

draft-ietf-netvc-testing-02

IETF 95

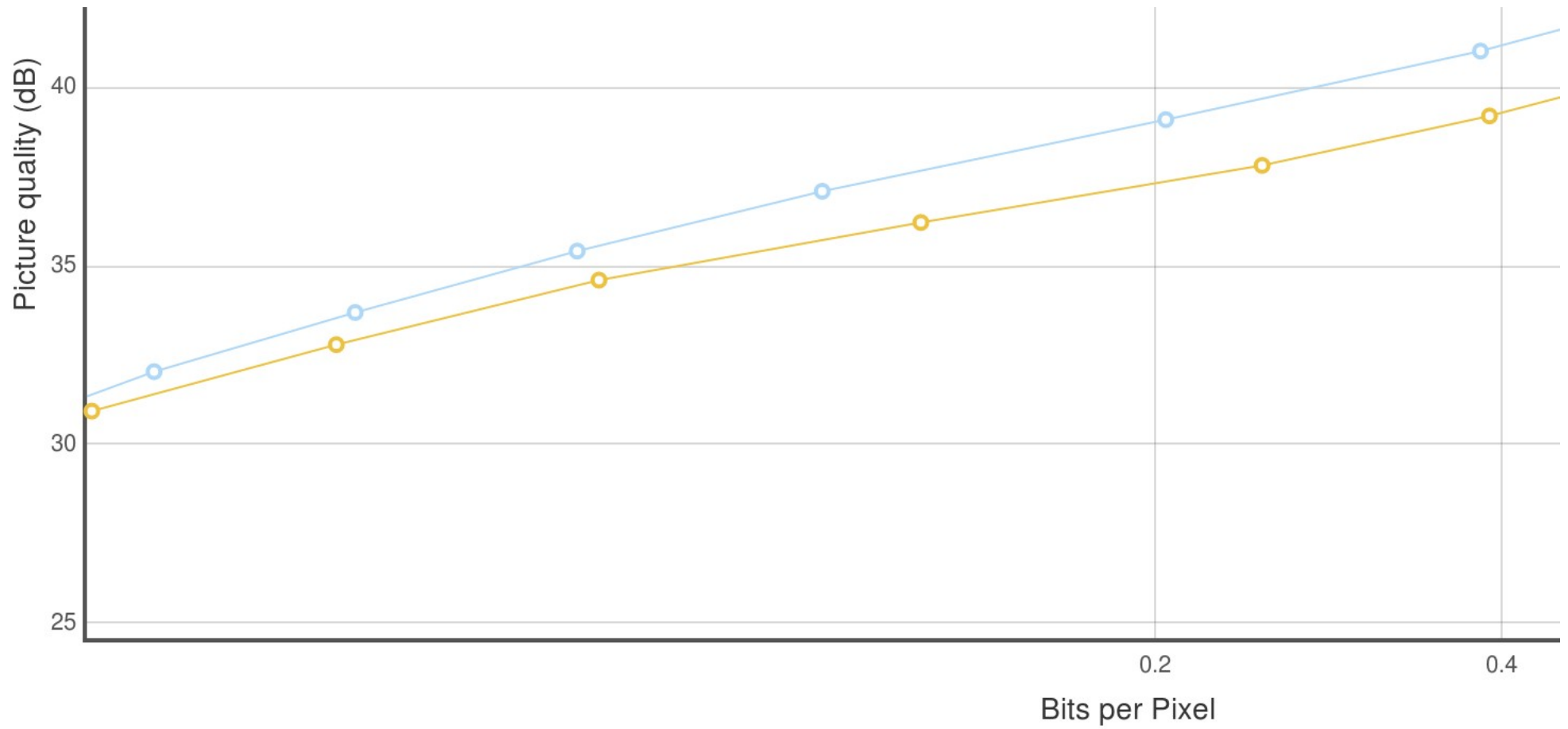
Sub-Agenda

- Improvements to set selections
- BD-Rate calculations
- Software updates
- TODO

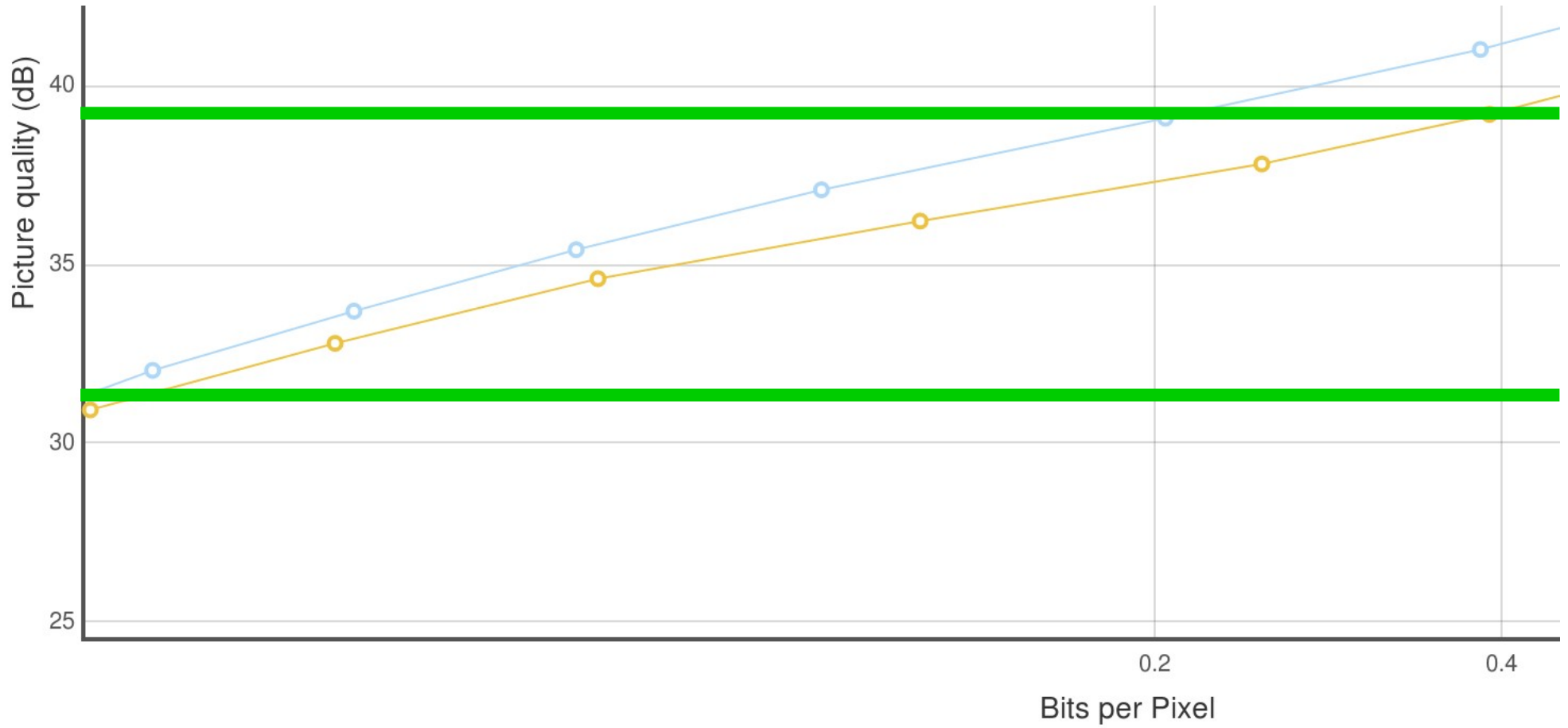
Set Selections

- 'Speedy' regression test
 - netflix-2k-1
 - vc-720p-1
- 'Objective testing' set
 - netflix-2k-1
 - video-hd-3
 - vc-720p-1
 - vc-360p-1
 - twitch-1
- Comprehensive periodic set

