

Анализ на цифрови видео формати и методи за конвертиране за пренос по комуникационни мрежи

Станимир Садинов, Йордан Кънев

Digital video format analysis and conversion methods for communication networks: In this paper is performed analysis of digital video encoding and methods for transcoding of video content for IPTV purposes. Live video traffic with different video codecs is simulated and obtained results are presented.

Key words: MPEG, IPTV, Transcoding, Video streaming.

ВЪВЕДЕНИЕ

За предаване на телевизионен сигнал по комуникационни мрежи е необходимо прилагане на подходящи техники за компресиране на видеопотока преди излъчването в зависимост от свободната честотна лента. Изследователи от ITU-T са сметнали, че MPEG компресията е най-подходящото решение за тази цел. Те са стандартизирали протокол H.264, който е еквивалентен на MPEG-4 (part 10) от групата Moving picture expert group (MPEG). Стандарта H.264/MPEG-4 се използва и за телевизия с висока резолюция (HDTV), докато по-стария стандарт MPEG-2 се използва все още за телевизия със стандартна резолюция (SDTV).

Друго решение е представено от Society of Motion Picture and Television Engineers (SMPTE), базирано на Windows Media 9 (WM-9) и е стандартизирано под името VC-1. Има характеристики подобни на MPEG-4, но по-добра интеграция с персоналните компютри.

Конвертирането (Transcoding) се използва за преобразуване скоростта на потока на входен видео сигнал с подходяща скорост за крайното устройство. Например за визуализиране на видео с висока резолюция на екрана на мобилен телефон. Входният видео сигнал може да бъде от файл или предаване на видеосигнал на живо.

ПРИНЦИП НА ВИДЕО КОДИРАНЕ

Целта на видео кодеците е да представят изначално аналоговата информация в цифров вид. Поради харектара на аналоговите видео сигнали, които представляват черно-бялата (luma) и цветовата информация поотделно, честа първа стъпка, залегната в дизайна на видео кодеците е представянето и съхраняването на изображението в YCbCr цветово пространство. Това конвертиране осигурява две ползи: първо - подобрява степента на компресия чрез осигуряване на независимостта на цветовите сигнали; и второ - отделя luma (черно-белия) сигнал, който е по-важен за възприятието, от chroma (цветовия) сигнал, който е по-маловажен и може да бъде представен с по-ниска резолюция за да се постигне по-ефикасна компресия. Често за представянето на различното количество информация за тези три канала (Y,Cb,Cr) се използва запис от 3 коефициента разделени помежду си с двоеточие - Y:Cb:Cr (chroma subsampling).

Различните кодеки използват различни chroma subsampling коефициенти, както е уместно в зависимост от техните нужди от компресия. Схемите за видео компресия при Web и DVD използват 4:2:0 шаблон, а DV стандарта - 4:1:1. Професионалните видео кодеки предназначени за работа при много по-високи bit rate-ове и за запис на много по-голямо количество цветова информация за нуждите на следпродажионна обработка използват 3:1:1 (рядко срещано), 4:2:2 и 4:1:1 съотношения. Пример за такива кодеки са DVCPRO50 и DVCPROHD на Panasonic (4:2:2), HDCAM-SR (4:4:4) на Sony или HDD5 (4:2:2) отново на Panasonic. Новият Prores HQ 422 кодек на Apple също използва 4:2:2 схема. Съществуват и още кодеки, които използват 4:4:4 шаблон, но се срещат по-рядко и изключително само в къщите за следпродажионна

обработка. Няма проблем обаче видео кодеците да използват и RGB цветово пространство. При тях не се цели семплиране на червения, зеления и синия канал в различни съотношения, тъй като има по-слаба мотивация за това от гледна точка на човешките възприятия - може единствено синият канал да бъде семплиран в по-ниско съотношение.

