



2008

MPEG2 to MPEG4 video conversion  
under OS/2 and eComStation  
Sjoerd Visser

## Subjects

The art of illusion:

Some notes about digital video, sound and pictures

DivX and Vidx

Video conversion in practice

Using FFMPEG to convert EmperoarTV recordings to avi.

Using mencoder to convert a DVD to avi.

## Words per minute

Reading, typing and speaking rates can be expressed in words per minute (wpm).

<i>Speed</i>	<i>User</i>	<i>Effect</i>
50 wpm	:Very slow speaker	Boring
150 wpm	:Speaker in audio book	Listening comfortable
200 wpm	:Compressed speech	Does allow for full comprehension
300 wpm	:Normal conversation	Asks full attention to keep a fraction of it
400 wpm	:Political debater	Wordflow as weapon, disinformation
100 wpm	:Slide presentations	Hiding information, let them sleep
300 wpm	:Normal reading	Giving attention to relations
400 wpm	:Fast reading	Scanning text for the essentials

Source: <http://www.answers.com/topic/words-per-minute-1>

There is a major difference between scanning words (automated listening when everything goes as expected) and attentive listening: hearing words you did not hear before and comprehend their relations, when the subject catches you.

## The bitrates of the human brain (compared to PC's)

The total estimated human sensory input processed by our nervous system, including proprioception (the awareness of bodily movement) and other internal receptors (gut feelings) is about 400 gigabit per second. Most of this input is processed automatically (unconsciously) by nuclei. Say brain DMA controllers. Coming in to the working memory of our consciousness via the external senses (compare: mouse , keyboard, webcam) is only a small part of it:

10,000,000 bits/s	Eyes	Compressed (!) DVD movie	6,000,000 bits/s
1,000,000 bits/s	Skin		
100,000 bits/s	Ears	384 KiBps AC3	400,000 bits/s
100,000 bits/s	Smell		
1,000 bits/s	Taste		

Only a fraction of it, about 2,000 bits/s (V.22bis 600 baud modem) has the focus of our attention. The rest is processed subconsciously. Just as your webcam can run without your notice in the background (:-)

So we may appear to see wide screen high definition video, but only a fraction of it can be perceived consciously. Ask a painter. Or get lost in a DVD movie.

Most auditory information is only listened to partly (ask your partner).

And the signals of our bladder are mostly ignored (don't run away now).

Remember that are **our eyes** use **10 MegaBit/s** of visual information to produce a very detailed 3D wide screen impression of the world in our brain.

Professional digital **High Definition TV** camera's allow for 1,5 GigaBit/s. This stream is **highly compressed to 4-8 MegaBit/s on a PAL MPEG-2 DVD**. But this only yields tiny 2D frames of 720 x 576 pixels, 25 times per second!

How does our brain do this trick? Why do we see a perfect world with only 10 MegaBit/s, whilst the higher uncompressed bitrate of a DVD is much larger?

Schopenhauer (*Die Welt als Wille und Vorstellung*) noticed that what the brain perceives, is not what the eye sees (the world upside down), but more what the person expects to see.

Our picture of the world is mainly composed of our knowledge and or fantasies about it. And not by what the eyes actually see. Magicians make use of it.

When new to the world, we are mentally (conceptually) blind. We must first and intensively exploit the objects, before we can see them. Watch with a toddler a toddler book or movie like Tik Tak. You are bored, but the toddler is excited.

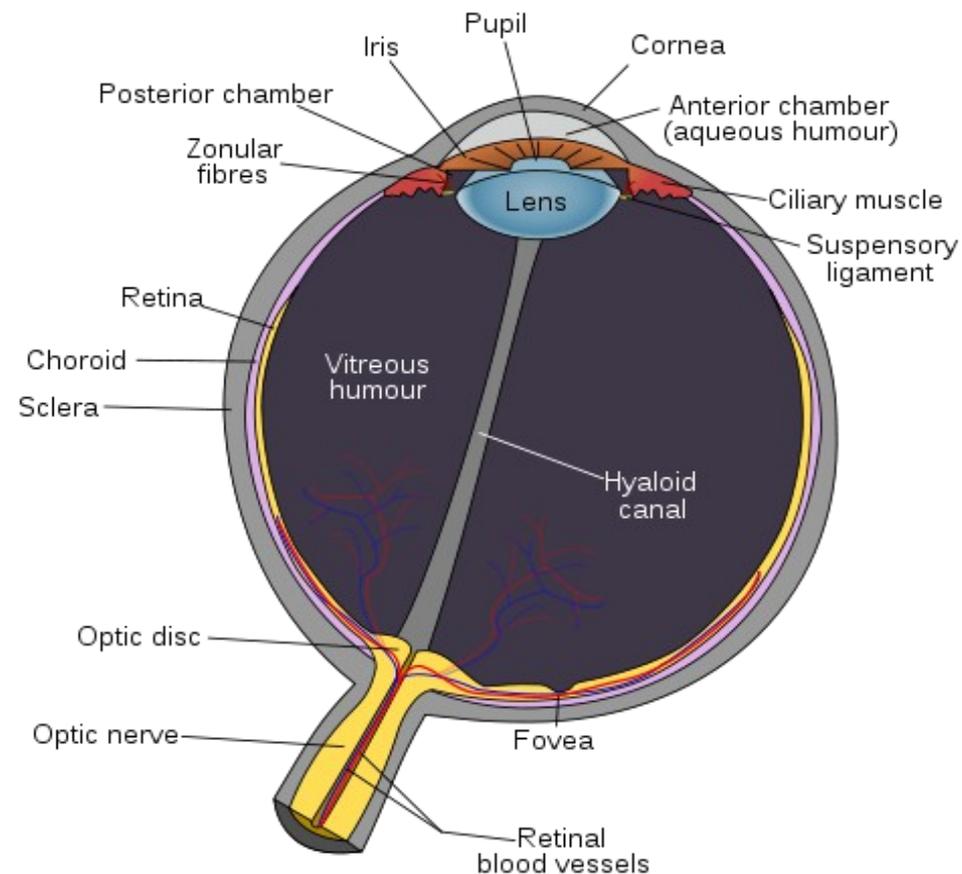
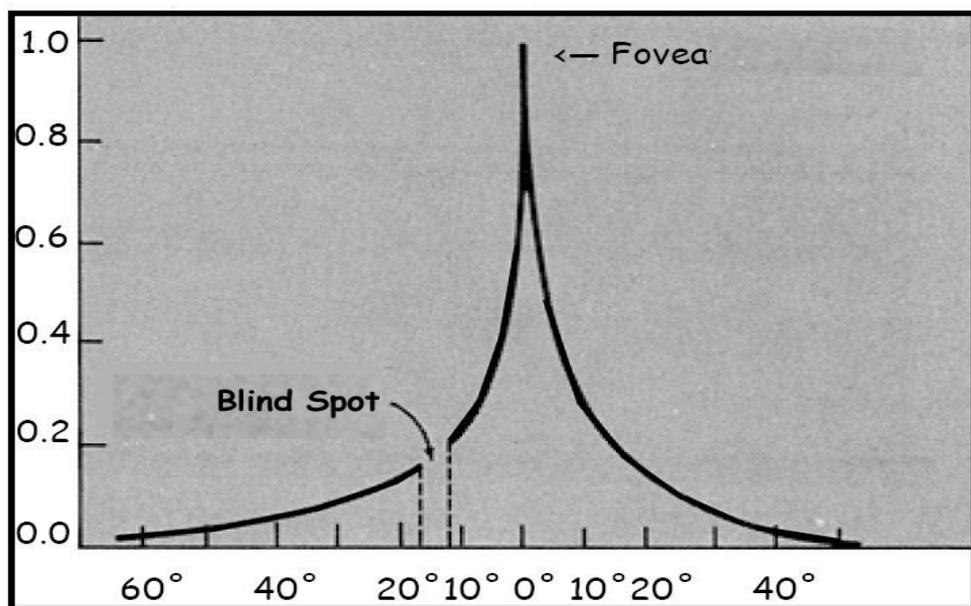
By the time we are grown up (conditioned by our culture), we perceive things without actually seeing much of them (result: cultural misunderstanding).