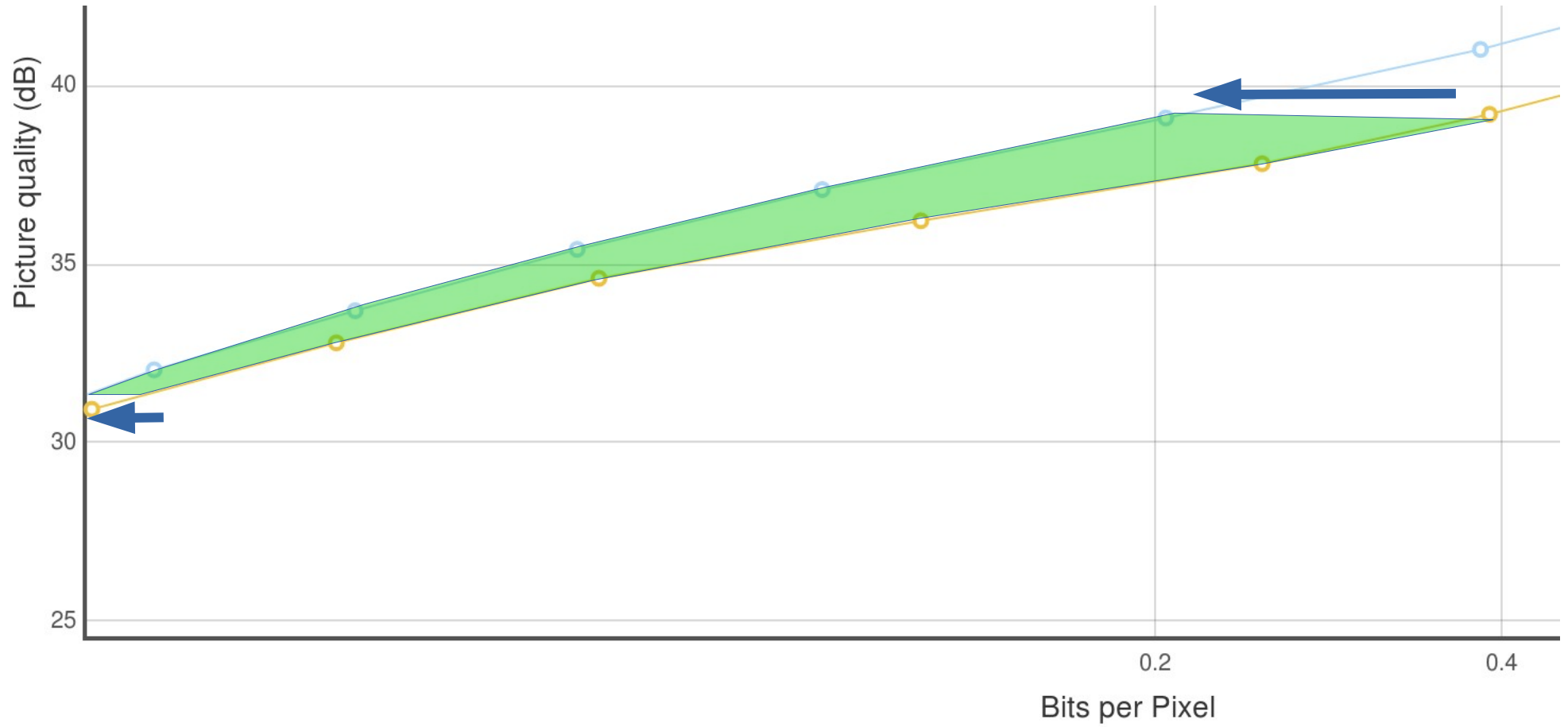
BD-Rate



BD-Rate



BD-Rate

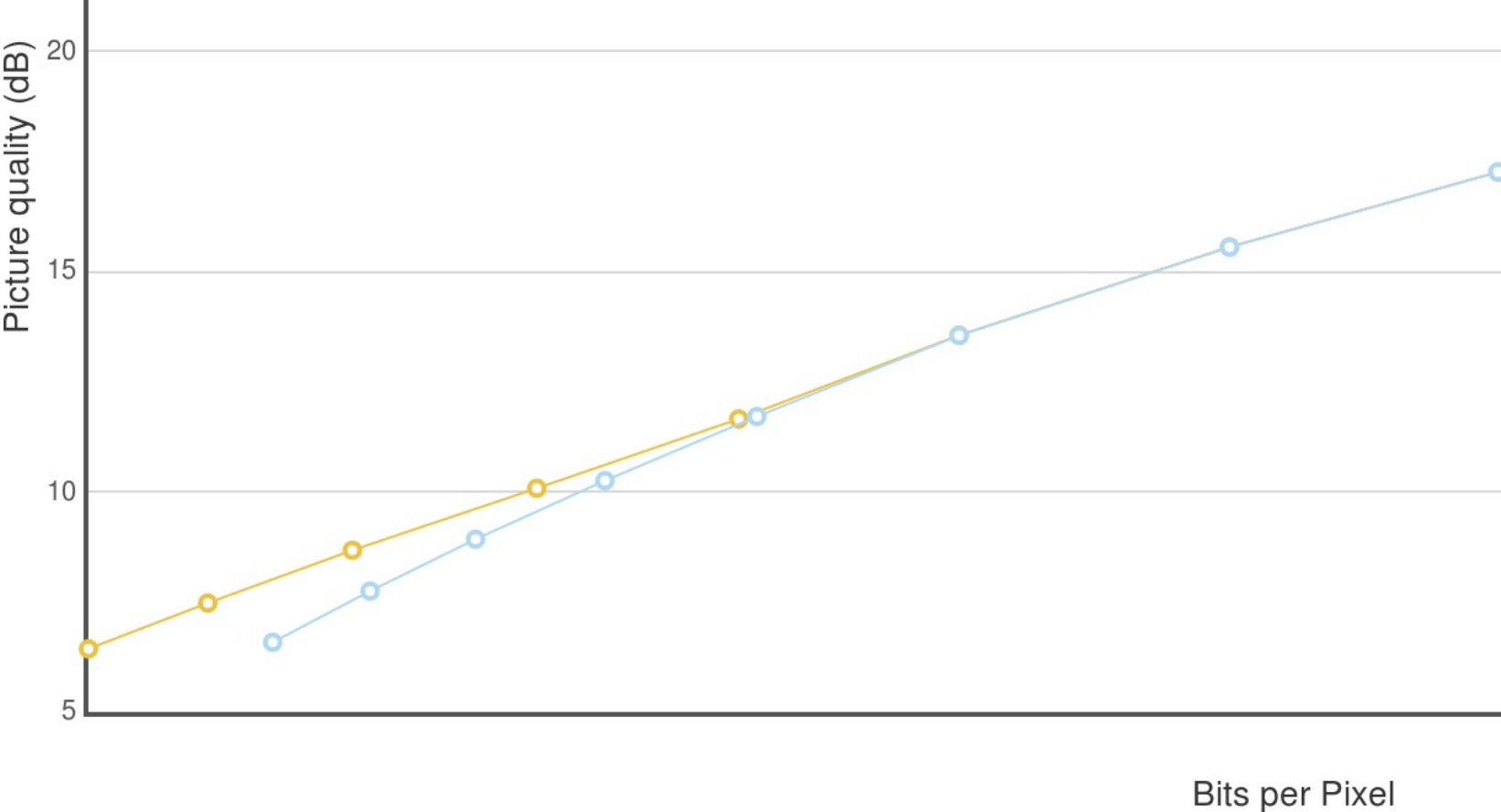


BD-Rate Range

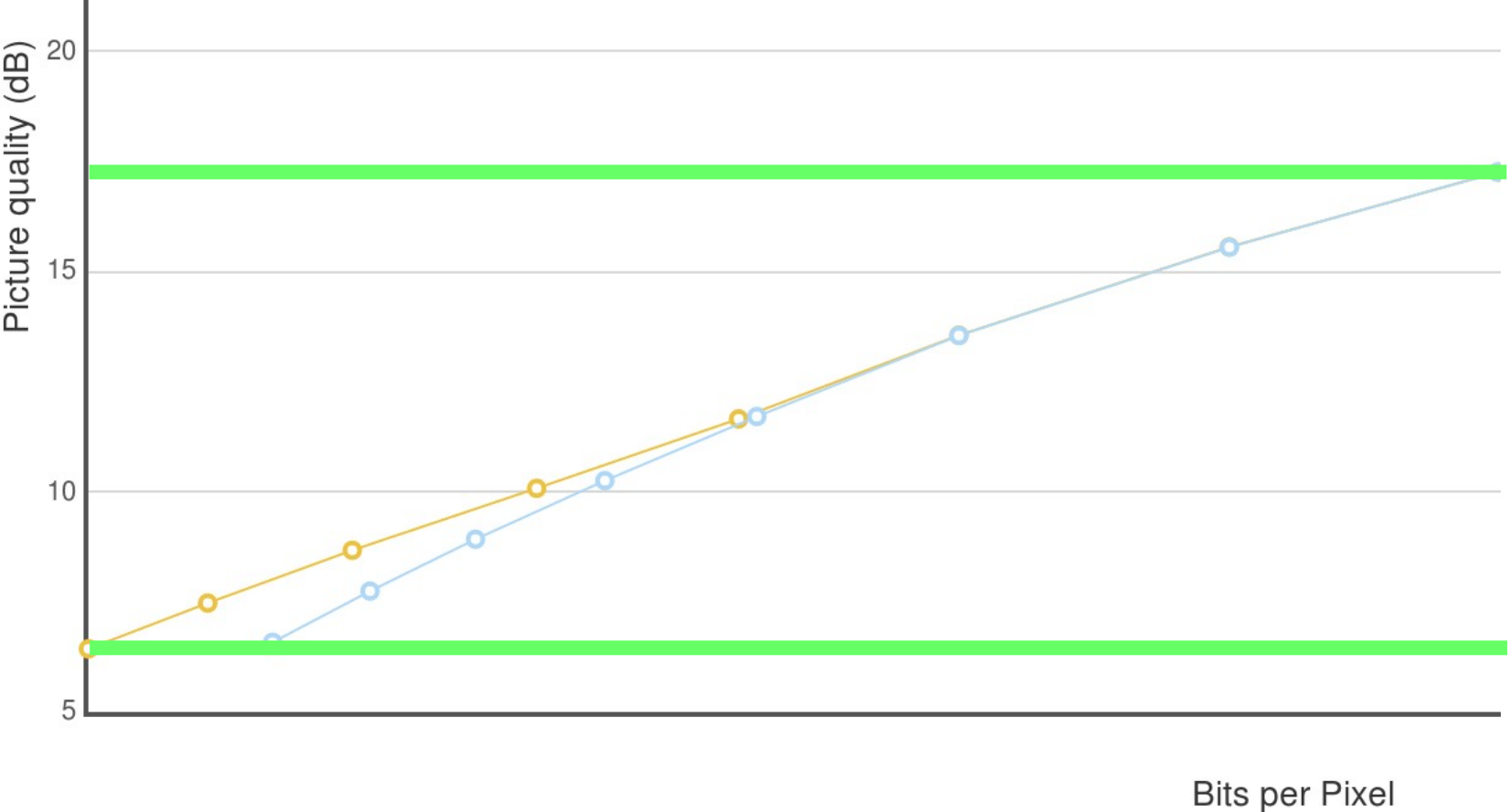
- Selecting a reasonable quality range is hard
 - We have a bunch of metrics on different scales
 - Metrics tend not to be valid on an absolute scale [1]
- There are a couple options to produce the range in a uniform manner

[1] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of PSNR in image/video quality assessment," in *Electronics Letters*, vol. 44, no. 13, pp. 800-801, June 19 2008.

