FAST TRANSRATING

BASED ON FLEXIBLE ENCODE INFRASTRUCTURE

April 2018

TABLE OF CONTENTS

1 INTRODUCTION

Flexible Encode Infrastructure (FEI) is an extension of Intel® Media SDK intended for direct access to hardware media processing pipeline. This paper compares a fast transrating technique based on FEI with conventional transcoding. Pros and cons are discussed, and recommendations for further improvement are given.

The paper starts with visual quality evaluation. Bitrate boundaries in which fast transrating can be used instead of conventional one, are identified. Quality dependency on stream content is investigated. Basic evaluation of subjective visual quality is performed.

Next, density of fast transratings and conventional transcodings is evaluated. Bottlenecks are identified and CPU, GPU, and memory usages are compared for both cases.

The paper concludes with a short summary and several appendices containing command lines used for analysis, system configuration, and other low-level details.

2 VISUAL QUALITY

2.1 PIPELINE DESCRIPTION

Figure 1 shows the pipeline used for visual quality evaluation.

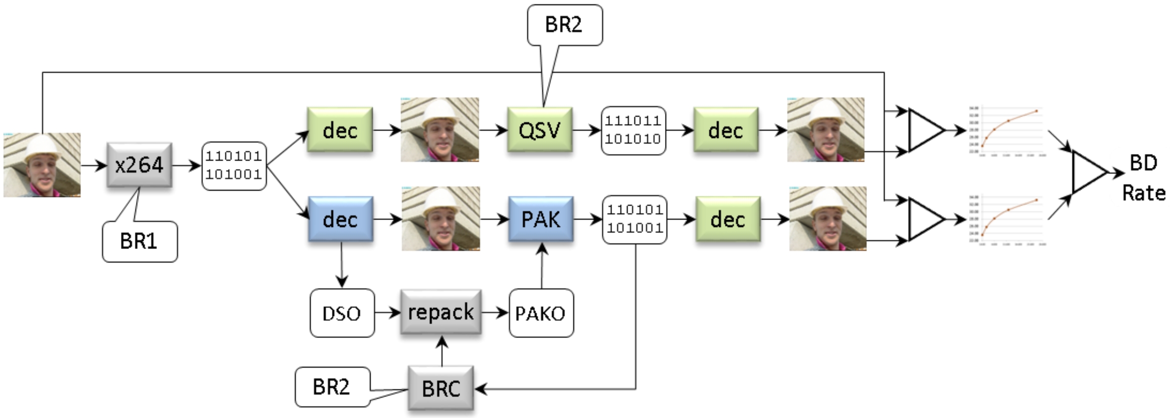


Figure 1. Pipeline used for VQ evaluation

In Figure 1, gray rectangles denote components developed by third parties, green rectangles represent standard Intel Media SDK components, and blue rectangles show the FEI extensions of Intel Media SDK. BR1 and BR2 are the input and output bitrates for transcoding.

On the first stage, original YUV stream is encoded to AVC with bitrate BR1. Preliminary experiments showed that the visual quality of fast transrating heavily depends on the quality of the source stream, so the high quality mode of x264 encoder was used. See “Appendix A. Quality evaluation details” for more details about command line options.

On the second stage, the actual comparison begins. The first pipeline consists of an Intel Media SDK-based decoder followed by an Intel Media SDK encoder. The source AVC stream is decoded to a raw YUV, then encoded again with different bitrate BR2. The second pipeline consists of an Intel Media SDK decoder with an FEI extension that delivers a so-called “decode stream out” (DSO) to the application. This DSO consists of MVs from source bitstream, plus MB level syntax elements. Afterwards, this DSO is being repacked into a PAK object which, along with output raw YUV frames and MVs, is fed to another FEI extension called PAK. It does actual packing of provided PAK object into the output bitstream.

In this particular experiment, no additional processing of PAK object is done. Syntax elements from the source bitstream are fed to PAK as-is, with minimum changes to satisfy PAK requirements and modification of frame-level QP by bitrate control. BRC is also straightforward, with no additional information from source bitstream (i.e., current source frame QP, complexity of incoming frames, etc.) used, since the usage of such information would make a comparison with QSV encoder unfair.

2.2 TEST CONTENT

The primary goal of this paper is to define the boundaries inside which the usage of fast transrating is beneficial. To achieve this goal, 23 full HD streams (1920x1080) were evaluated, which consisted of varied content (including fast and slow motion, static content, highly irregular scenes like waves on the water, and so on). Screen content was also included in the preliminary evaluation that was done in constant QP mode and showed very good quality in comparison with other types of content. However, screen content streams ultimately had to be excluded from the final evaluation, since the bitrate control algorithm used by the fast transrating pipeline was not adequate to deal with screen content.

Most of the streams were about 500 frames in length; several (used primarily for performance evaluation) had up to 15,000 frames. Evaluation of more resolutions, including SD for visual quality and 4K, is planned to be performed in the future.

2.3 VQ AS FUNCTION OF BITRATE

Different ranges of input and output bitrates and QP differences were evaluated. Some cases were discarded due to low quality of fast transrating (e.g., low input bitrate cases); others were deemed impractical (e.g., constant QP mode). The rest of the results are presented in this chapter.

2.3.1 40Mbps input

Figure 3 shows typical RD curve for fast transrating from a 40Mbps source (“tears_of_steel” stream) to 8 - 40 Mbps destination. The first frame of the stream is presented in Figure 2. About 75% of evaluated streams have similar RD curves. (We discuss the remaining 25% in section 2.5, “VQ dependency on content”.)

Figure 2. An example frame from the "tears_of_steel" stream

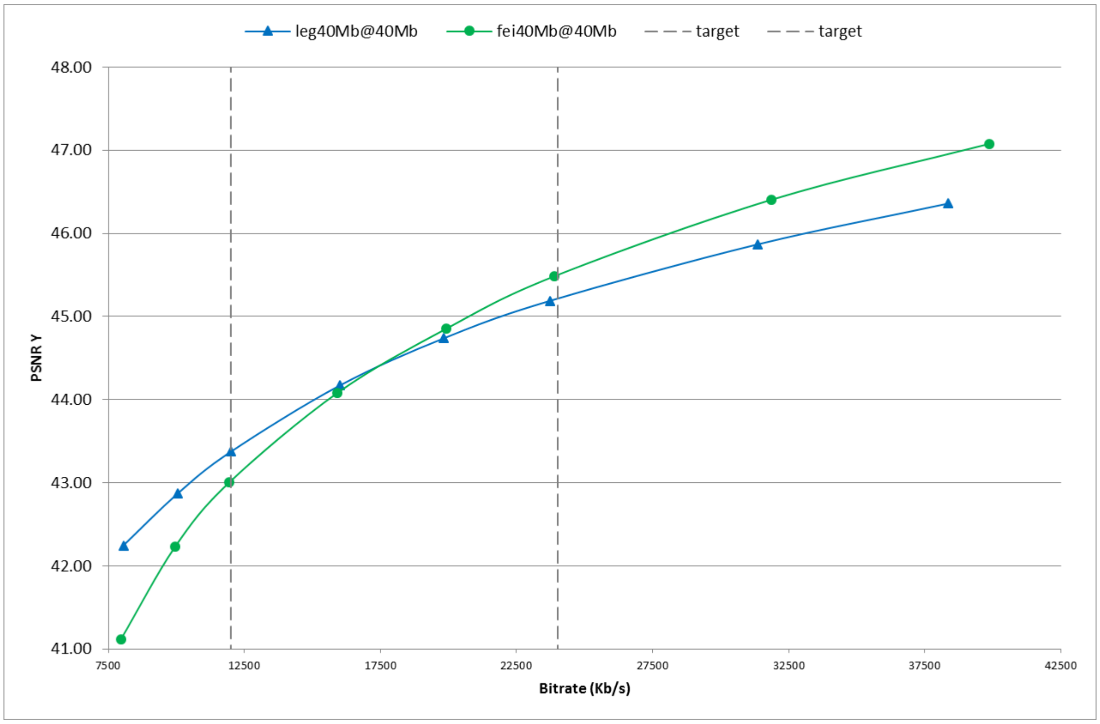


Figure 3. Typical RD curve for 40Mbps source

In this chart, the green line represents fast transrating and the blue line represents conventional transcoding. Two vertical dashed lines mark the bitrate range that was used for BD-Rate calculation. (BD-Rates for all evaluated streams and bitrates can be found in the “2.4 BD-Rates” section.)

As seen in the chart, for high bitrates the fast transrating quality becomes better than the conventional transcoding quality. This is because fast transrating preserves original MVs and mode decisions (MB types), which are made by the high-quality encoder used to encode the source stream. With the decrease in bitrate, the MVs, and especially mode decisions, become suboptimal, so the fast transrating quality degrades and goes below the conventional transcoding quality.

The overall bitrate reduction which conserves acceptable visual quality (VQ) is almost 4x, from 40Mbps to 12Mbps.

2.3.2 30Mbps input

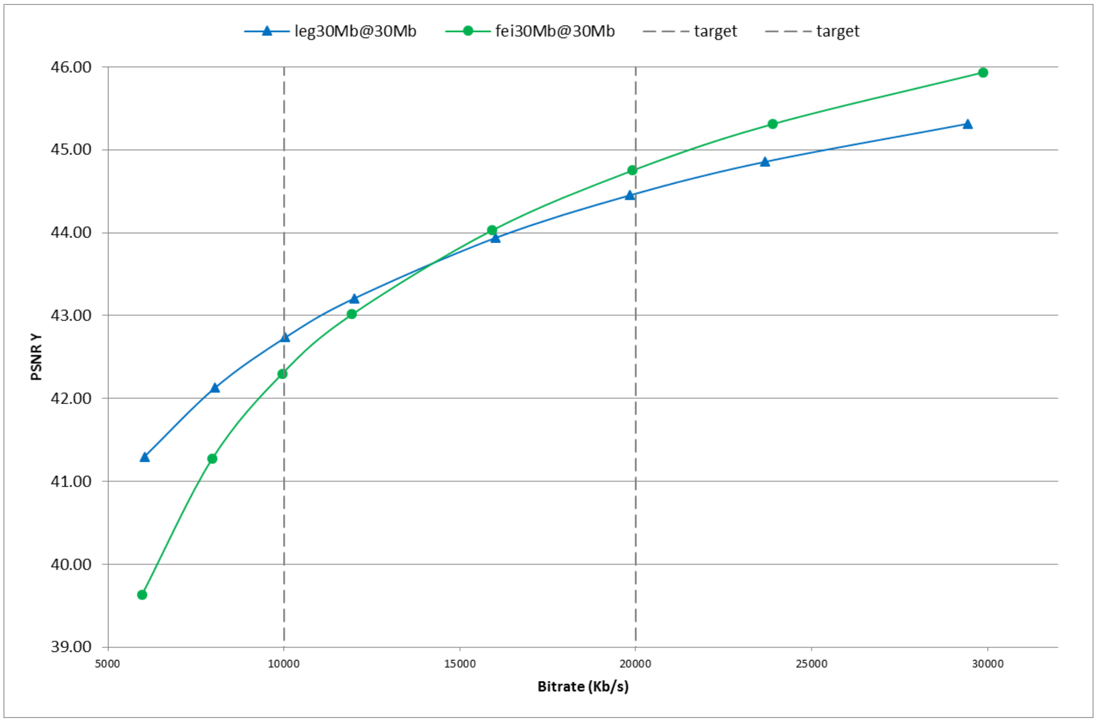


Figure 4. Typical RD curve for 30Mbps source

In Figure 4, the green line represents fast transrating, and the blue line represents conventional transcoding. Two vertical dashed lines mark the bitrate range that was used for BD-Rate calculation. (BD-Rates for all evaluated streams and bitrates can be found in “2.4 BD-Rates” section.)