Most of the eyes sensory input goes to the *fovea centralis* (**the yellow spot**) of the retina.

Here is a high density of coins that allow for sharp **central color vision**.

There is also a place (*macula*, **blind spot**) where we can see nothing. In the periphery we actually see little.

But when the eyes scan the environment, our brain fills in the gaps.



Pictures from:

<http://en.wikipedia.org/wiki/Fovea>

## **Slowdown for a moment...**

This Presentation could be considered as streaming media.

I hope it does not stream too fast, so you can follow the text.

We now go to the more practical aspects .

## Tools I use for video conversion

- **MPlayer** to view sources and results.
- **Mencoder** used to convert DVD's.
- **Ffmpeg** which is specialized in MPEG formats.

You find them at Hobbes (<http://hobbes.nmsu.edu>)  
and Paul Smedleys site: <http://smedley.info/os2ports>.

- **Priority** from Mark Kimes to set idle priorities.

<ftp://ftp.netlabs.org/pub/fm2/fm2utils.zip>

- **A batch scheduler** like **que**.

<http://hobbes.nmsu.edu/download/pub/os2/util/schedule/que09.zip>

**PRIORITY.EXE -- sets priority as specified, then runs command.**

A public domain launch-with-priority program by M. Kimes.

Usage: **PRIORITY [/Quiet] <class> <delta> [command and args...]**

Examples:

**PRIORITY /q 1 31 SWEEP DEL \*.BAK** (note /q before <class> and <delta>)

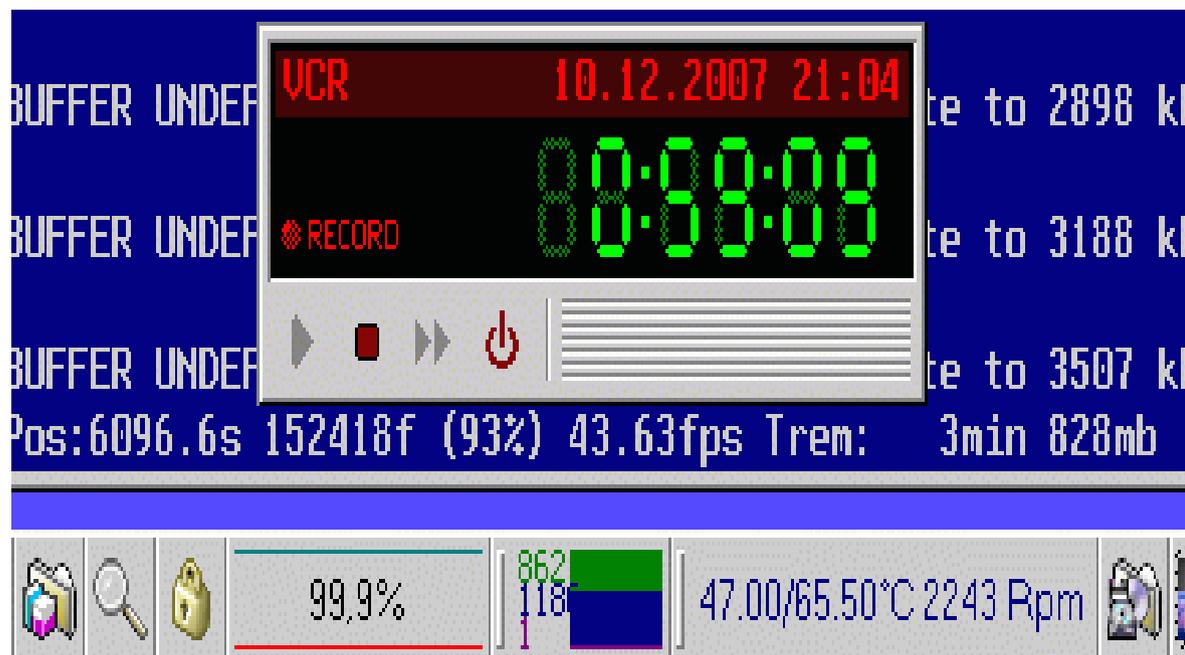
**PRIORITY 4 0 MYPROGRAM.EXE**

<class> must be between 1 and 4 (required argument).