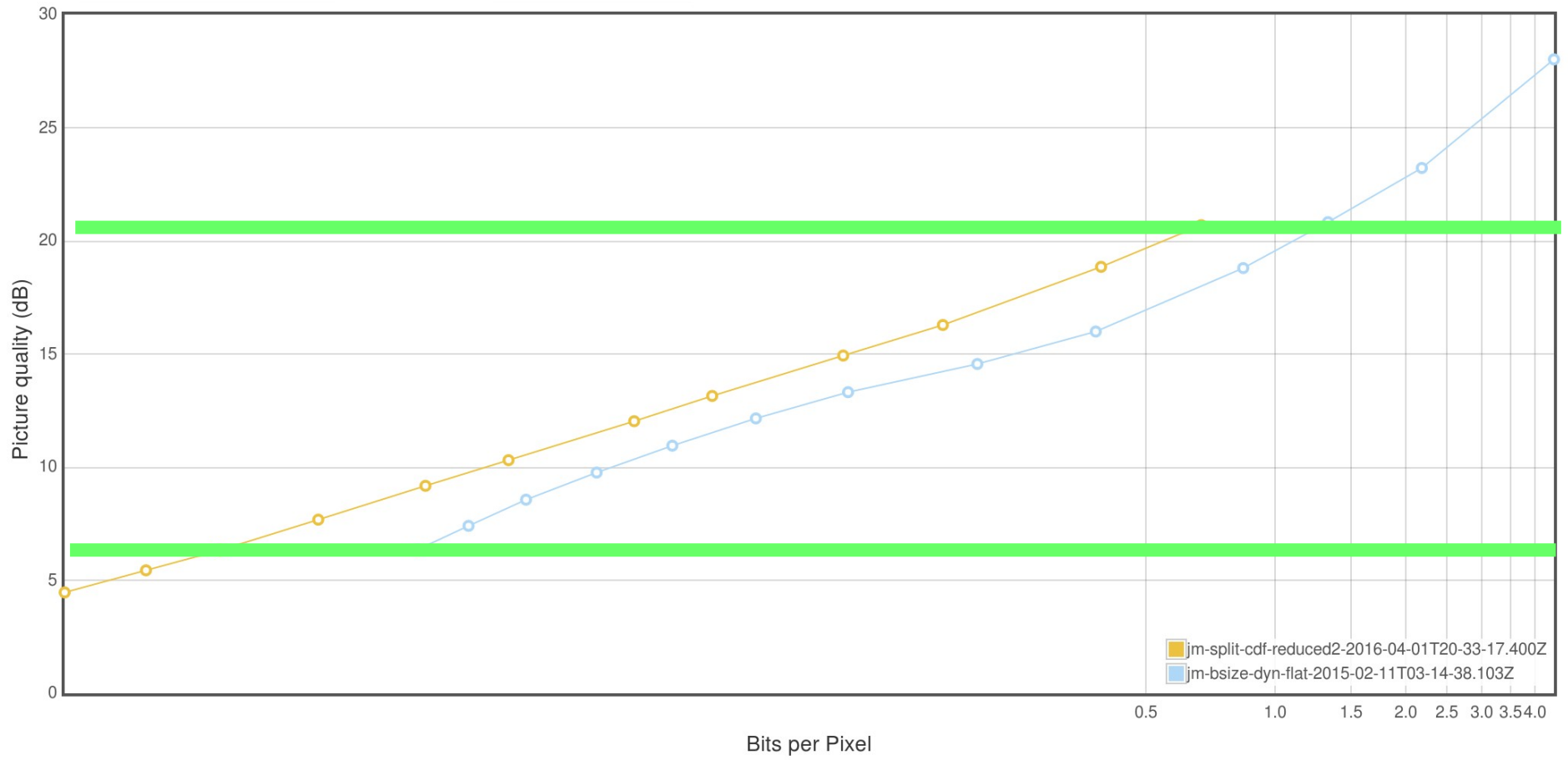
BD-Rate Range – Quantizer-derived



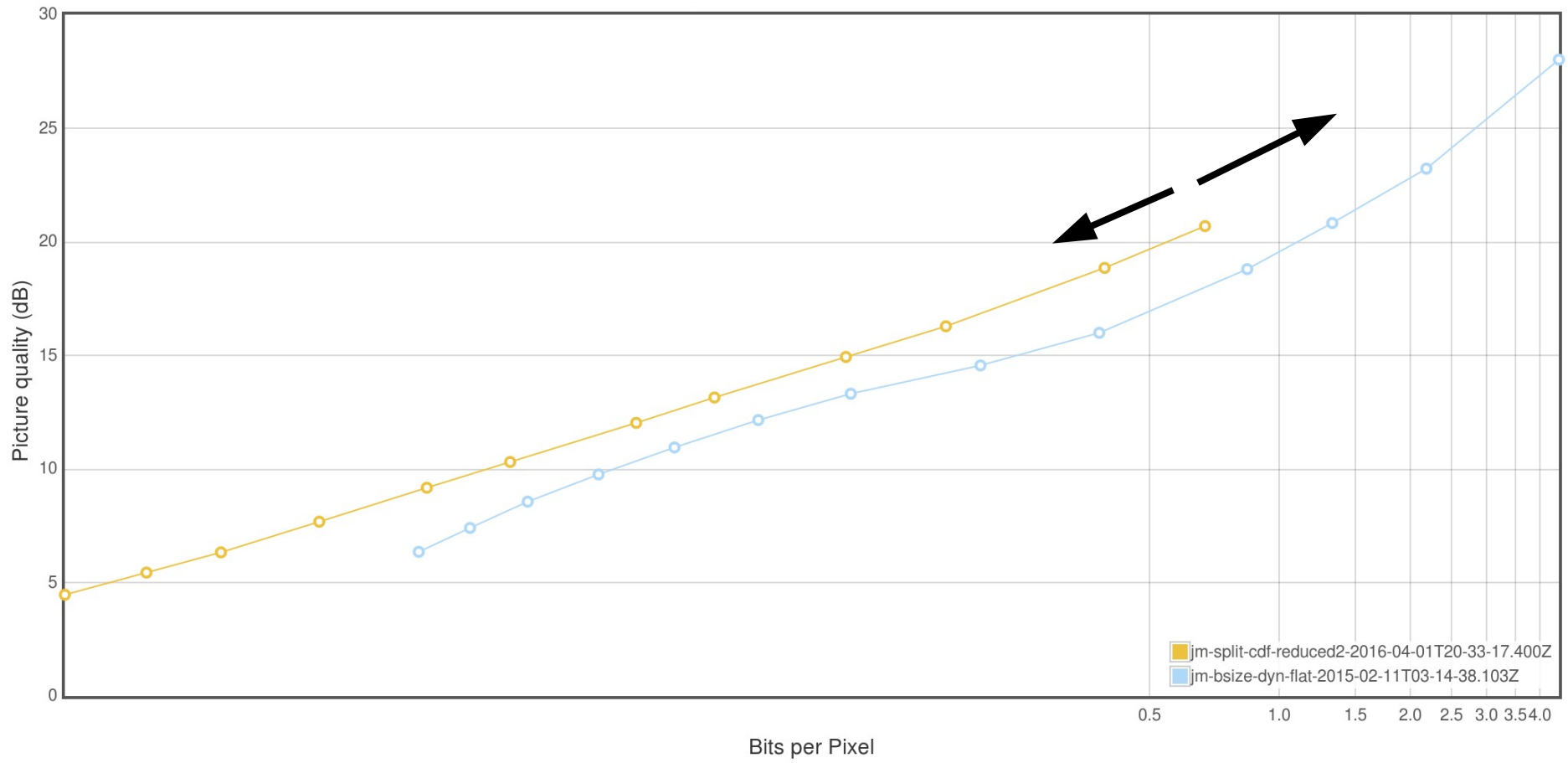
BD-Rate Range – Quantizer-derived



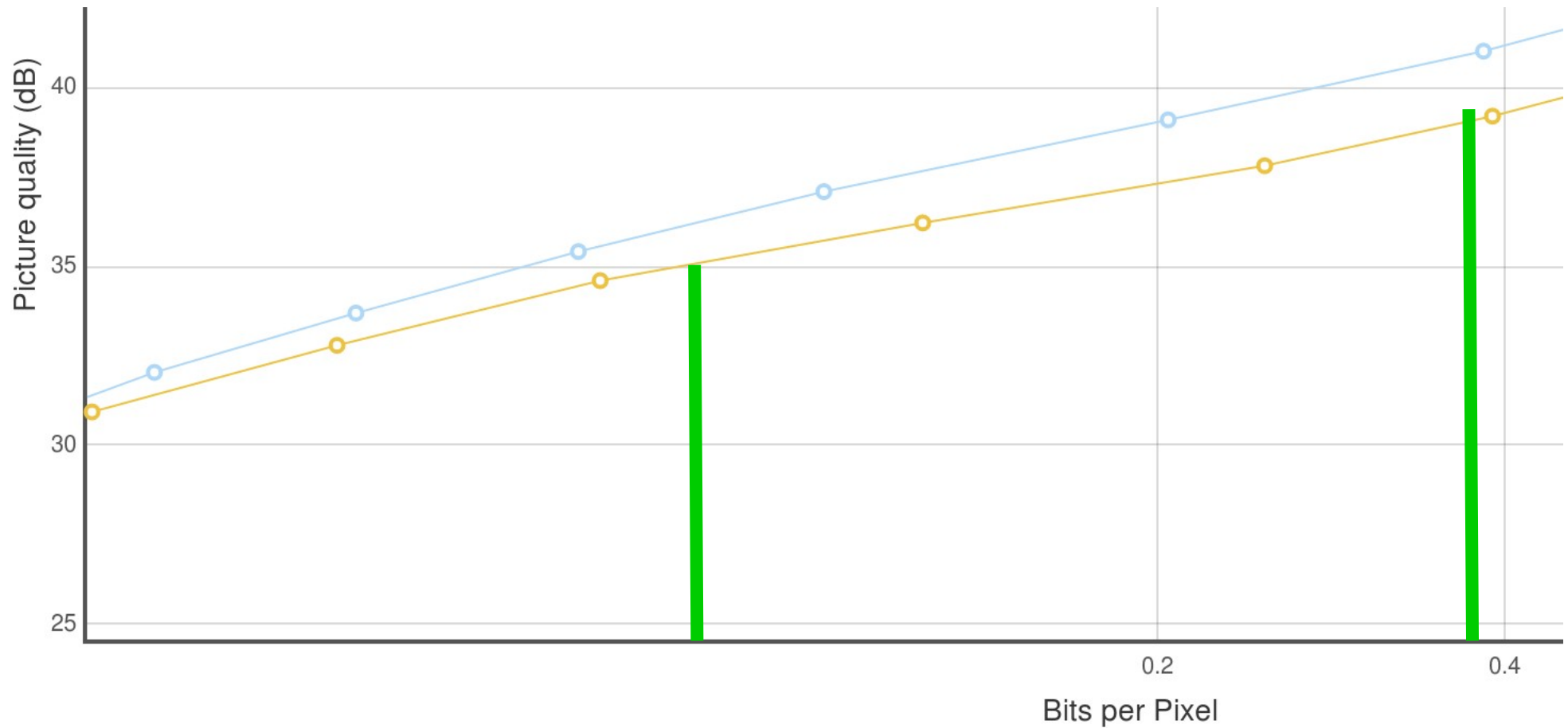
BD-Rate Range – Quantizer-derived



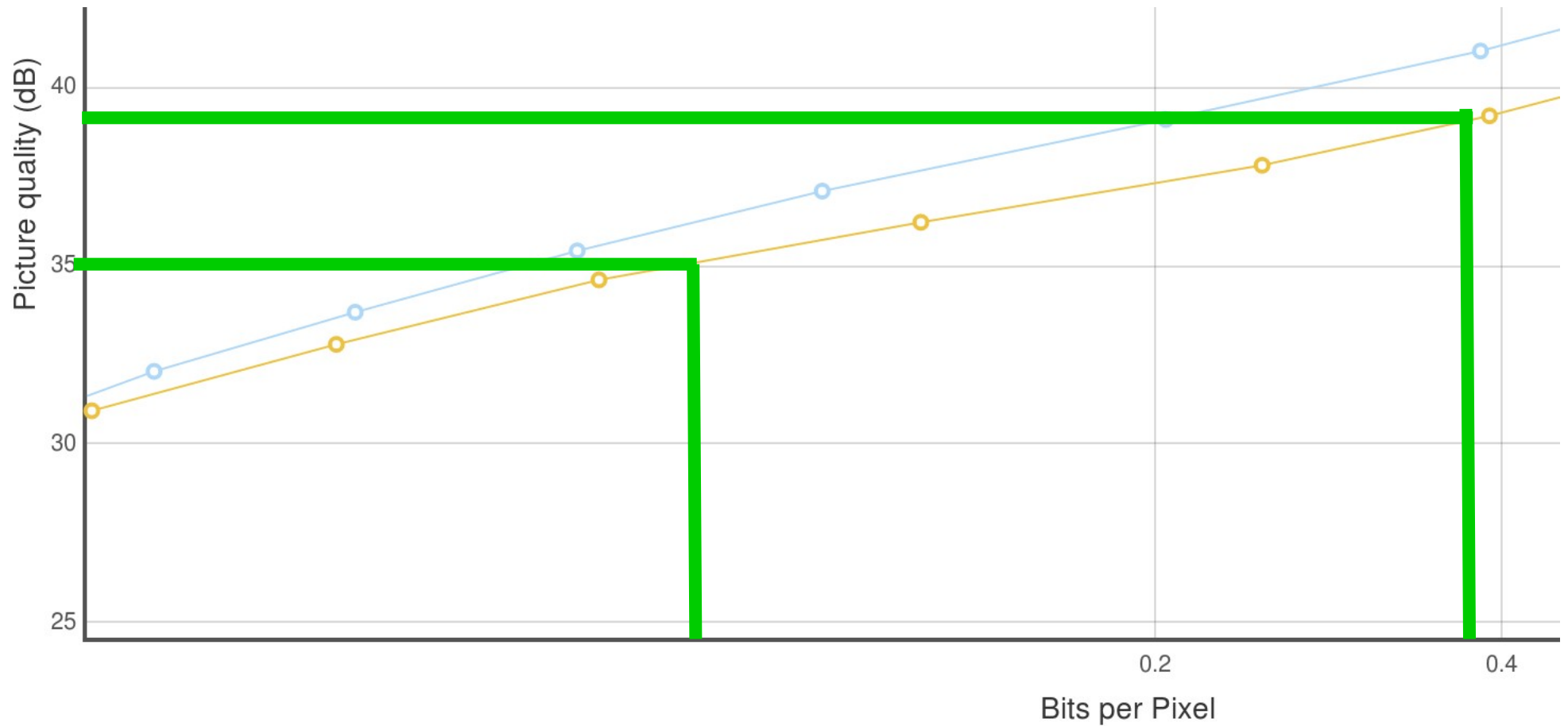
BD-Rate Range – Quantizer-derived



BD-Rate Range – Bitrate-derived



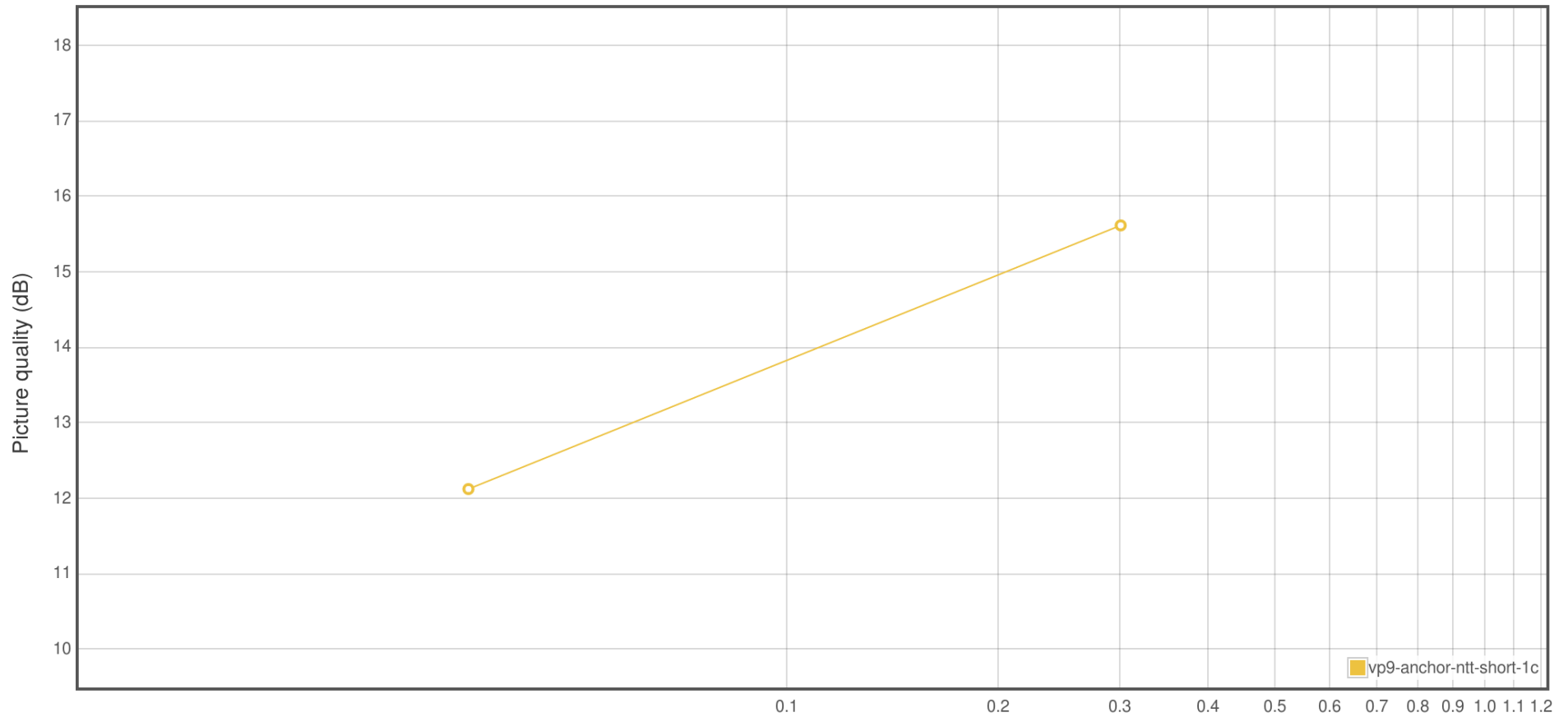
BD-Rate Range – Bitrate-derived



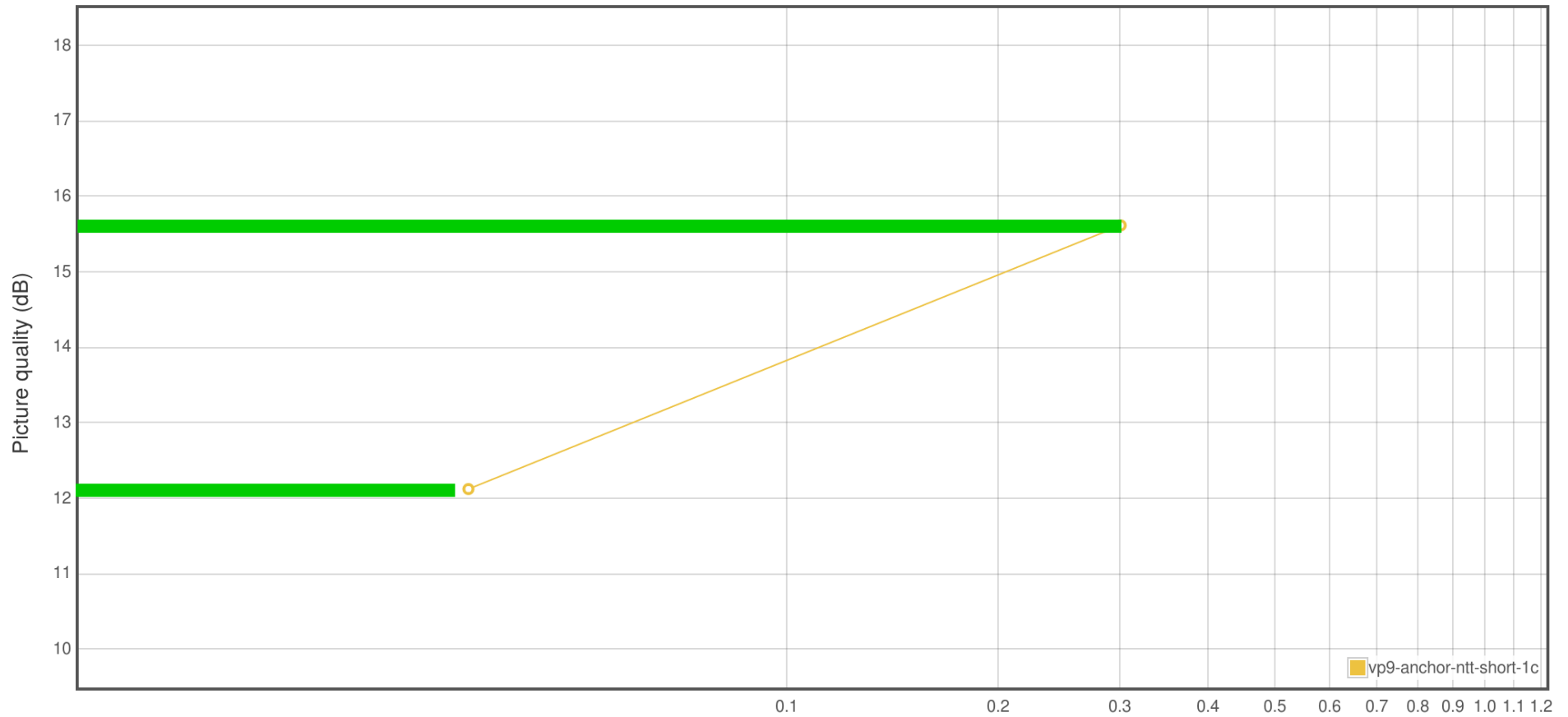


