NVIDIA VIDEO TECHNOLOGIES

Abhijit Patait, 3/20/2019

AGENDA

NVIDIA Video Technologies Overview

Turing Video Enhancements

Video Codec SDK Updates

Benchmarks

Roadmap

NVIDIA VIDEO TECHNOLOGIES

NVIDIA GPU VIDEO CAPABILITIES

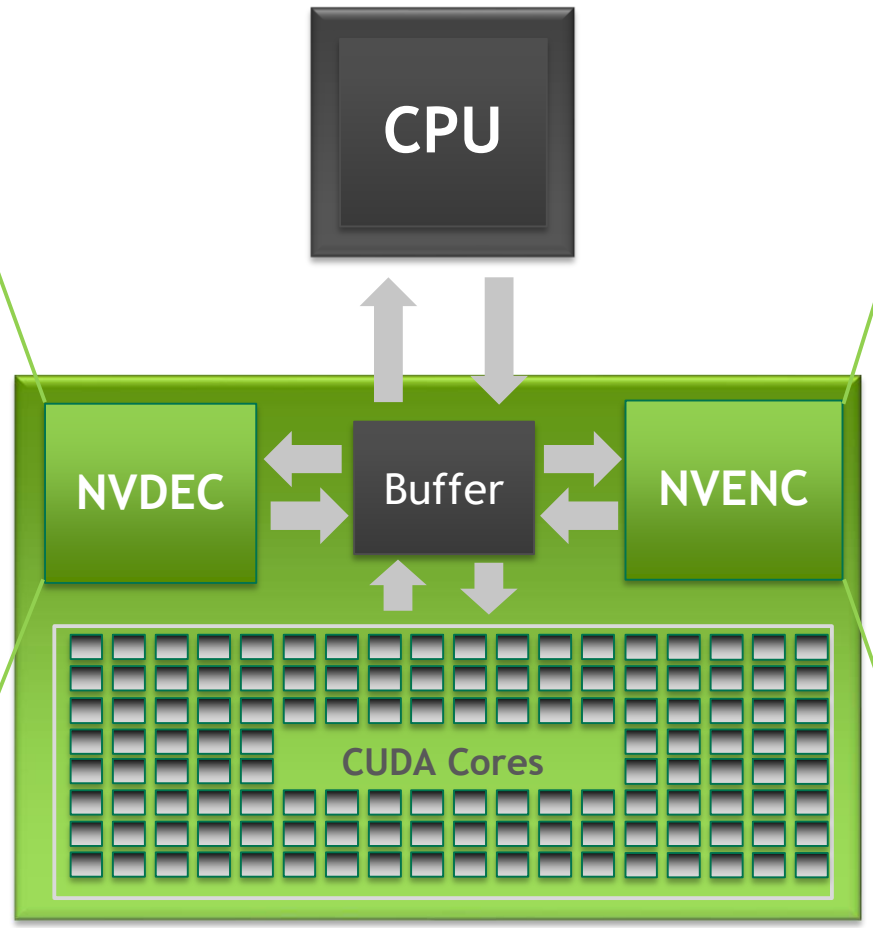
Decode HW*

Formats:
- MPEG-2
- VC1
- VP8
- VP9
- H.264
- H.265
- Lossless

Bit depth:
- 8/10/12 bit

Color**
- YUV 4:2:0
- YUV 4:4:4

Resolution
- Up to 8K***



Encode HW*

Formats:
- H.264
- H.265
- Lossless

Bit depth:
- 8 bit
- 10 bit

Color**
- YUV 4:4:4
- YUV 4:2:0

Resolution
- Up to 8K***

** See support diagram for previous NVIDIA HW generations*
*** 4:4:4 is supported only on HEVC for Turing; 4:2:2 is not natively supported on HW*
**** Support is codec dependent*

VIDEO CODEC SDK

A comprehensive set of APIs for GPU-accelerated video encode and decode

NVENCODE API for video encode acceleration

NVDECODE API for video & JPEG decode acceleration (formerly called NVCUVID API)

Independent of CUDA/3D cores on GPU for pre-/post-processing

- Gamestream
- Video transcoding
- Remote desktop streaming
- Intelligent video analytics
- Video archiving
- Video editing

NVIDIA VIDEO TECHNOLOGIES

SOFTWARE

FFMPEG — Easy access to GPU video acceleration

DeepStream SDK

DALI

cuDNN, TensorRT, cuBLAS, cuSPARSE

VIDEO CODEC, OPTICAL FLOW SDK

Video Encode and Decode for Windows and Linux
CUDA, DirectX, OpenGL interoperability

