FFMPEG

Presented by

Pooja Mishra

Supervised by

Dr. Chris Pollett

Basics of Audio/Video

- Huge amounts of storage space required
  - Assuming an NTSC standard video at 720x480 pixels, 30 frames per second and 24-bit RGB color, we're talking about 1,036,800 bytes per frame.
  - over 200GB for a 2-hour movie
- Traditional, lossless compression algorithms such as ZIP, gz and bzip2 don't work.
- Lossy compression
  - Compression that is far more efficient but with a trade-off in that the picture and sound quality

Why do we need codecs?

- Need of elaboration of new algorithms with lossy compression
- Algorithms that allow us to encode the data in order to transport it and to decode the data the other end

Codecs

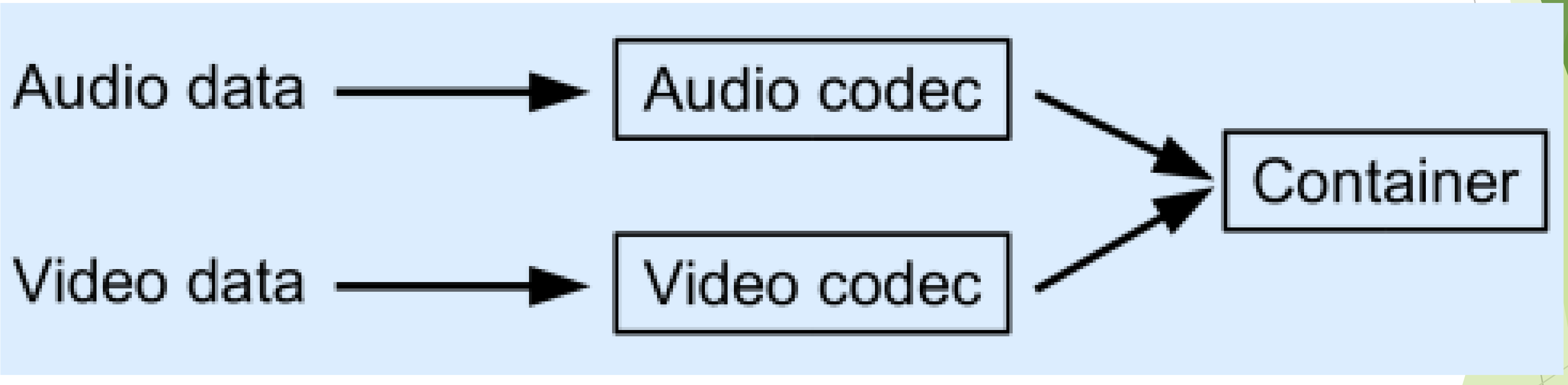
- Compressor/decompressor
- Compress
  - Transport and storage
- Decompress
  - Viewing or transcoding

Different types of codecs

- Audio Codecs
  - GSM – 13 Kbps (full rate), 20ms frame size.
  - iLBC – 15Kbps,20ms frame size: 13.3 Kbps, 30ms frame size.
  - ITU G.711 – 64 Kbps, sample-based. Also known as A-law/μ-law.
  - Speex – 2.15 to 44.2 Kbps.
  - LPC10 – 2.5 Kbps.
  - DoD CELP – 4.8 Kbps.
- Video Codecs
  - VP8 - free for use
  - H.264/MPEG-4 Part 10 or AVC (Advanced Video Coding)

Container

- Encapsulated encoded audio and video files into a single file. packaged, transported, and presented.
  - For example AVI , WAV files.
- Black boxes for holding a variety of media formats.
- Good container formats can handle files compressed with a variety of different codecs.


Audio data
Audio codec
Container
Video data
Video codec

Different types of containers

- 3GP (used by many mobile phones; based on the ISO base media file format)
- ASF (container for Microsoft WMA and WMV, which today usually do not use a container)
- AVI (the standard Microsoft Windows container, also based on RIFF)
- MP4 (standard audio and video container for the MPEG-4 multimedia portfolio, based on the ISO base media file format defined in MPEG-4 Part 12 and JPEG 2000 Part 12) which in turn was based on the QuickTime file format.
- Ogg (standard container for Xiph.org audio format Vorbis and video format Theora)

Usage of container

- Informs the media player about the audio and video codecs used
- Many different possible combinations of codecs that can be used within each type of container

Playback of a multimedia file

- The container is identified
- It tells which codecs are needed to decode the data
- The audio and video streams are then extracted from the container
- Fed through the appropriate codecs
- Get raw audio and video data that can be fed to the audio and display subsystems of the computer

What is ffmpeg?

- An application that allows Linux users to convert video files easily between a variety of different format
  - For example avi to mp4

- ffmpeg implements a decoder and then an encoder enabling the user to convert files from one container/codec combo to another

- VOB file from a DVD containing MPEG2 video and AC3 audio to an AVI file containing MPEG4 video and MP3 audio

How conversion takes place?

- The original container is examined and identified
- The encoded data extracted and fed through the codecs
- The newly-decoded data is then fed through the "target" codecs into the new container
- QuickTime file containing SVQ3 video and MP3 audio to a 3GP file containing H263 video and AMR wideband audio

Installation

- Download ffmpeg static for 64 bit from link and install the exe
- Set the path for ffmpeg
- Run commands like to verify the installation
  - ffmpeg -version
  - ffmpeg -formats

Usage of ffmpeg tool

- ffmpeg -i audio.wav

```
FFmpeg version SVN-r9607, Copyright (c) 2000-2007 Fabrice Bellard, et al.
  configuration:  {snipped for brevity}
  libavutil version: 49.4.1
  libavcodec version: 51.40.4
  libavformat version: 51.12.1
  built on Jul 12 2007 20:22:46, gcc: 3.4.6
Input #0, wav, from 'audio.wav':
  Duration: 00:05:08.1, start: 0.000000, bitrate: 1411 kb/s
  Stream #0.0: Audio: pcm_s16le, 44100 Hz, stereo, 1411 kb/s
Must supply at least one output file
```

Converting audio file

ffmpeg -i audio.wav -acodec mp3 -ab 192k audio.mp3

- -i audio.wav
  This tells ffmpeg that we want it to take audio.wav and process it.
- -acodec mp3
  This tells ffmpeg to use the "mp3" audio codec to create the target file.
- -ab 192k
  This tells ffmpeg to use an audio bitrate of 192 kbit/s. The higher this value, the better the audio quality, but the larger the resulting file. 192 kbit/s is pretty good quality audio.
- audio.mp3
  Dump the encoded audio data into a file called audio.mp3

Video file encoding

- ffmpeg -i kitty.flv
- ffmpeg -i kitty.flv -target ntsc-dvd -aspect 4:3 kitty.mpg

Additional functionalities

- Changing the sample rate of the audio and advancing or delaying it with respect to the video.
- Changing the frame rate of the resulting video, cropping it, resizing it, placing bars left and right and/or top and bottom in order to pad it when necessary, or changing the aspect ratio of the picture
- Allows importing audio and video from different sources, thus allowing dubbing for example

Delaying the audio or video

ffmpeg -i input_1 -itsoffset 00:00:03.5 -i input_2 .....

- n this example, input_2 will be delayed by 3.5 seconds
- The content of input_1 will start at the beginning of the movie generated by ffmpeg, and the content in input_2will start 3.5 seconds into the movie

Thank you!!

References

- http://howto-pages.org/ffmpeg/
- https://wiki.archlinux.org/index.php/FFmpeg
- http://superuser.com/questions/525249/convert-avi-to-mp4-keeping-the-same-quality
- https://www.youtube.com/watch?v=xcdTIDHm4KM