Hardware Transcoding Solutions For The Cloud

VES202

Jan Ozer
janozer@gmail.com

Agenda

- What you will learn
- Theory of testing
- H.264
  - NVIDIA
  - Quick Sync
  - X264 medium/veryfast
- HEVC
  - Xilinx – Field Programmable Gate Array-based codec (FPGA)
    - Formerly technology from NGCodec
  - Intel SVT-HEVC (not really hardware but topical)
  - X265 medium/veryfast

Results from This Study

http://bit.ly/hw_transcode

What You Will Learn

Technology-Specific

- H264
  - Hardware transcoders – NVIDIA and Intel Quick Sync
  - Software - x264 medium/very fast presets
- HEVC
  - NGCodec/Xilinx FPGA transcoding
  - Intel software-only SVT-HEVC
  - X265 medium and very fast

Methodology

- Considerations to incorporate when comparing transcoding technologies
  - Hourly cost
  - Quality (objective/subjective)
  - Identifying transient quality issues
  - Stream consistency
- How to apply objective quality metrics
- Inexpensive source for subjective evaluations
- How objective and subjective results can vary

Overview – Why We Tested

1. Cloud transcoding is the optimal workflow for many live producers
2. There are two options; software or hardware
    a. Software requires an expensive cloud computer with lots of CPUs
    b. Hardware (GPU, FPGA) requires lower CPU but may cost more
3. So, how do CPU-only and hardware systems compare?
    a. Quality-wise
    b. Cost-wise
4. The answers?
    a. Quality-wise: Hardware stacks up pretty well
    b. Cost-wise: It’s complicated; I couldn’t find a single machine that could perform all the hardware and software encodes

Theory of Testing

1. Derive most practical encoding configuration
2. Test capacity using encoding ladder
    a. Hardware - no dropped frames
    b. Software - 55 fps or higher
3. Test quality at those settings
    1. Rate distortion curves (VMAF/PSRN)
    2. BD-Rate functions (VMAF/PSNR)
    3. Subjective comparisions via Subjectify

Tuning for Metrics

- H.264
  - No way to tune with Intel Quick Sync so didn’t tune at all
- HEVC
  - Tuned for objective comparisons
  - Didn’t tune for subjective comparisons

NVIDIA H.264

- Instance
- Settings
- Capacity
- Quality

Instance - g3.4xlarge

Name	GPUs	vCPU	Memory (GiB)	GPU Memory (GiB)	Price/hr* (Linux)
g3s.xlarge	1	4	30.5	8	$0.75
g3.4xlarge	1	16	122	8	$1.14
g3.8xlarge	2	32	244	16	$2.28
g3.16xlarge	4	64	488	32	$4.56

- Instance selected and configured by engineers at Softvelum, who run the Nimble Streamer cloud transcoder. They have my undying gratitude and appreciation.

Finding the Right Settings

- Best source - Using FFmpeg With NVIDIA GPU HW Acceleration
  - https://developer.nvidia.com/designworks/dl/Using_FFmpeg_with_NVIDIA_GPU_Hardware_Acceleration-pdf (registration required)
- Recommended string:

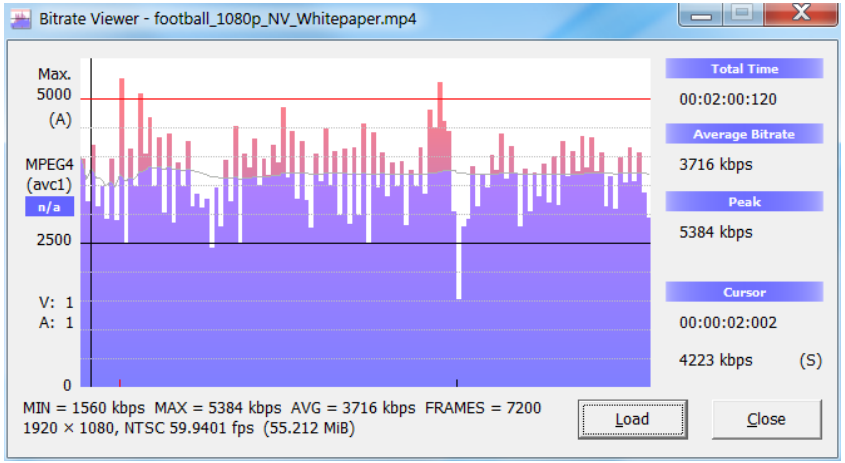
```
ffmpeg -y -vsync 0 -hwaccel cuvid -c:v h264_cuvid -i
input.mp4 -c:a copy -c:v h264_nvenc -preset slow -profile
high -b:v 5M -bufsize 5M -maxrat                250 -bf 2 -
temporal-aq 1 -rc-lo                            b_qfactor
1.1 output.mp4
```

Slow preset could
erformance

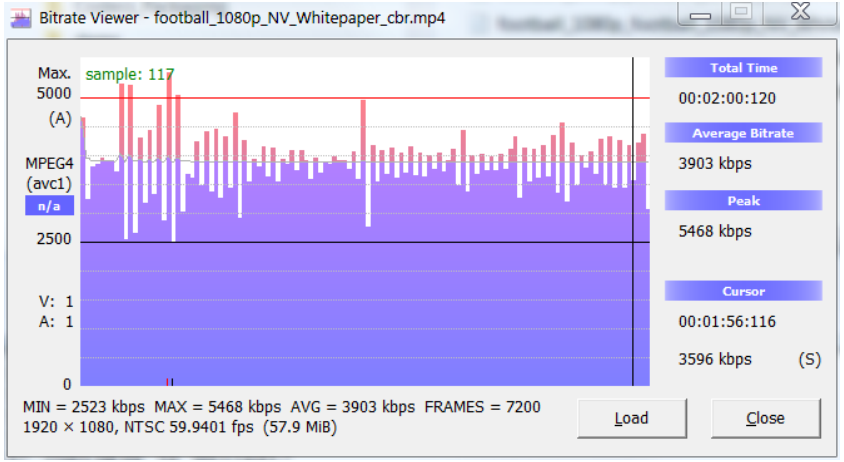
Max rate could
increase variability

- Concerns:
  - Performance - slow preset
  - Data rate fluctuations due to 2 second VBV buffer

Switch to 1 Second VBV Buffer



2 second buffer

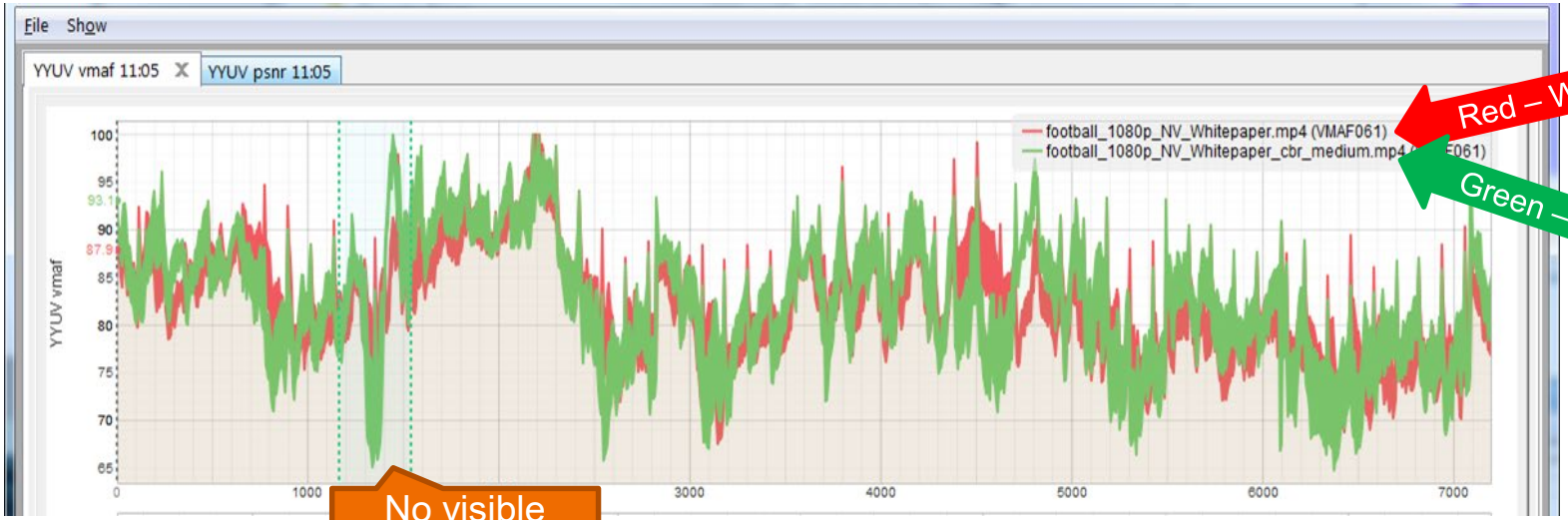


1 second buffer

- 1 second buffer delivered slightly higher overall bitrate and slightly more uniform stream

- Tried Medium preset to optimize capacity
  - VMAF dropped from 82.35 to 82.19

Check for Transient Quality Issues



- VMAF plot in VQMT
- Pretty similar throughout
- Conclusions: no major quality delta with updated settings

Comparisons

White paper

WP – Slow/CBR

WP – CBR/ Medium

	Original White Paper (Slow)	White Paper with CBR (Slow)	White Paper with CBR/Medium
Bitrate	3716	3903	3896
Peak	5384	5468	5123
VMAF	81.82	82.35	82.19
PSNR	33.65	33.83	33.74
CPU%	15%	15%	15%

Lowest peak

Quality higher than white paper

- Very little difference in quality/CPU with Slow or Medium

NVIDIA Encoding String - Final

- Hardware decode to CUVID, then encode

```
ffmpeg  -y -vsync 0 -hwaccel cuvid -c:v h264_cuvid -i input.mp4 -c:v
h264_nvenc -preset medium  -b:v 5M -bufsize 5M -maxrate 5M -qmin 0 -g
120 -bf 2 -temporal-aq 1 -rc-lookahead 20 -i_qfactor 0.75 -b_qfactor 1.1
output.mp4
```

Testing Capacity

- Tested with this encoding ladder
- Kept opening instances and running until frame rate dropped to below 60fps
- Nvidia achieved two 60 fps encodes on G3.4 xlarge

Rez	Data rate
1080p60	6 mbps
1080p30	4 mbps
720p30	2.5 mbps
540p30	1.2 mbps
360p30	.8 mbps

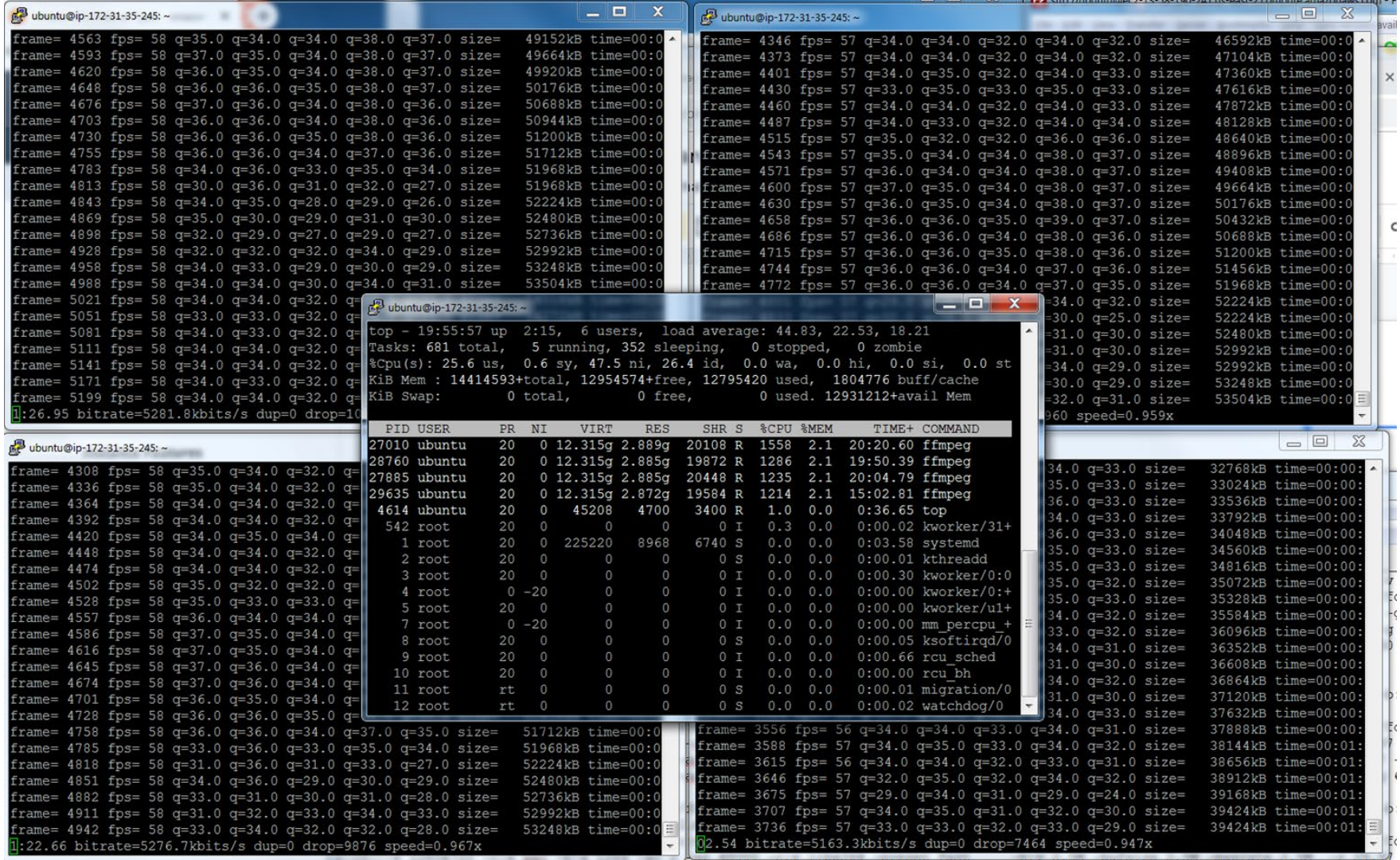
x264 Encodes

- Simple x264 conversion script
  - Tested with Medium and veryfast

```
ffmpeg  -y -re -i input.mp4  -c:v libx264 -preset medium  -b:v 5M -
bufsize 5M -maxrate 5M  -g 120  output.mp4
```