Известно количество пространствено и времево подсемплиране може да бъде извършено за да се намали първичния темп на данните (data rate, bit rate) преди основния кодиращ процес. Най-популярната такава трансформация е 8x8 дискретното косинусово преобразуване. Кодеки използващи вълничкова трансформация (wavelet transform) също навлизат на пазара, особено при камери с обработка на RAW изображения в отъски с движение. Изходът от трансформацията първо бива квантуван по ниво (квантизиран, quantized), след което върху получените квантувани стойности се прилага entropy кодиране. При използване на дискретно косинусово преобразуване, коефициентите обикновено се сканират зигзаг-образно и entropy кодирането обикновено комбинира броя на последователните нулеви квантувани коефициенти и стойността на следващия ненулев такъв в един символ и освен това има начини да индицира, че всички оставащи квантувани коефициенти са 0. Entropy кодиращия метод обикновено използва таблици с променливи по дължина кодове (variable-length coding tables). Някои кодери могат да извършват компресирането на видеото в няколко стъпки (n-стъпково кодиране, обикновено n=2), което е по-бавен процес, но води до потенциално по-добро качество.

Декодирация процес включва, в максимално възможна степен, инверсия на всеки един етап извършен при кодирането. Стъпка, която не може напълно да бъде обяснена е квантуването по ниво (квантизирането). Извършва се възможно най-добрия начин за обръщане на процеса и доближаване на получените стойности максимално до първоначалните. Този процес често се нарича "обратно квантуване" или "деквантуване", въпреки че в същността си квантуването е необрратимо.

MPlayer или VLC Media Player съдържат множество популярни кодеки в преносима единична библиотека, налична за много операционни системи, включително Windows, Linux и Mac OS X. Това решава и много проблеми в Windows, свързани с недобре инсталирани кодеки или конфликти между тях [5,7].

ХАРАКТЕРИСТИКИ НА КОМПРЕСИРАНО ВИДЕО

Цифровото видео се състои от серии правоъгълни растерни цифрови изображения, бързо редуващи се с постоянна скорост. В контекста на видеото тези изображения се наричат кадри. Скоростта, с която те се сменят се измерва в кадри в секунда (FPS - frames per second).

Тъй като всеки кадър е правоъгълно цифрово растерно изображение, то той се състои от растер от пиксели. Ако растера има ширина W пиксела и височина H пикела, казваме че размера на кадъра е WxH.

Пикселите имат само една характеристика - техният цвят. Цветът на пиксела е представен чрез фиксирано количество битове. Колкото повече са тези битове, толкова по фини вариации на цветовете може да пресъздадем. Това се нарича дълбочина на цвета на видеото (CD - color depth).

Примерен видео материал може да има продължителност (T) 1 час (3600 сек.), размер на кадъра 640x480 пиксела (WxH) при дълбочина на цвета 24 бита (CD) и скорост на кадрите 25 кад/сек (FPS). Това видео има следните характеристики:

- **пиксели за кадър** = $640 * 480 = 307\ 200$
- **битове за кадър** = $307\ 200 * 24 = 7\ 372\ 800 = 7.37$ Mbits
- **скорост на цифровия поток** (BR - bit rate) = $7.37 * 25 = 184.25$ Mbits/sec
- **размер на целия видео материал** (VS - video size) = 184 Mbits/sec * 3600 sec = $662\ 400$ Mbits = $82\ 800$ MBytes = 82.8 GBytes

Най-важните характеристика са скоростта на цифровия поток (BR - bit rate) и размера на целия видео материал (VS - video size). Формулите, свързващи тези параметри с всички останали са:

$$BR = W * H * CD * FPS$$

$$VS = BR * T = W * H * CD * FPS * T, \text{ някои второстепенни формули са:}$$

$$\text{пиксели за кадър} = W * H$$

$$\text{пиксели за секунда} = W * H * FPS$$

$$\text{битове за кадър} = W * H * CD$$

Описаните формули са коректни за некомпресирано видео. Заради относително високата му скорост на цифровия поток, екстензивно се използва видео компресия. При компресираното видео всеки кадър изисква по-малък процент битове от оригинала. Приемайки, че алгоритъм за компресия сбива входните данни с коефициент CF (compression factor), то тогава скоростта на цифровия поток (BR - bit rate) и видео размера (VS - video size) стават:

$$BR = W * H * CD * FPS / CF$$

$$VS = BR * T / CF$$

Трябва да се отбележи, че не е необходимо всички кадри да са компресирани с един и същи коефициент (CF). На практика не са, така че коефициента на компресия (CF) е средната стойност на компресия за всички кадри взети заедно.

