Encoding Video for the iPhone/iPod Touch (using mencoder/ffmpeg)

Brenton “B-Train” Fletcher

Why not QuickTime/ HandBrake?

- Some videos will just not encode with these tools
- Encoding videos with softsubs doesn't work
- QuickTime is for sissies
- Because you'll learn way more about video encoding this way

Why mencoder/ffmpeg?

- mencoder/ffmpeg IS FUCKIN' HARDCORE
- mplayer/mencoder manpage is 158 pages long, the "swiss army knife" of video encoding
- mencoder uses ffmpeg internally
- ffmpeg is the decoding/encoding core, ffmpeg mapage is 19 pages long

How

- I will demonstrate the final command we'll be running
- Then I will break it down into all the options you need
- I'll also cover installing mencoder/mplayer/ffmpeg

Example of what we'll end up with

- The complete command is:
- `mencoder -o angelic_layer19.m4v -of lavf -vf scale=480:270,expand=0:-50:::1,harddup -noskip -mc 0 -oac faac -faacopts mpeg=4:object=2:raw:br=128 -ovc x264 -x264encopts bframes=0:nocabac:global_header:no8x8dct -subfont-autoscale 0 -subfont-text-scale 20 -subpos 99 -subfont-blur 2 -subfont-outline 1 'Movies/Angelic Layer.fansub.1-26.complete/angelic_layer19.avi'`

mencoder -o angelic_layer19.m4v

- First part of command is:
- `mencoder -o angelic_layer19.m4v`
- mencoder is the command we are running
- -o angelic_layer19.m4v is file we are outputting to
- .m4v filename is important to iTunes

-of lavf

- This tells mencoder to use the libavformat library for encoding

-vf scale=480:270,expand=0:-50:::1

- -vf <options> tells mencoder to give options to the video encoder
- scale=480:270 scales the input video to 480 wide and 270 height
- expand=0:-50:::1 adds 50 pixels of black bars to the bottom of the video, so the end video size is 480 x 320 (the native resolution of the iPhone and iPod Touch)

,harddup

- In a movie, if frame n is the same as frame n + 1, some video encoders will just mark frame n + 1 as being a duplicate of frame n
- harddup ensures that if the input video has frames marked as duplicate, they are actually duplicated properly in the output video
- Because the iPhone and iPod Touch don’t support just marking the frames as duplicated

-noskip -mc 0

- -noskip prevents mencoder from skipping any frames in the input (similar to harddup)
- -mc 0 tells mencoder to prevent the video from getting out of sync with the audio

-oac faac

- Tells mencoder to use the libfaac library encoder for audio encoding
- iPhone/iPod Touch uses AAC for audio

-faacopts mpeg=4:object=2:raw:br=12

- -faacopts <options> tells mencoder to pass the options to libfaac
- mpeg=4 I don't even know
- object=2 tells libfaac to output Low-Complexity AAC, which the iPhone/iPod Touch requires
- raw is magic string that makes it work
- br=128 outputs 128kbps audio, which the iPhone/

-ovc x264

- -ovc x264 tells mencoder to use the libx264 library for video encoding
- iPhone/iPod Touch requires you to use DivX or H.264 video
- H.264 is better

-x264encopts bframes=0:nocabac

- -x264encopts <options> tells mencoder to pass the options to libx264
- iPhone requires “Baseline” H.264 video, which means a specific set of options
- bframes=0 tells libx264 not to use B-Frames
- nocabac tells libx264 not to use CABAC

:global_header:no8x8dct

- global_header tells libx264 to put some headers only at the start of the file, not also during the middle
- no8x8dct tells libx264 not to use 8x8 DCT's

-subfont-autoscale 0 -subfont-text-scale 20 -subpos 99 -subfont-blur 2 -subfont-outline 1

- -subfont-autoscale 0 tells mencoder not to scale subtitles automatically
- -subfont-text-scale 20 says to make the subtitle font size 20% the height of the movie
- -subpos 99 says to put the subtitle at 99% of the movie height
- -subfont-blur 2 adds a gaussian blur of 2px around the subtitle text
- -subfont-outline 1 adds a 1px outline around the subtitle text

'Movies/Angelic Layer.fansub.1-26.complete/

- this is the input file

Installing mencoder

- Install libfaac, libx264
- If input files will have mp3 audio stream, install libmp3lame
- Then download the latest version of mplayer you can get
- ./configure; make; sudo make install

Some videos crash mencoder

- For these videos, we use ffmpeg in combination with mencoder
- See me for info on this

But wait

- I've written a ruby script that handles all this
- Depends on mplayer
- Usage: `encode.rb Movies/movie.avi`
- Get it from me

Takeaways

- Video encoding is HARD SHIT
- iPhone/iPod Touch are very pedantic about what they require as input
- mencoder/ffmpeg have insane number of options