The RD curves behave similarly to the 40Mbps input case, but the quality for low bitrates degrades faster and the overall bitrate reduction limited by VQ degradation is less—about 3x, from 30Mbps to 10Mbps.

2.3.3 20Mbps input

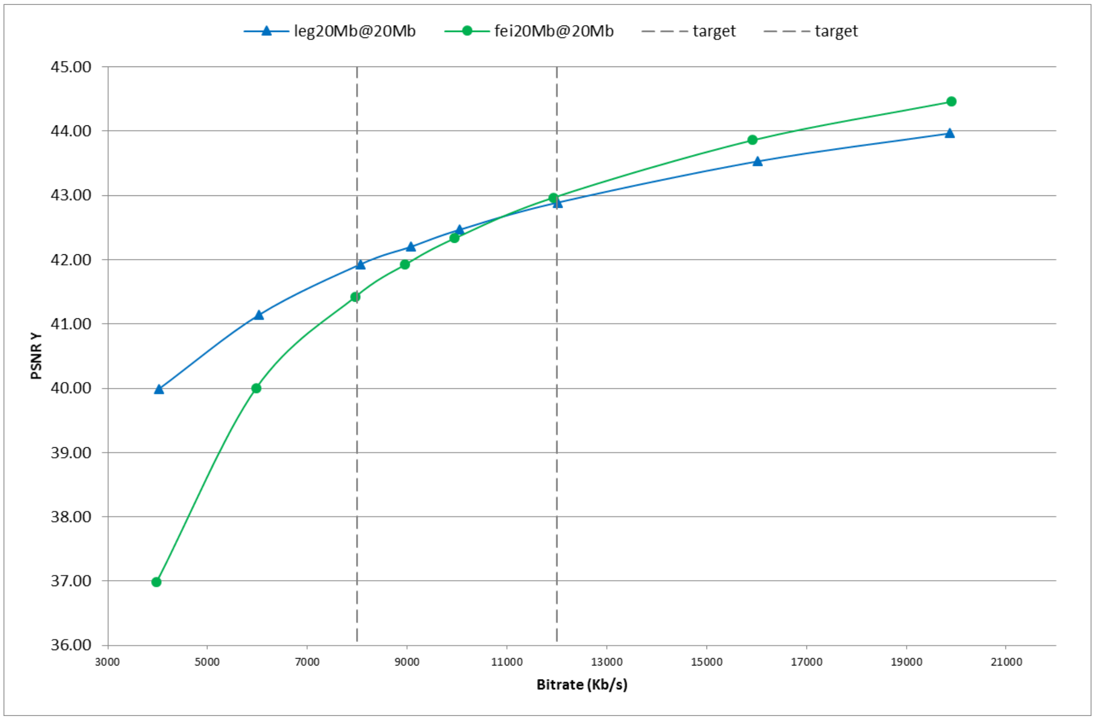


Figure 5 typical RD curve for 20Mbps source

In Figure 5, the green line represents fast transrating and the blue line represents conventional transcoding. Two vertical dashed lines mark the bitrate range that was used for BD-Rate calculation. (BD-Rates for all evaluated streams and bitrates can be found in "2.4 BD-Rates" section.)

Again, RD curves are similar to previous cases, but for low bitrates the quality degrades even faster and the bitrate reduction is smaller—about 2.5x (from 20Mbps to 8Mbps).

2.3.4 10Mbps input

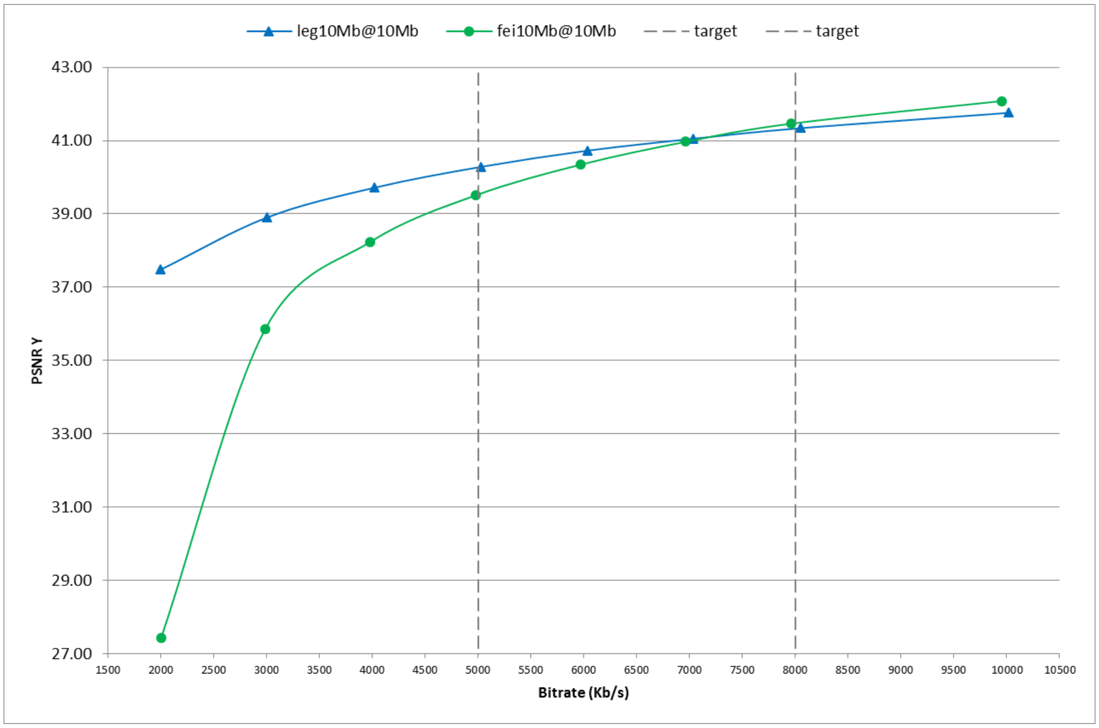


Figure 6 typical RD curve for 10Mbps source

In Figure 6, the green line represents fast transrating and the blue line represents conventional transcoding. Two vertical dashed lines mark the bitrate range that was used for BD-Rate calculation. (BD-Rates for all evaluated streams and bitrates can be found in “2.4 BD-Rates” section.)

Although the general shape of the RD curve remains the same as for higher input bitrates, the useful bitrate reduction had become even smaller—only 2x, from 10 Mbps to 5Mbps.

2.4 BD-RATES

The useful bitrate ranges for BD-Rate calculation (Table 1) were defined from the RD curves gathered on the first stage of investigation. The BD-Rates for each such range were estimated and presented in Table 2.

source bitrate Mbps	**target bitrate Mbps**
40	12-24
30	10-20
20	8-12
10	5-8

Table 1. Useful ranges for different bitrates

	YUV BD Rate (PSNR)			
Sequence	**10Mbps**	**20Mbps**	**30Mbps**	**40Mbps**
Beauty	4.62%	4.06%	12.62%	14.70%
big_buck_bunny	-2.96%	-1.37%	1.97%	2.12%
blue_sky	4.04%	4.92%	8.40%	6.93%
Bosphorus	-2.43%	-9.57%	-8.29%	-9.35%
crowd_run	-7.00%	-13.39%	-9.50%	-10.43%
ducks_take_off	-4.54%	-5.46%	-2.13%	-2.41%
elephants_dream	-22.95%	-24.89%	-19.30%	-20.14%
HoneyBee	18.62%	11.84%	9.23%	9.31%
in_to_tree	-3.84%	-9.10%	-4.47%	-3.51%
Jockey	-4.23%	-4.67%	0.05%	-0.28%
kimono1	4.11%	5.08%	9.10%	8.46%
old_town_cross	7.63%	1.57%	19.63%	17.44%
park_joy	-2.90%	-10.29%	-7.39%	-7.76%
park_scene	2.57%	3.27%	6.24%	3.80%
pedestrian_area	5.52%	4.06%	5.30%	3.50%
ReadySteadyGo	-8.55%	-10.24%	-6.79%	-6.84%
riverbed	-2.27%	-3.27%	-1.01%	-0.54%
rush_hour	10.25%	13.39%	15.82%	13.43%
station2	13.43%	10.19%	10.43%	7.79%
sunflower	10.21%	11.99%	13.32%	12.25%
tears-of-steel	-6.56%	-4.29%	-1.11%	-1.53%
tractor	2.98%	1.75%	4.63%	4.64%
YachtRide	-9.67%	-17.79%	-13.73%	-15.39%
Average	0.26%	-1.84%	1.87%	1.14%

Table 2. BD-Rates for different input bitrates

Positive numbers in the table mean that fast transrating quality is better than conventional transcoding for the specified range of destination bitrates. Negative numbers mean the converse.

As seen in Table 3, smaller input bitrates should be used to achieve smaller output bitrates. However, the scalability is quite bad – while for the 40Mbps input the output bitrate can be reduced up to 4x, the 10Mbps input only allows for only about 2x bitrate reduction. Thus, the bitrate reduction through changing quantization parameters only is limited to high-bitrate cases. For lower bitrates the mode decisions made by the source encoder should be reevaluated as well (e.g., Intra->Inter-> skip MB type conversion should be implemented, although this processing is out of the scope of this paper).