Capacity

- On GPU optimized computer, couldn’t produce a single x264 ladder with any preset
- Compared software performance to a C5.18 xlarge, which cost about the same ($1.25/hour compared to $1.14).
- Achieved 4 simultaneous encodes


ubuntu@ip-172-31-35-245: ~
frame= 4563 fps= 58 q=35.0 q=34.0 q=34.0 q=38.0 q=37.0 size= 49152kB time=00:0
frame= 4593 fps= 58 q=37.0 q=35.0 q=34.0 q=38.0 q=37.0 size= 49664kB time=00:0
frame= 4620 fps= 58 q=36.0 q=35.0 q=34.0 q=38.0 q=37.0 size= 49920kB time=00:0
frame= 4648 fps= 58 q=36.0 q=36.0 q=35.0 q=38.0 q=37.0 size= 50176kB time=00:0
frame= 4676 fps= 58 q=37.0 q=36.0 q=34.0 q=38.0 q=36.0 size= 50688kB time=00:0
frame= 4703 fps= 58 q=36.0 q=36.0 q=34.0 q=38.0 q=36.0 size= 50944kB time=00:0
frame= 4730 fps= 58 q=36.0 q=36.0 q=35.0 q=38.0 q=36.0 size= 51200kB time=00:0
frame= 4755 fps= 58 q=36.0 q=36.0 q=34.0 q=37.0 q=36.0 size= 51712kB time=00:0
frame= 4783 fps= 58 q=34.0 q=36.0 q=33.0 q=35.0 q=34.0 size= 51968kB time=00:0
frame= 4813 fps= 58 q=30.0 q=36.0 q=31.0 q=32.0 q=27.0 size= 51968kB time=00:0
frame= 4843 fps= 58 q=34.0 q=35.0 q=28.0 q=29.0 q=26.0 size= 52224kB time=00:0
frame= 4869 fps= 58 q=35.0 q=30.0 q=29.0 q=31.0 q=30.0 size= 52480kB time=00:0
frame= 4898 fps= 58 q=32.0 q=29.0 q=27.0 q=29.0 q=27.0 size= 52736kB time=00:0
frame= 4928 fps= 58 q=32.0 q=32.0 q=32.0 q=34.0 q=29.0 size= 52992kB time=00:0
frame= 4958 fps= 58 q=34.0 q=33.0 q=29.0 q=30.0 q=29.0 size= 53248kB time=00:0
frame= 4988 fps= 58 q=34.0 q=34.0 q=30.0 q=34.0 q=31.0 size= 53504kB time=00:0
frame= 5021 fps= 58 q=34.0 q=34.0 q=32.0 q=
frame= 5051 fps= 58 q=33.0 q=33.0 q=32.0 q=
frame= 5081 fps= 58 q=34.0 q=33.0 q=32.0 q=
frame= 5111 fps= 58 q=34.0 q=34.0 q=32.0 q=
frame= 5141 fps= 58 q=34.0 q=34.0 q=32.0 q=
frame= 5171 fps= 58 q=34.0 q=33.0 q=32.0 q=
frame= 5199 fps= 58 q=34.0 q=34.0 q=32.0 q=
]:26.95 bitrate=5281.8kbits/s dup=0 drop=10
ubuntu@ip-172-31-35-245: ~
frame= 4346 fps= 57 q=34.0 q=34.0 q=32.0 q=34.0 q=32.0 size= 46592kB time=00:0
frame= 4373 fps= 57 q=34.0 q=34.0 q=32.0 q=34.0 q=32.0 size= 47104kB time=00:0
frame= 4401 fps= 57 q=34.0 q=35.0 q=32.0 q=34.0 q=33.0 size= 47360kB time=00:0
frame= 4430 fps= 57 q=33.0 q=35.0 q=33.0 q=35.0 q=33.0 size= 47616kB time=00:0
frame= 4460 fps= 57 q=34.0 q=34.0 q=32.0 q=34.0 q=33.0 size= 47872kB time=00:0
frame= 4487 fps= 57 q=34.0 q=33.0 q=32.0 q=34.0 q=34.0 size= 48128kB time=00:0
frame= 4515 fps= 57 q=35.0 q=32.0 q=32.0 q=36.0 q=36.0 size= 48640kB time=00:0
frame= 4543 fps= 57 q=35.0 q=34.0 q=34.0 q=38.0 q=37.0 size= 48896kB time=00:0
frame= 4571 fps= 57 q=36.0 q=34.0 q=34.0 q=38.0 q=37.0 size= 49408kB time=00:0
frame= 4600 fps= 57 q=37.0 q=35.0 q=34.0 q=38.0 q=37.0 size= 49664kB time=00:0
frame= 4630 fps= 57 q=36.0 q=35.0 q=34.0 q=38.0 q=37.0 size= 50176kB time=00:0
frame= 4658 fps= 57 q=36.0 q=36.0 q=35.0 q=39.0 q=37.0 size= 50432kB time=00:0
frame= 4686 fps= 57 q=36.0 q=36.0 q=34.0 q=38.0 q=36.0 size= 50688kB time=00:0
frame= 4715 fps= 57 q=36.0 q=36.0 q=35.0 q=38.0 q=36.0 size= 51200kB time=00:0
frame= 4744 fps= 57 q=36.0 q=36.0 q=34.0 q=37.0 q=36.0 size= 51456kB time=00:0
frame= 4772 fps= 57 q=36.0 q=36.0 q=34.0 q=37.0 q=35.0 size= 51968kB time=00:0
960 speed=0.959x
ubuntu@ip-172-31-35-245: ~
top - 19:55:57 up 2:15, 6 users, load average: 44.83, 22.53, 18.21
Tasks: 681 total, 5 running, 352 sleeping, 0 stopped, 0 zombie
%Cpu(s): 25.6 us, 0.6 sy, 47.5 ni, 26.4 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 14414593+total, 12954574+free, 12795420 used, 1804776 buff/cache
KiB Swap: 0 total, 0 free, 0 used. 12931212+avail Mem
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
27010 ubuntu 20 0 12.315g 2.889g 20108 R 1558 2.1 20:20.60 ffmpeg
28760 ubuntu 20 0 12.315g 2.885g 19872 R 1286 2.1 19:50.39 ffmpeg
27885 ubuntu 20 0 12.315g 2.885g 20448 R 1235 2.1 20:04.79 ffmpeg
29635 ubuntu 20 0 12.315g 2.872g 19584 R 1214 2.1 15:02.81 ffmpeg
4614 ubuntu 20 0 45208 4700 3400 R 1.0 0.0 0:36.65 top
542 root 20 0 0 0 0 I 0.3 0.0 0:00.02 kworker/31+
1 root 20 0 225220 8968 6740 S 0.0 0.0 0:03.58 systemd
2 root 20 0 0 0 0 S 0.0 0.0 0:00.01 kthreadd
3 root 20 0 0 0 0 I 0.0 0.0 0:00.30 kworker/0:0
4 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/0:+
5 root 20 0 0 0 0 I 0.0 0.0 0:00.00 kworker/u1+
7 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 mm_percpu_+
8 root 20 0 0 0 0 S 0.0 0.0 0:00.05 ksoftirqd/0
9 root 20 0 0 0 0 I 0.0 0.0 0:00.66 rcu_sched
10 root 20 0 0 0 0 I 0.0 0.0 0:00.00 rcu_bh
11 root rt 0 0 0 0 S 0.0 0.0 0:00.01 migration/0
12 root rt 0 0 0 0 S 0.0 0.0 0:00.02 watchdog/0
ubuntu@ip-172-31-35-245: ~
frame= 4308 fps= 58 q=35.0 q=34.0 q=32.0 q=
frame= 4336 fps= 58 q=35.0 q=34.0 q=32.0 q=
frame= 4364 fps= 58 q=34.0 q=34.0 q=32.0 q=
frame= 4392 fps= 58 q=34.0 q=34.0 q=32.0 q=
frame= 4420 fps= 58 q=34.0 q=35.0 q=34.0 q=
frame= 4448 fps= 58 q=34.0 q=34.0 q=32.0 q=
frame= 4474 fps= 58 q=34.0 q=34.0 q=32.0 q=
frame= 4502 fps= 58 q=35.0 q=32.0 q=32.0 q=
frame= 4528 fps= 58 q=35.0 q=33.0 q=33.0 q=
frame= 4557 fps= 58 q=36.0 q=34.0 q=34.0 q=
frame= 4586 fps= 58 q=37.0 q=35.0 q=34.0 q=
frame= 4616 fps= 58 q=37.0 q=35.0 q=34.0 q=
frame= 4645 fps= 58 q=37.0 q=36.0 q=34.0 q=
frame= 4674 fps= 58 q=37.0 q=36.0 q=34.0 q=
frame= 4701 fps= 58 q=36.0 q=35.0 q=34.0 q=
frame= 4728 fps= 58 q=36.0 q=36.0 q=35.0 q=
frame= 4758 fps= 58 q=36.0 q=36.0 q=34.0 q=37.0 q=35.0 size= 51712kB time=00:0
frame= 4785 fps= 58 q=33.0 q=36.0 q=33.0 q=35.0 q=34.0 size= 51968kB time=00:0
frame= 4818 fps= 58 q=31.0 q=36.0 q=31.0 q=33.0 q=27.0 size= 52224kB time=00:0
frame= 4851 fps= 58 q=34.0 q=36.0 q=29.0 q=30.0 q=29.0 size= 52480kB time=00:0
frame= 4882 fps= 58 q=33.0 q=31.0 q=30.0 q=31.0 q=28.0 size= 52736kB time=00:0
frame= 4911 fps= 58 q=31.0 q=32.0 q=33.0 q=34.0 q=33.0 size= 52992kB time=00:0
frame= 4942 fps= 58 q=33.0 q=34.0 q=32.0 q=32.0 q=28.0 size= 53248kB time=00:0
1:22.66 bitrate=5276.7kbits/s dup=0 drop=9876 speed=0.967x
frame= 3556 fps= 56 q=34.0 q=34.0 q=33.0 q=34.0 q=31.0 size= 37888kB time=00:0
frame= 3588 fps= 57 q=34.0 q=35.0 q=33.0 q=34.0 q=32.0 size= 38144kB time=00:01:
frame= 3615 fps= 56 q=34.0 q=34.0 q=32.0 q=33.0 q=31.0 size= 38656kB time=00:01:
frame= 3646 fps= 57 q=32.0 q=35.0 q=32.0 q=34.0 q=32.0 size= 38912kB time=00:01:
frame= 3675 fps= 57 q=29.0 q=34.0 q=31.0 q=29.0 q=24.0 size= 39168kB time=00:01:
frame= 3707 fps= 57 q=34.0 q=35.0 q=31.0 q=32.0 q=30.0 size= 39424kB time=00:01:
frame= 3736 fps= 57 q=33.0 q=33.0 q=32.0 q=33.0 q=29.0 size= 39424kB time=00:01:
02.54 bitrate=5163.3kbits/s dup=0 drop=7464 speed=0.947x

Capacity

- Four encodes compared to 2 with NVIDIA, so about 1/2 the cost, though plenty of dropped frames
- Much higher-performance NVIDIA hardware is now available, so you'll have to perform your own cost analysis

Intel Quick Sync Encoding

- System
- Command line
- Preset/throughput/cost

Intel Quick Sync Encoding

- System:
  - Single-socket Intel Xeon CPU E3-1585L v5 @ 3.00 GHz
  - Integrated Intel Iris Pro Graphics
  - System sourced at PhoenixNAP for $250/month
  - Divided by 720 (30*24) = $0.35/hour

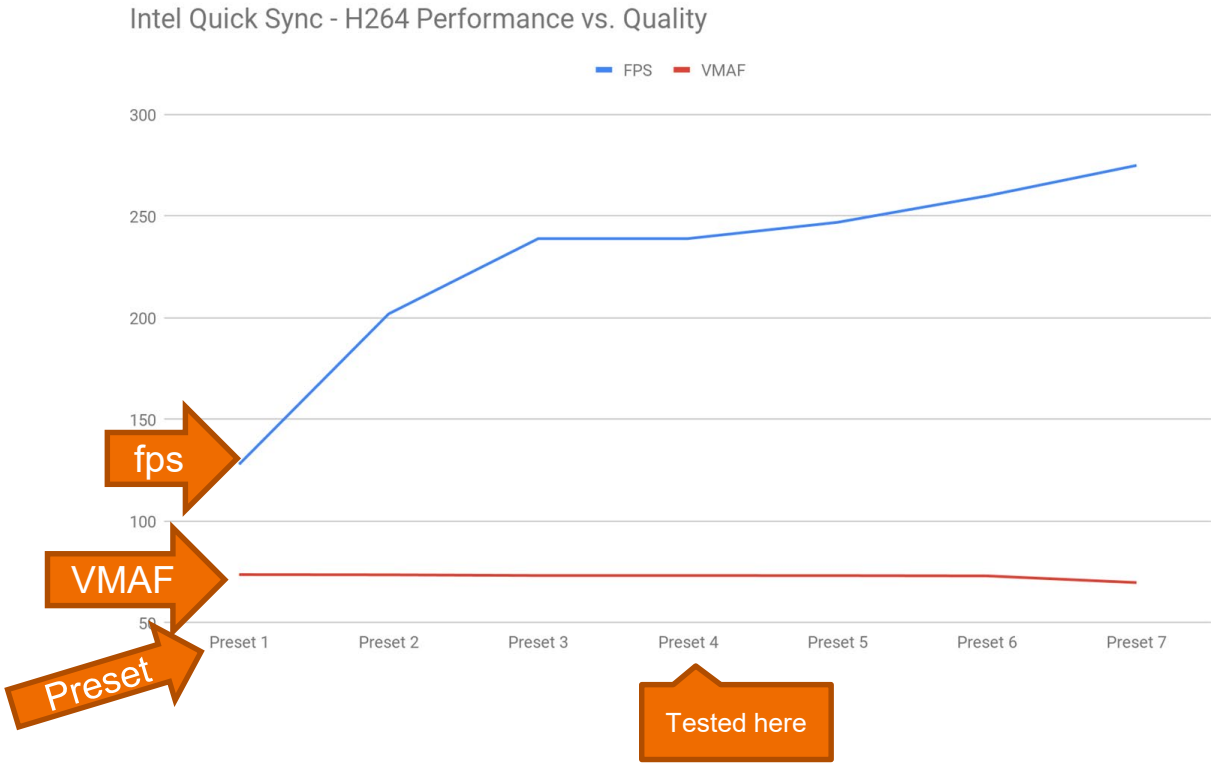
FFmpeg Script (Intel Provided)

```
ffmpeg -re -hwaccel qsv -c:v h264_qsv -y -i input.mp4 -filter_scale_threads 4
-c:v h264_qsv -vf hwupload=extra_hw_frames=64,format=qsv -preset 4 -b:v 5M
-maxrate 5M -bufsize 5M -g 120 -idr_interval 2 -async_depth 5 -look_ahead 1
-look_ahead_depth 30 output.mp4
```

Which Preset ? - Performance vs. Quality

	FPS	VMAF
Preset 1	128	73.75
Preset 2	202	73.64
Preset 3	239	73.29
Preset 4	239	73.29
Preset 5	247	73.25
Preset 6	260	73.11
Preset 7	275	69.82