PedestrianCrossing and ToddlerFountain from Netflix
CC-BY-NC-ND 4.0 <https://media.xiph.org/video/derf/>

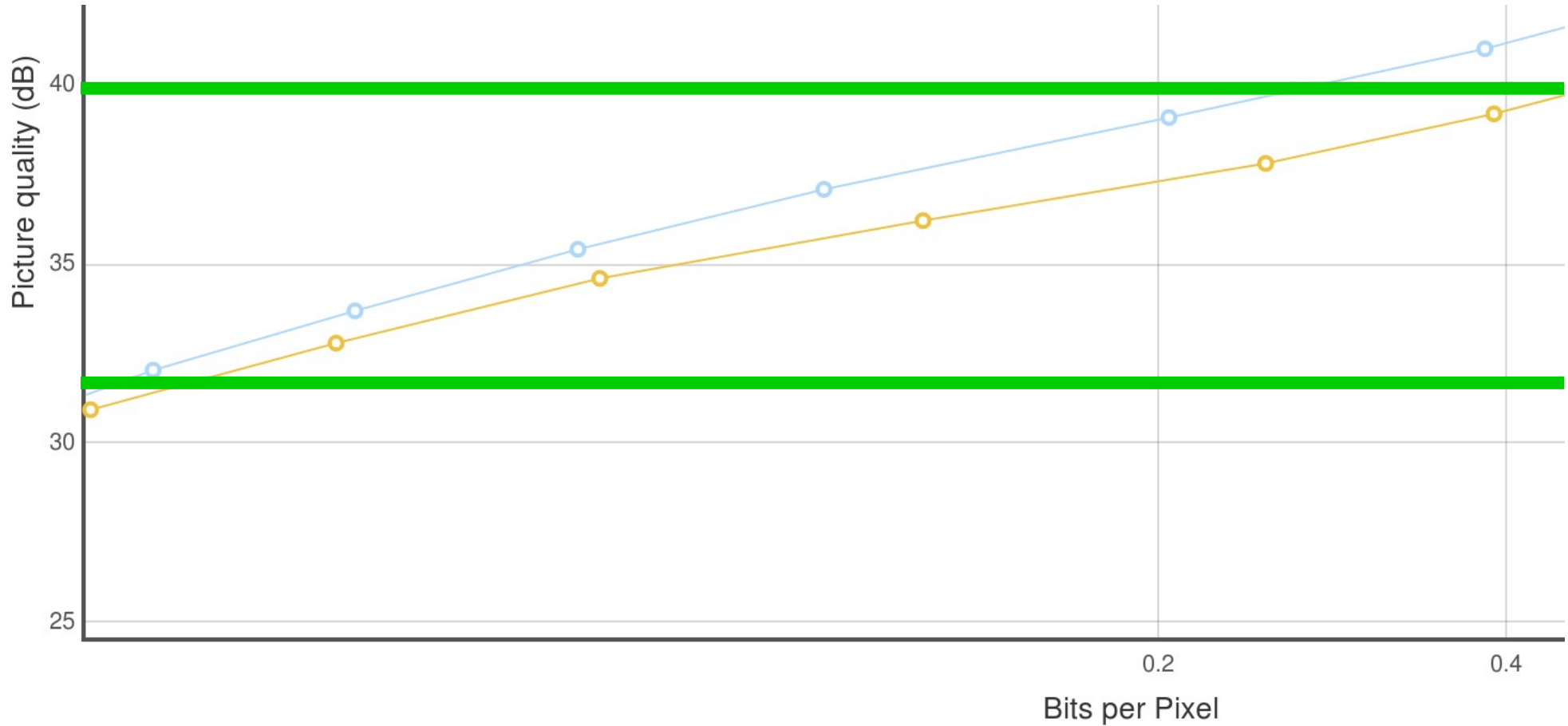
BD-Rate Range – Anchor-derived



BD-Rate Range – Anchor-derived




BD-Rate Range – Anchor-derived



BD-Rate Range – Anchor-derived

- Using libvpx-vp9 as anchor
- Not perfect – quality settings for libvpx-vp9 don't exactly map to quality either



**THIS SUCKS LESS
THAN PREVIOUSLY**

Software updates

- The AWCY repo has been updated with `bd_rate_report.py` to produce the new BD-Rate numbers [1]
- The `rd_tool.py` distributed encoding tool has been updated to allow a local configuration of workers [2]

[1] <https://github.com/tdaede/awcy>

[2] https://github.com/tdaede/rd_tool

TODO for -03

- Refine video selection based on actual computational limitations
- Add explicit 4:2:2 and 4:4:4, 10 bit test sets
- 2nd implementation of curve calculation

draft-ietf-netvc-testing-02

IETF 95

1

This is the update to the NETVC draft update.