<delta> must be between 0 and 31 (required argument).

If you use **mencoder** or **ffmpeg** without priority with standard priority (2 0) and **PRIORITY\_DISK\_IO=YES** your system will respond very slow.

Running them in lowest idle (0 0) priority, is fast (100% cpu-use) enough!



## Hardware needed to do video encoding

- The **encoding** work is done using **CLI** applications (ffmpeg, mencoder).
- So unlike video-editing, you don't need a fast video card for it.
- But to view the results with **MPlayer** the 2D acceleration of **SNAP** is handy.
- A processor and BIOS that support **thermal throttling** are necessary to prevent the processor overheating (expect hours of 100% cpu-use).
- A fast processor if you don't want to wait days (and make the conversions very energy costly).
- Gigabytes of free **hard disk** space are needed, preferably on a dedicated **JFS Volume**. Doing encoding on your boot drive with the WPS is very risky.

## Hardware used to view the results

- Personal computers with software media players (like MPlayer).
- Hardware media players (handheld or connected to external TV/monitor).
- Network and removable storage to transport the files.

## DVD sources can be found in libraries and on the internet.

Serenity (2005) PAL DVD9 (download torrent) - TPB - PmW-Fx

File Edit View History Bookmarks Tools Help

http://thepiratebay.org/torrent/4504026/Serenity\_(2005)\_PAL\_DVD9

Archief Actueel Computers Fora en NG OS/2 OS Nieuws Radio/TV Spel Psychiatrie Zoeken zolder Zwolle

[Zoek Torrents](#) | [Browse Torrents](#) | [Recente Torrents](#) | [TV-programma's](#) | [Muziek](#) | [Top 100](#)

Alles  Audio  Video  Programma's  Games  Anders  (zoek alleen in titels)

### Details voor deze torrent

http://zolder:4080 - MLdonkey: Wel

Parsing HTTP url

http://torrents.thepiratebay.org/4504026/Serenity\_

### Serenity (2005) PAL DVD9

Type: [Video > Films DVDR](#)

Files: 1

Grootte: 7.67 GiB (8239415296 Bytes)

Gesproken taal: Engels

Ondertiteling taal: Engels, Zweeds, Noors, Deens, Fins, Nederlands

Tag(s): [space](#) [pirate](#) [pirates](#)

Kwaliteit: +0 / -0 (0)

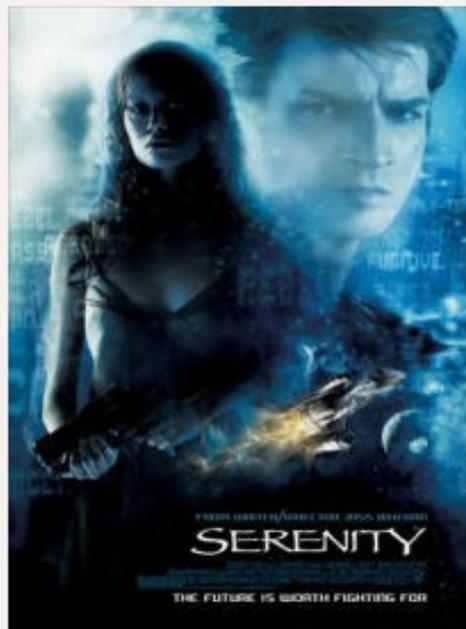
Geupload: 2008-11-12 20:00:37 GMT

Door: [Ubehage](#) MOD

Seeders: 1

Leechers: 22

Opmerkingen 0



THE FUTURE IS WORTH FIGHTING FOR

(D:10.0) (U:31.8) | sjoerd | MLNet 2.9.6: Multi-Network p2p client (Global Shares Direct Connect Gnutella G2 Fasttrack FileTP)

File Edit View History Bookmarks Tools Help

http://zolder:4080/

Archief Actueel Computers Fora en NG OS/2 OS Nieuws Radio/TV Spel Psychiatrie Zoeken zolder

Transfers Search Servers Statistics Options Help+ DL

eDonkey Table eMule MODs Bittorrent Table Overnet Kademia Gnutella

Down: 10.0 KB/s (0|10267), 12.74G Up: 31.8 KB/s (7|32546), 8.68G Shared(186): 308.38G

Brand	Seen %	Reqs %	B %	UL %	KB/s
	0 (0%)			0.0	
	0 (0%)			0.0	
	2.53G (89%)			9.1	
	57.9M (2%)			0.2	
	9.3M (0%)			0.0	
	44.5M (2%)			0.2	
	137.8M (5%)			0.5	
	79.7M (3%)			0.3	
	2.86G (100%)			10.2	

FC/2: H:\TVIX-SMB

H:\TVIX-SMB

..	UP-DIR	12-11-08	19:50
\$RECYCLE.BIN	SUBDIR	20-05-08	13:51
AUDIO	SUBDIR	25-05-08	2:32
PHOTO	SUBDIR	25-05-08	2:09
RECYCLER	SUBDIR	25-05-08	2:29
System Volum	SUBDIR	24-05-08	12:57
tvixfw	SUBDIR	24-05-08	13:21
VIDEO	SUBDIR	25-05-08	3:26

H:\MLDonkey

..	UP-DIR	12-11-08	19:50
directories	SUBDIR	12-11-08	20:55
files	SUBDIR	12-11-08	20:55

I use MLDonkey under Linux which has a nice web interface and supports many protocols

## Why take the effort of video encoding?

- Reducing a 4-8 GiB MPEG-2 DVD (4000-8000 kbit/s) to a more compressed MPEG-4 AVI-file (700-1300 kbit/s) saves space and bandwidth.

VIDEO: [XVID] 624x352 12bpp 23.976 fps 982.0 kbps (**119.9 kbyte/s**)

Opening audio decoder: [mp3lib] MPEG layer-2, layer-3

AUDIO: 48000 Hz, 2 ch, s16le, **128.0 kbit/8.33%** (ratio: 16000->192000)

Selected audio codec: [mp3] afm: mp3lib (mp3lib **MPEG layer-2, layer-3**)

- Because double layered DVD's were expensive, **DivX** became popular to put movies on cheap CD's and play them with a DivX Media Player.
- Today, with cheap (single layer) DVD writers you can put six 720 MiB AVI files on one DVD. Or six times as much movies on a removable disk.
- Most software and hardware **Media players** will support them.
- You can play them via the **network** (home media centre).
- And transport them faster.