- Tested at preset 4 (per Intel)
- Delivered single ladder
- Cost ~ $0.35/hour



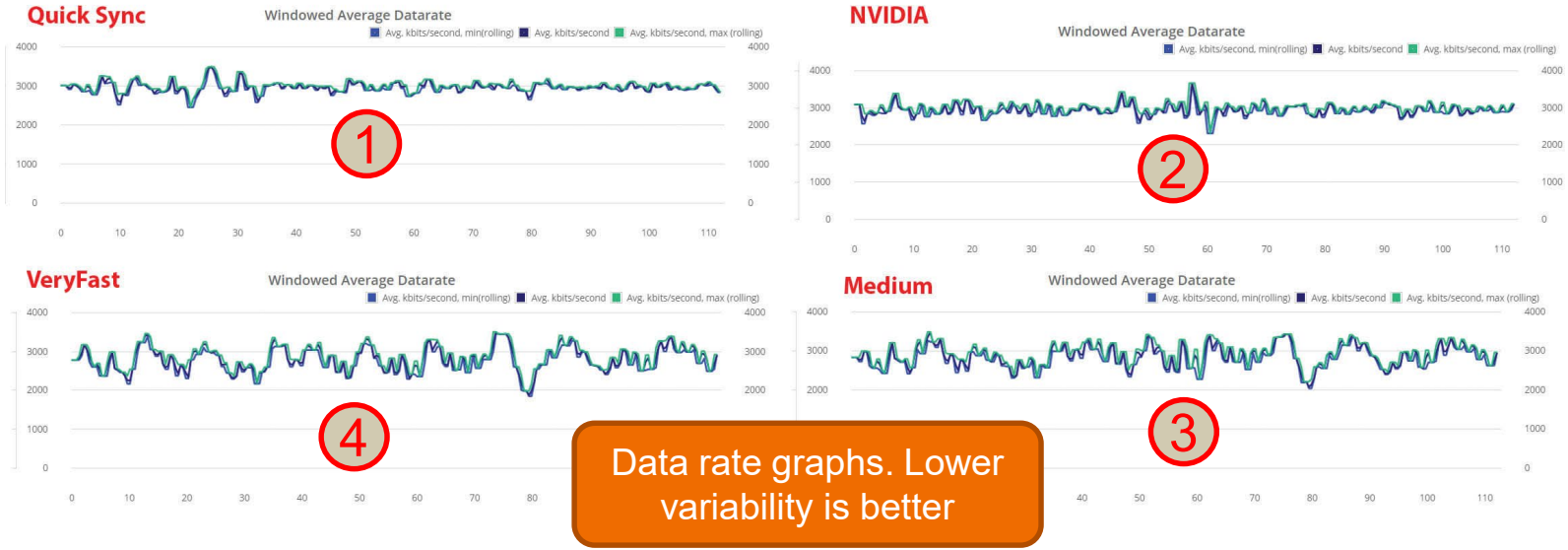
On Tested Computer

- 1 encoding ladder with Quick Sync at preset 4
  - Using preset 7 did not deliver 2 full ladders
- No ladders with x264, even using veryfast preset
- Obviously could get higher performance with other systems
- Had hoped to use exclusively AWS computers to get pricing, but went with Intel supplied computers for simplicity

Data Rate Consistency

- Important for very large streaming sites (like Twitch)
  - If working with fixed pipes at close to maximum capacity, data rate spikes can interrupt the stream
  - Stats/graphs shown generated by Hybrik cloud encoding/analysis platform
  - Can get visualizations from other tools like Bitrate viewer (H.264 only), Telestream Switch, and Zond 265

Data Rate Consistency (3 Mbps Football File)



Data rate graphs. Lower variability is better

		Data Rate	Standard Deviation	Max Data Rate
1	Intel Quick Sync	2969	139	3486
2	NVIDIA	2965	160	3669
3	x264 medium	2885	295	3497
4	x264 veryfast	2818	327	3514

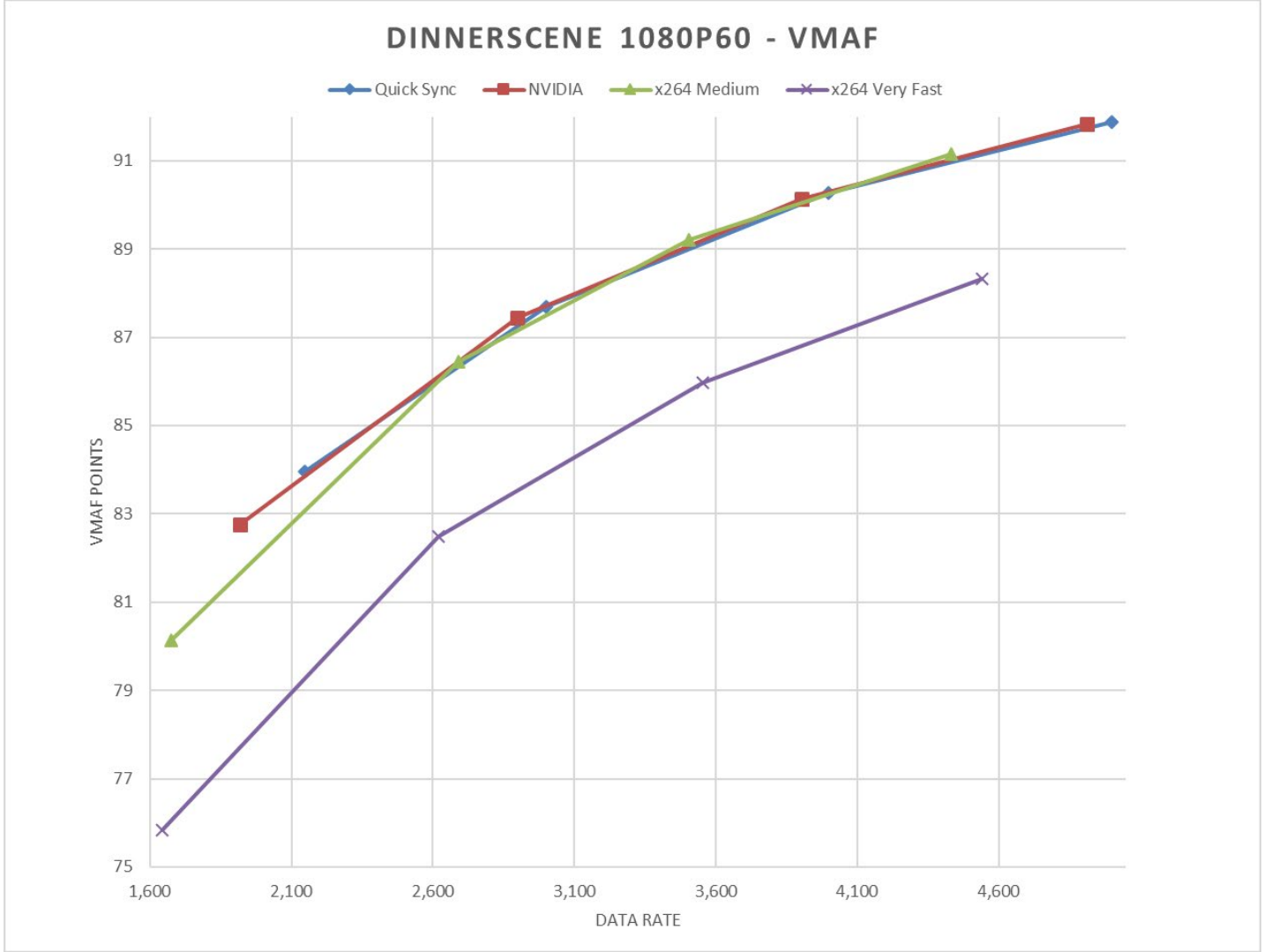
Lowest Flucuation

Lowest Max Data Rate

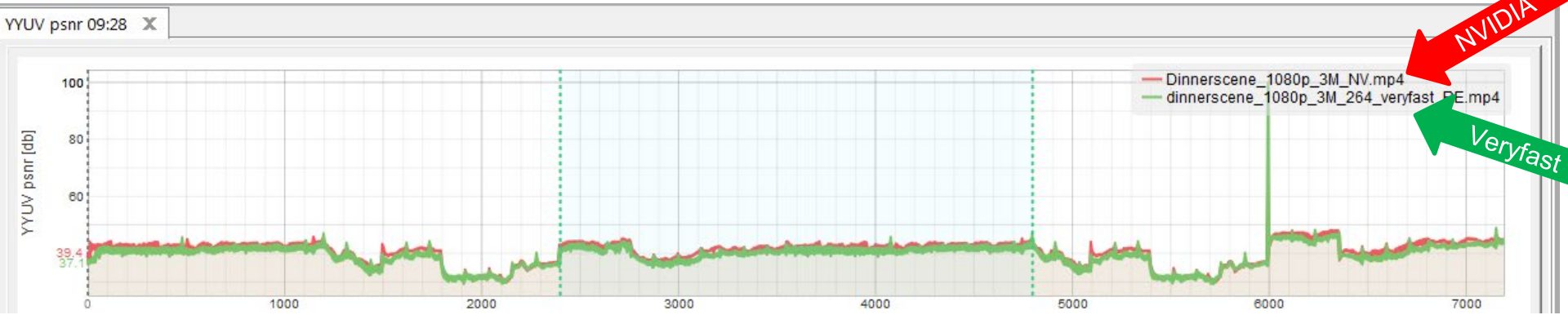
H.264 Quality Results

- Four videos
  - Netflix Dinner Scene
  - Harmonic football
  - GTAV
  - Netflix Meridian
  - All 1080p60
- Tested at 2-5 Mbps

- Four tested codecs
  - NVIDIA NVENC at Medium
  - Intel Quick Sync at Preset 4
  - x264 at Medium and Veryfast


DINNERSCENE 1080P60 - VMAF
Quick Sync
NVIDIA
x264 Medium
x264 Very Fast
VMAF POINTS
91
89
87
85
83
81
79
77
75
1,600
2,100
2,600
3,100
3,600
4,100
4,600
DATA RATE

Actual Visible Differences



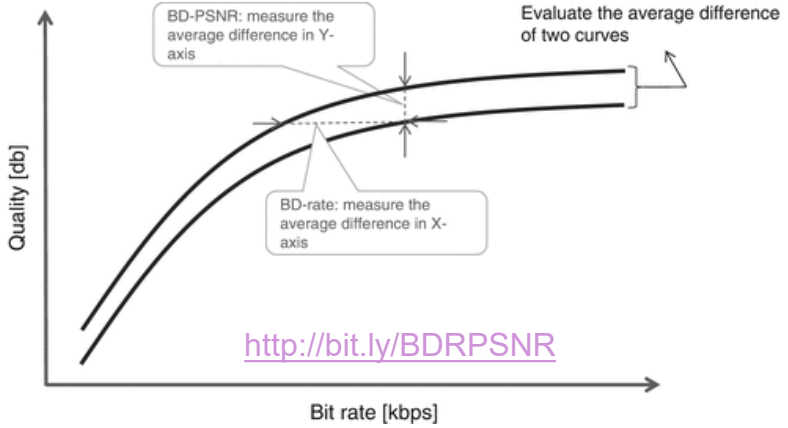
- No major deltas in graph; typically means no major quality deltas
- No significant qualitative differences

Then Compute Bjontegaard Functions (BD-Rate)

- Quantifies differences between two curves
  - BD-Rate – data rate saving for the **same quality**
  - BD-PSRN – quality disparity for same **bitrate**
    - Can use with any metric (not just PSNR)

- Following stats generated from Excel plugin available here (http://bit.ly/BD_functions - free)

- Encoding procedure and plug-in documented and explained in course, Computing and Using Video Quality Metrics: A Course for Encoding Professionals (http://bit.ly/SLC_VM - $99)



http://bit.ly/BDRPSNR

http://bit.ly/SLC_VM

BD-Rate Comparisons

- Generated from Excel plugin available here (http://bit.ly/BD_functions - free)

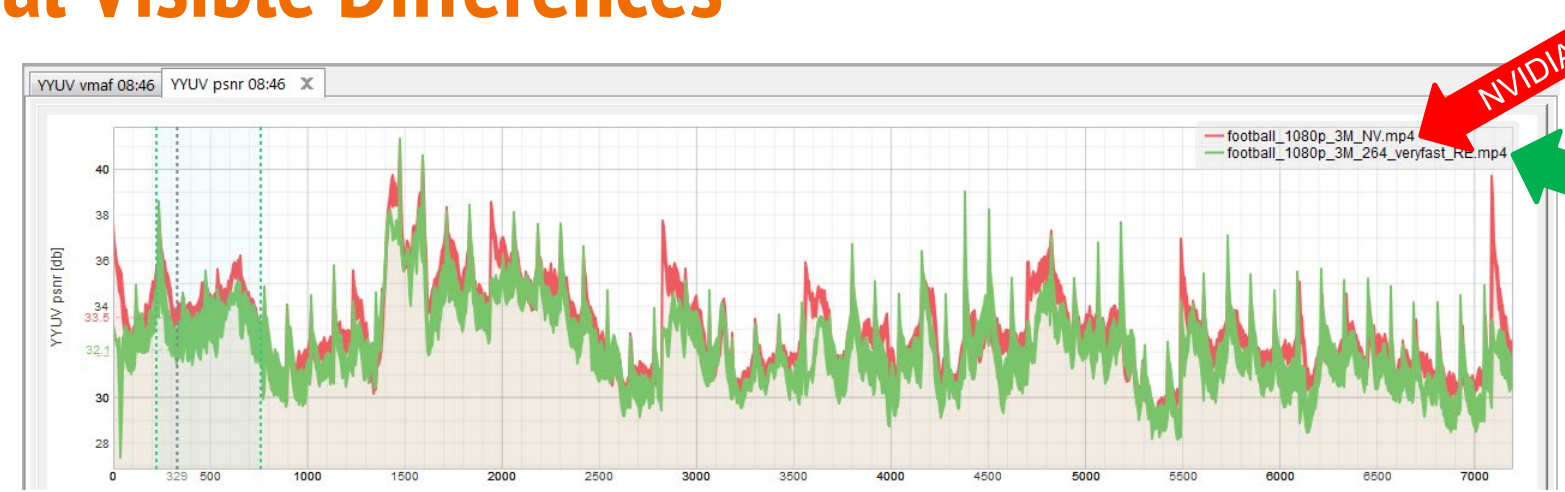
Dinner Scene - BD-Rate Computations

VMAF	Quick Sync	NVIDIA	x264 Medium	x264 Very Fast
Quick Sync	X	1.04 (2)	-0.48	-28.24
NVIDIA	(1) -1.03	X	-2.00	-28.95
x264 Medium	0.49	2.04	X	(3) -25.82
x264 Very Fast	(4) 39.36	40.74	34.80	X

PSNR	Quick Sync	NVIDIA	x264 Medium	x264 Very Fast
Quick Sync	X	-1.16	(2) 4.78	-9.24
NVIDIA	1.17	X	5.83	(3) -8.19
x264 Medium	(1) -4.57	-5.51	X	-13.09
x264 Very Fast	(4) 10.18	8.92	15.06	X


FOOTBALL 1080P60 - VMAF
Quick Sync
NVIDIA
x264 Medium
x264 Very Fast
87
82
77
72
67
62
57
52
VMAF POINTS
1850
2350
2850
3350
3850
4350
4850
DATA RATE