We've jumped a number this time, that means this update is twice as good.

Sub-Agenda

- Improvements to set selections
- BD-Rate calculations
- Software updates
- TODO

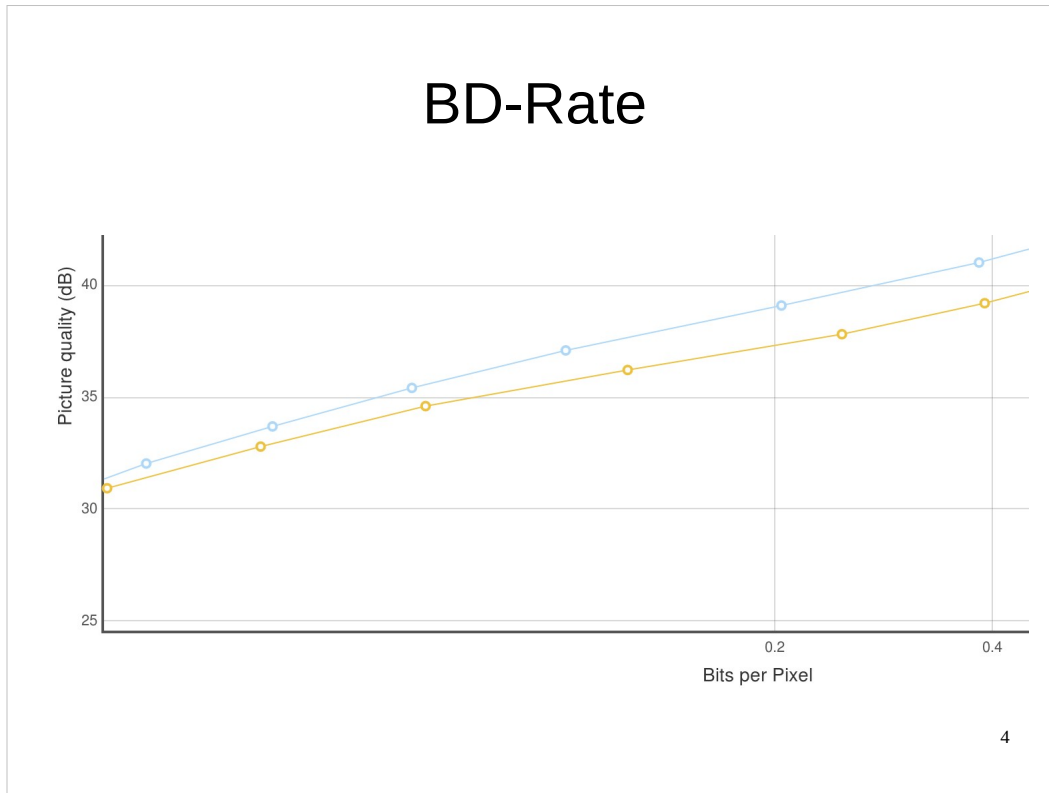
Set Selections

- 'Speedy' regression test
 - netflix-2k-1
 - vc-720p-1
- 'Objective testing' set
 - netflix-2k-1
 - video-hd-3
 - vc-720p-1
 - vc-360p-1
 - twitch-1
- Comprehensive periodic set

3

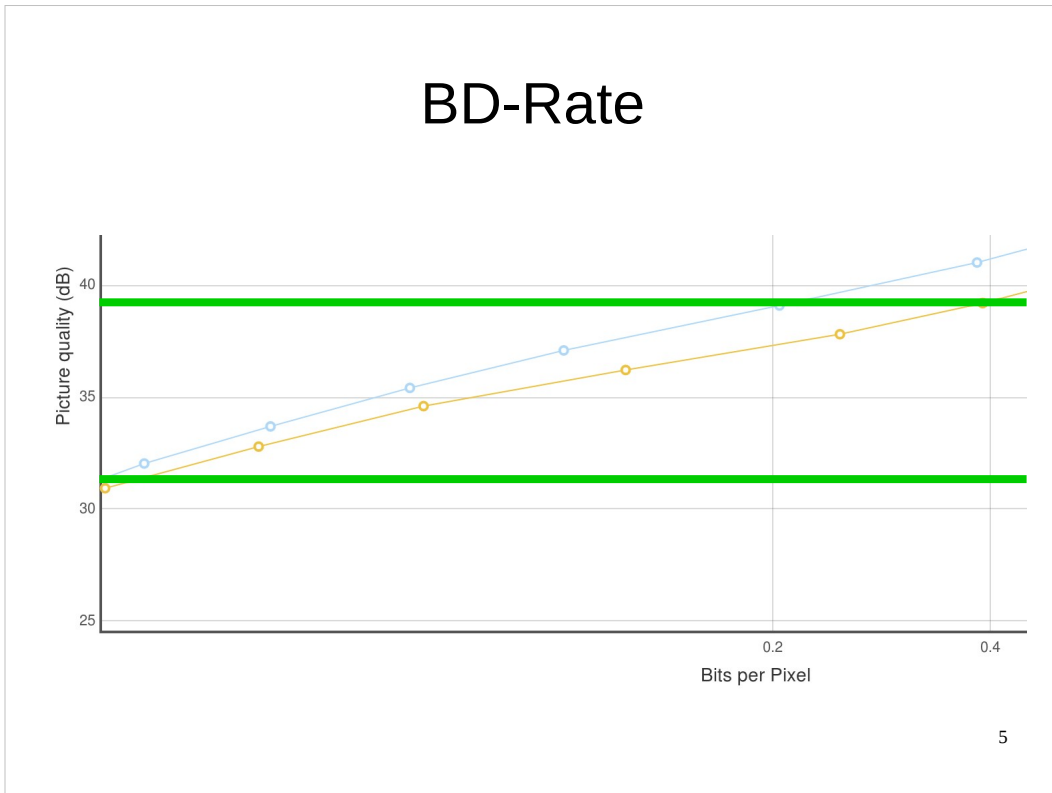
There are now specific sets for particular scenarios. A fast 'Regression test' set and a slow 'Objective testing' set. The slow 'Objective testing' set is actually too slow at the moment – it will need to be cropped down a bit more.

BD-Rate

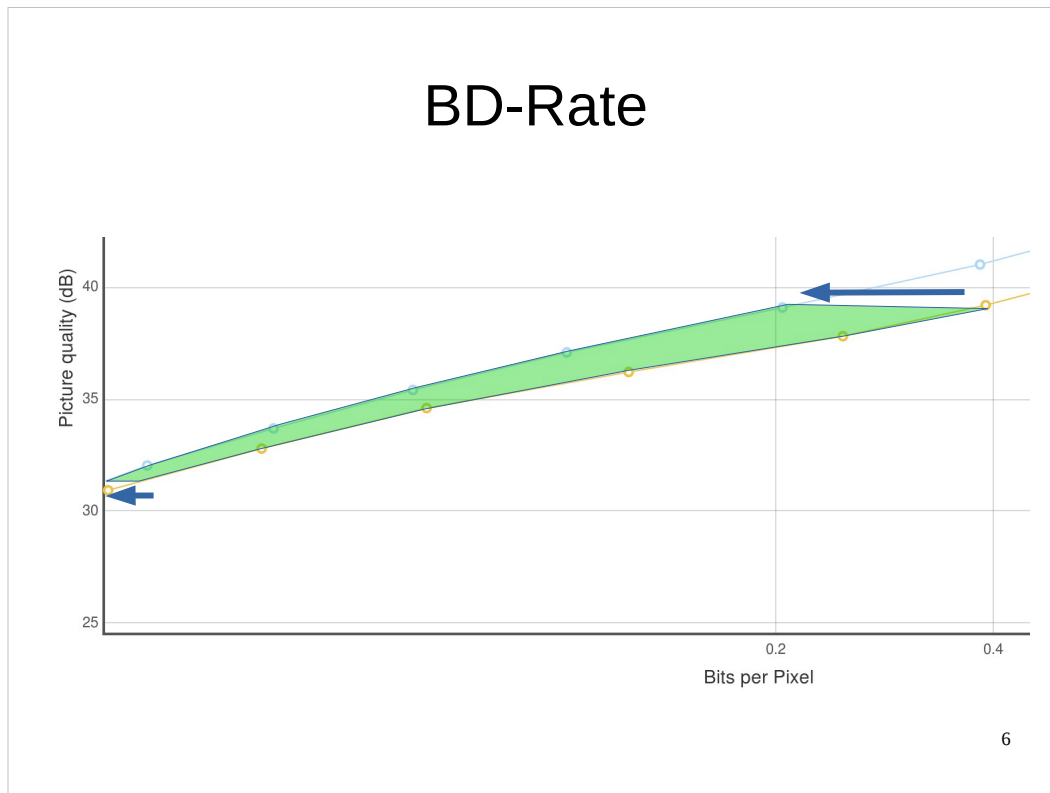


The general principle of Bjontegaard rate is to compute a single number from these curves, with the X axis being quality and Y being picture quality.