Poor quality of fast transrating, specifically on “elephant dream,” can be explained by the complex nature of the stream. On many occasions, the content quickly changes from static dark scenes to fast motion with lots of fine details. The bitrate control used for fast transrating was not tuned for this particular use case; the dependency between frame QP, bitrate, and distortion was different in comparison to conventional transcoding, but unaccounted for. As a result, the bitrate control often assigned suboptimal QPs, leading to significant quality drop on certain frames—up to 8dB in comparison with the conventional encoder.

2.5 VQ DEPENDENCY ON CONTENT

As seen from the results above, most streams have almost the same behavior with regard to output visual quality and RD curves. The rest falls into two extreme categories described in this section.

2.5.1 Noise-like textures

Figure 7. Frame 150 of the "blue_sky" stream.

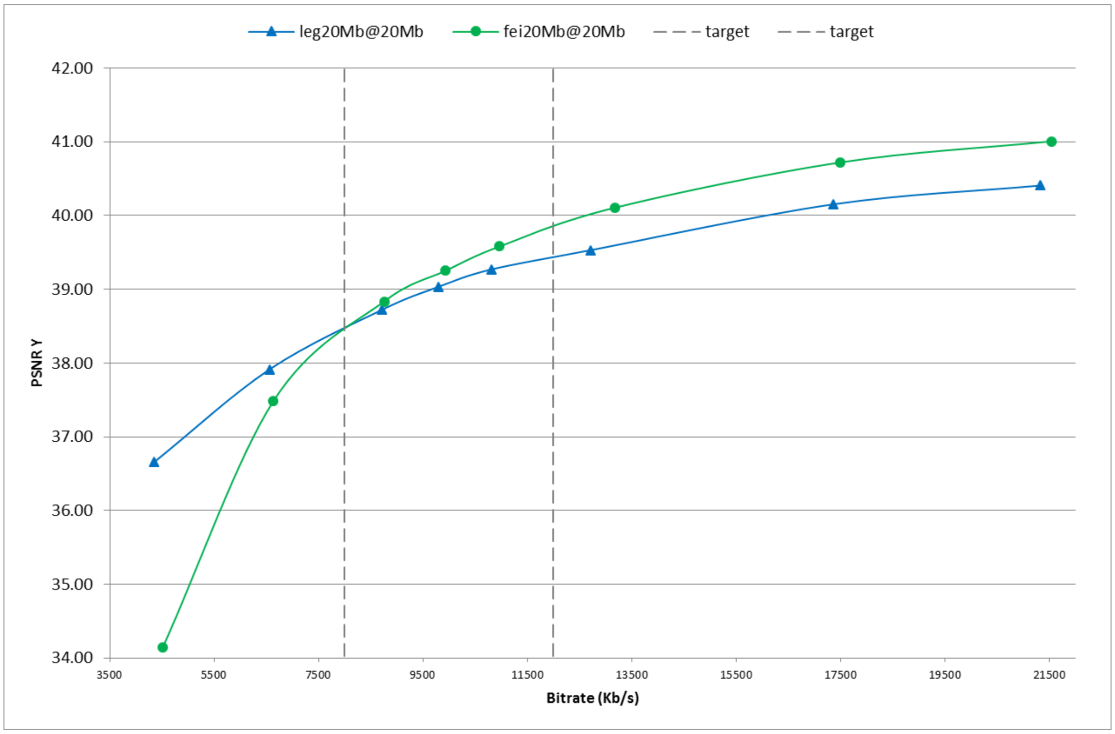


Figure 8. RD curve for "blue_sky" stream, source is 20Mbps.

For these streams, most bits are spent on encoding the transform coefficients. This number can be significantly reduced simply by changing quantization parameters (that is why such streams are well suited for fast transrating). The bitrate of source stream can be significantly reduced without noticeable loss of visual quality in comparison with conventional transcoding. And due to the better mode decision of the source stream, the VQ of fast transrated streams is better in the broad range of bitrates.

For this particular stream, the *(number of bits for MVs) /(number of bit for coefficients)* value is equal to 0.36 (8Mbps target bitrate). In other words, 3x as many bits are spent on coding transform coefficients and MB modes than for coding motion vectors.

2.5.2 Prevalent local movement

Figure 9. First frame of the “rush_hour” stream.

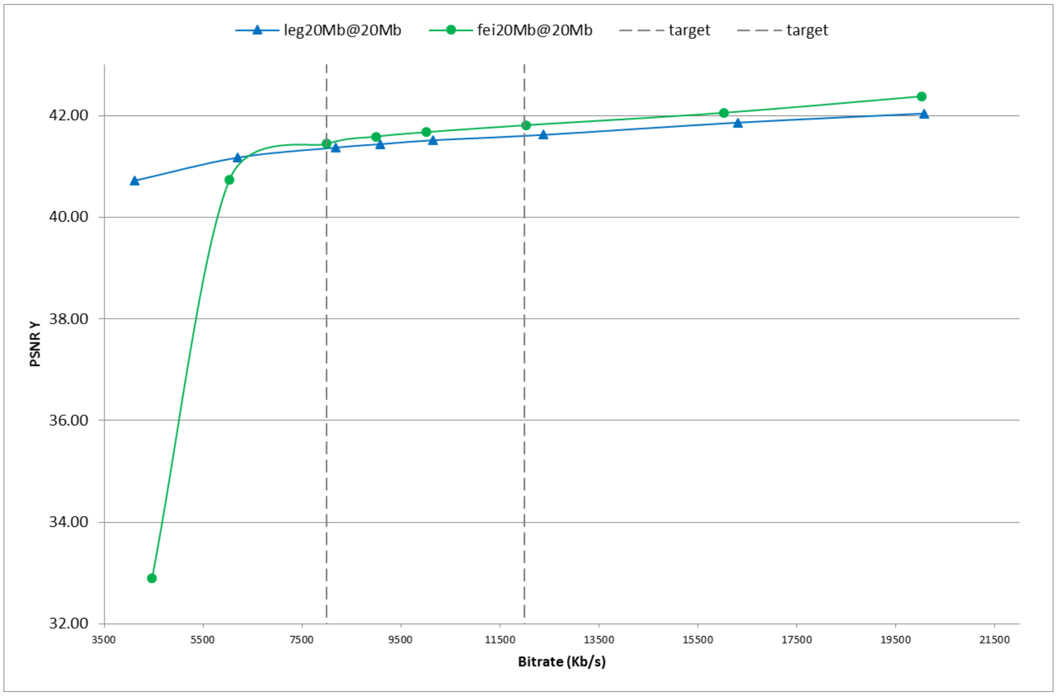


Figure 10 RD curve for "rush_hour" stream, source is 20Mbps.

The streams of this kind have a lot of small, random MVs. This particular stream was shot using a long-focus lens which, combined with the shimmering of hot air, produced a randomized MV field. A lot of bits were spent on MV coding, so fast transrating was powerless to significantly reduce bitrate beyond a certain point. For this stream, judging from the RD curve, it is evident that at around 6Mbps the most of the transform coefficients had been quantized to zero and the VQ plummeted down without gathering any significant bitrate reduction.

For this stream, the *(number of bits for MVs) /(number of bit for coefficients)* value is equal to 0.88 (8Mbps target). This means about the same number of bits were spent on coding transform coefficients and coding MVs.

2.6 SUBJECTIVE QUALITY

For most of the 23 streams mentioned above, we also evaluated subjective visual quality. During the evaluation, a stream transcoded by the conventional transcoder was played side-by-side with the corresponding fast transrated stream. Due to the huge amount of different input/output bitrate combinations, the only two combinations used for evaluations were 40Mbps to 12Mbps and 10Mbps to 4Mbps.

For high bitrates, no significant differences between conventional and fast transrating were observed.

For low bitrates, several cases were identified where fast transrater quality was below that of the conventional transcoder. The root cause of all these cases was poor bitrate control behavior. For example, in frame 2,916 of the "elephants_dream" sequence, the conventional transcoder uses QP 29, while the fast transrater uses QP 40—which obviously leads to significantly worse visual quality. Such poor behavior of the bitrate control algorithm is probably explained by different dependencies between

QP and frame size for the fast transrater. As we already mentioned, we made no special attempts to optimize or tune bitrate control for fast transrater usage.

For the rest of the low bitrate cases, the fast transrater quality is on par with or better than the conventional transcoder quality. The fast transrater quality is especially good for streams with steady global motion on regions with uniform textures, like water or grass. Conventional transcoding often produces “glass wall” effects for such regions (frozen textures) accompanied by arbitrary jumps and minor artifacts. The fast transrater keeps its original smooth motion due to preservation of MVs from the original stream.

2.6.1 Broken background

Figure 11 shows frame 142 of the “park_joy” sequence, 10Mbps to 4Mbps. Here, a group of people move from left to right with the camera following them so that the group is almost steadily located in the center of the screen, while the background moves form right to left. In the stream encoded by a conventional transcoder, the grass directly below the moving figures is static most of the time with regular, short, abrupt jumps to accommodate the actual movement. This produces a very noticeable effect. In contrast, the fast transrated stream has the grass below group of people moving smoothly together with the rest of background. Figure 12 and Figure 13 show the motion vectors for this region of the frame for both streams.

Figure 11. Frame 142 of "park_joy" sequence, broken background effect.

Figure 12. MVs for frame 142 of "park_joy", conventional transcoder. The grass below moving figures has wrong MVs, similar to MVs of figures.

Figure 13. MVs for frame 142 of "park_joy", fast transrater. The grass below figures has correct, global motion.

2.6.2 Jerkiness

Figure 14 shows frame 23 of the “park_scene” sequence, 10Mbps to 4Mbps, as another example of visual artifacts mitigated by fast transrating. Here, the background is relatively steady, but the tree trunks quickly move from right to left across the frame. Conventional transcoding produces very noticeable “shimmering” or “jerking” effects on the trunks. As Figure 15 shows, conventional transcoding ignores trunk movement and propagates background MVs onto the trunk. At the same time, fast transrating preserves the true motion (Figure 16) and the trunk moves smoothly without jerkiness.