Actual Visible Differences



- No significant transient issues
- Quality differences not that significant

Sample Differential- Source

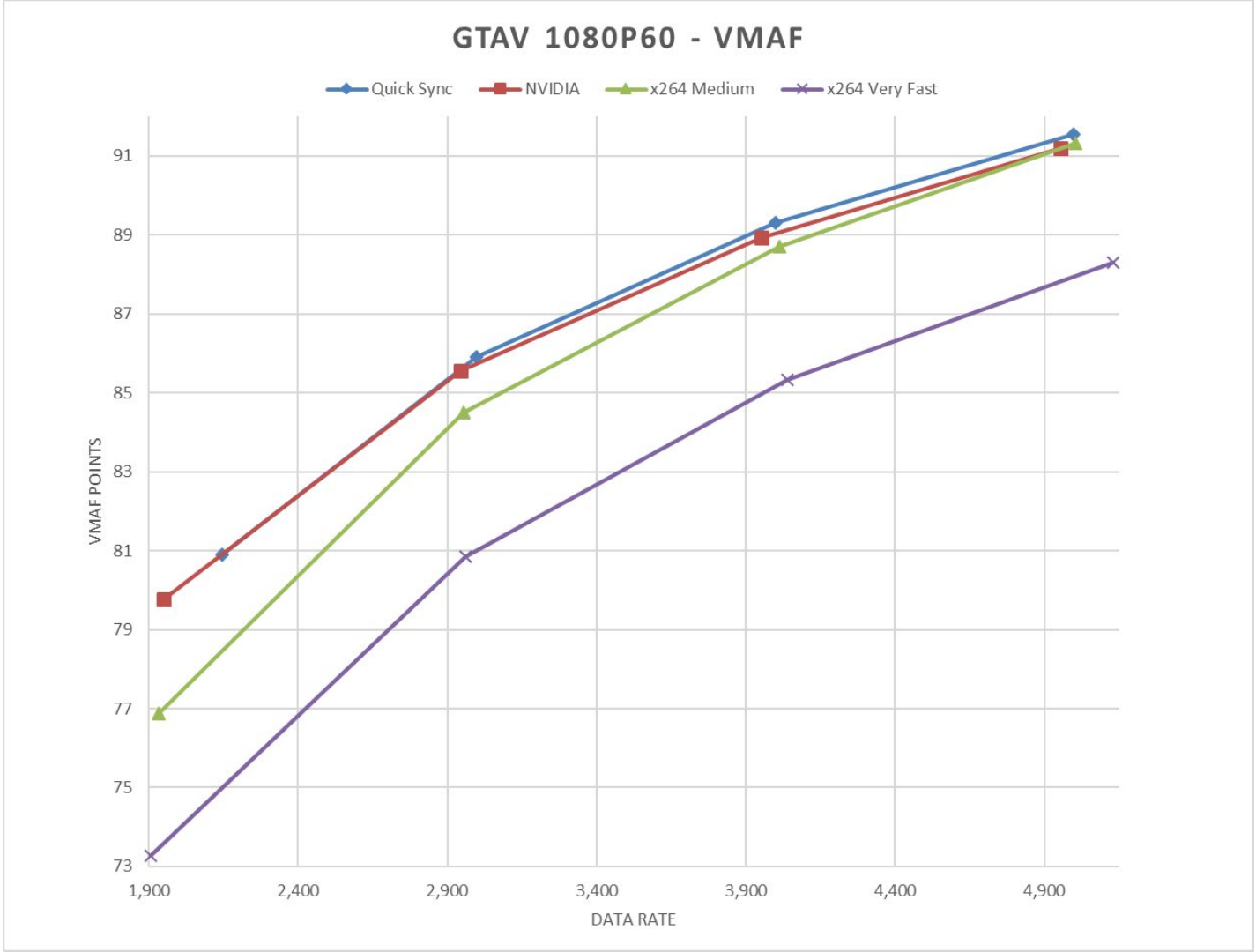

NVIDIA
00:00:12:57
harmonic


Very Fast
00:00:12:57
harmonic

Football - BD-Rate Computations

VMAF	Quick Sync	NVIDIA	x264 Medium	x264 Very Fast
Quick Sync	X	6.75	3.25	(3) -16.94
NVIDIA	(1) -6.33	X	-3.34	-22.09
x264 Medium	-3.14	3.46 (2)	X	-19.04
x264 Very Fast	(4) 20.40	28.35	23.52	X

PSNR	Quick Sync	NVIDIA	x264 Medium	x264 Very Fast
Quick Sync	X	5.07	4.09	(3) -7.71
NVIDIA	(1) -4.82	X	-0.95	-12.48
x264 Medium	-3.93	0.96 (2)	X	-10.81
x264 Very Fast	(4) 8.35	14.26	12.12	X

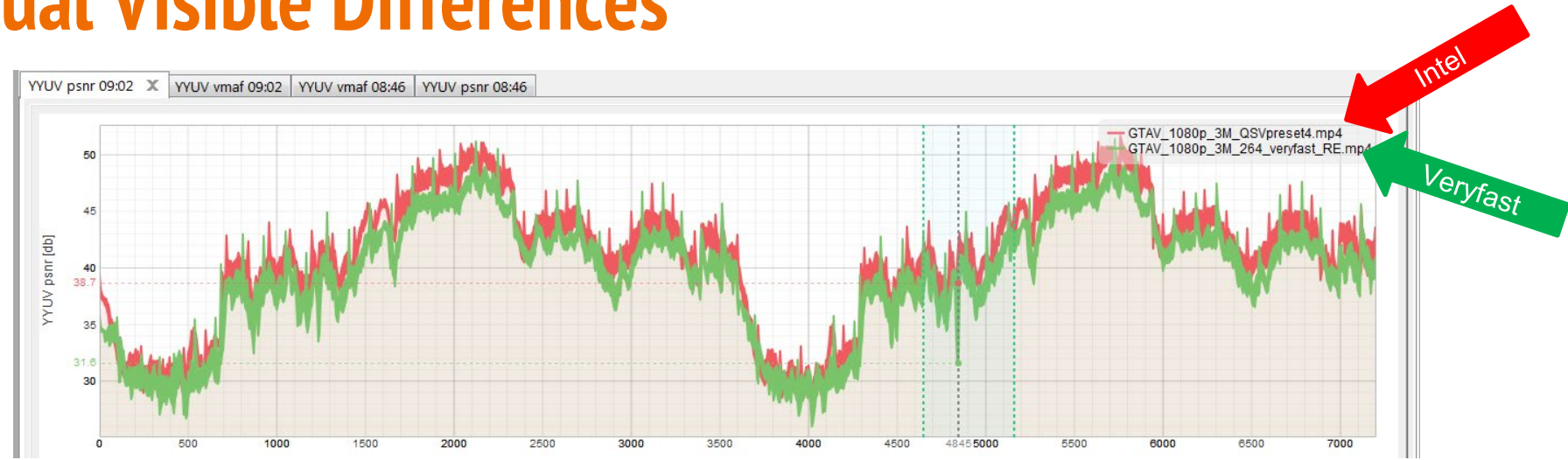

GTAV 1080P60 - VMAF
Quick Sync
NVIDIA
x264 Medium
x264 Very Fast
VMAF POINTS
91
89
87
85
83
81
79
77
75
73
1,900
2,400
2,900
3,400
3,900
4,400
4,900
DATA RATE

GTAV - BD-Rate Computations

VMAF	Quick Sync	NVIDIA	x264 Medium	x264 Very Fast
Quick Sync	1 X	-0.88	-7.17	-28.70
NVIDIA	0.89 2	X	-6.96	-28.64
x264 Medium	7.72	7.49	X	3 -21.01
x264 Very Fast	4 40.25	40.13	26.60	X

PSNR	Quick Sync	NVIDIA	x264 Medium	x264 Very Fast
Quick Sync	1 X	-1.56	-3.72	-18.98
NVIDIA	1.58 2	X	-2.33	-17.88
x264 Medium	3.86	2.39	X	3 -15.74
x264 Very Fast	4 23.42	21.77	18.68	X

Actual Visible Differences

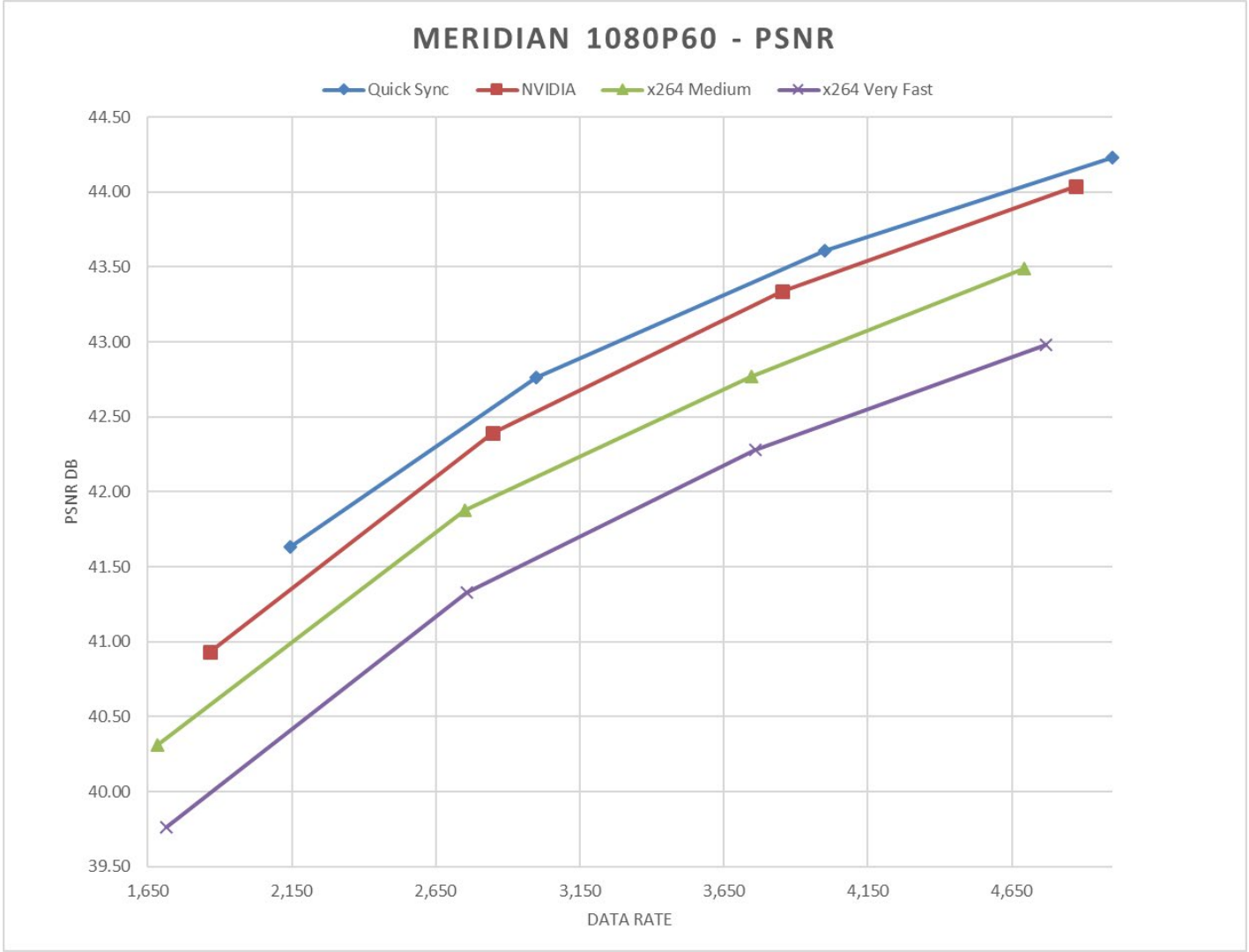


- Significant visual differences in one or two regions

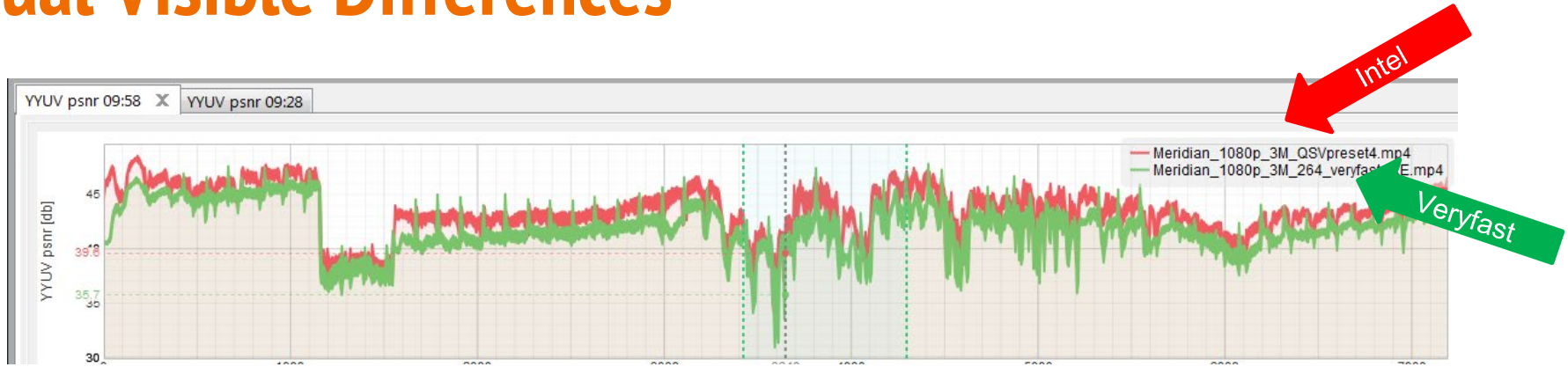
Sample Differential- Source


Intel
00;00;20;43


Very Fast
00;00;20;43


MERIDIAN 1080P60 - PSNR
Quick Sync
NVIDIA
x264 Medium
x264 Very Fast
44.50
44.00
43.50
43.00
42.50
42.00
41.50
41.00
40.50
40.00
39.50
PSNR DB
1,650
2,150
2,650
3,150
3,650
4,150
4,650
DATA RATE

Actual Visible Differences



- Issues very transient
- Probably not noticeable

- Frames brightened by 40%

Sample Differential- Source

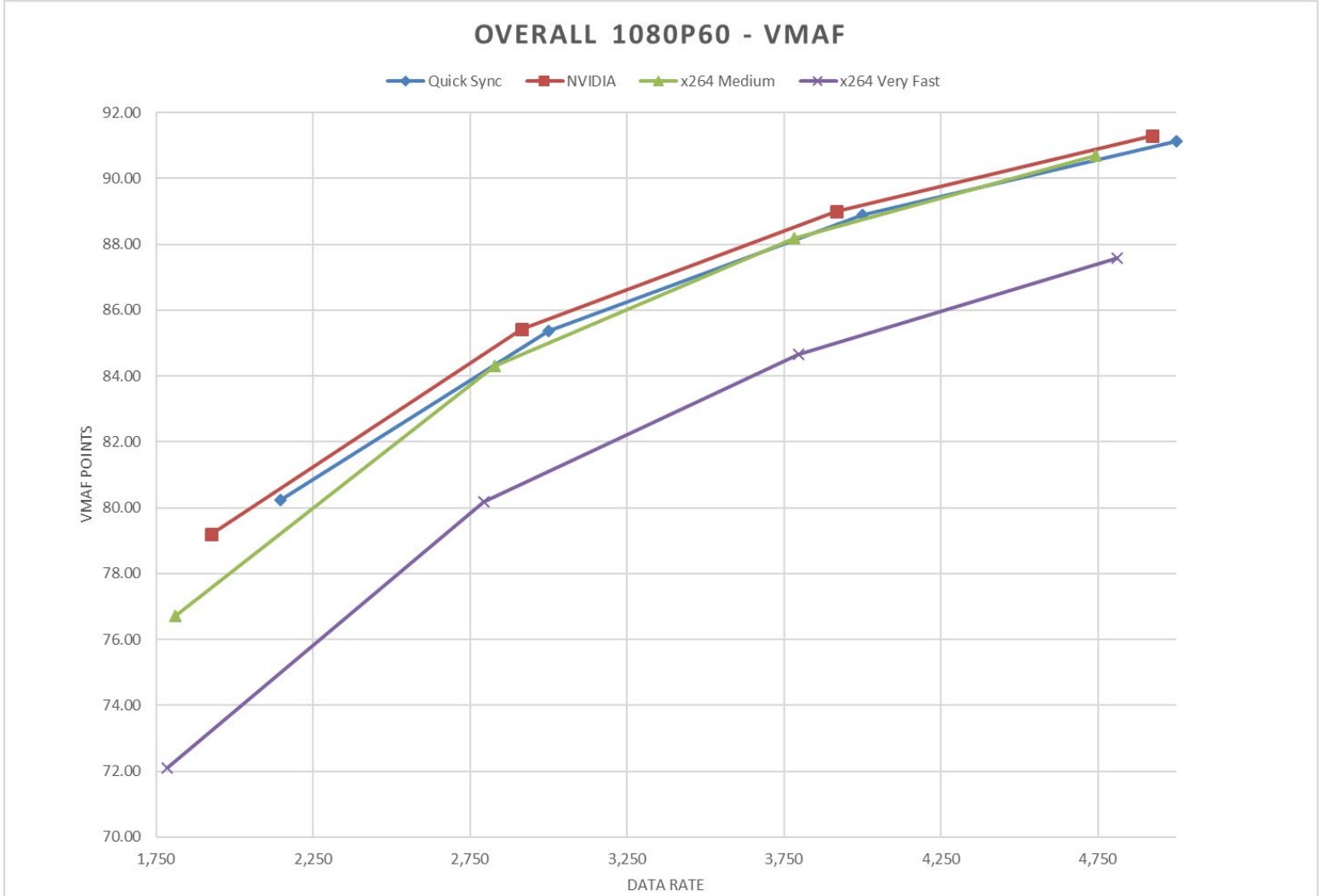

Intel
00;01;00;52


Very Fast
00;01;00;52

Meridian - BD Rate

VMAF	Quick Sync	NVIDIA	x264 Medium	x264 Very Fast
Quick Sync	1 X	-5.72	-22.12	-45.32
NVIDIA	6.07 2	X	-16.17	-40.62
x264 Medium	28.41	19.29	X	3 -29.22
x264 Very Fast	4 82.88	68.41	41.29	X