Уравнението за скоростта на цифровия поток може да бъде записано и по следния начин, комбинирайки коефициента на компресия (CF) и дълбочината на цвета (CD):

$$BR = W * H * (CD / CF) * FPS$$

Стойността (CD/CF) представлява средното количество битове за пиксел (BPP - bits per pixel). Например ако имаме дълбочина на цвета 12 bits/pixel и алгоритъм, който постига 40x компресия, то тогава BPP е 0.3 (12/40). Така, че в случая с компресирано видео, формулата за скоростта на цифровия поток е:

$$BR = W * H * BPP * FPS$$

Въщност тази формула е валидна и за некомпресирано видео, тъй като може да приемем коефициента на компресия (CF) за 1, което води до равенство между средното количество битове за пиксел (BPP) и дълбочината на цвета (CD).

Скоростта на цифровия поток е мярка за темпото на данните в този поток. В случай на некомпресиран видео материал, тази скорост кореспондира директно с качеството на видеото (скоростта на цифровия поток е правопропорционална на всяка характеристика, засягаща видео качеството). Скоростта на цифровия поток е важен параметър при пренасянето на видео по канал, тъй като този канал трябва да бъде в състояние да поддържа тази скорост. Скоростта на цифровия поток е важна и при съхранението на видеото, защото размера на видео материала е пропорционален, както именно на тази скорост, така и на продължителността. Скоростта на цифровия поток на некомпресиран материал в много случаи е твърде висока за да бъде пренесена по комуникационни мрежи. За да се намали значително се използва видео компресия.

Средното количество битове за пиксел (BPP) е мярка за ефикасността на компресията. Пълноцветно (true color) видео без никаква компресия може да има BPP 24 bits/pixel. Цветовото подсемплиране може да намали BPP до 16 или 12

bits/pixel. Използването на JPEG компресия върху всеки кадър може да намали ВРР до 8 или дори до 1 bits/pixel. MPEG-1, MPEG-2 или MPEG-4 схемите за компресия може да постигнат дробни стойности (по-малки от 1) за ВРР.

Има алгоритми на компресия, които поддържат ВРР стойността постоянна през цялата продължителност на материала. В този случай и видео изхода е с постоянна скорост на цифровия поток (CBR, constant bit rate). Това видео е подходящо за пренасяне в реално време по канал с фиксирана честотна лента, където няма възможност и за буфериране.

Възможно е обаче не всички кадри да бъдат компресирани с един и същи фактор (кофициент). Тъй като видео качеството е по-силно засегнато при по-сложните сцени, някои алгоритми постоянно се опитват да нагласят ВРР стойността. Поддържат я висока при компресирането на сложни сцени и ниска при другите. По този начин се постига най-доброто качество при най-ниска средна скорост на цифровия поток. Използвайки този метод, скоростта на цифровия поток също е променлива, тъй като следва изменението на ВРР.

КОНВЕРТИРАНЕ В РАЗЛИЧНИ ВИДЕО ФОРМАТИ ЧРЕЗ СПЕЦИАЛИЗИРАН СОФТУЕР - FFmpeg, MEncoder

Тестваната постановка включва персонален компютър с подходяща операционна система, с инсталирани програмни продукти FFmpeg и MEncoder и подходящ виде файл за образец. Конвертирането се изпълнява със следният набор от команди:

MPEG-1: `ffmpeg -i input_file.avi -qscale 3 -vcodec mpeg1video -r 24000/1001 -aspect 16:9 output_file.avi`

`mencoder -force-avi-aspect 16:9 -ofps 24000/1001 -ovc lavc -lavcopts vcodec=mpeg1video:vqscale=3 input_file.avi -o output_file.avi`

Конвертира образца с MPEG-1 кодек при постоянна степен на квантуване 3, задава 16:9 формат на картината и 23.98 кад/сек за изходното видео.