**A fine NAS / media player with NTFS support, 2 USB ports, Gigabit LAN with SMB and NFS networking support (avoid NAS devices with FAT32).**

## About containers and codecs

The files that contain the streaming media are called containers.

**Examples of containers** are:

AVI	MS Audio Video Interleave
divx	DivX 6 format
FLS	Adobe's Flash video
MKV,MKA	Matroska
MOV,QT	Apple's Quicktime movie
MP4	MPEG standard based on Apple's Quicktime
OGG,OGM	Ogg Vorbis Media
VOB	MPEG Video Object file of DVD

[http://en.wikipedia.org/wiki/Comparison\\_of\\_container\\_formats](http://en.wikipedia.org/wiki/Comparison_of_container_formats)

**The container** holds audio streams (tracks), a video stream and metadata and can also contain chapters, support menu's, subtitles and the like. The container **header** shows their content.

→The streams are put together in one file or stream by a **multiplexer**.

→The reverse process, **demuxing**, is done by the media player

The audio and video streams are **compressed** and **decompressed** with **codecs**: Devices (chips) or programs (DLLs) that can encode or decode digital data streams. A media player or encoder can use codecs as **plug-ins**.

**Video codec standards**: MPEG-1, MPEG-2, MPEG-4 (Advanced) Simple Profile, H.264, VC-1/WMV, RealVideo, Theora.

Note that the **DivX**, **Xvid**, **FFmpeg MPEG-4 codecs** are special implementations of the MPEG-4 Part 2 standard.

**Lossy audio codecs**: MP3, WMA, RealAudio, Vorbis, AAC, AC3, DTS.

**Lossless audio codecs**: APE, FLAC, ALAC, SHN, WV.

→Visual content including subtitles and menu's is represented graphically. You need an OCR to extract the content as text (srt-file).

The container header (here AVI) defines how the content should be interpreted. So Mplayer and mencoder do not need file-extensions.

```

FC/2: Matrix1.avi
View: L:\DVD\Matrix_U1400_SUB\Matrix1.avi
00000  52 49 46 46   FA DB FF 3F   41 56 49 20   4C 49 53 54   RIFF- ?AVI LIST
00010  CA 01 00 00   68 64 72 6C   61 76 69 68   38 00 00 00   ..hdrlavih8...
00020  40 9C 00 00   00 00 00 00   00 00 00 00   10 09 00 00   @f.....
00030  3D 41 02 00   00 00 00 00   02 00 00 00   00 00 00 00   =A.....
00040  C0 02 00 00   30 01 00 00   00 00 00 00   00 00 00 00   L...0.....
00050  00 00 00 00   00 00 00 00   4C 49 53 54   B4 00 00 00   .....LIST|...
00060  73 74 72 6C   73 74 72 68   38 00 00 00   76 69 64 73   strlstrh8...vids
00070  58 56 49 44   00 00 00 00   00 00 00 00   00 00 00 00   XVID.....
00080  01 00 00 00   19 00 00 00   00 00 00 00   03 A2 02 00   .....ó..
00090  92 1D 01 00   00 00 00 00   00 00 00 00   00 00 00 00   fe.....
000A0  C0 02 30 01   73 74 72 66   28 00 00 00   28 00 00 00   L.0.strf(...(...
000B0  C0 02 00 00   30 01 00 00   01 00 18 00   58 56 49 44   L...0.....XVID
000C0  00 CC 09 00   00 00 00 00   00 00 00 00   00 00 00 00   .||.....
000D0  00 00 00 00   69 6E 64 78   38 00 00 00   04 00 00 00   ...indx8.....
000E0  02 00 00 00   30 30 64 63   00 00 00 00   00 00 00 00   ....00dc.....
000F0  00 00 00 00   06 BE 2C 4F   00 00 00 00   18 0A 12 00   ....¥,0.....
00100  3F 41 02 00   1E C8 3E 4F   00 00 00 00   40 06 03 00   ?A...L>0....@...
00110  C4 60 00 00   4C 49 53 54   AA 00 00 00   73 74 72 6C   -`..LIST~...strl
00120  73 74 72 68   38 00 00 00   61 75 64 73   55 00 00 00   strh8...audsU...
    
```

## Short history of DivX

- DivX ('DIV-ex') started in 1998 when an Microsoft codec named MPEG-4v3 was reverse engineered by Jérôme Rota and Max Morice .
- Unlike the name suggests, **MPEG-4v3** was not MPEG-4 compliant codec.
- Microsoft spread it in two forms: A free crippled version that only could decode ASF Files in Media Player. And a payware version that could encode.
- But as it was spread by Microsoft it could have become a standard.
- The hacked **DivX ;-)** v3 codec could both, but was illegal.
- To develop an improved legal version **DivXNetworks**, Inc.(2000) hired Rota and others. With the help of some open source programmers (Project Mayo) they developed an **OpenDivX codec** (jan. 2001), which allowed for more containers incl. Audio Video Interleave (AVI) and 2 pass encoding techniques.
- Some months later an improved version of the codec ("encore2"), written by an DivX Networks employee was suddenly removed from the open source repository. This frustrated the open source developers.
- From this DivXNetworks (renamed to **DivX, Inc.** in 2005) developed the **DivX 4.0** (2001), **DivX 5.0** (2002) and **DivX 6.0** (2005) codecs.
- The closed source DivX codecs became standards that were supported by many hardware and software media players and encoders.

## From OpenDivX to Xvid (2001)

- In 2001 the open source developers used the "encore2" source code to start a new project: **Xvid** (an acronym for Divx).
- The Xvid homepage (<http://www.xvid.org>) releases the GPL-ed Xvid source, but no binary codecs to prevent legal issues with patents.
- But encoders and media players compiled with the in Hungary hosted **libavcodec** library of F(ast)F(oward)MPEG developer Fabrice Bellard and others (<http://mplayerhq.hu>) include the codec. They encompass ffmpeg, mencoder, mplayer, smplayer, kplayer (and WarpVision in the Hall of shame).
- The current Xvid codec has a better compression rate than the commercially available DivX 5, but these extra features are not supported by DivX certified hardware media players. But is still looks well.
- The Xvid video codec uses MPEG-4 Advanced Simple Profile (ASP) compression, which can be decoded by all MPEG-4 ASP compliant decoders.
- To be recognized by hardware media players as XviD 4 compatible the with ffmpeg end mencoder encoded video can be FFOURCE tagged as XviD instead of with *libavcodec's* FMP4 (Fast MP4) house label .