PSNR	Quick Sync	NVIDIA	x264 Medium	x264 Very Fast
Quick Sync	1 X	-5.69	-18.43	-31.24
NVIDIA	6.03 2	X	-11.95	-25.98
x264 Medium	22.60	13.58	X	3 -16.69
x264 Very Fast	4 45.44	35.10	20.03	X


OVERALL 1080P60 - VMAF
Quick Sync
NVIDIA
x264 Medium
x264 Very Fast
92.00
90.00
88.00
86.00
84.00
82.00
80.00
78.00
76.00
74.00
72.00
70.00
VMAF POINTS
1,750
2,250
2,750
3,250
3,750
4,250
4,750
DATA RATE

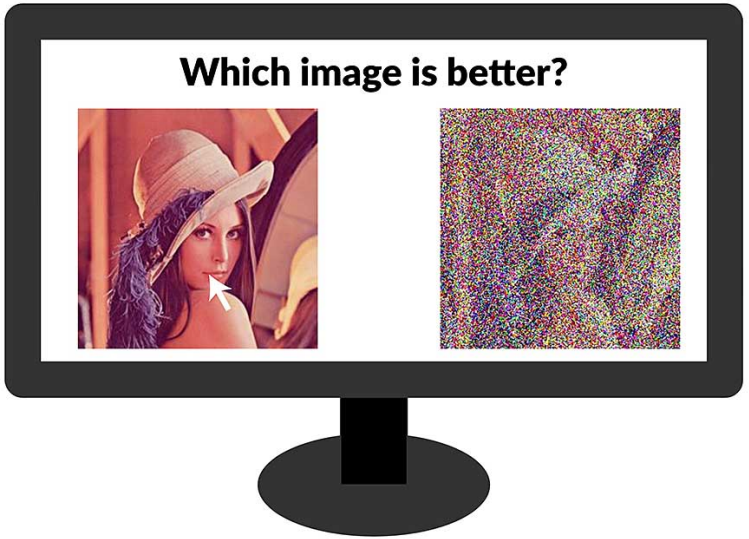
Overall - BD Rate

VMAF	Quick Sync	NVIDIA	x264 Medium	x264 Very Fast
Quick Sync	X	3.44 (2)	-1.26	-24.65
NVIDIA	(1) -3.33	X	-4.75	-27.23
x264 Medium	1.27	4.99	X	(3) -22.67
x264 Very Fast	(4) 32.71	37.41	29.31	X

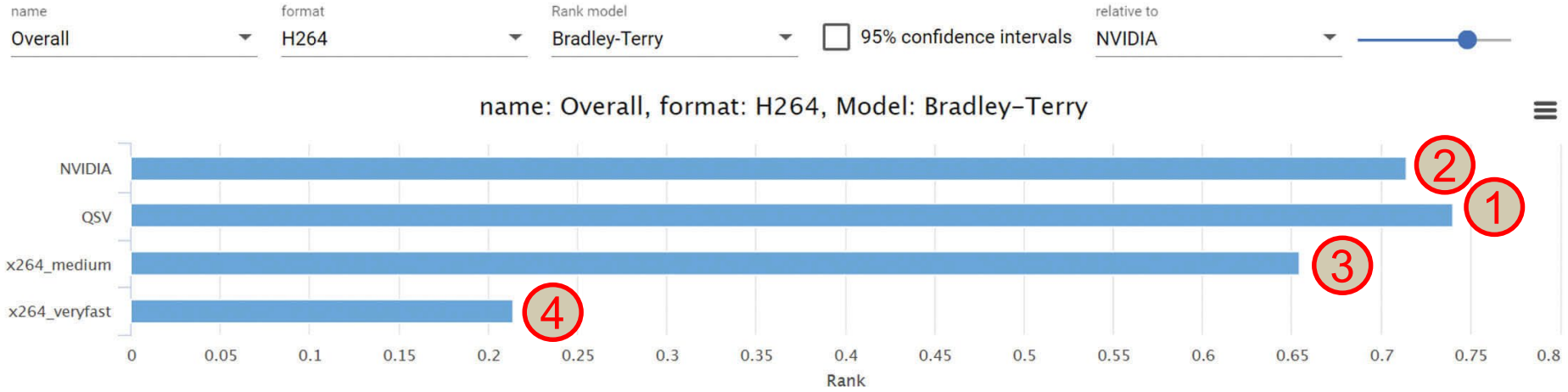
PSNR	Quick Sync	NVIDIA	x264 Medium	x264 Very Fast
Quick Sync	(1) X	-0.46	-2.12	-16.09
NVIDIA	0.46 (2)	X	-1.81	-15.85
x264 Medium	2.17	1.85	X	(3) -14.03
x264 Very Fast	(4) 19.18	18.84	16.32	X

Subjective Ratings via Subjectify

- Subjectify is a service from Moscow State University that recruits viewers to compare video and still images
- How it works:
  - You send them test files
  - They recruit viewers to run A:B tests
  - They return stats like you're about to see
  - Cost is ~$3/viewer (who can compare ten 20-second A:B comparisons per session)
  - Total cost for work done for this article – under $300 (paid for by NGCodec and Intel)
  - Website: http://www.subjectify.us/
  - My review: http://bit.ly/Ozer_Subjectify



Subjective Ratings (First 20 Seconds of Each File)



H.264 Summary

	Quick Sync	NVIDIA	Medium	Very Fast
Cost per hour	$0.35	$0.57	$0.47	$0.24
Stream consistency	1	2	3	4
VMAF quality rank	2	1	3	4
PSNR quality rank	1	2	3	4
Subjective quality	1	2	3	4
Overall	**1**	**2**	**3**	**4**

HEVC

- Compared:
  - Xilinx - FPGA-based encoding (was NGCodec)
  - Intel SVT-HEVC - preset 6
  - X265 medium
  - x265 veryfast

Xilinx

- Test spec - 16 core AMD EPYC CPU based machine with 32GB of DDR4 RAM and 1TB of SSD
- Two FPGAs
- Full PCIe 16 lanes communication speed between CPU and both FPGAs.

- Performance
  - One full encoding ladder for each FPGA

Xilinx Script

```
ffmpeg -y -re -i football_1080p.mp4 -c:a aac -b:a 128k -ac 2 -ar 48000 -
c:v NGC265 -b:v 3M -g 0 -idr-period 120 football_1080p_3M_ngc265.mp4
```

- Xilinx provided
- No preset to toggle quality vs. encoding speed
  - Either live and full quality or not live
  - Buffer setting is fixed
- Tuning
  - `-aq-mode 0` switch to disable adaptive quantization for objective tests (per Xilinx)

Xilinx – Capacity/Cost

- Tested on FPGA-based cloud computer (AS-f1.2fx8c) hosted by Altered Silicon:
  - Two FPGA cards
  - Cost $2.21 per hour
- Our tests
  - One encoding ladder
  - Xilinx claimed 2 streams per FPGA possible with planned upgrade
  - We used $0.054/hour ($2.21/4)
    - If you consider the Xilinx system, you should verify this performance up front

Intel SVT-HEVC

- What is SVT-HEVC?
  - “The Scalable Video Technology for HEVC Encoder (SVT-HEVC Encoder) is an HEVC-compliant encoder library core that achieves excellent density-quality tradeoffs, and is highly optimized for Intel® Xeon Scalable Processor and Xeon D processors”
  - bit.ly/GY-SVT-HEVC
  - Basically, a highly efficient codec for multi-threaded operation

Which Preset?

- Tested Preset 6 at Intel's request

Intel Script

```
ffmpeg -SVTnew -i input.mp4 -c:v libsvt_hevc -tune 0 -rc 1 -preset 6
-b:v 5M -maxrate 5M -bufsize 10M -g 120 output.mp4
```

- Intel supplied
- Tuning
  - 0 – visual quality (used for subjective)
  - 1 – PSNR/SSIM
  - 2 – VMAF (used for objective)
- Doubled buffer size wherever possible on HEVC encodes

Intel Capacity/Cost

- Tested on a C5.9xlarge system with an Intel Xeon Platinum 8000 series (Skylake-SP) processor
- Produced two simultaneous encodes of the full encoding ladder using preset 6 tune 0
- Spot pricing was $0.3466 per hour, so cost/ladder was $0.1733.

X265 Script

```
ffmpeg -re -i input.mp4 -c:v libx265 -preset veryfast -x265-params
keyint=120:bitrate=5000k:vbv-maxrate=5000k:vbv-bufsize=10000k -pix_fmt yuv420p
output.mp4
```

- Simple as possible
- Changed to medium preset for those tests
- Tuned for PSNR for objective tests (`-tune psnr`)

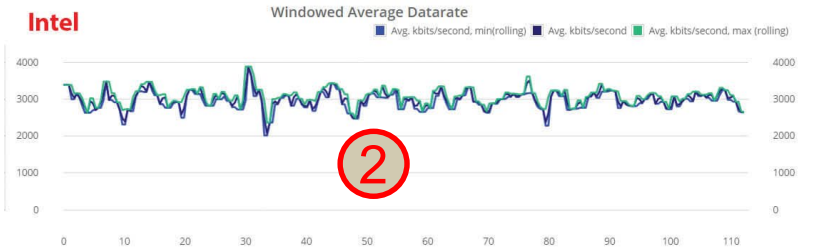
x265 Capacity/Cost

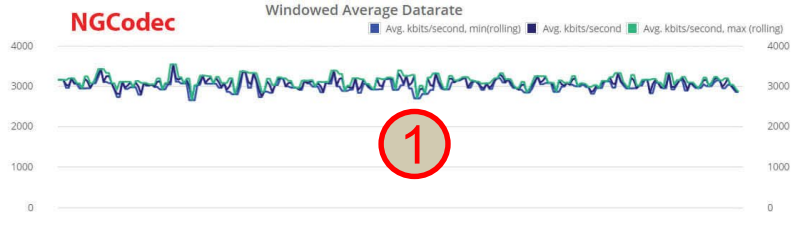
- Tested on a C5.9xlarge system with an Intel Xeon Platinum 8000 series (Skylake-SP) processor ($0.3466 per hour)
- Very fast produced no complete encoding ladder
- Cost/hour will exceed SVT-HEVC

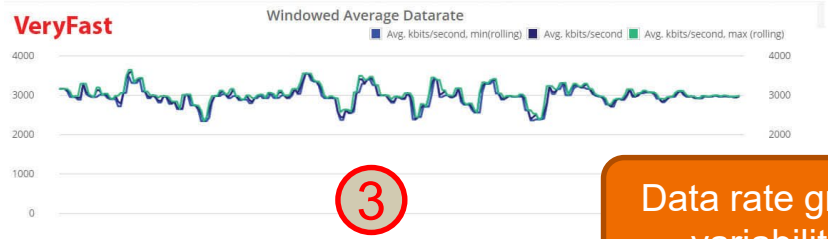
Data Rate Consistency

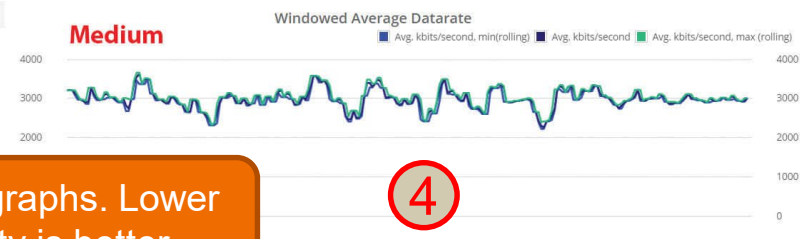
- Important for very large streaming sites (like Twitch)
    - If working with fixed pipes close to maximum capacity, data rate spikes can interrupt the stream

Data Rate Consistency (3 Mbps Football File)









Data rate graphs. Lower variability is better

	Data Rate	Standard Deviation	Max Data Rate
2 Intel SVT-HEVC	3013	253	3897
1 NGCodec	3076	149	3548
4 x265 medium	2990	253	3661
3 x265 veryfast	2989	240	3652