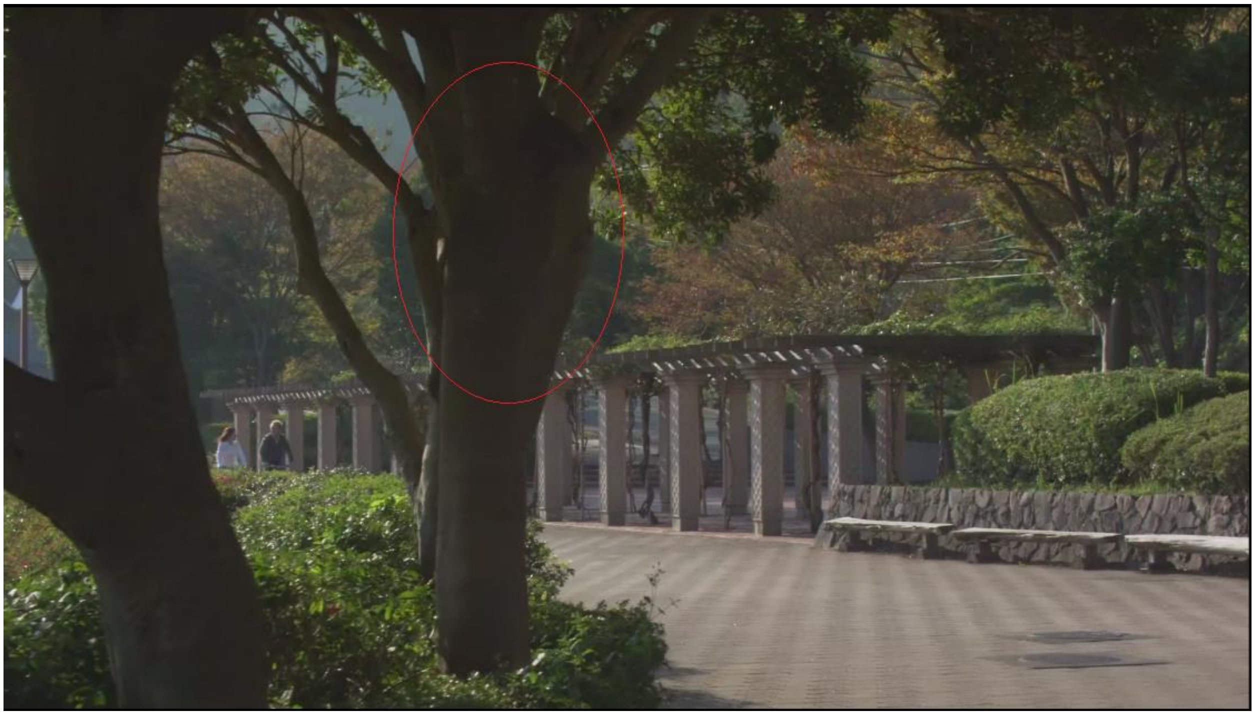


Figure 14. Frame 23 of "park_scene" sequence, jerking bark on the tree.

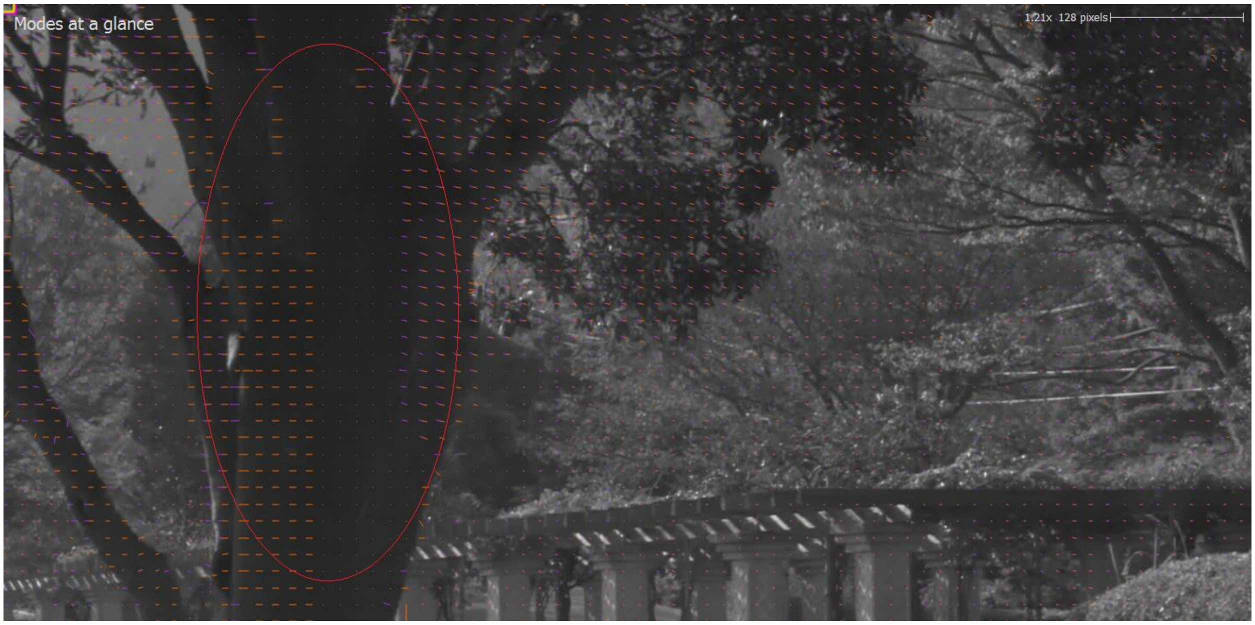


Figure 15. MVs for frame 23 of "park_scene", conventional transcoder. The bark on the tree has wrong MVs, same as the background MVs.

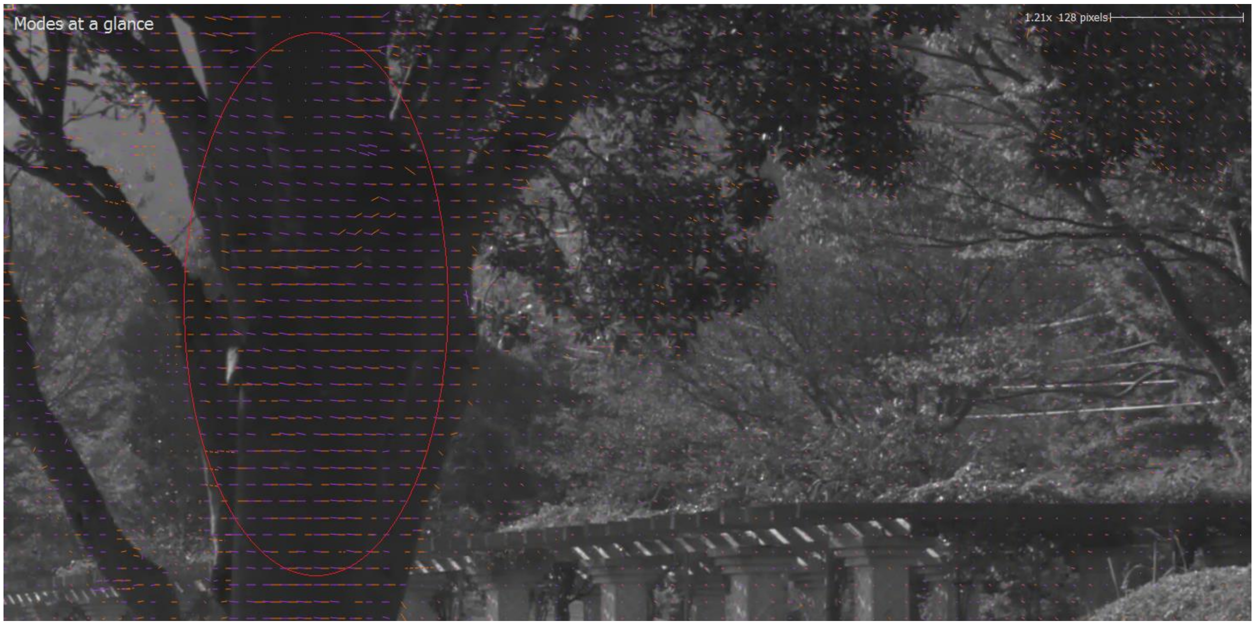


Figure 16 MVs for frame 23 of "park_scene", fast transrater. The bark on the tree has correct MVs, equal to true motion.

3 PERFORMANCE

The bitrate range useful for fast transrating was identified during quality evaluation. The next step was to estimate the transcoding density for both for fast transrating and conventional transcoding using a subset of streams and bitrates employed in quality evaluation. The command lines used for quality and performance estimation were exactly the same, so the quality/performance tradeoff is easily estimated.

Only long input streams were used in performance evaluation, from 1,500 to 15,000 frames long—some obtained from shorter streams via concatenating the short source YUV stream to itself multiple times (using standard Linux* cat command) and then encoding the result YUV file with a high-quality encoder.

Both HD and SD input stream cases were measured. The SD input streams were produced from the same source YUV files used to produce the HD streams, but with prior resizing using sample_vpp (see Appendix B for the command line).

3.1 PIPELINE DESCRIPTION

Figure 17 shows the pipeline that was used for density evaluation of transcoder.

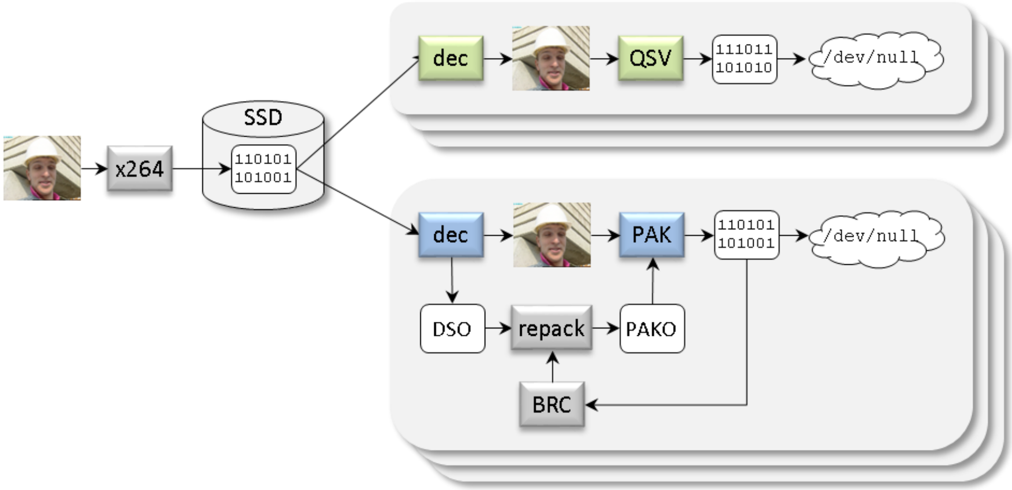


Figure 17. Pipeline used for density evaluation.

The source YUV stream is encoded by x264 encoder and stored on SSD. Next, several simultaneous runs of the same application (with each one representing a single channel) are performed and a per-application transcoding frame rate is estimated. Output (transcoded) streams are discarded (redirected to /dev/null to reduce the disk I/O influence on the density values). The number of applications N_{app} is then slowly increased (starting from some initial channel count) until the desired per application frame rate F_{app} is achieved[1]. Finally, the total execution time T is obtained and the density value calculated according to Equation 1:

$$D = \frac{N_{app} N_{fr}}{T F_{app}}$$

Equation 1. Density estimation.

