního kódovacího algoritmu založeného na DCT ..	31
Obrázek č. 12 - Rozdělení do bloku 8 x 8 pixelů.....	31
Obrázek č. 13 – Cik-cak pořadí DCT koeficientů.....	32
Obrázek č. 14 - Dvoudimenzní pole hodnot pro Huffmanovo kódování.....	33
Obrázek č. 15 - Progresivní kódování pomocí spektrální selekce a postupné aproximace.....	35

Obrázek č. 16 – Prostorová vazba mezi pixelem, který bude kódován, a třemi dekódovanými susedy	36
Obrázek č. 17 - Struktura makrobloku	39
Obrázek č. 18 - MPEG-1 skupina obrázků (GOP - group of pictures)	40
Obrázek č. 19 - MPEG-1 skupina obrázků - seřazení kotvících obrázků	40
Obrázek č. 20 - MPEG-1 hierarchie	42
Obrázek č. 21 - Příklad MPEG-1 plátů	42
Obrázek č. 22 - DCT kódování snímků (a) a polí (b).....	44
Obrázek č. 23 - Dočasná škálovatelnost.....	45
Obrázek č. 24 - Video scéna ukazující více video objektů	47
Obrázek č. 25 - MPEG-4 složení vrstev	48
Obrázek č. 26 – Neprůhledný (a) a průhledný (b)VOP.....	49
Obrázek č. 27 - Makroblok PB-snímku	55
Obrázek č. 28 - Porovnání datových toků při různých rozlišeních	72
Obrázek č. 29 - Test javascriptu pro obnovení obrázku v prohlížeči Mozilla Firefox.....	77
Obrázek č. 30 - Zobrazení obrazu z kamery pomocí komponenty ActiveX.....	79
Obrázek č. 31 - Zobrazení všech parametrů pomocí VitaminCtrl	80
Obrázek č. 32 – VLC player zatížení sítě, měřeno pomocí programu NetLimiter3	83
Obrázek č. 33 – VLC player zatížení procesoru	83
Obrázek č. 34 - Rozhraní programu BroadCam Video Streaming Server	84
Obrázek č. 35 - BroadCam zatížení sítě, měřeno pomocí programu NetLimiter3.....	85
Obrázek č. 36 - BroadCam zatížení procesoru.....	85
Obrázek č. 37 - Rozhraní aplikace ST7501	86
Obrázek č. 38 - ST7501 zatížení sítě, měřeno pomocí programu NetLimiter3	87
Obrázek č. 39 - ST7501 zatížení procesoru	87

11 Přílohy

11.1 Slovník cizích pojmů

16CIF – rozlišení obrázku 1408×1152

3GPP – partnerský projekt, jehož cílem bylo vyvinout síť třetí generace mobilních telefonů

4CIF – rozlišení obrázku 704×576

ADSL – v současnosti nejčastěji využívaný typ připojení k internetu

AIX – proprietární UNIXový operační systém od společnosti IBM

Apache server – webový server s otevřeným kódem

Aproximace – přibližná hodnota čísla nebo nahrazení čísla číslem blízkým

ASCII – kódová tabulka definující znaky anglické abecedy a jiné znaky používané v informatice

Autentizace – proces ověření identity subjektu

Autorizace – proces prověření a získání přístupu k informacím nebo funkcím

Bps – jednotka přenosové rychlosti; bit za sekundu

CCTV – uzavřený televizní okruh; užití kamer ke sledování majetku

CIF – rozlišení obrázku 352×288

CMYK – barevný model založený na míchání barev, složeno z barev – azurová, purpurová, žlutá, černá

CPU – procesor, základní součást počítače, která vykonává strojový kód

CRT – klasická katodová obrazovka

DCT – diskrétní kosinová transformace

DIFF – rozdíl mezi dvěma soubory

Doména – jednoznačný identifikátor počítače v síti internetu

Dpi – jednotka určující počet obrazových bodů (pixelů) na palec (dots per inch)

DVR – zkratka pro digitální video rekordér

DVS – zkratka pro digitální video zabezpečení

EDGE – technologie datových přenosů pomocí GPRS v mobilních sítích

Entropie – jeden z nejdůležitějších a nejzákladnějších termínů v teorii pravděpodobnosti, teorii informace, matematice, fyzice a mnoha dalších oblastech vědy. Entropie je používána všude tam, kde se hovoří o pravděpodobnosti možného stavu daného systému nebo soustavy.