CUDA TOOLKIT

APIs, libraries, tools, samples

NVIDIA DRIVER

HARDWARE

NVENC

Video encode

H.264 MPEG-4/AVC

H.265 HEVC

NVDEC

Video decode

MPEG2

H.264 MPEG-4/AVC

H.265 HEVC

VP8 VP9

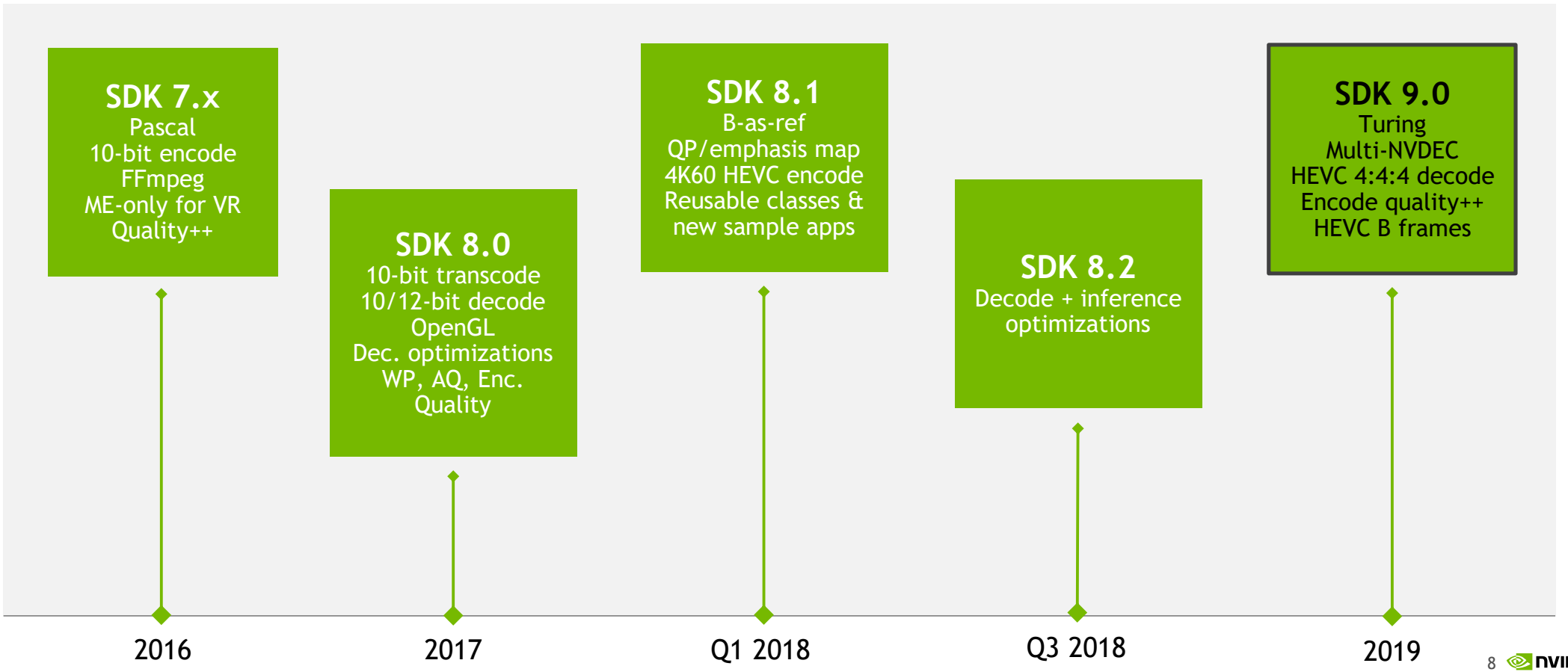
CUDA

High-performance computing on GPU

NVIDIA CUDA NVIDIA.COM

VIDEO CODEC SDK UPDATE

VIDEO CODEC SDK UPDATE



VIDEO CODEC SDK 9.0

Soul

Feature	Who it benefits
Higher video encode quality HEVC B-frames Higher encode quality	Cloud gaming Game broadcasting (e.g. Twitch) Video transcoding (e.g. Youtube, Facebook) OTT/M&E
HEVC 4:4:4 decode	End-to-end high-quality remote desktop
Mutiple NVDECs	Higher decode + inference throughput
Direct output to vidmem	Higher perf with post-processing
Power 9 + Tesla V100 SXM2	Video SDK for IBM platforms