N_{app} *- number of applications,* N_{fr} *- number of frames in stream,* T *- total execution time from the start of the first channel processing to the end of the last channel processing,* F_{app} *- target frame rate.*

Figure 18 shows an example of the CPU usage overview (gathered by Intel® VTune Amplifier 2018 Update 1 for 16 simultaneously running channels. Each channel (application) has three active threads. Note that the start time is about the same, but the end time is slightly different.

---

[1] In reality, density estimation occurred in two steps – in the first step, the running time of 5 channels was measured; from this, the system frame rate was calculated, producing an initial density estimation. Afterwards, as the second step, the estimation was refined by setting the initial channel count to the density predicted in the first step, and then increasing the channel count until the target FPS is achieved. This significantly speeds up measurements by eliminating redundant iterations.

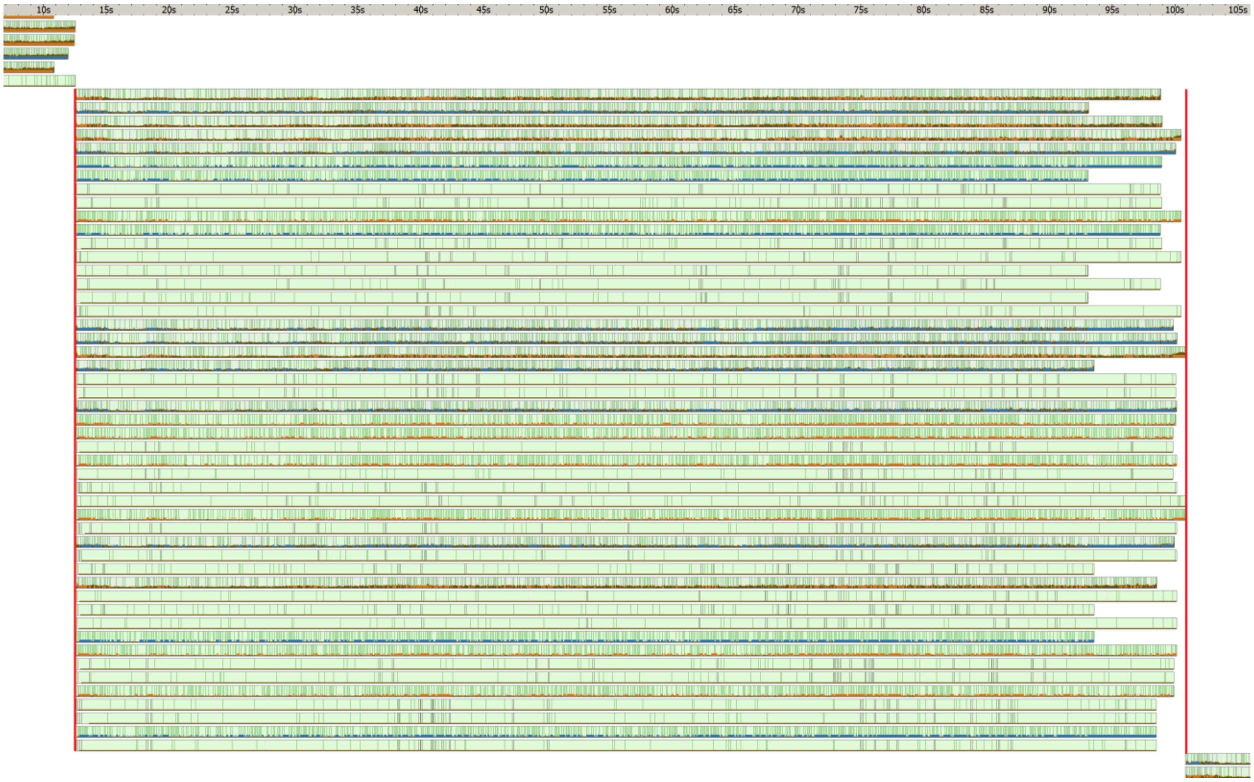


Figure 18. VTune Amplifier view of the system running 16 simultaneous fast transrating channels.

CPU, GPU, and uncore frequencies were fixed with power-saving mode set to performance. (For more about used command lines and system configuration see Appendix B. Performance evaluation details".)

3.2 HD PERFORMANCE

3.2.1 Density

The results of density estimation for HD streams are shown in Figure 19. Six different streams were used, each one was profiled for four different bitrates.

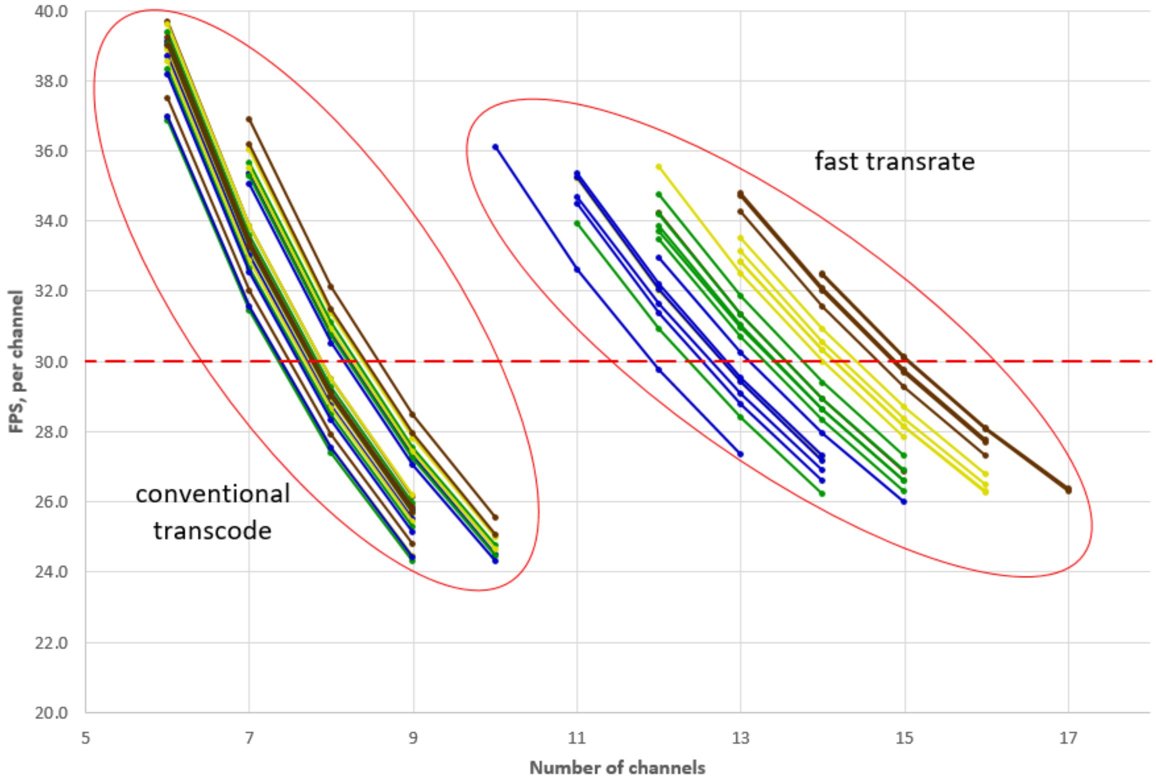


Figure 19. Fast transrater versus conventional transcoder density for six HD streams and four bitrates

In Figure 19, blue lines represent 40Mbps to 12Mbps transcoding, green lines represent 30Mbps to 10Mbps, yellow lines represent 20Mbps to 8Mbps, and brown lines represent 10Mbps to 5Mbps. There are multiple lines of the same color in the chart, representing different streams. The group of lines on the left represents conventional transcoding; the one on the right represents fast transrating.

As the chart shows, fast transrating is more sensitive to bitrate than to stream content. For conventional transcoding, the converse is true: the performance is less sensitive to bitrate and more sensitive to stream content.

The average number of channels is 13.60 for the fast transrater and 7.9 for the conventional transcoder; thus, in average, by using fast transrater we can increase the system density by five to six channels.

3.2.2 Density for “elephants_dream” stream

Figure 20 shows density for “elephants_dream” stream. (RD curves for this stream can be found in section 2.3, “VQ as function of bitrate”.)

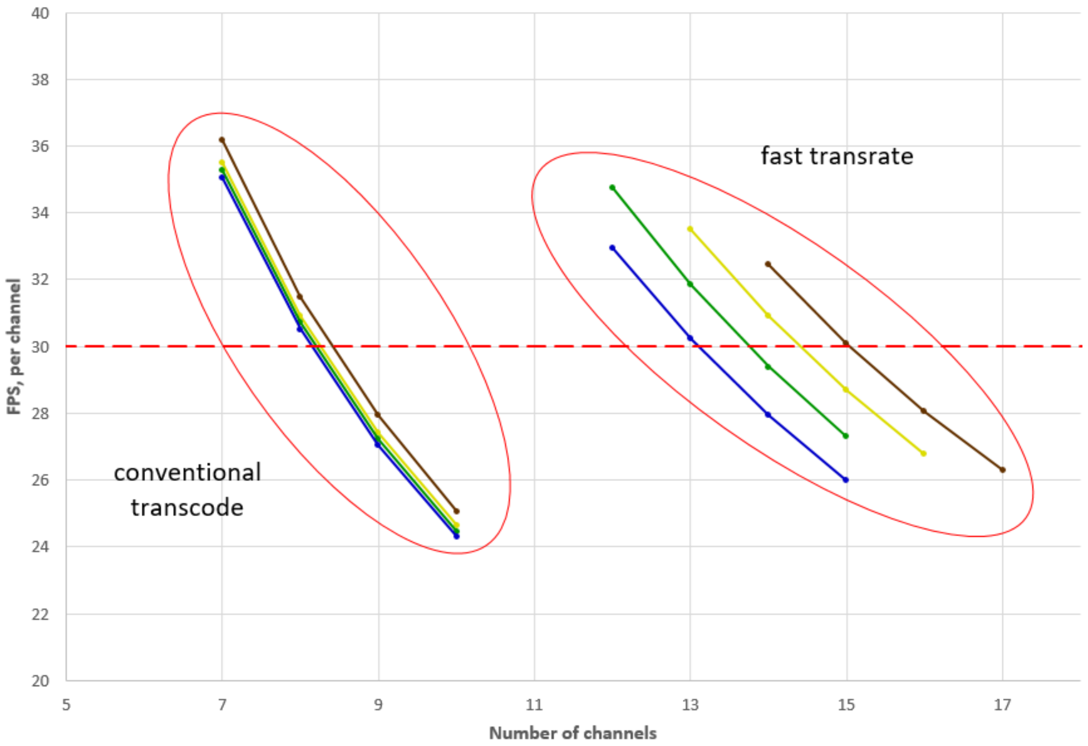


Figure 20. Density for “elephants_dream” stream.