MPEG-2: `ffmpeg -i input_file.avi -qscale 3 -vcodec mpeg2video -r 24000/1001 -aspect 16:9 output_file.avi`

`mencoder -force-avi-aspect 16:9 -ofps 24000/1001 -ovc lavc -lavcopts vcodec=mpeg2video:vqscale=3 input_file.avi -o output_file.avi`

Конвертира образца с MPEG-2 кодек при постоянна степен на квантуване 3, задава 16:9 формат на картината и 23.98 кад/сек за изходното видео.

H.263+: `ffmpeg -i input_file.avi -qscale 3 -vcodec h263p -r 24000/1001 -aspect 16:9 output_file.avi`

`mencoder -force-avi-aspect 16:9 -ofps 24000/1001 -ovc lavc -lavcopts vcodec=h263p:vqscale=3 input_file.avi -o output_file.avi`

Конвертира образца с H.263+ кодек при постоянна степен на квантуване 3, задава 16:9 формат на картината и 23.98 кад/сек за изходното видео.

H.263 (във Flash Video): `ffmpeg -i input_file.avi -qscale 3 -vcodec flv -r 24000/1001 -aspect 16:9 output_file.avi`

`mencoder -force-avi-aspect 16:9 -ofps 24000/1001 -ovc lavc -lavcopts vcodec=flv:vqscale=3 input_file.avi -o output_file.avi`

Конвертира образца с H.263 кодек (използван във Flash Video) при постоянна степен на квантуване 3, задава 16:9 формат на картината и 23.98 кад/сек за изходното видео.

MPEG-4 Part 2 (DivX 4/5): `ffmpeg -i input_file.avi -qscale 3 -vcodec mpeg4 -r 24000/1001 -aspect 16:9 output_file.avi`

`mencoder -force-avi-aspect 16:9 -ofps 24000/1001 -ovc lavc -lavcopts vcodec=mpeg4:vqscale=3 input_file.avi -o output_file.avi`

Конвертира образца с MPEG-4 Part 2 (DivX 4/5) кодек при постоянна степен на квантуване 3, задава 16:9 формат на картината и 23.98 кад/сек за изходното видео.

MPEG-4 Part 2, Microsoft вариант вер. 3 (DivX 3): `ffmpeg -i input_file.avi -qscale`

```
3 -vcodec msmpeg4 -r 24000/1001 -aspect 16:9 output_file.avi  
    mencoder -force-avi-aspect 16:9 -ofps 24000/1001 -ovc lavc -lavcopts  
vcodec=msmpeg4:vqscale=3 input_file.avi -o output_file.avi
```

Конвертира образеца с MPEG-4 Part 2 (Microsoft вариант версия 3, DivX 3) кодек при постоянна степен на квантуване 3, задава 16:9 формат на картина и 23.98 кад/сек за изходното видео.

Windows Media Video, вер. 1 (известен още като WMV7): `ffmpeg -i input_file.avi -qscale 3 -vcodec wmv1 -r 24000/1001 -aspect 16:9 output_file.avi`

```
    mencoder -force-avi-aspect 16:9 -ofps 24000/1001 -ovc lavc -lavcopts  
vcodec=wmv1:vqscale=3 input_file.avi -o output_file.avi
```

Конвертира образеца с WMV1 (Windows Media Video, версия 1) кодек при постоянна степен на квантуване 3, задава 16:9 формат на картина и 23.98 кад/сек за изходното видео.

Windows Media Video, вер. 2 (известен още като WMV8): `ffmpeg -i input_file.avi -qscale 3 -vcodec wmv2 -r 24000/1001 -aspect 16:9 output_file.avi`

```
    mencoder -force-avi-aspect 16:9 -ofps 24000/1001 -ovc lavc -lavcopts  
vcodec=wmv2:vqscale=3 input_file.avi -o output_file.avi
```

Конвертира образеца с WMV2 (Windows Media Video, версия 2) кодек при постоянна степен на квантуване 3, задава 16:9 формат на картина и 23.98 кад/сек за изходното видео.