BD-Rate



To do that, we select a range of qualities



And integrate between the curves, to produce an average rate reduction number.

As you can see, often the rate reduction will be higher at one end than the other – this is why an average is performed over a range.

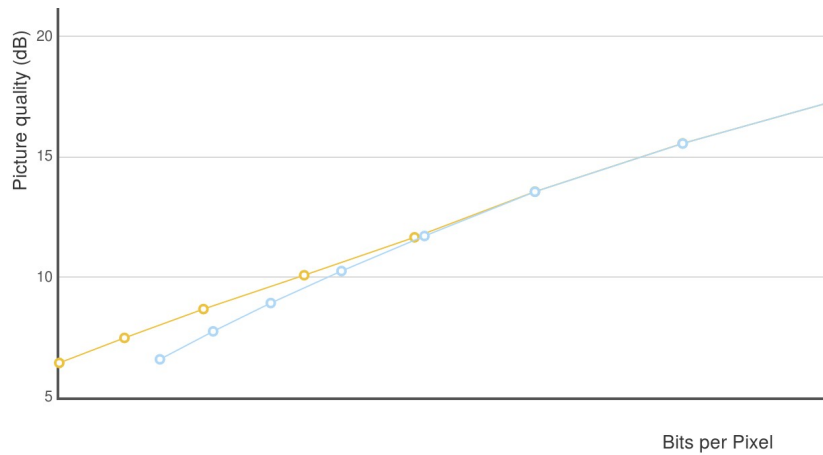
BD-Rate Range

- Selecting a reasonable quality range is hard
 - We have a bunch of metrics on different scales
 - Metrics tend not to be valid on an absolute scale [1]
- There are a couple options to produce the range in a uniform manner

[1] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of PSNR in image/video quality assessment," in *Electronics Letters*, vol. 44, no. 13, pp. 800-801, June 19 2008.

This is now the third try at this.

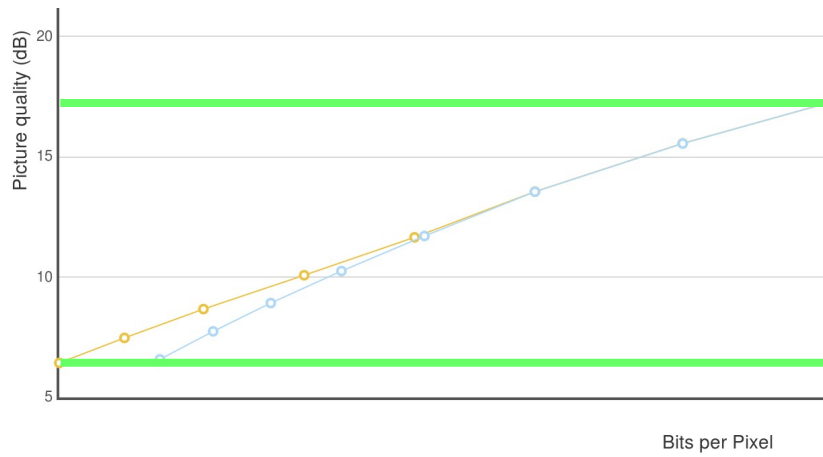
BD-Rate Range – Quantizer-derived



8

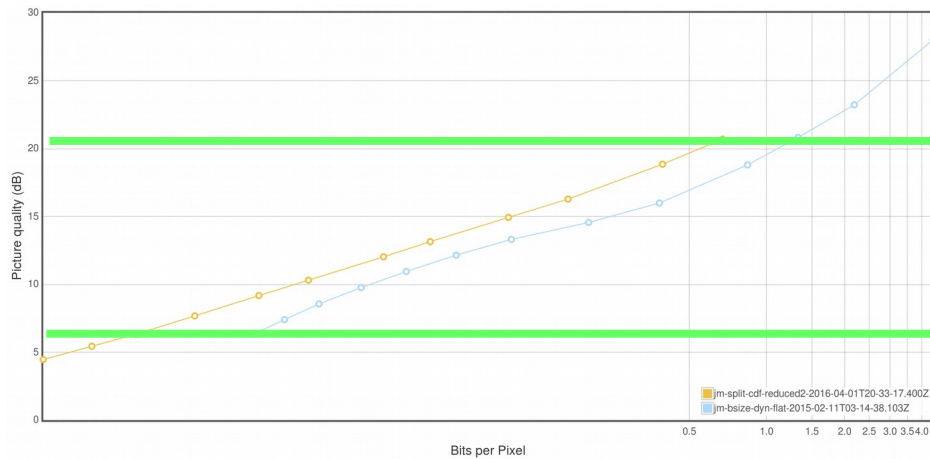
The classic way to do this is derive the range from quantizers. In this graph, we've made a change to the codec which cost a lot of extra bits at each quantizer. You can see that the quantizers line up nicely.

BD-Rate Range – Quantizer-derived



The range is obvious.

BD-Rate Range – Quantizer-derived

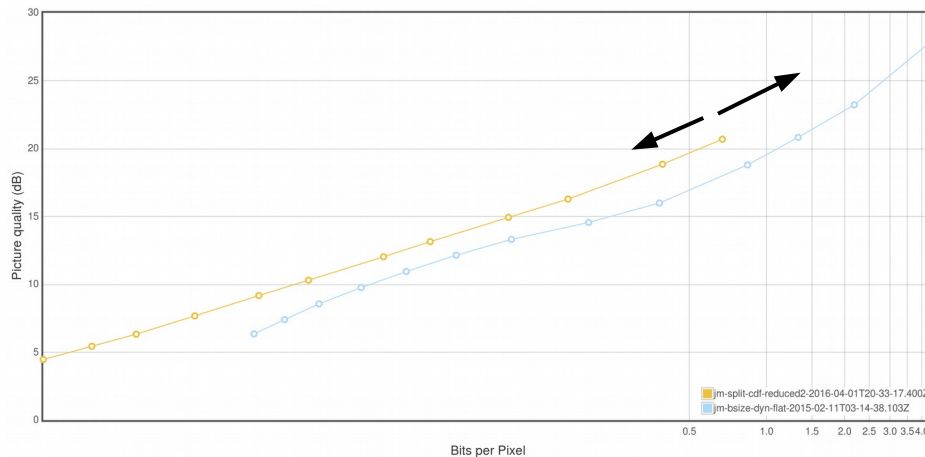


10

Now we swapped out the codec. The quantization method is totally different, leading to pretty quantizer correspondance. We pick the intersection of the ranges.

Unfortunately, the intersection is controlled by the exact quantizers.

BD-Rate Range – Quantizer-derived

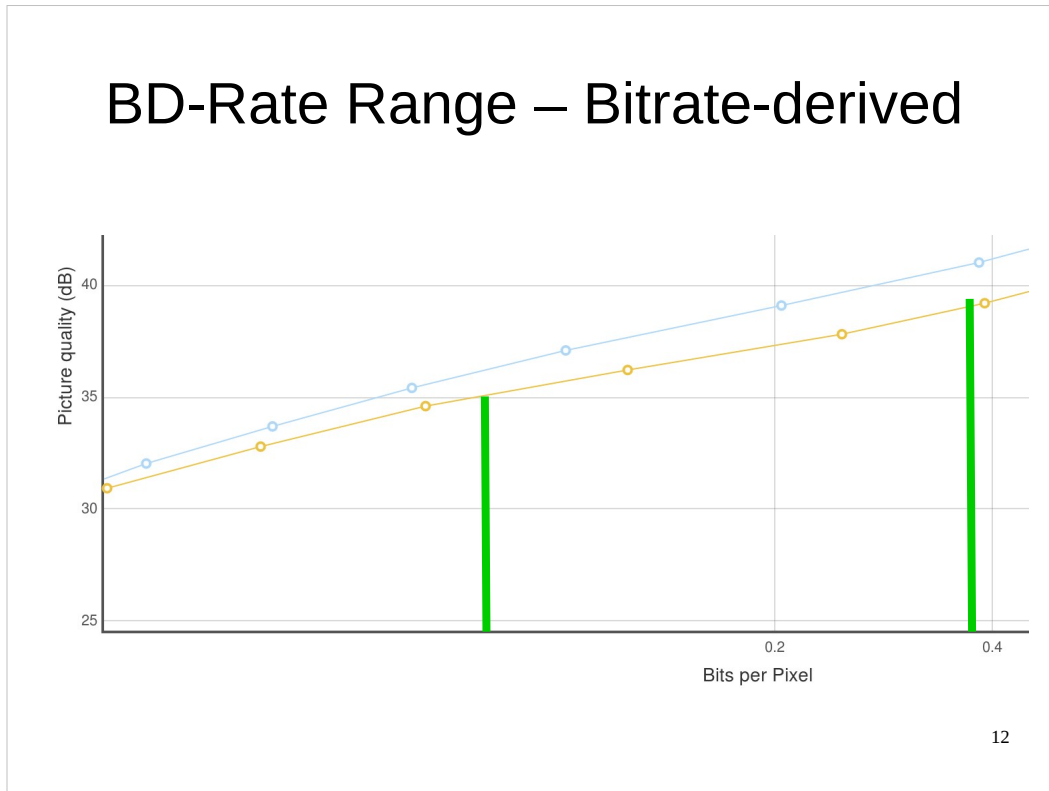


11

We can see that the top line is not only better than the bottom line, but the difference is larger at the top. By moving the quantizer along the direction of arrows, we can increase or decrease the area in the top range, and therefore the average.

