How can I extract a good quality JPEG image from an H264 video file with ffmpeg?

Asked 7 years, 11 months ago Active 5 months ago Viewed 112k times

97

41

Currently I am using this command to extract the images:

```
ffmpeg.exe -i 10fps.h264 -r 10 -f image2 10fps.h264_%03d.jpeg
```

But how can I improve the JPEG image quality?

video graphics ffmpeg computer-vision sharpffmpeg

asked Apr 19 '12 at 9:39

Daniel Gartmann
5,892 ● 8 ● 36 ● 51

1 What is wrong with the current quality, apart from that it is not "good"? – bjoernz Apr 19 '12 at 11:05

2 Answers

Active Oldest Votes

196

Use `-qscale:v`

Use `-qscale:v` (or the alias `-q:v`) as an output option. Effective range for JPEG is 2-31 with 31 being the worst quality. I recommend trying values of 2-5.

To output a series of images:

```
ffmpeg -i input.mp4 -qscale:v 2 output_%03d.jpg
```

To output a single image at ~60 seconds duration:

```
ffmpeg -ss 60 -i input.mp4 -qscale:v 4 -frames:v 1 output.jpg
```

MJPEG

If you input is MJPEG (Motion JPEG) then the images can be extracted without any quality loss.

The `ffmpeg` or `ffprobe` console output can tell you if your input is MJPEG:

```
$ ffprobe -v error -select_streams v:0 -show_entries stream=codec_name -of
default=nw=1 input.avi
codec_name=mjpeg
```

Then you can extract the frames using the `mjpeg2jpeg` bitstream filter:

```
$ ffmpeg -i input.avi -codec:v copy -bsf:v mjpeg2jpeg output_%03d.jpg
```

Also see

- FFmpeg FAQ: How do I encode movie to single pictures?
- FFmpeg Wiki: Create a thumbnail image every X seconds of the video

edited Oct 31 '18 at 21:54

answered Apr 19 '12 at 18:03

llogan

61.4k ● 18 ● 125 ● 155

This seems to have no effect for me-- qscale 1 and 2 both give identical file sizes and (to my naked eye) appear the same as without qscale at all. – felwithe Jan 28 '15 at 23:03

Can you post the complete commandline you're using? Also please post the complete, uncut output from ffmpeg on the commandline. Note that *placement* of options is relevant, so -qscale:v 2 needs to be placed after the -i inputfile option, but before the output file option, to have any effect. – Ronald S. Bultje Apr 12 '15 at 12:10

1 For me adding `-qmin 1 -qmax 1` in addition to `-q:v 1` doubled the file size. And I can seem to see a very slight improvement also. – complistic Jun 27 '15 at 0:43

1 @complistic: `-qmin 1 -qmax 1` resulted in larger file, but gives me an exact same image. I validated this via photoshop, 2 layers and difference filter. The pixels are the same. – cherouvim Nov 30 '15 at 15:41

@Kostanos You can try `-qmin 1 -q:v 1`. – llogan Oct 14 '17 at 23:00

0

Output the images in a lossless format such as PNG:

```
ffmpeg.exe -i 10fps.h264 -r 10 -f image2 10fps.h264_%03d.png
```

Then use another program (where you can more precisely specify quality, subsampling and DCT method – e.g. GIMP) to convert the PNGs you want to JPEG.

It is possible to obtain slightly sharper images in JPEG format this way than is possible with `-qmin 1 -q:v 1` and outputting as JPEG directly from `ffmpeg`.

558 ● 4 ● 12

`ffmpeg` outputs `PNG8` files which use only 256 colors (same as GIF). so it is actually *very* lossy. – lapin Feb 17 at 6:51

@lapin The PNGs I extracted using this method are 24 bit (even for frames with fewer than 256 colours, though others have many more). This was using version 4.2.1 x64 on Windows. Also written here is: "If I pull png's from an mp4, with this [`ffmpeg`] command, I get high quality png's that are of identical quality to the original video." What version of `ffmpeg` are you using that is outputting `PNG8` files, and what is your input format? – Jake Feb 20 at 0:29

Actually I think you're right, the problem was that `identify image.png` gives result "8-bit" when actually its not really single channel 8-bit, but 8-bit for R, G and B. IDK how the average `identify` user is supposed to understand that tho. – lapin Feb 20 at 11:12