In Figure 20, blue lines represent 40Mbps to 12Mbps transcoding, green lines represent 30Mbps to 10Mbps, yellow lines represent 20Mbps to 8Mbps, and brown lines represent 10Mbps to 5Mbps. The group of lines on the left is for conventional transcoding; the group on the right is for fast transrating.

Type\Input->output BR	**40 Mbps ->12 Mbps**	**30 Mbps ->10 Mbps**	**20 Mbps ->8 Mbps**	**10 Mbps ->5 Mbps**
Fast transrating	13.04	13.72	14.35	14.97
Conventional transcoding	8.12	8.17	8.23	8.38

Table 3. Density (in channel units) for the “elephants_dream" stream.

As the chart and Table 3 show, the fast transrater density significantly depends on bitrate. The density difference between 10Mbps and 40Mbps sources is 15%, whereas for the conventional transcoder, the difference is only 3%.

3.2.3 Conventional transcoding

Figure 21 and Table 4 shows the example of system utilization during an eight-channel conventional transcoding of the “elephants_dream” stream from 30Mbps to 10 Mbps. It is clear that the performance is limited by EUs and render (their utilizations are both 100%), while VDBOXes are at about 50% utilization and the CPU is at about 29%. All data was gathered by 5 seconds Intel® VTune™ Amplifier sampling performed in the middle of the transcoding process.

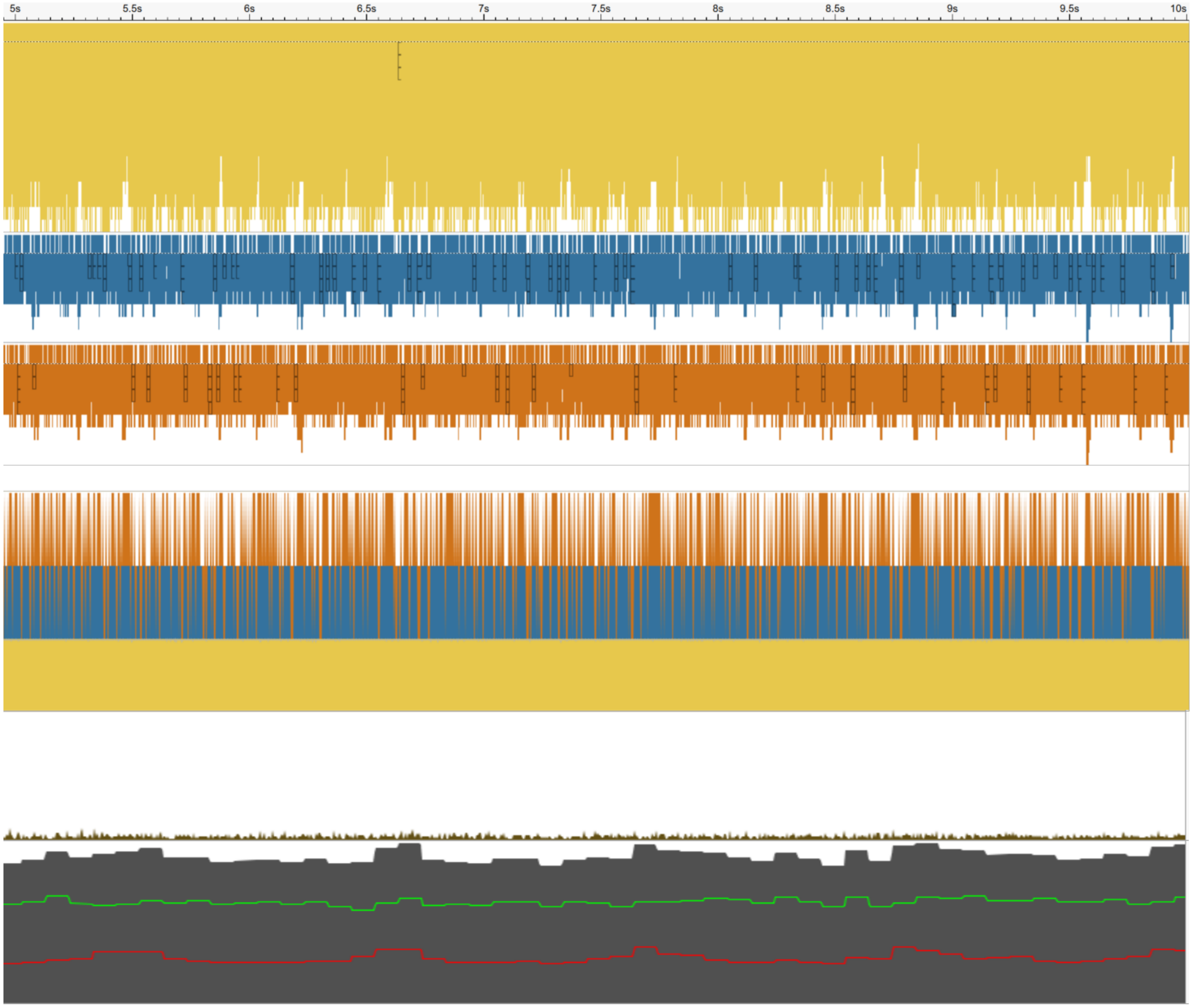


Figure 21. Intel VTune Amplifier view of the system performing an eight-channel conventional transcoding. The first graph from the top (yellow) is the EUs/render queue, the second (blue) is the VDBOX1 queue, the third (orange) is VDBOX2, the fourth - (orange/blue/yellow) is EU&render/VDBOX1/VDBOX2 utilization, the fifth (brown) is CPU usage, and the sixth is DRAM bandwidth (gray is the total; the green line is read; the red line is write).

Indicator	Conventional transcoding	Fast transrating (top performance)	Fast transrating (channel-aligned)
Number of channels	8	14	8
System frame rate	247 fps	417 fps	415 fps
CPU usage (800% max)	29%	300%	300%
VDBOX1 usage	59%	**99%**	**94%**
VDBOX2 usage	45%	**99%**	**92%**
EU/render usage	**100%**	0%	0%
Average DRAM bandwidth: read write	5.7 GBps 3.95 GBps 1.15 GBps	11.8 GBps 7.8 GBps 4.0 GBps	11.3 GBps 7.2 GBps 4.1 GBps
Total DRAM bandwidth per frame	23 MBpf	28 MBpf	27 MBpf

Table 4. Performance indicators for conventional transcoding.

3.2.4 Fast transrating

Figure 22 and Table 5 show the example of system utilization during a 14-channel fast transrating of the “elephants_dream” stream from 30Mbps to 10 Mbps. Judging from the graph, the performance in this case is limited by VDBOXes (almost 100% utilization according to Intel VTune Amplifier). The EUs and render are completely idle, 0% usage, and are not shown on the graph. CPU usage is 300% (out of a maximum 800% possible for the test CPU), which is much larger than for conventional transcoding. This is mostly attributable to repacking the “decode stream out” to PAK objects. The code was not optimized and the CPU usage may be reduced at least couple of times by performance tuning.

The higher memory usage of the fast transrater is due to a larger system frame rate (i.e., the gross number of frames transcoded in one second). The per-frame memory usage for the fast transrater is slightly higher due to more data structures used. The fast transrater uses several additional buffers in memory:

- A DSO buffer, which amounts to 0.5 MByte per frame and includes MVs from decoder
- A PAK object buffer, which also has a size of 0.5 MByte per frame
- A PAK MVs buffer (1 MByte per frame)

In total, the fast transrater uses an extra 2 Mbytes per frame (4 Mbytes if accounting for both read and write operations).

All data was gathered by 5 seconds of Intel VTune Amplifier sampling in the middle of the transcoding. The measurement error were not estimated due to the lack of time to gather a necessary amount of data. Only a couple of runs were done, but it seems that the confidence interval should be in 10% range for CPU/GPU usage and memory bandwidth. Therefore, the CPU/GPU usage and memory bandwidth for 8 and 14 channels for fast transrating should be considered the same.

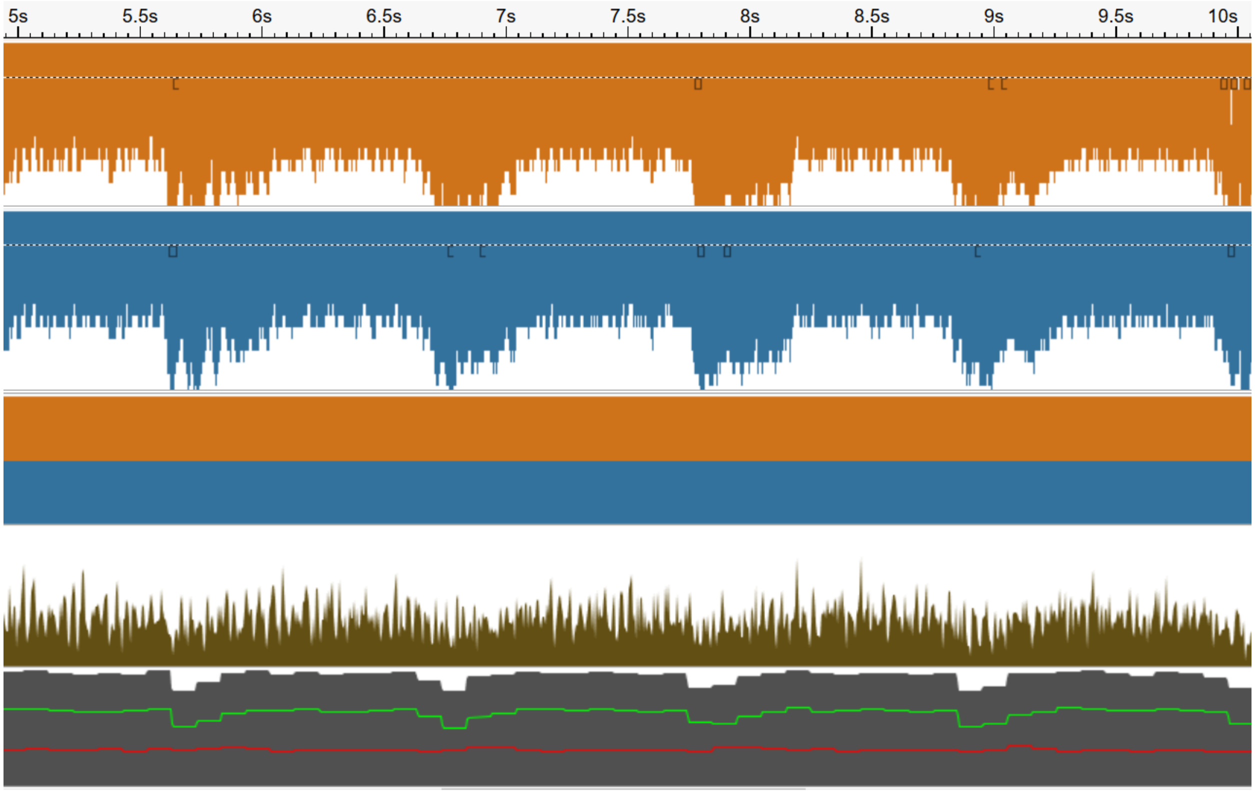


Figure 22. Intel VTune Amplifier view on a system performing a 13-channel fast transrating. From the top, the first (orange) graph is the VDBOX1 queue, the second (blue) is the VDBOX2 queue, the third and the fourth (orange/blue) are VDBOXes utilization, the fifth (brown) is CPU usage and the bottom graph (gray) is total DRAM bandwidth, with the green line denoting read and the red line denoting write.