## Video compression is done essentially by :

- **Sampling** digital pictures (**frames**) at a frame rate that just prevents flicker.
- **Compressing frames** more or less loosely to **JPG** ('JAY-peg').
- **Comparing** subsequent **frames for change** (moving mouth on static background, zooming effects, vector movement of fore- or background).
- Trying to find the **best algorithm** that *only* encodes the change compared to a preceding (P-frame) or coming (B frame) of a full picture (I frame) .

Your movie could then have:

→ **I-Frames (Intra coded pictures)** which are full sized JPG frames that occur at a regular interval. They are the reference material for P- and B-frames.

→ **P-Frames (Predicted pictures)** store only the changes made in respect to the previous I-Frame. Consecutive P-frames tend to grow.

→ The best compression rates are achieved with **B-Frames (Bi-predictive pictures or delta-frames)** that allow the video encoder to store data with reference to both previous and coming I- and P-frames.

• The ways these changes should be detected and encoded are defined globally by the **Moving Pictures Experts Group**. But of course the implementations of the mostly closed source video encoders (codecs) will differ.

## Two pass encoding

Audio samples can be re-encoded in one pass: WAV to MP3 conversion.  
Because of the **lower bitrates** involved and the **acoustic model** (what the human ear cannot hear is discarded), less computing is needed.

Video can be re-encoded too in one pass using I- and P-frames, but for high quality video encoding using low bitrates the source has to be processed twice.

Analyzing the movie beforehand, allows for **B-frames** that reference to the *best fitting past and future frames*.

- During the **first pass** the input video stream is decoded and superficially re-encoded with the chosen codec in the output container.
- Meanwhile a **ffmpeg2pass-0.log** is made which analyses the amount of change encountered between consecutive frames.
- During the final **second pass** the optimal encoding algorithm for each frame can be predicted from the information about near past and future frames in **ffmpeg2pass-0.log**.

What can we expect of the by Player, mencoder and ffmpeg used **libavcodec** and lame libraries? And which container to choose?

First notice that most **container formats** that define the structure of the video/audio stream are **closed source**.

- The rather outdated **Audio Video Interleave AVI**-format from Microsoft, in which mencoder is specialized, is broadly supported, but has known limitations.
- The closed source Advanced Systems Format **WMV**-containers using **Windows Media Video (WMV)** codecs can more or less be played by MPlayer, but they make you dependent on Microsoft (a less realized limitation).
- MPlayer can play some closed source **RealMedia** content, but to use this closed source format as an output to mencoder will easily fail
- **Matroska** ([www.matroska.org](http://www.matroska.org)) is an promising open source video (mkv) and audio (mka) multimedia container format. But open source standards are not yet embraced by the industry. I always wonder why.

As the codecs concern, the choice is easier: The by FFMPEG and mencoder used ISO standard MPEG-4 **mpeg4** video codec is compatible with XVID and DivX. And the superior compressing **MP3** audio codec supported by *libmp3lame* and the less compressing Dolby 5.1 surround **AC3** audio codec supported by *libavcodec* are well supported ISO standards too.

## FFMPEG syntax

Transform MPG(2) to MP(eg)4:

```
ffmpeg -i input.mpg -acodec mp3 -vcodec mpeg4 output.mp4
```

Being more specific (one long string):

```
ffmpeg -i input.avi  
-acodec libmp3lame -ac 2 -ab 96  
-vcodec mpeg4 -b 800 -s 320x240 -r 24  
output.mp4
```

Explanation:

**-acodec libmp3lame -ac 2 -ab 96** : use for audio the libmp3lame codec with two audio channels (default mono) with an audio bitrate of 96 kbits/s (default 64 kbits/s).

**-vcodec mpeg4 -b 800 -s 320x240 -r 24** : use for video the mpeg4 codec with a bitrate of 800 kbits/s (default 200 kbits/s) and a frame size of 320x240 pixels at a frame rate of 24 times a second (default 25 fps).

FFMPEG will autodetect the container format of the input file. It will infer the container format of the output file from its extension (mp4, avi).

## More FFMPEG options

Use "ffmpeg | more" to see the options.

or "ffmpeg -formats | more" to see formats, codecs, filters and protocols.

This version of FFMPEG supports lame mp3 and many other codecs, devices and formats via its built-in libraries (statically compiled).

```
[F:\]ffmpeg > ffmpeg.txt
```

```
FFmpeg version SVN-r11573, Copyright (c) 2000-2008 Fabrice Bellard, et al.
```

```
configuration: --enable-gpl --enable-libmp3lame --enable-liba52 --enable-libgsm  
--enable-pp --enable-libfaad --enable-libfaac --prefix=/ffmpeg
```

```
libavutil version: 49.6.0
```

```
libavcodec version: 51.49.0
```

```
libavformat version: 52.4.0
```

```
libavdevice version: 52.0.0
```

```
built on Jan 20 2008 14:59:50, gcc: 3.3.5 (Bird Build 2007-06-10 14:30)
```

## My strategy to encode

Make **reusable batches** with options (output format, codecs, filters) that are supported by your media players.

- Quality options that worked before, should work again.
- Less typing and thus less typing errors.
- Use variables for parameters that are changed (name, bitrates)
- Promote the best batches to the **Templates** folder.

I also made two interactive REXX scripts that create batches.