Xvid (MPEG-4 Part 2): `ffmpeg -i input_file.avi -qscale 3 -vcodec libxvid -r 24000/1001 -aspect 16:9 output_file.avi`

```
    mencoder -force-avi-aspect 16:9 -ofps 24000/1001 -ovc xvid -xvidencopts  
fixed_quant=3:notrellis:vhq=0:nochroma_me input_file.avi -o output_file.avi
```

ffmpeg: Конвертира образеца с Xvid MPEG-4 Part 2 (ASP) кодек при постоянна степен на квантуване 3, задава 16:9 формат на картина и 23.98 кад/сек за изходното видео.

mencoder: Конвертира образеца с Xvid кодек при постоянна степен на квантуване 3. Изключени са Trellis квантуването (notrellis), цветовото изчисляване на движението (nochroma_me) и честотната основа (DCT) при изчисляване на движението (vhq=0). Задава 16:9 формат на картина и 23.98 кад/сек за изходното видео.

x264 (MPEG-4 Part 10 / AVC / H.264): `ffmpeg -i input_file.avi -vcodec libx264 -vpre hq -crf 20 -r 24000/1001 -aspect 16:9 output_file.avi`

```
    mencoder -force-avi-aspect 16:9 -ofps 24000/1001 -ovc x264 -x264encopts crf=22  
input_file.avi -o output_file.avi
```

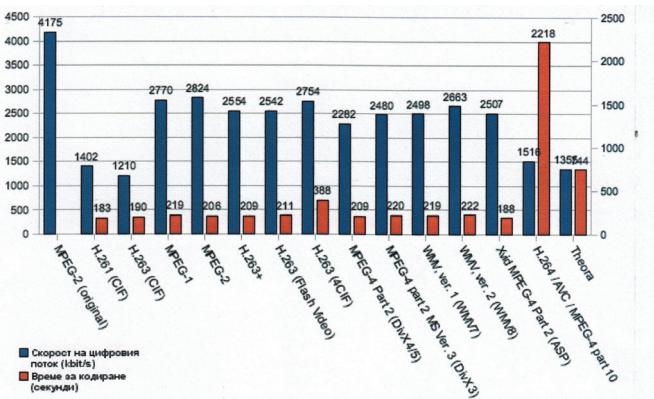
ffmpeg: Конвертира образеца с x264 видео кодек с фактор на постоянна скорост 20 (-crf 20, това е аналогичния метод за указване на постоянна степен на квантуване при x264) и предефиниран профил за качество hq (-vpre hq), задава 16:9 формат на картина и 23.98 кад/сек за изходното видео.

mencoder: Конвертира образеца с x264 кодек при фактор на постоянна скорост 22 (crf=22, аналогичен метод за указване на постоянна степен на квантуване при x264), задава 16:9 формат на картина и 23.98 кад/сек за изходното видео.

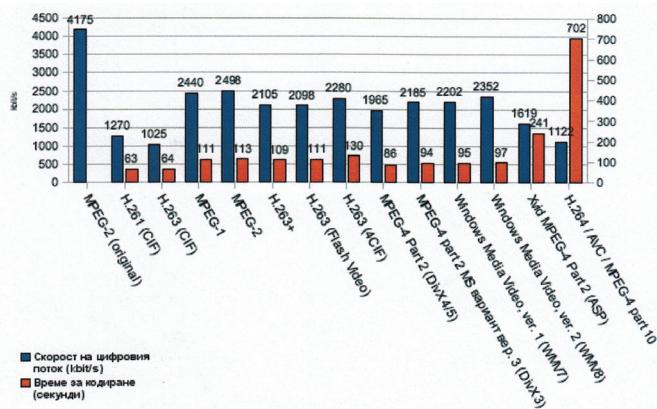
Theora: `ffmpeg2theora -v 5 --aspect 16:9 --framerate 24000:1001 -o output_file.ogv input_file.avi`

Конвертира образеца с Theora кодек при степен на качество 5, задава 16:9 формат на картина и 23.98 кад/сек за изходното видео.

Получените резултати са показани на фиг.1 и фиг.2.



