

Technical support for `ffmpeg`

Fritz Francisco
fritz.francisco@hu-berlin.de

April 7, 2020

1 `ffmpeg`

`ffmpeg`: “A complete, cross-platform solution to record, convert and stream audio and video.”

Since `ffmpeg` is a program with command line interface you will need to open a terminal in order to use it. From here you can type `ffmpeg -h` for further information about all possible options or input the commands listed below.

1.1 Installation

Windows:

<https://www.wikihow.com/Install-FFmpeg-on-Windows>

Mac OS:

<https://github.com/fluent-ffmpeg/node-fluent-ffmpeg/wiki/Installing-ffmpeg-on-Mac-OS-X>

Linux:

https://github.com/adaptlearning/adapt_authoring/wiki/Installing-FFmpeg

2 Most important `ffmpeg` commands

2.1 Cut

Cutting smaller parts from one large video:

Command:

```
ffmpeg -i input.MP4 -ss 00:01:00 -t 00:02:00 -c copy output.mp4
```

Explanation:

```
-i input
-ss seek to (starting point)
-t time to (ending point)
-c codec
output.mp4 designated output file
```

Examples: <https://superuser.com/questions/138331/using-ffmpeg-to-cut-up-video>

2.2 Crop

Cropping a video to a region of interest:

Command:

```
ffmpeg -i input.MP4 -filter:v "crop=out_w:out_h:x:y" output.mp4
```

Explanation:

```
-i input
-filter:v video filter
out_w width of the output rectangle
out_h height of the output rectangle
x and y specify the top left corner of the output rectangle
output.mp4 designated output file
```

Examples: <https://video.stackexchange.com/questions/4563/how-can-i-crop-a-video-with-ffmpeg>

2.3 Concatenate

Concatenating refers to appending short videos in a sequence in order to create one long video. This is useful for all filming done with GoPros since they output the continuous video stream to small video files.

Command:

```
ffmpeg -f concat -safe 0
-i <(for f in /*.MP4; do echo "file '$PWD/$f'"; done)
-c copy output.mp4
```

Explanation:

```
-i input (here with a for loop to search
through all files in current directory)
-f video filter
-safe 0 safety option
output.mp4 designated output file
```

Examples: <https://trac.ffmpeg.org/wiki/Concatenate>

2.4 Extract Frames

Extracting frames is very easy with **ffmpeg**. Input is a video and output are a series of frames dependent on the set rate. Be sure to output into a separate folder/directory for convenience and better overview. The file format of the output images can be determined by changing the file ending to jpg, png, bmp, tiff etc.

Command:

```
ffmpeg -i input.MP4 -r 1 output_%03d.jpg
```

Explanation:

```
-i input
-r rate at which frames are saved (per second)
output_%03d.jpg designated output file(s)
```

Examples: <https://stackoverflow.com/questions/10957412/fastest-way-to-extract-frames-using-ffmpeg>

2.5 Encode

Changing codecs of video files can be convenient when using these across various applications.

Command:

```
ffmpeg -i file.MP4 -c:v libx264 -c:a copy output.avi
```

Explanation:

```
-i input
-c:v video codec (set to `copy' if only changing file type)
-c:a audio codec
```

Examples: <https://trac.ffmpeg.org/wiki/Encode/H.264>
<https://superuser.com/questions/193379/batch-convert-videos-from-mpg-to-avi>

2.6 Create Video from Frames

Converting images to a continuous video.

Command:

```
ffmpeg -r 60 -f image2 -s 1920x1080 -i pic%04d.png \
-vcodec libx264 -crf 25 -pix_fmt yuv420p output.mp4
```

Explanation:

```
-i input (%04d defines continuous index number of 4 digits)
-s resolution
-f filter
-r framerate (fps)
-vcodec video codec
-crf quality, lower means better quality, 15-25 is usually good
```

Examples: <http://hamelot.io/visualization/using-ffmpeg-to-convert-a-set-of-images-into-a-video/>

2.7 Stack Videos

Stacking videos horizontally or vertically into one single file.

Command:

```
ffmpeg -i input0 -i input1 -filter_complex hstack=inputs=2 output
```

Explanation:

```
-i      input
-filter_complex      filter defining stacking method (hstack, vstack)
```

Examples: <http://hamelot.io/visualization/using-ffmpeg-to-convert-a-set-of-images-into-a-video/>

3 Further notes

As some might be working on linux computers or may encounter a linux command line from time to time it may be useful to know some basic linux commands:

ls : ["list"] lists all files in the current directory

cd : ["change directory"] changes into the directory following the command. If no directory is specified it will change into the basal **home** directory

cp : ["copy"] copy specified files to designated directory. **cp** is followed by the filenames to be copied and then the designated directory

mv : ["move"] moves specified files to designated directory which does not replicate them. **mv** is followed by the filenames to be moved and then the designated directory

pwd : ["print working directory"] shows the current directory

4 Documentation

Also see <https://www.ffmpeg.org/documentation.html>

If you have any questions please feel free to contact me and I'll try my best to help!