Examples can be found in:

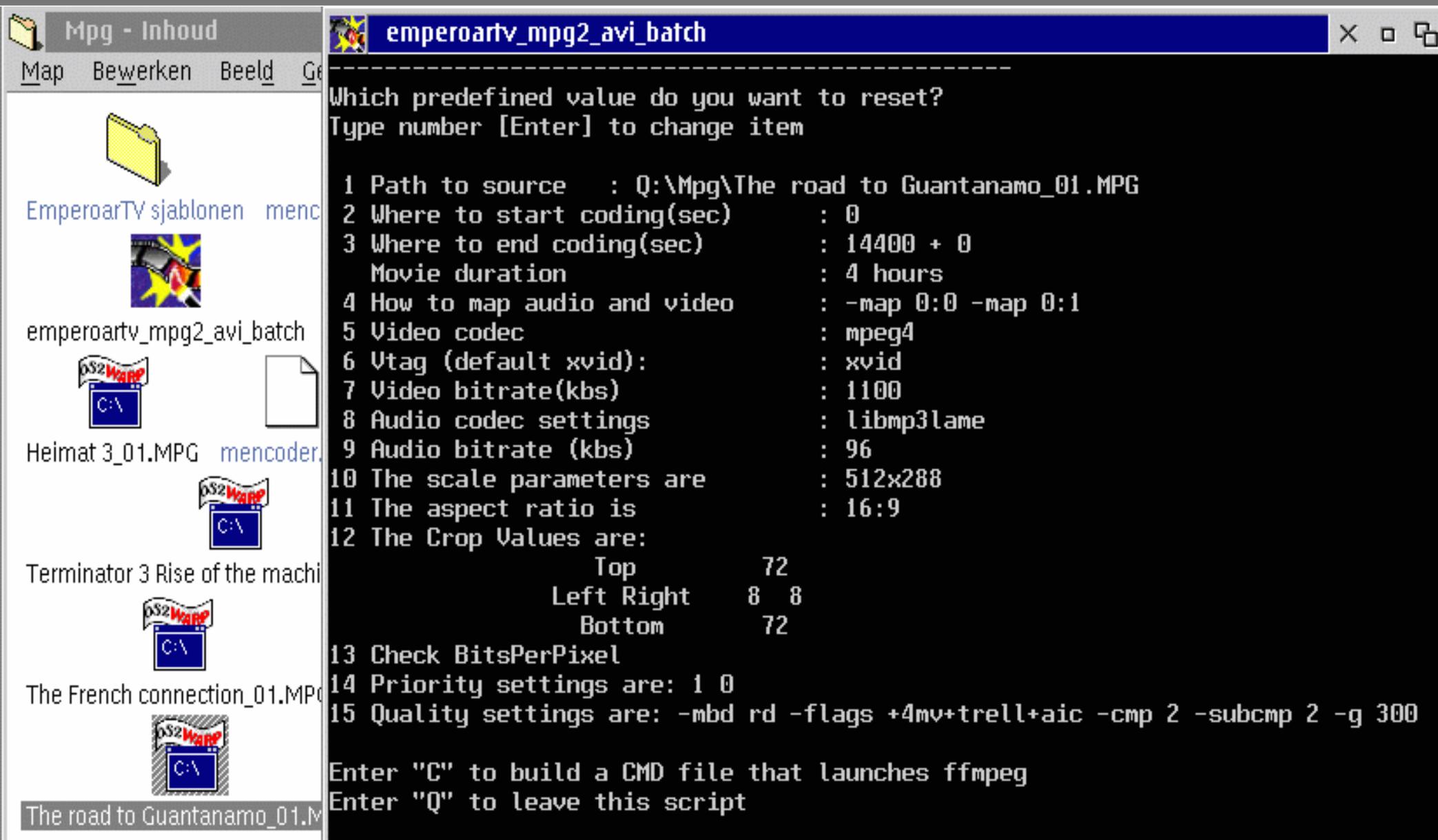
<http://www.sjoerd-visser.demon.nl/mplayer/index.html>

<http://www.sjoerd-visser.demon.nl/mplayer/extra>

## Description of EmperoarTV\_MPG2\_AVI\_BATCH

An interactive REXX script that creates batches to convert the by EmperorTV created MPEG-2 files to XVID compatible AVI.

- It does not do the encoding, it creates a batch-file for it.
  - The batch that starts FFMPEG can be investigated and tweaked first, before it is scheduled to run.
  - The two pass conversion, say 8 GiB DVD format to 1 GiB AVI will take several hours, so must run using idle priority.
- The OS/2 scheduler can run many batches at the same time, but to use the system resources more efficiently and loose less encoding time when PM crashes, I schedule them to run one after another with the batch scheduler que.



The screenshot shows two windows from an OS/2 desktop:

- File Explorer (Mpg - Inhoud):** Displays a folder containing several MPEG-2 files:
  - EmperoarTV sjablonen mencoder
  - emperoartv\_mpg2\_avi\_batch
  - Heimat 3\_01.MPG mencoder
  - Terminator 3 Rise of the machi
  - The French connection\_01.MPG
  - The road to Guantanamo\_01.M
- Terminal (emperoartv\_mpg2\_avi\_batch):** Shows the output of a script for converting an MPEG-2 file to MPEG-4. The text is as follows:
 

```

-----
Which predefined value do you want to reset?
Type number [Enter] to change item

 1 Path to source      : Q:\Mpg\The road to Guantanamo_01.MPG
 2 Where to start coding(sec)      : 0
 3 Where to end coding(sec)      : 14400 + 0
   Movie duration              : 4 hours
 4 How to map audio and video     : -map 0:0 -map 0:1
 5 Video codec                 : mpeg4
 6 Utag (default xvid):         : xvid
 7 Video bitrate(kbs)           : 1100
 8 Audio codec settings         : libmp3lame
 9 Audio bitrate (kbs)          : 96
10 The scale parameters are      : 512x288
11 The aspect ratio is          : 16:9
12 The Crop Values are:
      Top              72
      Left Right      8 8
      Bottom          72
13 Check BitsPerPixel
14 Priority settings are: 1 0
15 Quality settings are: -mbd rd -flags +4mv+trell+aic -cmp 2 -subcmp 2 -g 300

Enter "C" to build a CMD file that launches ffmpeg
Enter "Q" to leave this script
      
```

Usage: Drag the MPEG-2 file to the program object.  
Set options and create the batch file.

