Table of Contents

Interpolate video to higher framerate 2

Interpolate video to higher framerate

Ffmpeg has a filter called 'minterpolate' that can interpolate videos to a higher framerate. You get get not-terrible results by playing with the parameters, it's better then the 60 or 120hz interpolation done by TV's that just fades/blends the frames.

Here is a command that will take the input, interpolate it to 60 fps, and encode it with libx264 at crf 18 (visually lossless)

```
 ffmpeg -i "input.mkv" -c:a copy -c:v libx264 -preset medium -crf 18 -vf
"minterpolate=fps=60:mi_mode=mci:mc_mode=aobmc:me_mode=bidir:vsbmc=1"
"output.mkv"
```

Interpolation is very single threaded, and very slow. Most modern CPUs have multiple cores so it would benefit performance to run multiple threads. We can do this by splitting the video into 20 or 30 second chunks (depending on video length) and running multiple instances of ffmpeg in parallel.

This can be done using my video splitting method listed below, then using the following GNU parallel command to run 16 instances of FFMPEG (for my 16 core processors - be careful since it uses a lot of RAM).

```
 parallel -j 16 -q -a files ffmpeg -i {} -c:a copy -c:v libx264 -preset medium
-crf 18 -vf
"minterpolate=fps=60:mi_mode=mci:mc_mode=aobmc:me_mode=bidir:vsbmc=1" -
max_muxing_queue_size 1024 -y "{.}.done.mkv"
```

where -j is the number of threads to run, and the “files” file contains a list of the parts, one file per line. You can generate the list of files by running a find command

```
find ./ -name '*.mkv'>> files
```

Finally, you can merge all the files to merge using ffmpeg. (see other page)

(Or just leave it on overnight on the CS club servers)

From:
https://wiki.tonytascioglu.com/ - **Tony Tascioglu Wiki**

Permanent link:
https://wiki.tonytascioglu.com/scripts/ffmpeg/interpolate_video_to_higher_framerate

Last update: **2022-04-06 21:46**