Фиг. 1 - Конвертиране с FFmpeg



Фиг. 2 - Конвертиране с MEncoder

ЗАКЛЮЧЕНИЕ

Основният параметър, който трябва да се има предвид при подготвяне на видео материал за излъчване в реално време, е поддържаната скорост на комуникационния канал, по който ще се разпространява той. Изяснявайки този параметър, следващата ни задача е да определим кой видео кодек е най-подходящия за кодиране на видеото, така че да постигнем най-добро качество на картината при тази скорост. Най-простият начин за установяване на това е да конвертираме образец от материала в различни формати, указвайки предварително желаната скорост на изходния видео поток чрез използвания инструмент (FFmpeg или MEncoder) [6,7]. Трябва да се има определи дали скоростта може да варира в някакви граници. След приключване на конвертирането, преценяваме кой изходен образец е с най-добро качество и избираме съответния кодек. Освен с изходното качество, трябва да се съобразим и с фактори като изчислителна мощ, необходима за декодиране на видеото, защита от грешки, възможност за скалируемост и други, важни за конкретното приложение. Обикновено кодеците, постигащи добро качество при ниска скорост на цифровия поток имат и по-голяма сложност.

Задавайки директно желаната скорост на видеопотока, инструментите за кодиране динамично променят за всеки кадър ред параметри, влияещи върху

степента на компресия, а от там и върху качеството на изображението. Друг начин за понижаване скоростта на цифровия поток (освен чрез коефициента на компресия) е чрез понижаване на размера на изображението, броя кадри в секунда или дълбочината на цвета.

Чрез мощнни инструменти, като FFmpeg и MEncoder [6,7] всички тези обработки и много други могат да бъдат извършени без проблемно. Командният интерфейс на тези програми ги прави изключително гъвкави и удобни за употреба. Благодарение на него те могат да бъдат използвани в автоматизирани процеси по конвертиране и обработка на цифрово видео. Този софтуер е свободен и с отворен код, тъй като се разпространява под GNU GPL лиценз.

При проведените конвертирации, най-ниска скорост на цифровия поток, при запазване на еднакво качество на картина, се постига чрез видео кодеците H.263+, MPEG-4 part 2, Xvid , x264 и Theora. При x264 (реализация на MPEG-4 Part 10 /AVC/ H.264 стандарта) и Theora, обаче, и времето необходимо за кодиране е значително по-голямо. Наблюдава се, че видео формата x264 изисква по-голяма процесорна мощ за декодиране.

БЛАГОДАРНОСТИ

Настоящата статия е изготвена с финансова помощ на Европейския социален фонд. Русенският университет „Ангел Кънчев“ носи цялата отговорност за съдържанието на настоящия документ, и при никакви обстоятелства не може да се приеме като официална позиция на Европейския съюз или Министерството на образованието и науката.

Проект: № BG051PO001-3.3.06-0008 „Подпомагане израстването на научните кадри в инженерните науки и информационните технологии“

ЛИТЕРАТУРА

- [1] O'Driscoll, G., Next Generation IPTV Services and Technologies, John Wiley&Sons Inc., 2008.
- [2] Kanev J., Sadinov S., “Analyzing RTSP protocol features for managing video on demand server”, UNITECH 2012, Gabrovo Bulgaria, 2012.
- [3] Kanev J., Sadinov S., “Analysis of current methods and technologies for encoding, distribution and consumption of IPTV services”, ICEST 2012, Proc. of Papers, V.Tarnovo, Bulgaria, 2012
- [4] Балабанова И., Симулационно моделиране на трафика в телекомуникационните мрежи., сп. „Компютърни науки и комуникации“, Том 2, №2 (2013), БСУ, Бургас, ISSN:1314-7846, (11-20)стр.
- [5] <http://www.videolan.org>, VideoLan Project, Sep 2013
- [6] <http://www.ffmpeg.org>, FFmpeg Project, Sep 2013
- [7] <http://www.mplayerhq.hu>, MEncoder, Sep 2013

За контакти:

доц. д-р инж. Станимир Михайлов Садинов, Катедра “Комуникационна техника и технологии”, Технически университет - Габрово, тел.: 066/827-388, e-mail: murry@tugab.bg

маг. инж. Йордан Христов Кънев, редовен докторант в катедра “Комуникационна техника и технологии”, Технически университет – Габрово, тел.: 0888-142-987, e-mail: danchokanev@gmail.com

Докладът е рецензиран.