Video Editing

Lossless Cut Cut/Trimming

LosslessCut aims to be the ultimate cross platform ffmpeg GUI for extremely fast and lossless operations on video, audio, subtitle and other related media files. The main feature is lossless trimming and cutting of video and audio files, which is great for saving space by rough-cutting your large video files.

https://github.com/mifi/lossless-cut https://github.com/mifi/lossless-cut/releases

Edit MKV Files

To remove audio or subtitle tracks from mkv files, the mkvmerge command can be used.

First, identify the track ids contained in the mkv file:

```
mkvmerge -i videofile.mkv
```

The output will look something like this:

```
kax i am at 6718
File 'videofile.mkv': container: Matroska
Track ID 0: audio (AC3/EAC3)
Track ID 1: subtitles (SubRip/SRT)
Track ID 2: subtitles (SubRip/SRT)
Track ID 3: video (MPEG-4p10/AVC/h.264)
Track ID 4: audio (DTS)
Chapters: 17 entries
Tags for track ID 0: 7 entries
Tags for track ID 1: 7 entries
Tags for track ID 2: 7 entries
Tags for track ID 3: 7 entries
Tags for track ID 4: 7 entries
```

For more detailed information the mkvinfo command can be used:

```
mkvinfo videofile.mkv
```

To create a new mkv file with the subtitles removed and only keeping the second audio stream, use the following command:

```
mkvmerge -o newvideofile.mkv -a 4 -s 0 videofile.mkv
```

To extract tracks from an mkv file, use the mkvextract command using the IDs obtained from mkvinfo:

```
mkvextract tracks videofile.mkv 2:subtitletrack2.srt 3:subtitletrack3.srt
```

Fix broken header/index

```
ffmpeg -i 'input' -map 0 -c copy 'output.mkv'
```

Flip video

To Flip Video Vertically (upside down):

```
ffmpeg -i INPUT.mp4 -vf vflip -c:a copy OUTPUT.mp4
```

To Flip Video Horizontally (left/right):

```
ffmpeg -i INPUT.mp4 -vf hflip -c:a copy OUTPUT.mp4
```

Convert DVD VIDEO_TS folder

```
ffmpeg -i
"concat:VTS_01_1.VOB|VTS_01_2.VOB|VTS_01_3.VOB|VTS_01_4.VOB|VTS_01_5.VOB|VTS
_01_6.VOB|VTS_01_7.VOB" -filter_complex
"scale=ceil(iw*min(1\,min('1080'/iw\,'720'/ih))/2)*2:-2" -c:v libx264 -r 30
-crf 28 -c:a aac  -b:v '4M' -maxrate '4M' -bufsize '4M' -f mp4 output.mp4

or

cat ./VIDEO_TS/*.VOB | ffmpeg -i - -filter_complex
"scale=ceil(iw*min(1\,min('1080'/iw\,'720'/ih))/2)*2:-2" -c:v libx264 -r 30
-crf 28 -c:a aac  -b:v '4M' -maxrate '4M' -bufsize '4M' -f mp4 output.mp4
```