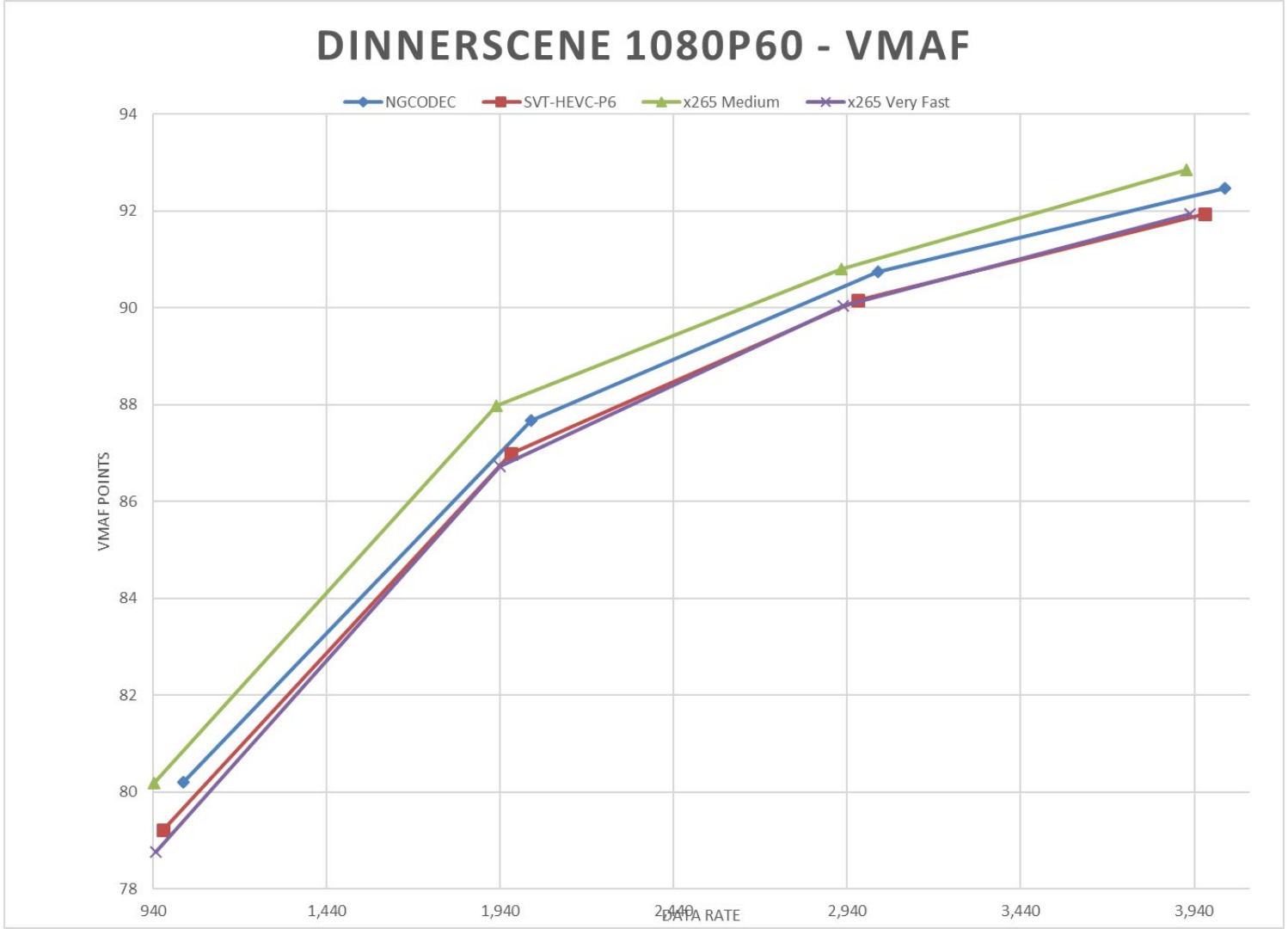
Lowest Flucuation

Lowest Max Data Rate

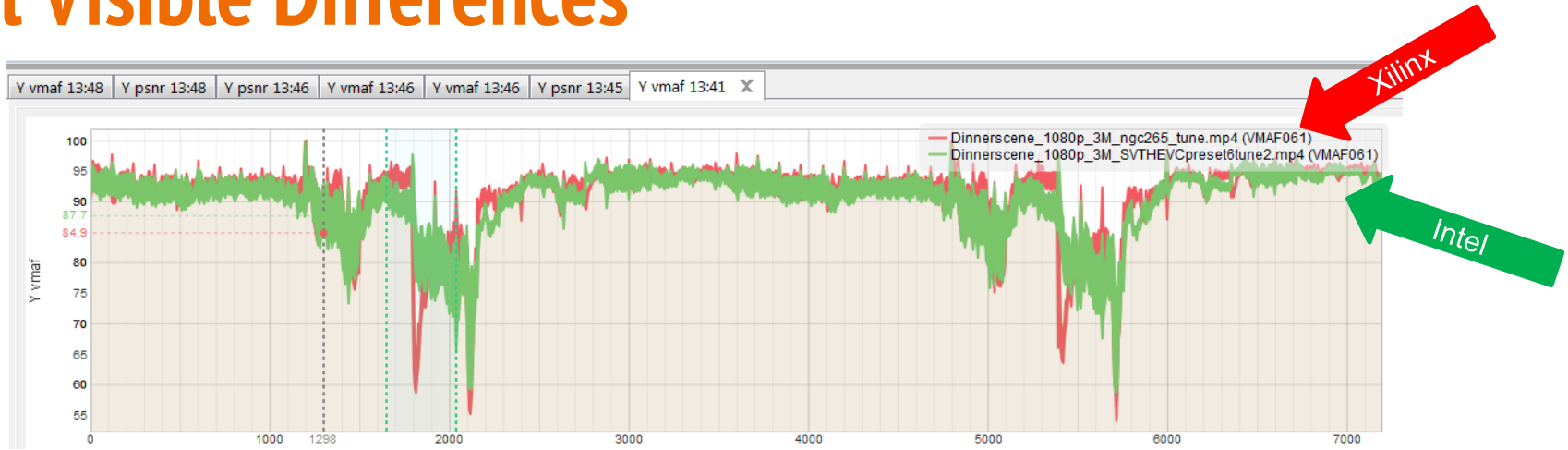
HEVC Quality Results

- Four videos
  - Netflix Dinner Scene
  - Harmonic football
  - GTAV
  - Netflix Meridian
  - All 1080p60
- Tested at 1-4 Mbps

- Four tested codecs
  - Xilinx
  - SVT-HEVC @ 6
  - X265 at medium and veryfast


DINNERSCENE 1080P60 - VMAF
NGCODEC
SVT-HEVC-P6
x265 Medium
x265 Very Fast
VMAF POINTS
94
92
90
88
86
84
82
80
78
940
1,440
1,940
2,440
2,940
3,440
3,940
DATA RATE

Actual Visible Differences



- Some major scoring differences
- Busy scene so not really visible.

- Xilinx scored higher but had a couple of low quality regions

Sample Differential- Source

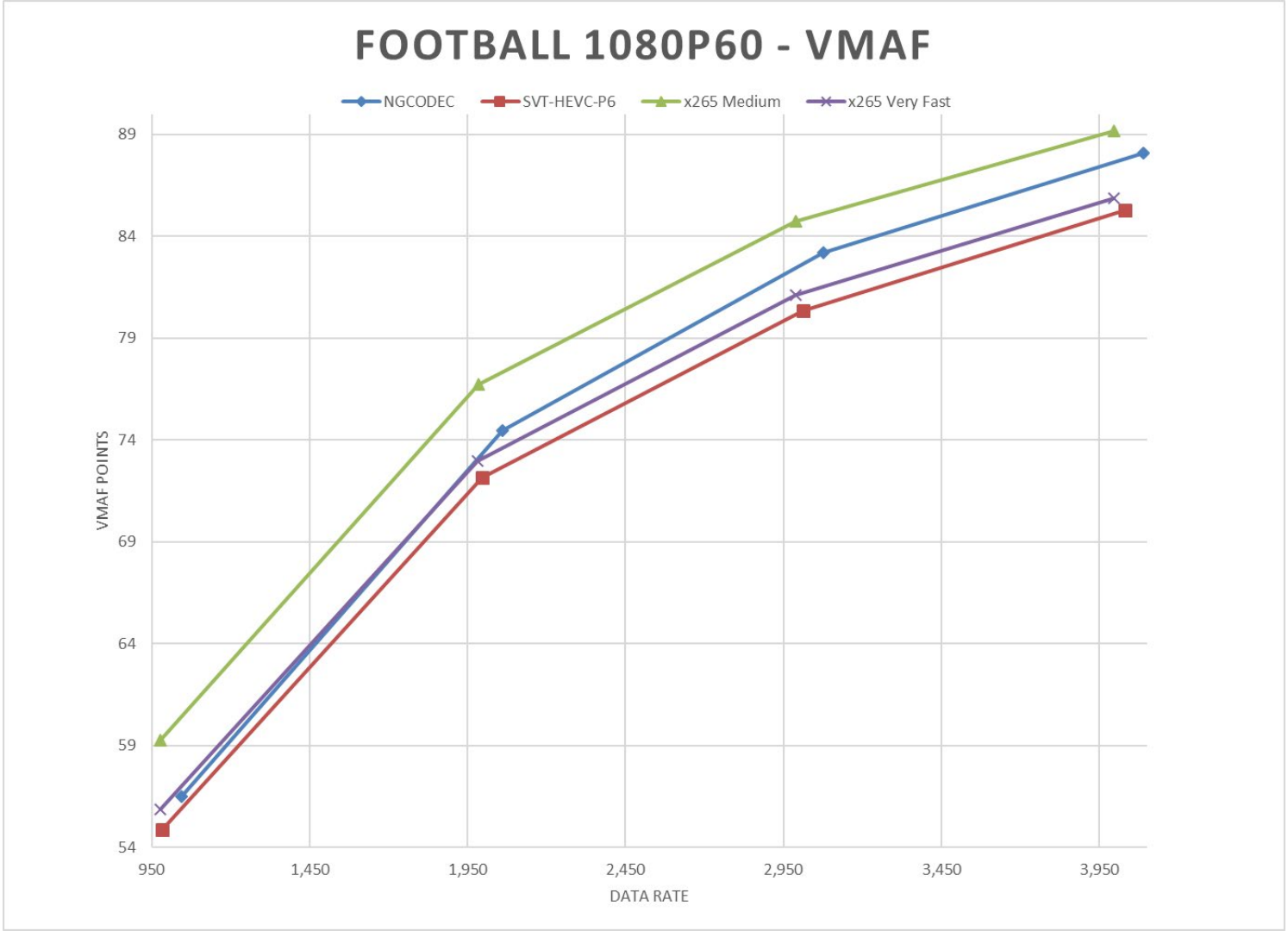

Xilinx
01:00:30:18


Intel
01:00:30:18

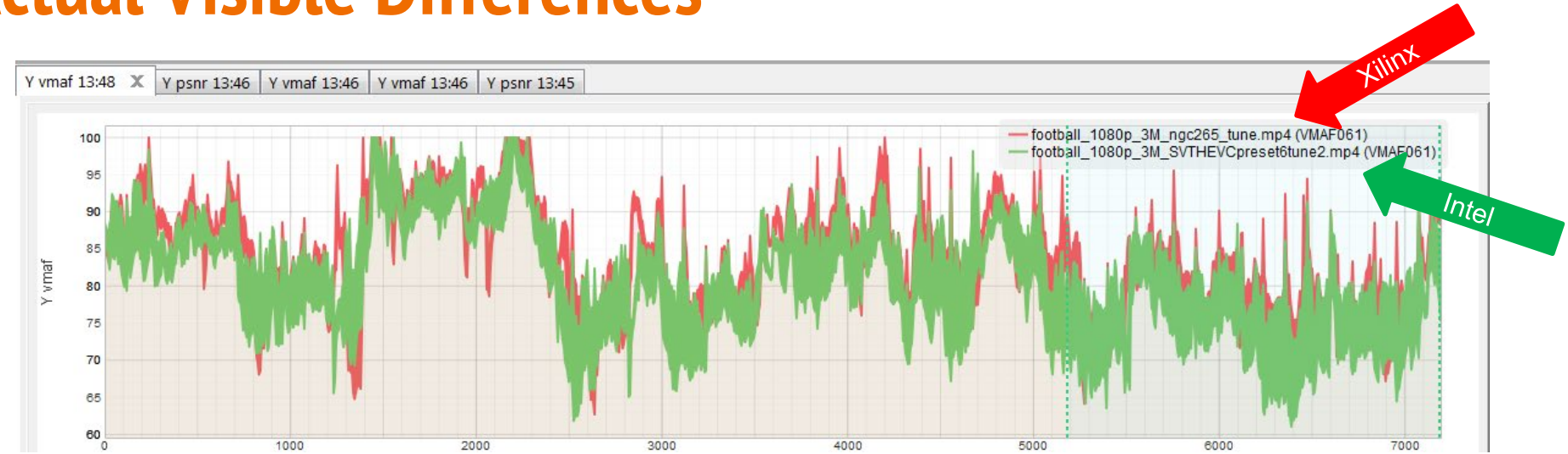
HEVC - Dinner Scene - BD-Rate Computations

VMAF	NGCODEC	SVT-HEVC-P6	x265 Medium	x265 Very Fast
NGCODEC	X	-4.21	12.28 (2)	-5.19
SVT-HEVC-P6	4.40	X	17.45	(3) -1.07
x265 Medium (1)	-10.93	-14.86	X	-15.73
x265 Very Fast	(4) 5.48	1.08	18.66	X

PSNR	NGCODEC	SVT-HEVC-P6	x265 Medium	x265 Very Fast
NGCODEC	X	-13.90	5.16 (2)	-4.92
SVT-HEVC-P6 (4)	16.15	X	21.64	9.78
x265 Medium (1)	-4.90	-17.79	X	-9.48
x265 Very Fast	5.17	-8.91 (3)	10.47	X

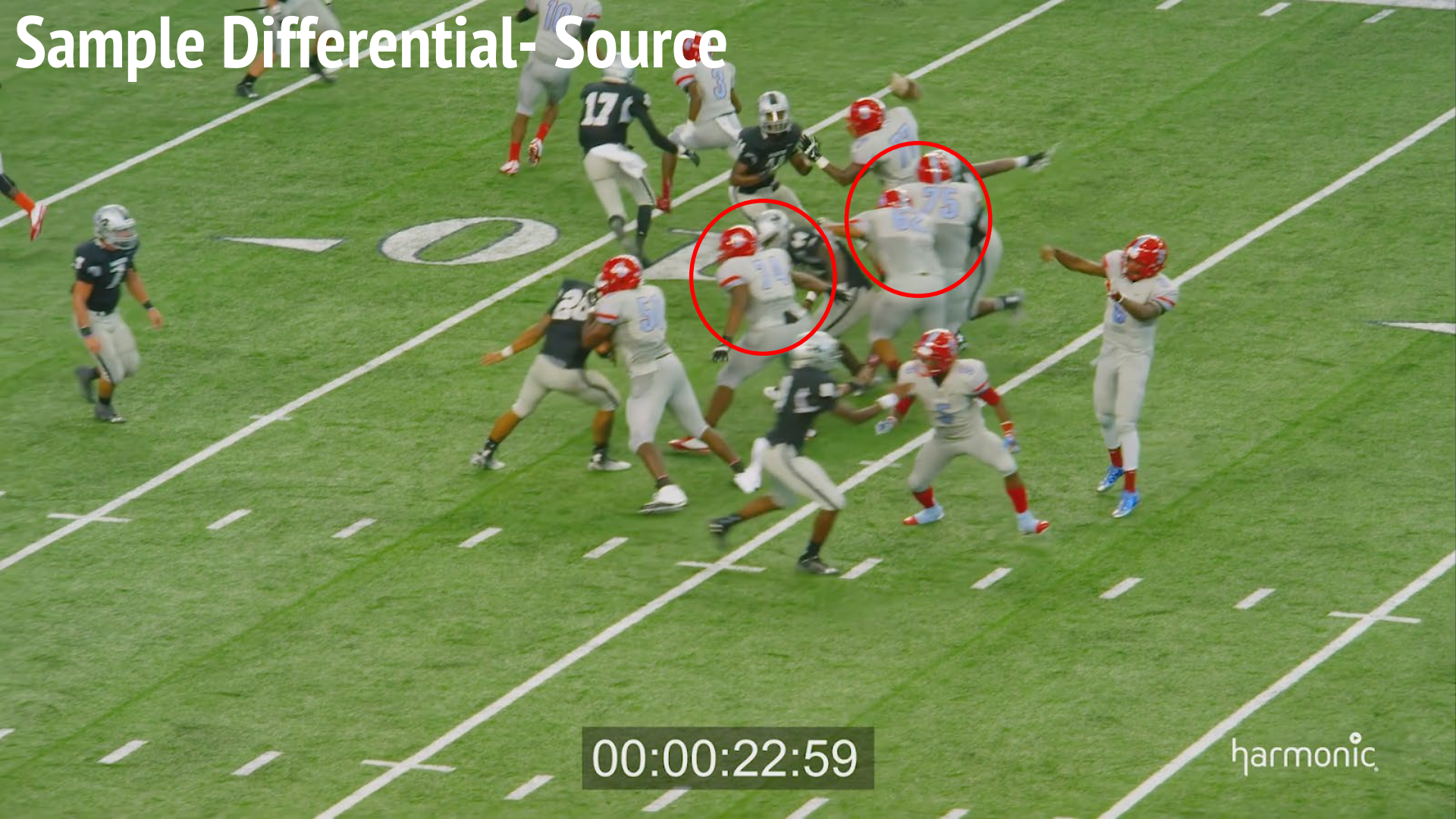

FOOTBALL 1080P60 - VMAF
NGCODEC
SVT-HEVC-P6
x265 Medium
x265 Very Fast
VMAF POINTS
89
84
79
74
69
64
59
54
950
1,450
1,950
2,450
2,950
3,450
3,950
DATA RATE