FDCT – zkratka pro dopřednou diskrétní kosinovu transformaci

Firewall – síťové zařízení sloužící k zabezpečení a řízení provozu na počítačové síti

Fps – snímkovací frekvence, počet snímků za sekundu

Framework – softwarová struktura sloužící jako podpora při programování, většinou obsahuje návrhové vzory, podpůrné programy, API knihovny

FreeBSD – UNIXový operační systém

FTP – protokol pro přenos souborů mezi počítači pomocí počítačové sítě

GOB – zkratka pro skupinu bloků

GOP – zkratka pro skupinu obrázků

GPRS - mobilní datová služba

HDTV – formát vysílání s výrazně vyšším rozlišením proti klasickým formátům (NTSC, PAL, SECAM), v současnosti nabízí rozlišení 1920×1080

Hosting – prostor pro webové stránky na cizím serveru v síti internet

HP-UX – operační systém od společnosti Hewlett-Packard založený na UNIXu

HTML – značkovací jazyk pro vytváření www stránek

HTTP – internetový protokol pro výměnu hypertextových dokumentů ve formátu HTML

HTTPS – nadstavba http, umožňující zabezpečit spojení

Hub – aktivní prvek počítačové sítě, umožňující její větvení

Hyperlink – hypertextový odkaz

IDCT – zkratka pro inverzní diskretní kosinovu transformaci

IETF – organizace vyvíjející a podporující internetové standardy

IP – datový protokol používaný pro přenos dat přes paketové sítě

ISDN – digitální síť integrovaných služeb

ITU – mezinárodní telekomunikační unie

JPEG – metoda ztrátové komprese používané pro ukládání počítačových obrázků

Kbps – jednotka přenosové rychlosti

Konvergence – pojem označující sblížování, propojování technologií

Korelace – vzájemný vztah mezi dvěma veličinami nebo procesy.

Kvantovaný signál – signál, který nemá spojitý průběh, ale skokem se mění, přičemž může nabývat pouze omezeného počtu úrovní.

LCD – tenké ploché zobrazovací zařízení, nástupce CRT monitorů

Metadata – strukturovaná data o datech

MIME – typ internetového média, dvoudílný identifikátor formátu souboru na internetu

MMS – internetový streamovací protokol vytvořený společností Microsoft

Multicast – metoda přeposílání IP dat z jednoho zdroje skupině více koncových stanic

Multiplexování – proces, ve kterém je více analogových signálů nebo digitálních datových toků kombinováno do jednoho signálu

NTSC – standard kódování analogového televizního signálu, vznikl v Americe

Open source – počítačový software s otevřeným zdrojovým kódem

Outsourcing – vyčlenění různých podpůrných a vedlejších činností společnosti a smluvní svěřeni jejich správy jinému subjektu

Paket – v informatice blok dat přenášený v počítačových sítích

PAL - standard kódování analogového televizního signálu, využívaný v Evropě

Peer-to-peer – architektura počítačových sítí, ve které spolu komunikují přímo jednotliví uživatelé, klient-klient

Pixel – nejmenší jednotka digitální rastrové grafiky

Plugin – software, který pracuje jako doplňkový modul jiné aplikace a rozšiřuje tak její funkčnost

QCIF - rozlišení obrázku 176×144

QXGA - rozlišení obrázku 2048×1536

Redundance – informační nebo funkční nadbytek, větší množství prvků než je potřeba

RGB – barevný model, používaný ve všech monitorech a projektorech, skládá se z barev: červená, zelená, modrá

Router – aktivní síťové zařízení, které přeposílá IP diagramy směrem k jejich cíli

RTCP – řídicí protokol pro distribuci zvuku a videa v reálném čase

RTP – protokol standardizující paketové doručování po internetu

RTSP – protokol pro streamování videa po internetu

SECAM - standard kódování analogového televizního signálu, vznikl ve Francii

Klient-server – síťová architektura, ve které je jeden počítač nadřazen jinému počítači

Smart karty – plastová karta s integrovaným čipem

Solaris – UNIXový operační systém společnosti Sun Microsystems

SQCIF – rozlišení obrázku 128×96

TCP – jeden ze sady internetových protokolů, protokol TCP garantuje spolehlivé doručování a doručování ve správném pořadí

Textura – technika, umožňující dodat realistický vzhled virtuálnímu trojrozměrnému modelu

UDP – jeden ze sady internetových protokolů, protokol UDP nedává záruky na spolehlivé doručování a doručování ve správném pořadí

Unicast – zasílání paketů pouze jedinému cíli

URL – řetězec znaků s definovanou strukturou, sloužící k jednoznačné specifikaci umístění zdroje informací na internetu

VCEG – zkratka pro Video Coding Experts Group, neboli skupinu odborníků, která pracovala na vývoji standardů kódování videa

VCR – zařízení sloužící k záznamu televizního vysílání na magnetický pásek - ve formě videokazety

VGA - rozlišení obrázku 640 × 480

VHS – standard pro nahrávání a přehrávání na kazetových videorekordérech

VLC – kódování s variabilní délkou

VO – zkratka pro video objekt

VoIP – technologie umožňující telefonování po internetu

VOP – zkratka pro video object plane, neboli zobrazení celé scény

VS – zkratka pro video scénu