This isn't done on purpose – many things can cause this, like changing lambda. It introduces unwanted noise in the results. Let's try to cure this.

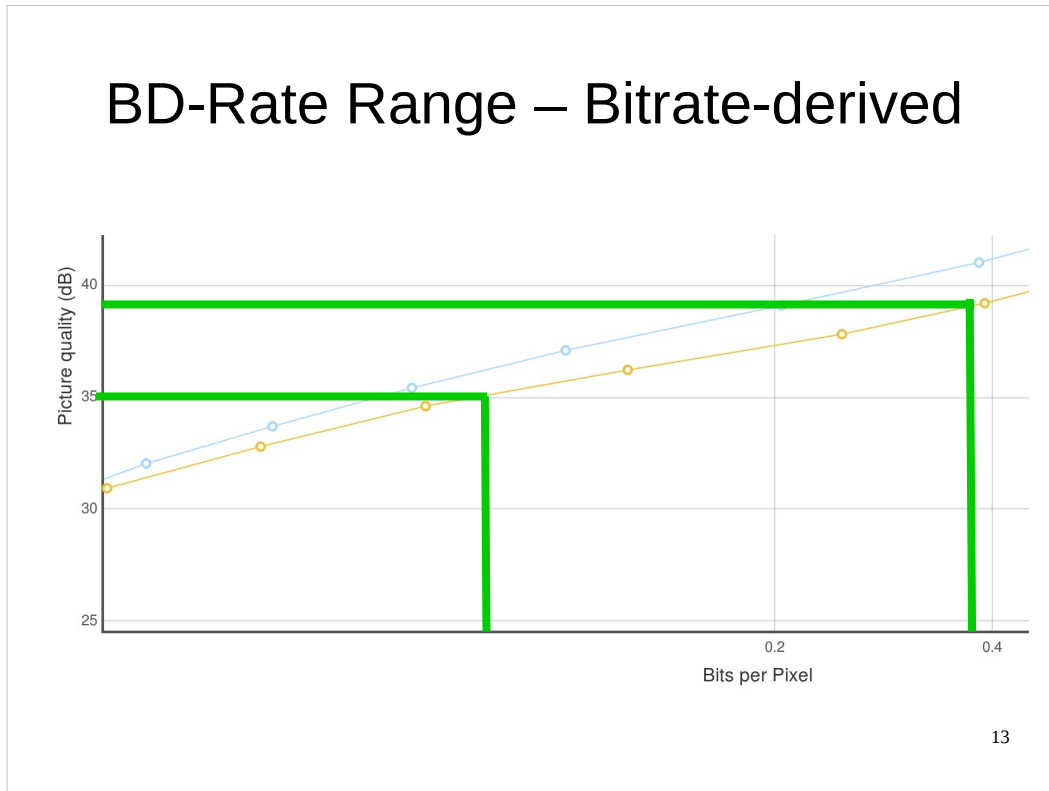
BD-Rate Range – Bitrate-derived



12

I came up with this pretty cool method to produce a constant quality range in the last version of the draft. We pick a range of bitrates instead of qualities.

BD-Rate Range – Bitrate-derived



13

Then we look at the qualities of those bitrates. Note that it matters which curve you pick for this. But there's a bigger problem.



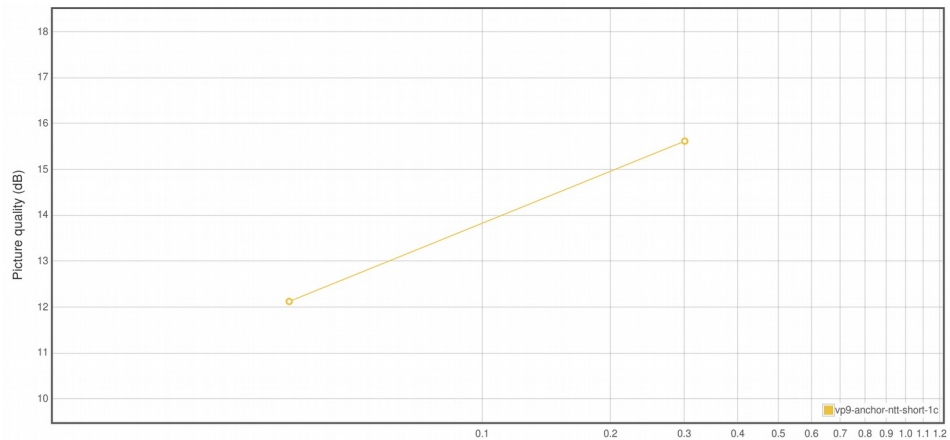
PedestrianCrossing and ToddlerFountain from Netflix
CC-BY-NC-ND 4.0 <https://media.xiph.org/video/derf/>

We got a lot of video! A lot of variety! But it all needed different bitrates.

At the same bitrate, we can have an impeccable talking head, or the ugliest Minecraft pixels you have ever seen. It turns out real-world users are smart and will do things like use appropriate bitrates, scaling the video and framerate when necessary.

It's possible to choose a bitrate for each video. But that would be insane, and I don't trust my judgement **THAT** well.

BD-Rate Range – Anchor-derived

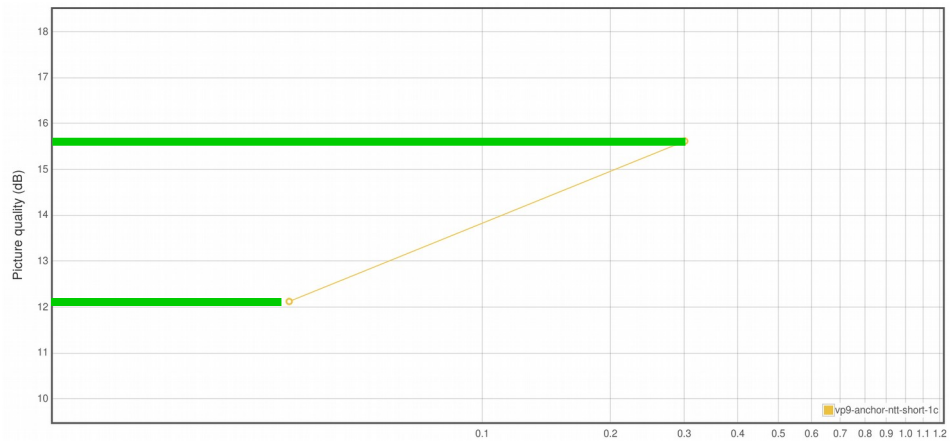


15

What if we could get a more constant quality, but avoid the problems of the endpoints moving about?

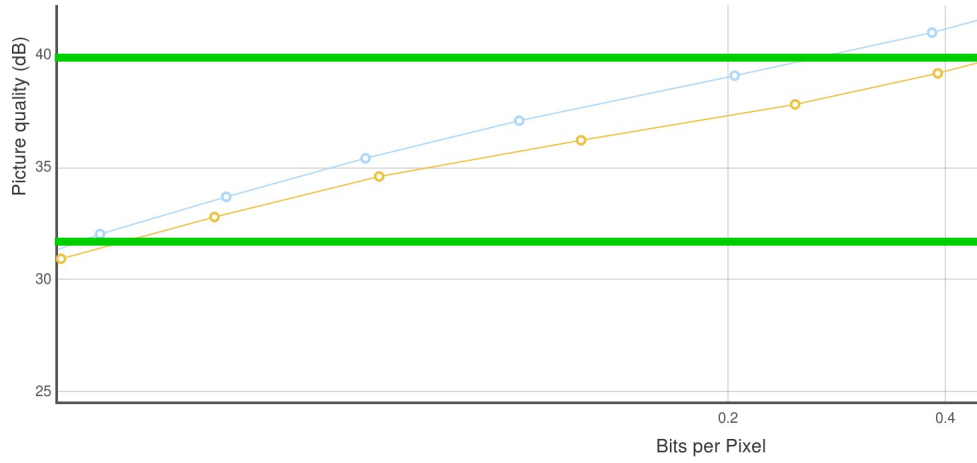
Well, there's a solution – rather than use the endpoints of the codec we're using, use the endpoints of a reference codec. Here I've run two points through VP9.

BD-Rate Range – Anchor-derived



And we can select a range for that.

BD-Rate Range – Anchor-derived



BD-Rate Range – Anchor-derived

- Using libvpx-vp9 as anchor
- Not perfect – quality settings for libvpx-vp9 don't exactly map to quality either



Software updates

- The AWCY repo has been updated with `bd_rate_report.py` to produce the new BD-Rate numbers [1]
- The `rd_tool.py` distributed encoding tool has been updated to allow a local configuration of workers [2]

[1] <https://github.com/tdaede/awcy>

[2] https://github.com/tdaede/rd_tool

TODO for -03

- Refine video selection based on actual computational limitations
- Add explicit 4:2:2 and 4:4:4, 10 bit test sets
- 2nd implementation of curve calculation