Indicator	Conventional transcoding	Fast transrating (top performance)	Fast transrating (channel-aligned)
Number of channels	8	14	8
System frame rate[2]	247 fps	417 fps	415 fps
CPU usage	29%	300%	300%
VDBOX1 usage	59%	**99%**	**94%**
VDBOX2 usage	45%	**99%**	**92%**
EU/render usage	**100%**	0%	0%
Total memory bandwidth read write	5.7 GBps 3.95 GBps 1.15 GBps	11.8 GBps 7.8 GBps 4.0 GBps	11.3 GBps 7.2 GBps 4.1 GBps
Total memory bandwidth per frame	23 MBpf	28 MBpf	27 MBpf

Table 5. Performance indicators for fast transrating.

---

[2] Gross number of frames transcoded in one second, which was calculated as the number of channels multiplied by the number of frames in the transcoded stream and divided by the total duration of transcoding.

3.3 SD PERFORMANCE

3.3.1 Density

The results of our density estimation for SD streams are shown in Figure 23. Six streams were used, with exactly the same content and frame count as their HD counterparts. Each stream was profiled for three different bitrates.

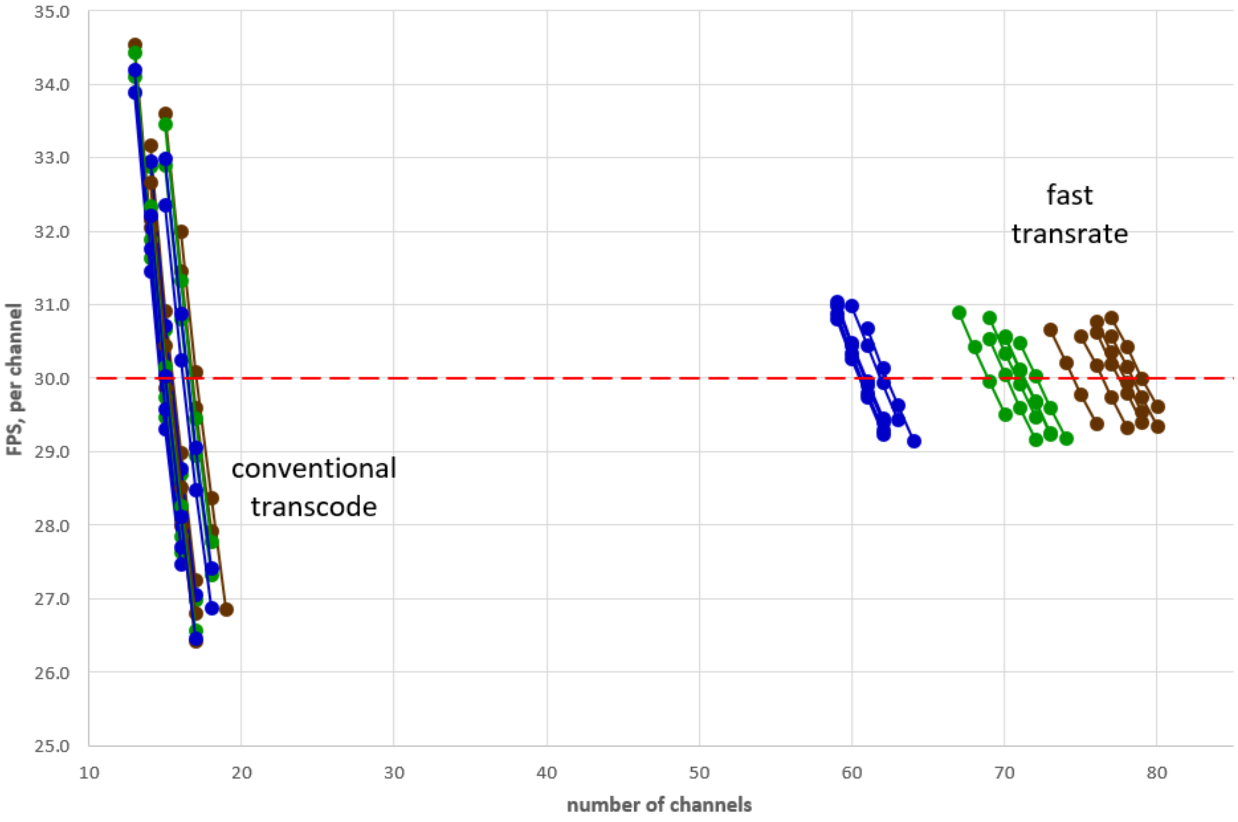


Figure 23. Fast transrater versus conventional transcoder density for six SD streams and three bitrates.

In Figure 23, the blue lines represent 12Mbps to 3Mbps transcoding. The green lines represent 6Mbps to 2Mbps. And the brown lines represent 2Mbps to 1Mbps. The group of lines on the left represents conventional transcoding; the group on the right represents fast transrating.

The average number of channels is 69.7 for the fast transrater and 15.5 for conventional transcoding. Therefore, on average, by using the fast transrater, we can increase the system density by 54 channels (i.e., more than 4.5x).

The data in Table 6 shows that the fast transrating has an almost linear dependency on stream resolution (specifically, the number of pixels in the frame).

Case	Pixels per frame	Number of channels	
		Fast transrate	Conventional transcode
SD 720x480	345600	69.7	15.5

HD 1920x1080	2073600	13.6	7.9
Ratio	6	5.1	1.96

Table 6. Transcoder density dependency on stream resolution.

4 CONCLUSION

Fast transrating based on a flexible encode infrastructure outperforms conventional transcoding in the trans-rate use case if the bitrate difference between source and destination is moderate. For full HD content, we identified four such ranges. Inside these ranges, the average BD rate between both kinds of transcoding is negligible/

source bitrate Mbps	target bitrate Mbps
40	12-24
30	10-20
20	8-12
10	5-8

For these ranges, the fast transrating density is five channels higher—13.6 channels for fast transrating versus 7.9 channels for conventional transcoding, on average.

Fast transrating also has significantly better subjective visual quality on low bitrates for streams with steady global motion in the regions with uniform textures like water or grass.

Conventional transcoding density is limited by EUs (100% usage), while fast transrating is limited by VDBOXes (100% usage as well). In case of fast transrating, EUs are completely idle and can be used for additional processing. The memory bandwidth required for both kinds of transcoding is about the same.

The CPU usage for fast transrater is 5x higher (300% versus 29% out of a maximum of 800% for the tested CPU), but can be significantly reduced by code optimization and/or by disabling minor partitions in the source stream.

The visual quality of fast transrating can be further improved by optimizing bitrate control (using source stream QPs and frame/MB level statistics).

Bitrate ranges suitable for fast transrating can be extended to lower values if additional processing is implemented (i.e., Inter to Skip MB conversion).

5 APPENDIX A. QUALITY EVALUATION DETAILS

The current fast transrater implementation does not change the “stream out” data received from decoder and merely repacks it into a format that is suitable for the FEI PAK component. This imposes significant restrictions on the source bitstream, since its GOP structure and reference lists should be exactly the same as the PAK expects. As a result, scene change detection and weighted prediction were disabled in the x264 encoder used to produce input bitstreams. There is also no support in the current implementation for reference B frames and MB level QP , which were disabled as well.

Input bitstream encoder (x264) command line:

```
./bin/x264-r2901-7d0ff22 --input-res $WIDTH"x"$HEIGHT  --bitrate $BR1
--no-mbtree  -I 32 --no-fast-pskip  --no-dct-decimate --no-scenecut
--ref 4 --bframes 3 --b-pyramid none --b-adapt 0 --weightp 0
--no-weightb --ipratio 1.4  --pbratio 1.3  --merange 24  --me umh
--subme 11 --partitions all --trellis 0 --no-psy --psnr --deblock 0:0
--frames $FRAMES --fps $FPS -o $REF_AVC $IN_YUV
```

Conventional transcoder (sample_multi_transcode) command line:

```
./bin/sample_multi_transcode -i::h264 $REF_AVC -o::h264 $XCODED_STREAM -hw -u
4 -gop_size 32 -num_ref 4 -dist 4 -nobref -async 1 -l 1 -vbr -b $BR2K
```

Fast transrater (`sample_fei`) command line:

```
./bin/sample_fei -i::h264 $REF_AVC -o $XCODED_STREAM -w $WIDTH -h $HEIGHT -n
$FRAMES -f $FPS -vbr -TargetKbps $BR2 -l 1 -NumRefFrame 4 -g 32 -GopRefDist 4
-idr_interval 0 -nobref -pak -streamout /dev/null -num_active_P 4 -
num_active_BL0 3 -num_active_BL1 1
```

6 APPENDIX B. PERFORMANCE EVALUATION DETAILS

6.1 SYSTEM CONFIGURATION

OS

- CentOS* Linux release 7.4.1708
- Kernel: 3.10.0-693.17.1.el7.x86_64
- UMD: 16.8.69021
- KMD: i915 1.6.0 20180222-16.8-69021-k0e37c29
- MSS version: Intel® Media Server Studio Essentials 2018R1

CPU

- Intel® Core™ i7-6770HQ processor @ 1.80GHz
- CPU(s): 4
- Thread(s) per core: 2
- Stepping: 3
- L1d cache: 128 KB
- L1i cache: 128 KB
- L2 cache: 1 MB
- L3 cache: 6 MB
- L4 cache: 128 MB