**REM Batch to convert " Q:\Mpg\The French connection\_01.MPG " to " Q:\Mpg\The French connection\_1100\The French connection.avi "**

**MD "The French connection\_1100"**

**CD "The French connection\_1100"**

**REM Here starts the first pass encoding which analyses the movie content and creates a log**

**PRIORITY 1 0 FFMPEG -i "Q:\Mpg\The French connection\_01.MPG" -map 0:0 -map 0:1 -ss 5 -t 7453 -pass 1 -vcodec mpeg4 -vtag xvid -f avi -b 1100k -mbd rd -flags +4mv+trell+aic -cmp 2 -subcmp 2 -g 300 -cropbottom 72 -cropleft 8 -cropright 8 -croptop 72 -s 592x336 -aspect 16:9 -acodec libmp3lame -ac 2 -ab 96k "Q:\Mpg\The French connection\_1100\The French connection.avi"**

**REM Here starts the second pass encoding that makes use of the log**

**DEL "Q:\Mpg\The French connection\_1100\The French connection.avi"**

**PRIORITY 1 0 FFMPEG -i "Q:\Mpg\The French connection\_01.MPG" -map 0:0 -map 0:1 -ss 5 -t 7453 -pass 2 -vcodec mpeg4 -vtag xvid -f avi -b 1100k -mbd rd -flags +4mv+trell+aic -cmp 2 -subcmp 2 -g 300 -cropbottom 72 -cropleft 8 -cropright 8 -croptop 72 -s 592x336 -aspect 16:9 -acodec libmp3lame -ac 2 -ab 96k "Q:\Mpg\The French connection\_1100\The French connection.avi"**

**EXIT**

## Something more about the options

**-ss 5 -t 7453** means start encoding at 4 seconds and stop 7453 seconds afterwards.

**-vcodec mpeg4 -vtag xvid -f avi** means use FFMPEG's standard mpeg4 codec, tag it as XVID (and thus Divx) compatible and put it in an avi format container.

**-mbd rd -flags +4mv+trell+aic -cmp 2 -subcmp 2 -g 300** these are high quality encoding options I cannot explain here.

**-s 592x336** means code for a 592x336 frame (screen).

A large frame allows for more details and encoding tricks, but also needs a higher bitrate.

A fast 2-3 GHz processor may be able to decompress a 500 kbs video stream on the fly to 768x432, but the slower processor of a hardware media player will not.

**-aspect 16:9** is the usual width:height aspect ratio of the DVD format.

If you chose the wrong aspect filter the figures on the movie will be too thick or thin.

**-cropbottom 72 -cropleft 8 -cropright 8 -croptop 72** means cut off 72 pixels from the top and bottom and 8 pixels from outer left and right borders. This is because we are not interested in the black screen borders.

**Question:** Would **-cropbottom 70 -cropleft 2 -cropright 2 -croptop 70** do too?

## Encoding DVD's with mencoder

### Why use mencoder?

- Mencoder works well together with Mplayer (ripping, cropdetect)
- Mencoder can work with subtitles (sub or hard coded in the video)
- Mencoder is specialized in AVI (a better AVI than the abandoned ;-).
- The AVI files can be made compatible with XVID and DivX.

### Procedure

The movie is ripped or copied to disk.

MPlayer (that knows css) is used to inspect the movie.

With this information you edit a mencoder batch from the Templates folder.

En then execute it using a low background priority.

The principles are the same as with FFMPEG.

The syntax is somewhat different.

## The structure of a Digital Video Disc

- CD's and DVD's are optical digital disks: laser scans surface of disk for pits.
- DVD's have many usage's: **Digital Versatile Disc**.
- The popular single-sided RW single-layer DVD has a capacity of 4.7 GB.
- DVD's can be formatted in an Universal Disk Format (requires UDF.IFS).
- Most DVD's are compatible with the CD **ISO 9660** file system too (CDFS.IFS).
- Most DVD-Videodisks are single-sided double-layer (8.54 GB)
- Bandwidth of a standard (1x) DVD player is 10.5 Mibit/s (1.35 MB/s)
  
- Most **DVD-Video disks** use the video standard of their **analogue 4:3 national TV broadcasting system**: NTSC (US, Japan), PAL (EU), Secam (France)
  - National Television System Committee (**NTSC**): 720×480 at 29.97 frames/s.
  - **Phase Alternating Line (PAL)**: 720×576 at 25 frames/s.
  - To display 16:9 aspect ratio MPEG-2 video, filters are used.
  - Audio streams are stored in Dolby Digital (AC-3), Digital Theater System (DTS) formats or MPEG-1 Layer 2.
- DVD-Video also support menus, multiple subtitles and audio tracks.

## Inspecting DVD's with MPlayer

If the DVD is in drive M:

```
mplayer -dvd-device M: dvd://
```

If the DVD is in a folder containing the VIDEO\_TS subfolder:

```
mplayer -dvd-device %1 dvd://
```

You make a batch for it and drag the DVD folder to its program object.

```
mode co100,5
```

```
mplayer -autosync 30 -sid 20 -dvd-device %1 dvd:// > NUL 2>&1
```

```
exit
```

To enable On Screen Display (OSD) you must give MPlayer a font:

```
copy c:\psfonts\arial.ttf c:\home\default\.mplayer\subfont.ttf
```

Or edit %HOME%/.mplayer/config for Paul Smedley's port:

```
# Write your default config options here!
```

```
font=F:\PSFONTS\tnrwt_j.ttf
```

```
subfont=F:\PSFONTS\VERA.TTF
```

```
prefer-ipv4 = yes
```

```
slang=nl
```

## You can give your DVD player batch a menu

"<-" or "->"	Seek backward/forward 10 secondsecho
down or up	Seek backward/forward 1 minute
pgdown or pgup	Seek backward/forward 10 minutes
"<" or ">"	Step backward/forward in playlist
p or SPACE	Pause movie (press any key to continue)
q or ESC	Next movie or quit program
o	Cycle OSD mode: none/seekbar/seekbar+timer
+ or -	Adjust audio delay by +/- 0.1 second
* or /	Increase or decrease PCM volume
F	Toggle displaying "forced subtitles".
v	Toggle subtitle visibility.
j or g	Cycle through the available subtitles.
x or z	Adjust subtitle delay by +/- 0.1 second
r or t	Adjust subtitle position up/down.