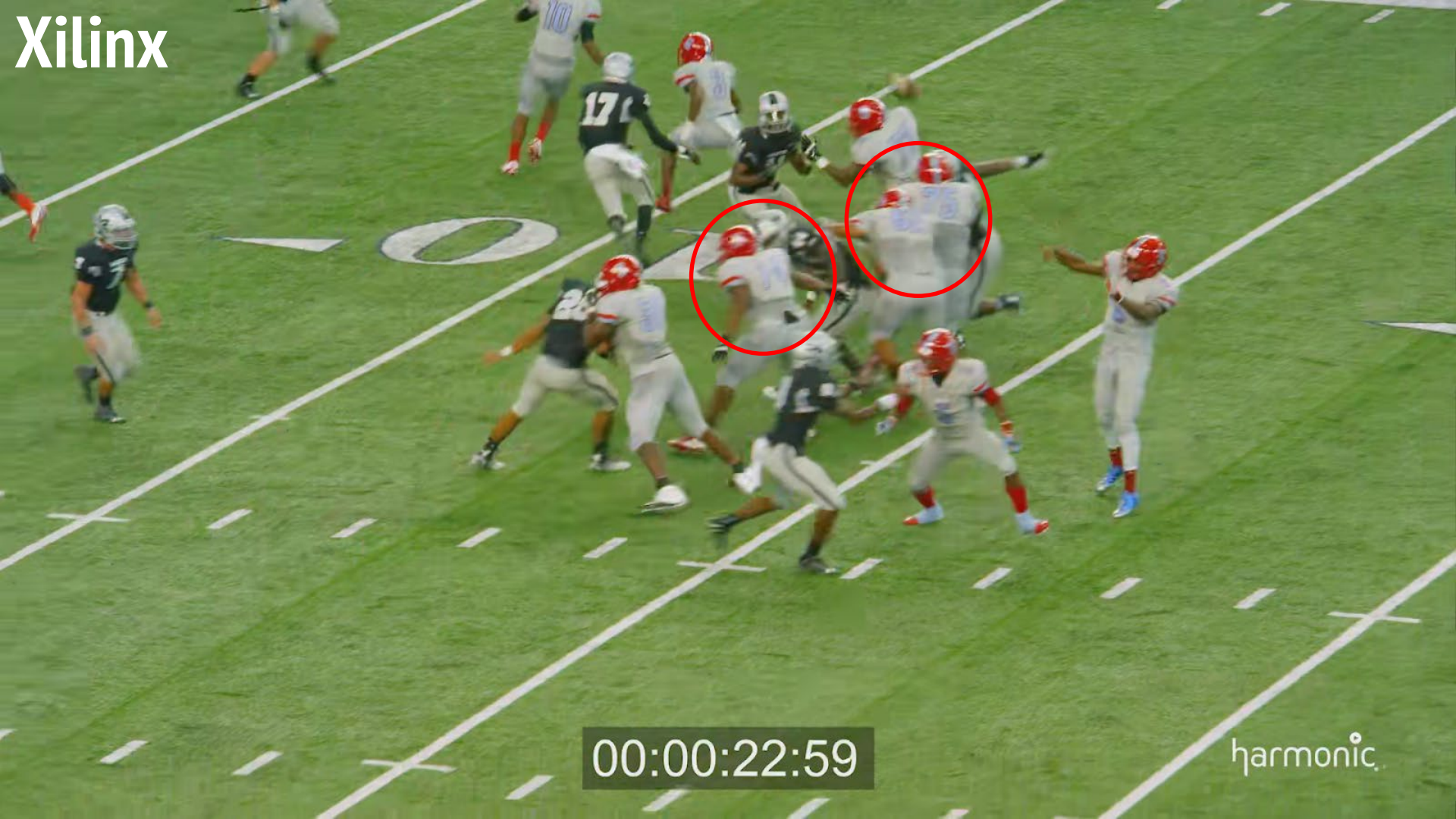
Actual Visible Differences

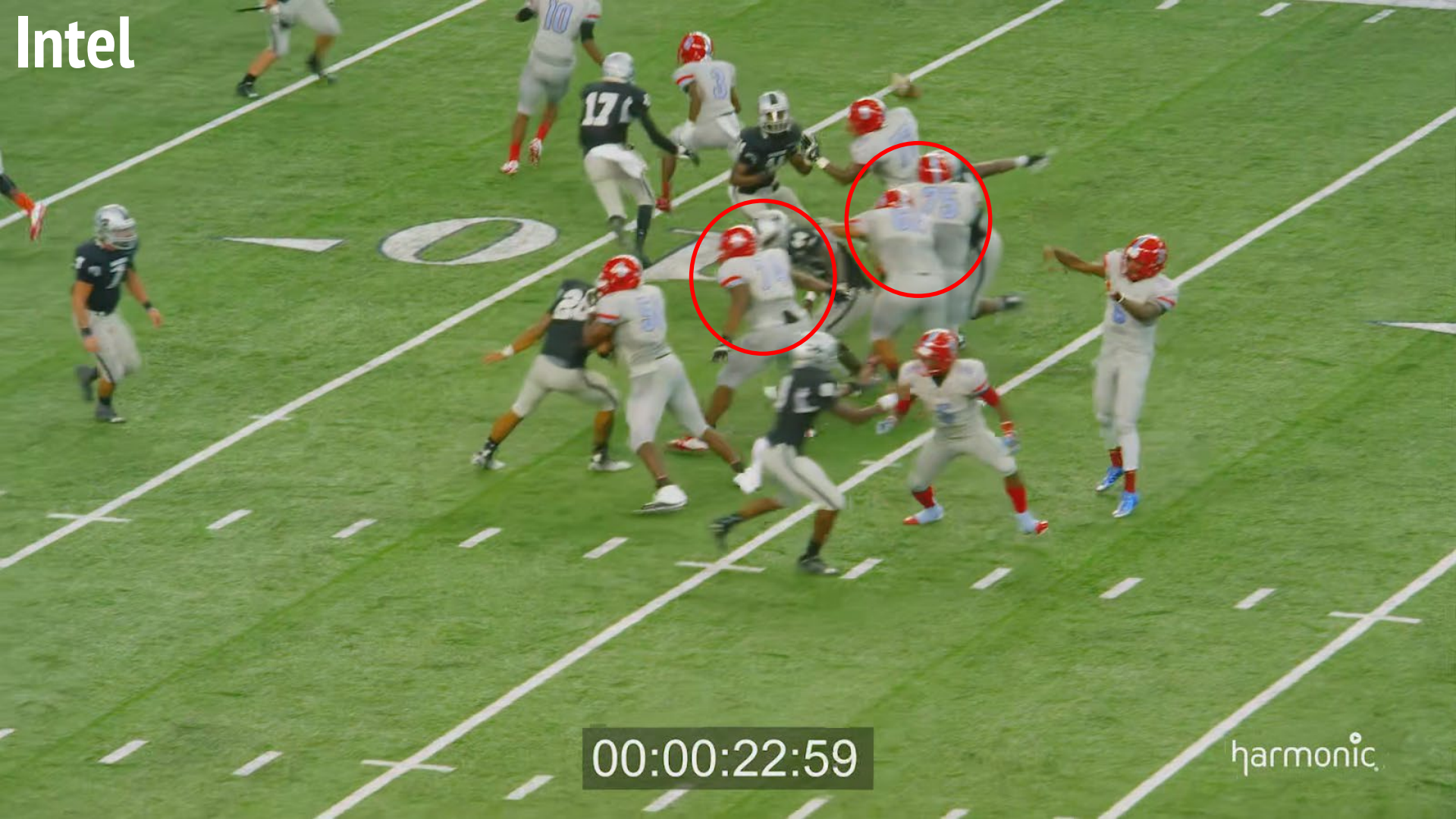


- Xilinx overall higher, but had some transient issues
- Very short and not really noticeable

Sample Differential- Source



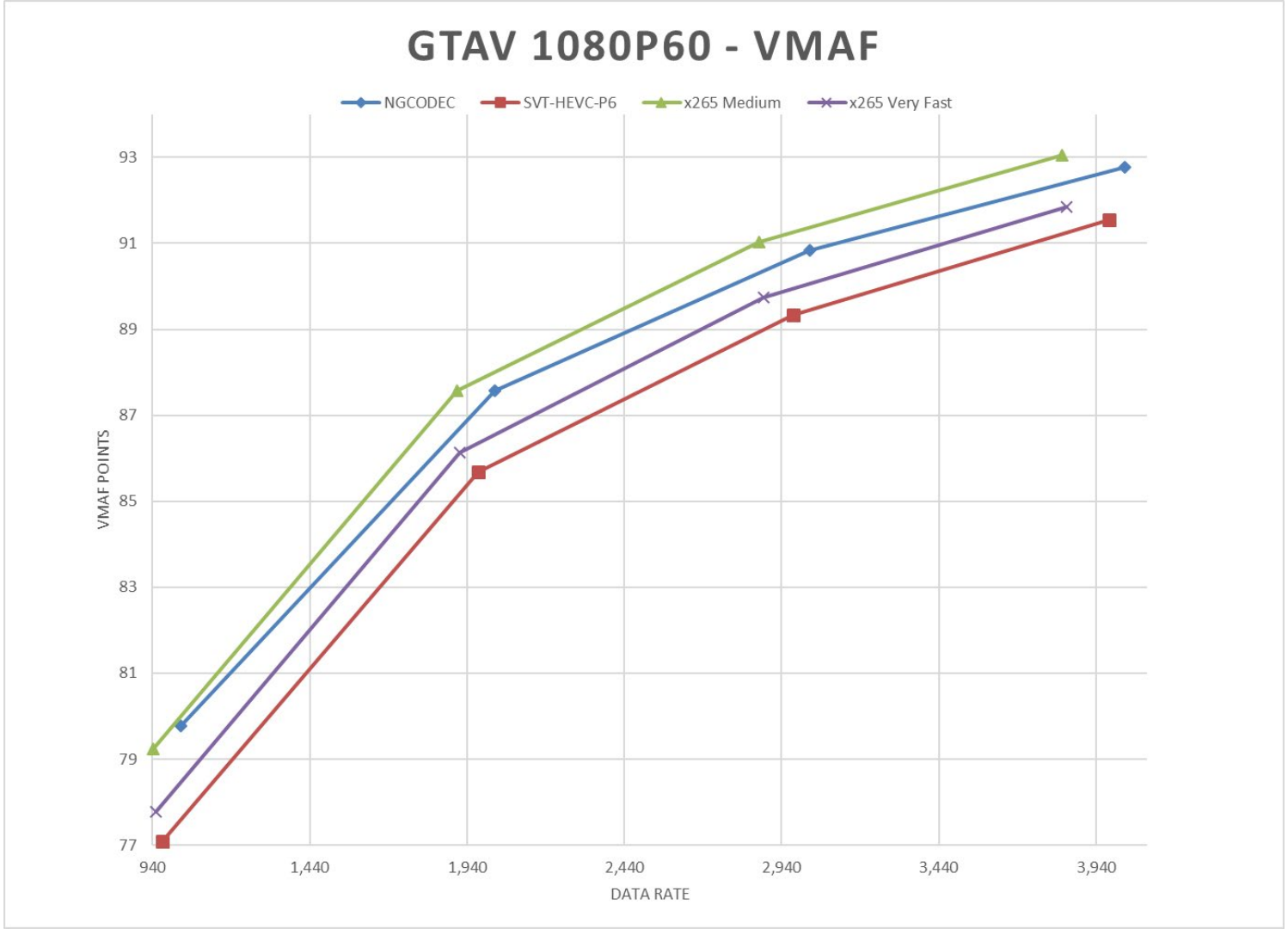

Xilinx
00:00:22:59
harmonic


Intel
00:00:22:59
harmonic

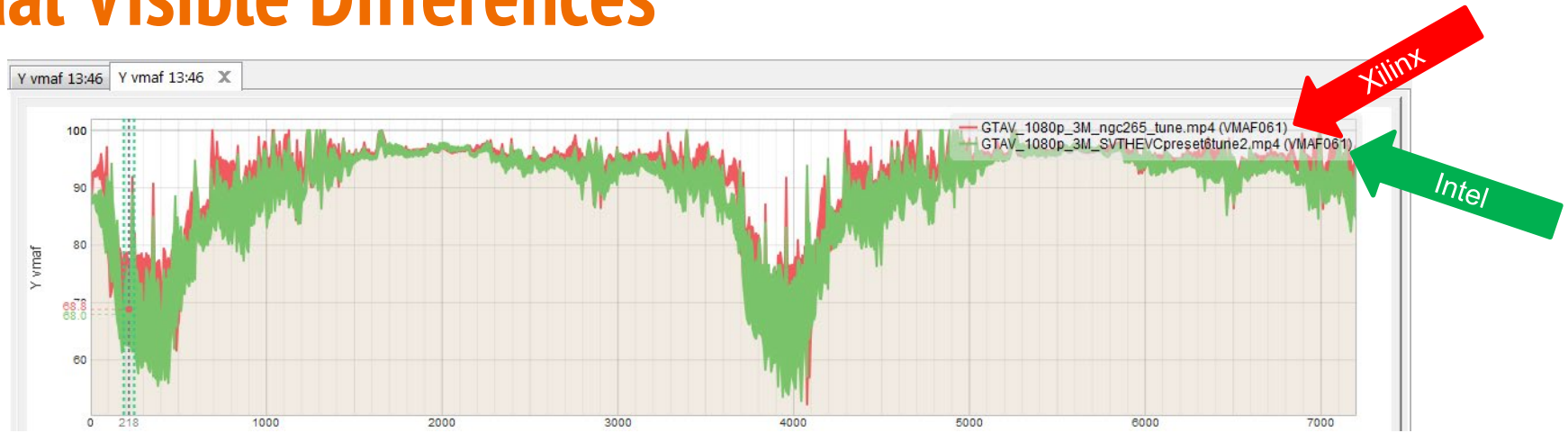
HEVC - Football - BD-Rate Computations

VMAF	NGCODEC	SVT-HEVC-P6	x265 Medium	x265 Very Fast
NGCODEC	X	-5.99	14.77 (2)	-1.90
SVT-HEVC-P6	(4) 6.37	X	23.28	4.55
x265 Medium	(1) -12.87	-18.88	X	-15.27
x265 Very Fast	1.94	-4.35 (3)	18.03	X

PSNR	NGCODEC	SVT-HEVC-P6	x265 Medium	x265 Very Fast
NGCODEC	X	-10.17	10.44 (2)	-3.44
SVT-HEVC-P6	(4) 11.32	X	24.91	7.94
x265 Medium	(1) -9.45	-19.94	X	-13.38
x265 Very Fast	3.57	-7.35 (3)	15.45	X


GTAV 1080P60 - VMAF
NGCODEC
SVT-HEVC-P6
x265 Medium
x265 Very Fast
VMAF POINTS
93
91
89
87
85
83
81
79
77
940
1,440
1,940
2,440
2,940
3,440
3,940
DATA RATE

Actual Visible Differences



- Very slight Blockiness in Xilinx clip
- Very short and not really noticeable

Sample Differential- Source


Xilinx
AMMU-NATION
Introducing a new breed of low-tech combat solution: our hand-sharpened machete is available in stores now.
00;00;03;38