GPU

- Intel® Iris™ Graphics 580
- eDRAM: 128 MB
- EU Count: 72
- Max Core Frequency: 950 MHz

Memory

- Number of channels: 1
- Size: 16384 MB
- Type: DDR4
- Speed: 2133 MHz

Two GPU slices were enabled, CPU, GPU, and uncore frequencies were fixed, performance mode was enabled. To do so next commands were executed:

```
echo 700 > /sys/class/drm/card0/gt_min_freq_mhz
echo 700 > /sys/class/drm/card0/gt_max_freq_mhz
cpupower frequency-set -d 1800000 -u 1800000 -g performance
wrmsr 0x620 0x1212
echo -e "[KEY]\n\t0x00000001\n\tUFKEY_INTERNAL\\LibVa\n\t[VALUE]\n\t\tDynamic
Slice Shutdown\n\t\t4\n\t\t2" > /etc/igfx_user_feature.txt
```

The SD streams for performance evaluation were obtained via downsampling using sample_vpp as follows:

```
./sample_vpp -sw 1920 -sh 1080 -scc i420 -dw 720 -dh 480 -dcc i420 -i
hd.yuv -o sd.yuv
```

6.2 HRD CONFORMANCE IMPACT ON DENSITY

There was no way to disable HRD conformance for the conventional transcoder during estimations and also no way to enable it for the fast transrater. To satisfy HRD conformance, the conventional encoder may sometimes reencode frames and this in turn can bias measurement to lower densities for conventional transcoding. To make comparisons fair, the transcoded streams were analyzed for possible signs of reencoding via evaluating HRD buffer fullness. Figure 24 shows an example of the HRD buffer fullness. We can see that most of the time, the buffer fullness is well away from the top or the bottom of the buffer, so the probability of reencoding is very small.

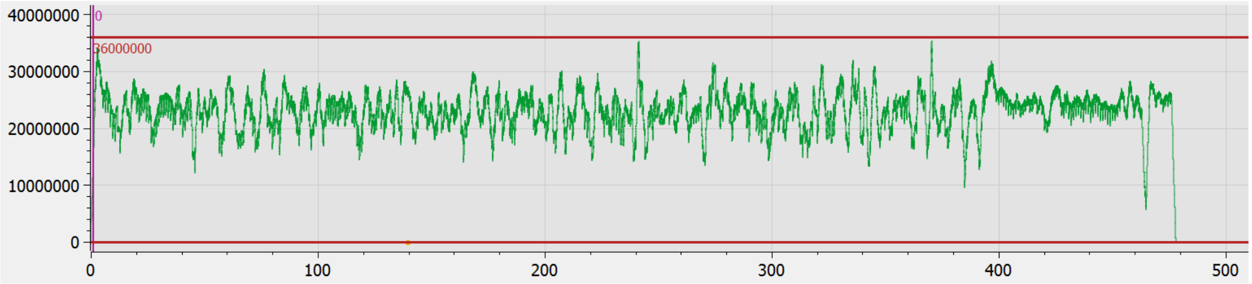


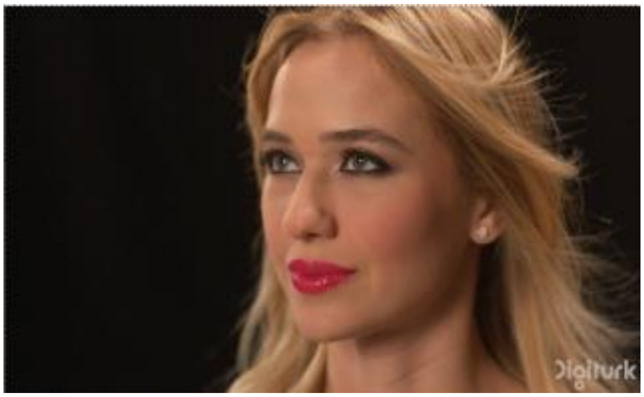
Figure 24. Example of HRD buffer fullness for the "big_buck_bunny" stream, conventional transcoding 20Mbps->12Mbps, 14315 frames. Vertical axis is buffer fullness in bits, horizontal axis is time in seconds.

To double-check this conclusion, we encoded the same stream in both VBR and constant QP modes (see Table 7). The VBR performance is lower, but the difference is small and was considered acceptable for the purposes of this paper.

Bitrate control	File size	Number of channels	Fps per channel
VBR (12 Mbps)	689.6 Mbytes	8	30.92
CQP (QP = 20)	552.4 Mbytes	8	32.33

Table 7. CQP vs VBR performance.

7 APPENDIX C. STREAM INFORMATION


Name: Beauty

Source: http://ultravideo.cs.tut.fi/#testsequences, frame conversion from 120 fps to 60 fps

Copyright: This sequence and all intellectual property rights therein remain the property of Digiturk.

Name: Bosphorus

Source: http://ultravideo.cs.tut.fi/#testsequences, frame conversion from 120 fps to 60 fps

Copyright: This sequence and all intellectual property rights therein remain the property of Digiturk.

Name: HoneyBee

Source: http://ultravideo.cs.tut.fi/#testsequences, frame conversion from 120 fps to 60 fps

Copyright: This sequence and all intellectual property rights therein remain the property of Digiturk.

Name: Jockey

Source: http://ultravideo.cs.tut.fi/#testsequences, frame conversion from 120 fps to 60 fps

Copyright: This sequence and all intellectual property rights therein remain the property of Digiturk.

Name: ReadySteadyGo

Source: http://ultravideo.cs.tut.fi/#testsequences, frame conversion from 120 fps to 60 fps

Copyright: This sequence and all intellectual property rights therein remain the property of Digiturk.


Name: YachtRide
Source: http://ultravideo.cs.tut.fi/#testsequences, frame conversion from 120 fps to 60 fps
Copyright: This sequence and all intellectual property rights therein remain the property of Digiturk.


Name: Blue sky
Source: https://xiph-media.net/video/derf/
Copyright: No Copyright


Name: Pedestrian Area
Source: https://xiph-media.net/video/derf/
Copyright: No Copyright


Name: rush_hour
Source: https://xiph-media.net/video/derf/
Copyright: No Copyright


Name: Station
Source: https://xiph-media.net/video/derf/
Copyright: No Copyright


Name: Sunflower
Source: https://xiph-media.net/video/derf/
Copyright: No Copyright


Name: Tractor
Source: https://xiph-media.net/video/derf/
Copyright: No Copyright


Name: Riverbed
Source: https://xiph-media.net/video/derf/
Copyright: No Copyright


Name: CrowdRun
Source: https://xiph-media.net/video/derf/
Copyright: The sequences and all intellectual property rights therein remain the property of Sveriges Television AB (SVT), Sweden. These sequences may only be used for the purpose of developing, testing and presenting technology standards. SVT makes no warranties with respect to the materials and expressly disclaim any warranties regarding their fitness for any purpose.


Name: DucksTakeOff
Source: https://xiph-media.net/video/derf/
Copyright: The sequences and all intellectual property rights therein remain the property of Sveriges Television AB (SVT), Sweden. These sequences may only be used for the purpose of developing,

testing and presenting technology standards. SVT makes no warranties with respect to the materials and expressly disclaim any warranties regarding their fitness for any purpose.

Name: IntoTree

Source: https://xiph-media.net/video/derf/

Copyright: The sequences and all intellectual property rights therein remain the property of Sveriges Television AB (SVT), Sweden. These sequences may only be used for the purpose of developing, testing and presenting technology standards. SVT makes no warranties with respect to the materials and expressly disclaim any warranties regarding their fitness for any purpose.

Name: Kimono

Source: Sequence is part of JCT-VC "Common test conditions and software reference configurations".

Copyright: Individuals and organizations extracting sequence from this archive agree that the sequences and all intellectual property rights therein remain the property of Tokyo Institute of Technology, Nakajima Laboratory. This material may only be used for the purpose of developing, testing and promulgating technology standards. The owner makes no warranties with respect to the material and expressly disclaims any warranties regarding its fitness for any purpose.

Name: OldTownCross

Source: https://xiph-media.net/video/derf/

Copyright: The sequences and all intellectual property rights therein remain the property of Sveriges Television AB (SVT), Sweden. These sequences may only be used for the purpose of developing, testing and presenting technology standards. SVT makes no warranties with respect to the materials and expressly disclaim any warranties regarding their fitness for any purpose.

Name: park_joy

Source: https://xiph-media.net/video/derf/

Copyright: The sequences and all intellectual property rights therein remain the property of Sveriges Television AB (SVT), Sweden. These sequences may only be used for the purpose of developing, testing and presenting technology standards. SVT makes no warranties with respect to the materials and expressly disclaim any warranties regarding their fitness for any purpose.

Name: park_scene

Source: Sequence is part of JCT-VC "Common test conditions and software reference configurations".

Copyright: Individuals and organizations extracting sequence from this archive agree that the sequences and all intellectual property rights therein remain the property of Tokyo Institute of Technology, Nakajima Laboratory. This material may only be used for the purpose of developing,

testing and promulgating technology standards. The owner makes no warranties with respect to the material and expressly disclaims any warranties regarding its fitness for any purpose.


Name: big_buck_bunny
Source: http://www.bigbuckbunny.org/
Copyright: (c) copyright 2008, Blender Foundation / www.bigbuckbunny.org


Name: elephants_dream
Source: http://www.elephantsdream.org/

Copyright: (c) copyright 2006, Blender Foundation / Netherlands Media Art Institute / www.elephantsdream.org


Name: tears_of_steel
Source: https://mango.blender.org/
Copyright: (CC) Blender Foundation | mango.blender.org

LEGAL DISCLAIMER

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS OTHERWISE AGREED IN WRITING BY INTEL, THE INTEL PRODUCTS ARE NOT DESIGNED NOR INTENDED FOR ANY APPLICATION IN WHICH THE FAILURE OF THE INTEL PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or by visiting **Intel's Web Site**.

MPEG is an international standard for video compression/decompression promoted by ISO. Implementations of MPEG CODECs, or MPEG enabled platforms may require licenses from various entities, including Intel Corporation.

Intel and the Intel logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

*Other names and brands may be claimed as the property of others.

Copyright © 2018, Intel Corporation. All Rights reserved.

The benchmark results reported in this paper may need to be revised as additional testing is conducted. The results depend on the specific platform configurations and workloads utilized in the testing, and may not be applicable to any particular user's components, computer system or workloads. The results are not necessarily representative of other benchmarks and other benchmark results may show greater or lesser impact from mitigations.

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit www.intel.com/benchmarks.

OPTIMIZATION NOTICE

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel.

Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Notice revision #20110804