## How to RIP (dump) the DVD (content) in M: with MPlayer

If you can't copy (fast) the content of DVD drive M: to the folder "movienamename" (with the subdir VIDEO\_TS) then use this (slow) template from your templates folder. If MPlayer cannot de-encrypt the DVD in M:, the project will probably fail.

```
SET NAME=movienamename
set MPLAYER_OPTIONS = -dvd-device M: -sid 20
rem Setting -sid 20 can prevent that subtitles are missed.
c:
cd mpg
md %name%
cd %name%
del *.*
mplayer dvd://1 -dumpstream -dumpfile c:\Mpg\%name%\%name%1.mpg
mplayer dvd://2 -dumpstream -dumpfile c:\Mpg\%name%\%name%2.mpg
mplayer dvd://3 -dumpstream -dumpfile c:\Mpg\%name%\%name%2.mpg
mplayer dvd://4 -dumpstream -dumpfile c:\Mpg\%name%\%name%3.mpg
mplayer dvd://5 -dumpstream -dumpfile c:\Mpg\%name%\%name%4.mpg
mplayer dvd://6 -dumpstream -dumpfile c:\Mpg\%name%\%name%5.mpg
etc for more tracks
```

Notice that VOB files are just MPEG-2 files. And that the MPEG-2 files created by MPlayer are very well understood by mencoder (and reverse)

## Encoding the DVD to AVI with mencoder

**Put the source on hard disk. One or more pass approaches can be used.**

One pass encoding (f.i. when recording internet streams):

1.) Encode the audio and the video in one run to moviename.avi

Classical two pass encoding (using B frames):

1.) Analyze video > divx2pass.log

2.) Encode the audio and the video using divx2pass.log > moviename.avi

Classical three pass encoding (using B frames):

1.) Encode the audio to AC3 or MP3 in the AVI container file

2.) Analyze video > divx2pass.log

3.) Encode the video using divx2pass.log

Tree pass encoding with subtitles (using B frames):

1.) Record chosen subtitle > moviename.sub

2.) Analyze video > divx2pass.log

3.) Encode the audio and the video using divx2pass.log > moviename.avi

## How to deal with subtitles?

- DVD (MPEG-2) subtitles are encoded as **graphical font streams**.
- An DVD player is not an OS: It has no knowledge of character sets.
- So you need an in MPEG-2 streams specialized **Optical Character Recognition (OCR)** program like **SubRip** to decode the displayed fonts to the by DivX media players used plain text **SubRip Text (SRT)** formats.
- Mplayer can display subtitles via it's **On Screen Display (OSD)** with your subfont, but it just displays the fonts on screen.
- **Mencoder** can encode your chosen subtitle fonts in the **encoded movie** or it can subtract it from the MPEG-2 stream and put them in a separate **SUB-file**. Both can be displayed by MPlayer.
- Separate sub-files show best in hardware Media Players. And you can have many of them (to learn a language). They can also be placed in the black areas outside the frame if the media player supports them.
- Fonts that are encoded straight away with the audio stream may appear small and blocky when the frame is scaled up in hardware Media Players.

```

SET MOVIE_NAME=Matrix
SET AID=128
SET SID=3
SET LAMEBR=128:cbr
SET LAMEVOL=6
SET VCODEC=mpeg4
SET VBITRATE=1400
rem ### 235:100 wide screen movie ###
rem SET SCALE=752:320
SET SCALE=704:304
rem ##### 16:9 DVD widescreen #####
rem SET SCALE=768:432
rem SET SCALE=512:288
rem ##### 4:3 television #####
rem SET SCALE=640:480
rem SET SCALE=384:288
SET CROP=720:416:0:82
SET FFOURCC=-ffourcc XVID
SET PRIORITY=1 3
mode co100,60

```

## Example of a batch that uses variables.

The batch can rather complex, but only 12 or less items need to be set per movie.

**Audio ID (AID)** tracks number from 128 (typically English) on.

**Sublanguage ID (SID):** you can determine it with the OSD of MPlayer.

**SCALE** :do an educated guess.

**CROP:** drag VOB to `mplayer % -vf cropdetect`

**-ffourcc XVID** makes the AVI Xvid and Divx compatible.

q: **First the subtitles are extracted.**

```
cd \
md var\rip\%MOVIENAME%_V%VBITRATE%_SUB
cd var\rip\%MOVIENAME%_V%VBITRATE%_SUB
del *.*
```

**Then we do a two pass encoding using lame (MP3) and libavcodec's FMP4 (tagged as XVID for better compatibility).**

```
REM Subtract subtitle with sublanguage id %SID% in %MOVIENAME%1.sub
priority %PRIORITY% mencoder dvd://1 -dvd-device %1 -nosound -ovc
frameno -sid %SID% -vobsubout %MOVIENAME%1 %FFOURCC% -o NUL
```

```
REM First pass creates divx2pass.log and %MOVIENAME%1.avi
priority %PRIORITY% mencoder dvd://1 -dvd-device %1 -aid %AID% -sws 2
-oac mp3lame -lameopts br=%LAMEBR%:vol=%LAMEVOL%:aq=9 -ovc lavc
-lavcopts vcodec=%VCODEC%:vbitrate=%VBITRATE
%:mbd=2:trell:v4mv:turbo:vpas=1 -vf crop=%CROP%,scale=%SCALE% %ENDPOS%
%FFOURCC% -o %MOVIENAME%1.avi
```

```
REM Second pass creates %MOVIENAME%1.avi
priority %PRIORITY% mencoder dvd://1 -dvd-device %1 -aid %AID% -sws 2
-oac mp3lame -lameopts br=%LAMEBR%:vol=%LAMEVOL%:aq=2 -ovc lavc
-lavcopts vcodec=%VCODEC%:vbitrate=%VBITRATE
%:mbd=2:trell:v4mv:turbo:vpas=2 -vf crop=%CROP%,scale=%SCALE% %ENDPOS%
%FFOURCC% -o %MOVIENAME%1.avi
```

## Conclusions

Re-encoding MPEG-2 video and audio to the more compact MPEG-4 formats can be easily be done by using simple commands.

For more complex repeated jobs batches can be used. The best batches becoming templates.