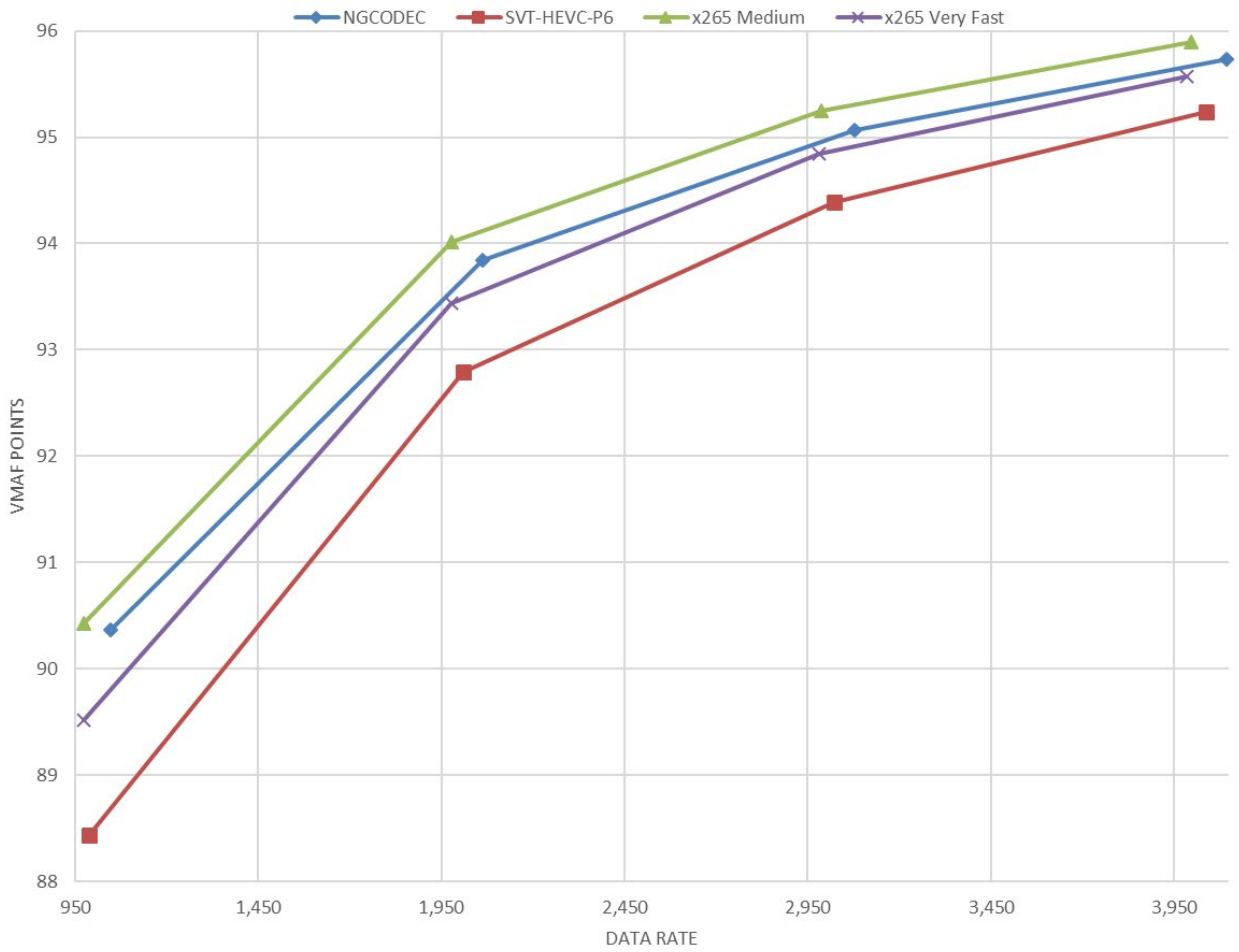

Intel
AMMU-NATION
Introducing a new breed of low-tech combat solution: our hand-sharpened machete is available in stores now.
00;00;03;38

HEVC - GTAV - BD-Rate Computations

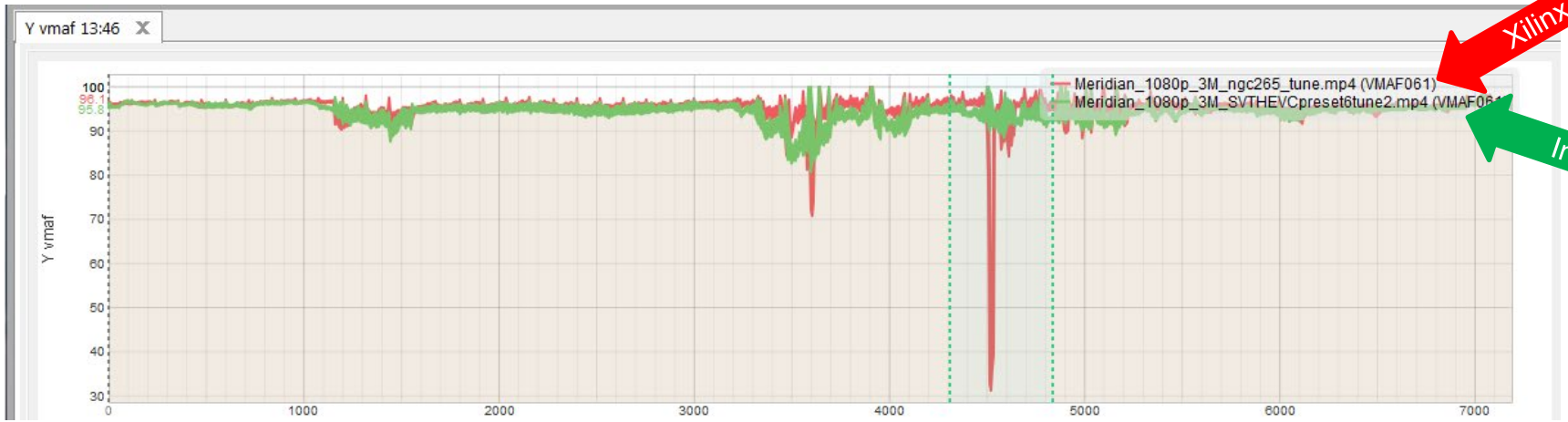
VMAF	NGCODEC	SVT-HEVC-P6	x265 Medium	x265 Very Fast
NGCODEC	X	-14.98	6.34 (2)	-8.63
SVT-HEVC-P6	(4) 17.61	X	24.53	7.33
x265 Medium	(1) -5.97	-19.70	X	-13.80
x265 Very Fast	9.44	-6.83 (3)	16.02	X

PSNR	NGCODEC	SVT-HEVC-P6	x265 Medium	x265 Very Fast
NGCODEC	(1) X	-22.50	-0.20	-18.79
SVT-HEVC-P6	(4) 29.04	X	27.59	3.99
x265 Medium	0.20 (2)	-21.62	X	-18.20
x265 Very Fast	23.13	-3.84 (3)	22.25	X

MERIDIAN 1080P60 - VMAF



Actual Visible Differences



- Xilinx overall higher, but had two transient issues, one very major
  - Hide your eyes
- Probably would be perceivable though very short

Sample Differential- Source

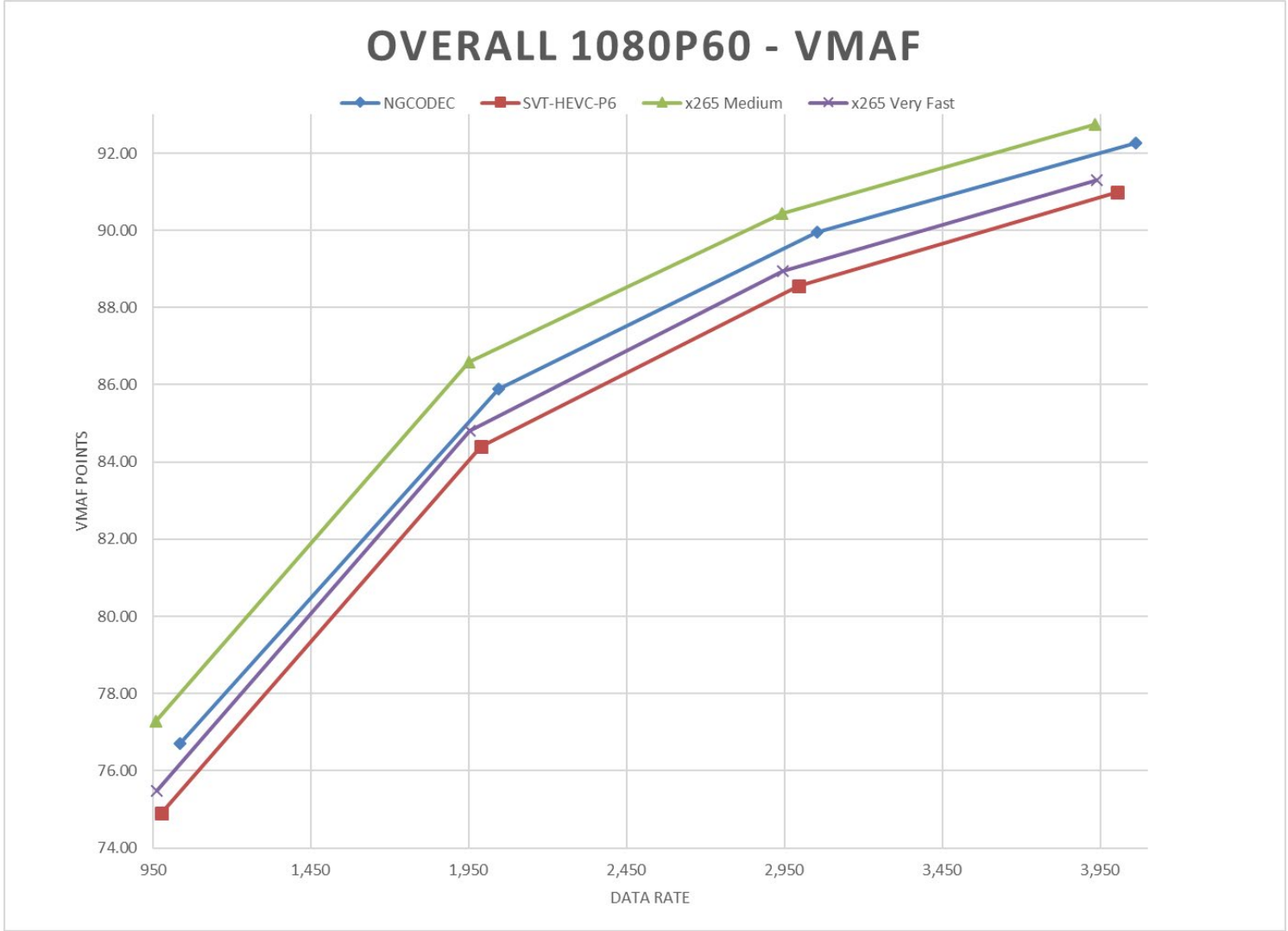

Xilinx
Help me, I'm melting !
00;01;15;24


Intel
00;01;15;24

HEVC - Meridian - BD Rate

VMAF	NGCODEC	SVT-HEVC-P6	x265 Medium	x265 Very Fast
NGCODEC	X	-20.23	8.69 (2)	-6.64
SVT-HEVC-P6	(4) 25.35	X	35.97	17.12
x265 Medium	(1) -7.99	-26.46	X	-14.04
x265 Very Fast	7.12	-14.62 (3)	16.33	X

PSNR	NGCODEC	SVT-HEVC-P6	x265 Medium	x265 Very Fast
NGCODEC	X	-25.35	7.79 (2)	-3.86
SVT-HEVC-P6	(4) 33.96	X	43.86	27.44
x265 Medium	(1) -7.23	-30.49	X	-10.77
x265 Very Fast	4.02	-21.53 (3)	12.07	X

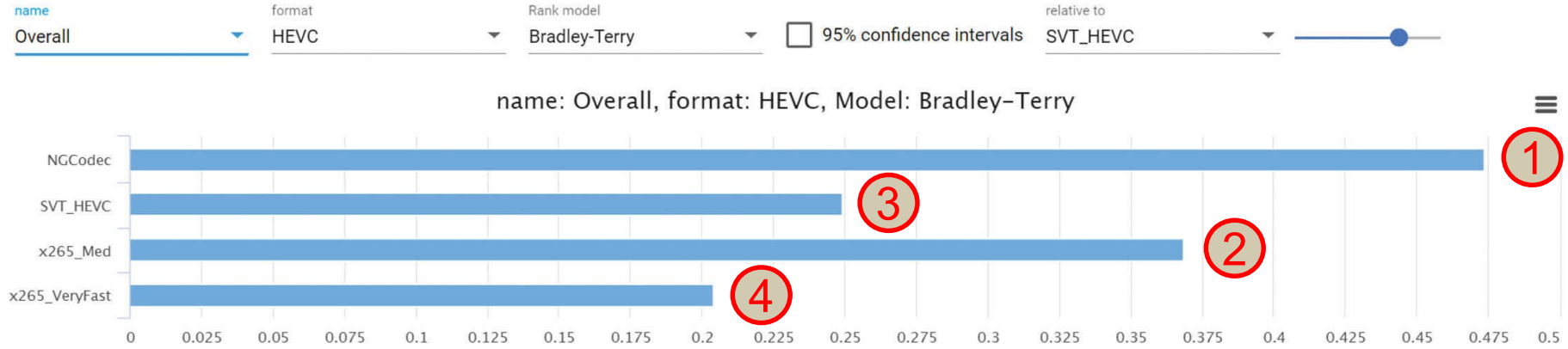

OVERALL 1080P60 - VMAF
NGCODEC
SVT-HEVC-P6
x265 Medium
x265 Very Fast
92.00
90.00
88.00
86.00
84.00
82.00
80.00
78.00
76.00
74.00
VMAF POINTS
950
1,450
1,950
2,450
2,950
3,450
3,950
DATA RATE

HEVC - Overall - BD Rate

VMAF	NGCODEC	SVT-HEVC-P6	x265 Medium	x265 Very Fast
NGCODEC	X	-9.18	12.06 (2)	-4.34
SVT-HEVC-P6	(4) 10.11	X	23.78	5.44
x265 Medium	(1) -10.76	-19.21	X	-14.86
x265 Very Fast	4.54	-5.16 (3)	17.45	X

PSNR	NGCODEC	SVT-HEVC-P6	x265 Medium	x265 Very Fast
NGCODEC	X	-17.64	5.86 (2)	-8.24
SVT-HEVC-P6	(4) 21.41	X	28.57	11.11
x265 Medium	(1) -5.53	-22.22	X	-13.33
x265 Very Fast	8.98	-10.00 (3)	15.38	X

Subjective Ratings (First 20 Seconds of Each File)



HEVC Summary

	Xilinx	SVT-HEVC	x265 Medium	X265 Very Fast
Cost per hour	$0.54	$0.1733	> $0.1733	> $0.1733
VMAF quality rank	2	4	1	3
PSNR quality rank	3	4	1	3
Subjective quality	1	3	2	4
Transient issues	Yes	No	No	No
Stream consistency	1	2	2	2

- X265 good option if affordable
- Xilinx expensive but good quality
  - Transient issues a concern

- SVT-HEVC is a work in process
  - Impressive debut, should advance nicely

What's the Bottom Line?

- Hardware encoding showed great promise
  - H.264 - NVIDIA was worth exploring
    - Intel not so much - lower quality and transient issues
  - HEVC - Xilinx - best for live encoding
    - SVT - Real time quality needs improvement (but codec is new)
    - Best quality looks competitive with x265 (but need to compare at x.265 Medium to Slow for true comparison)
    - Will run these tests for upcoming article in Streaming Media

Suggested Procedure

- Test capacity using current encoding ladder to compute cost/hour
- Test quality using four files at relevant intervals (four data points needed for rate distortion graph)
  - Performance/quality graphs should provide a good starting point
  - Look underneath the numbers (visualization tool is essential to identify problem areas and compare actual frames)
- Strongly consider subjective evaluations for key technology decisions
  - Subjective quality usually tracks objective, but not always