TURING UPDATES - NVDEC

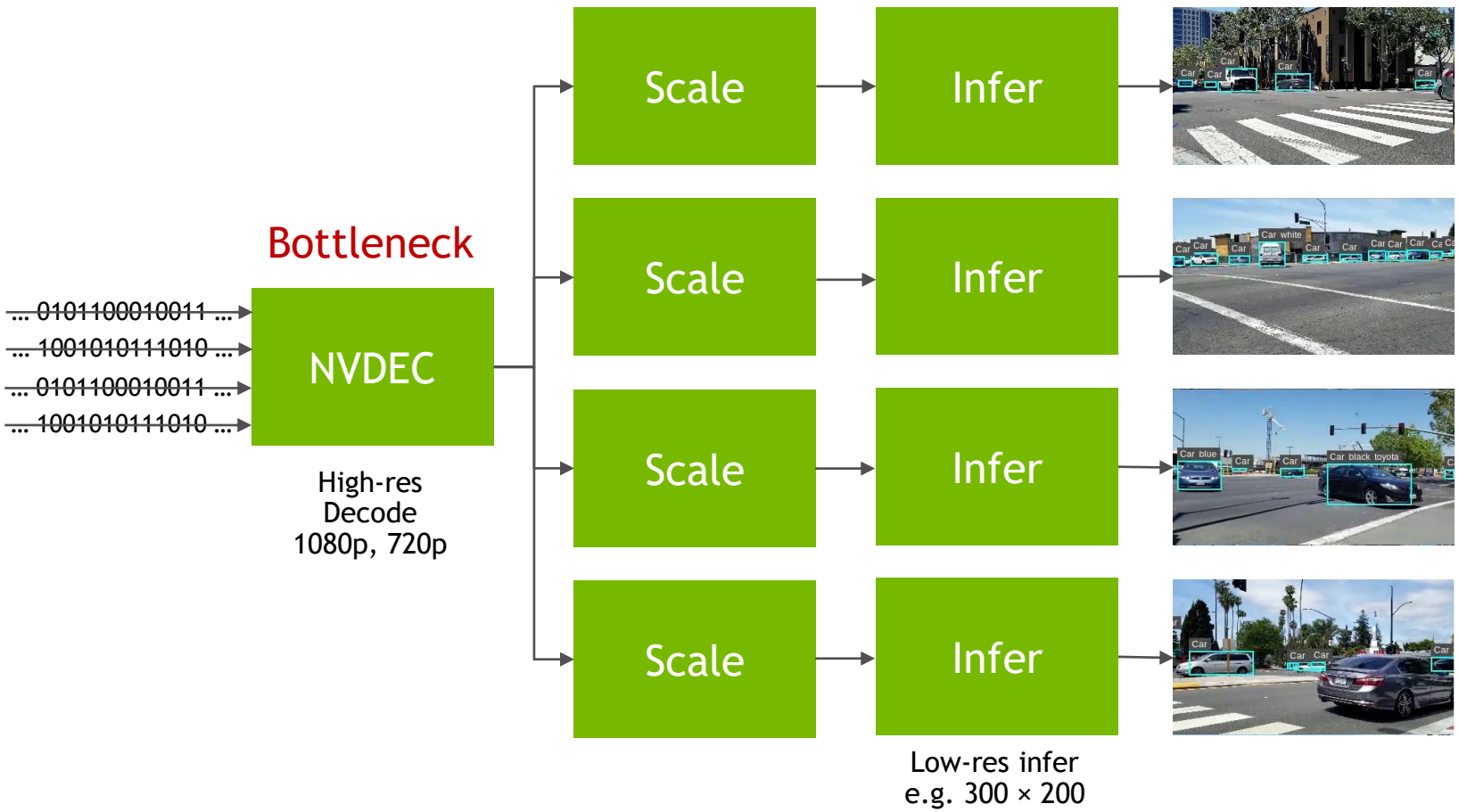
MULTIPLE NVDECS IN TURING

GPU	Number of NVDECs per GPU
Volta, Pascal & earlier	1
Turing – GeForce (RTX)	1
Turing – Quadro & Tesla (TU106)	3
Turing – Quadro & Tesla (TU104)	2
Turing – others	1

- Quadro & Tesla feature
- Auto-load-balanced by driver

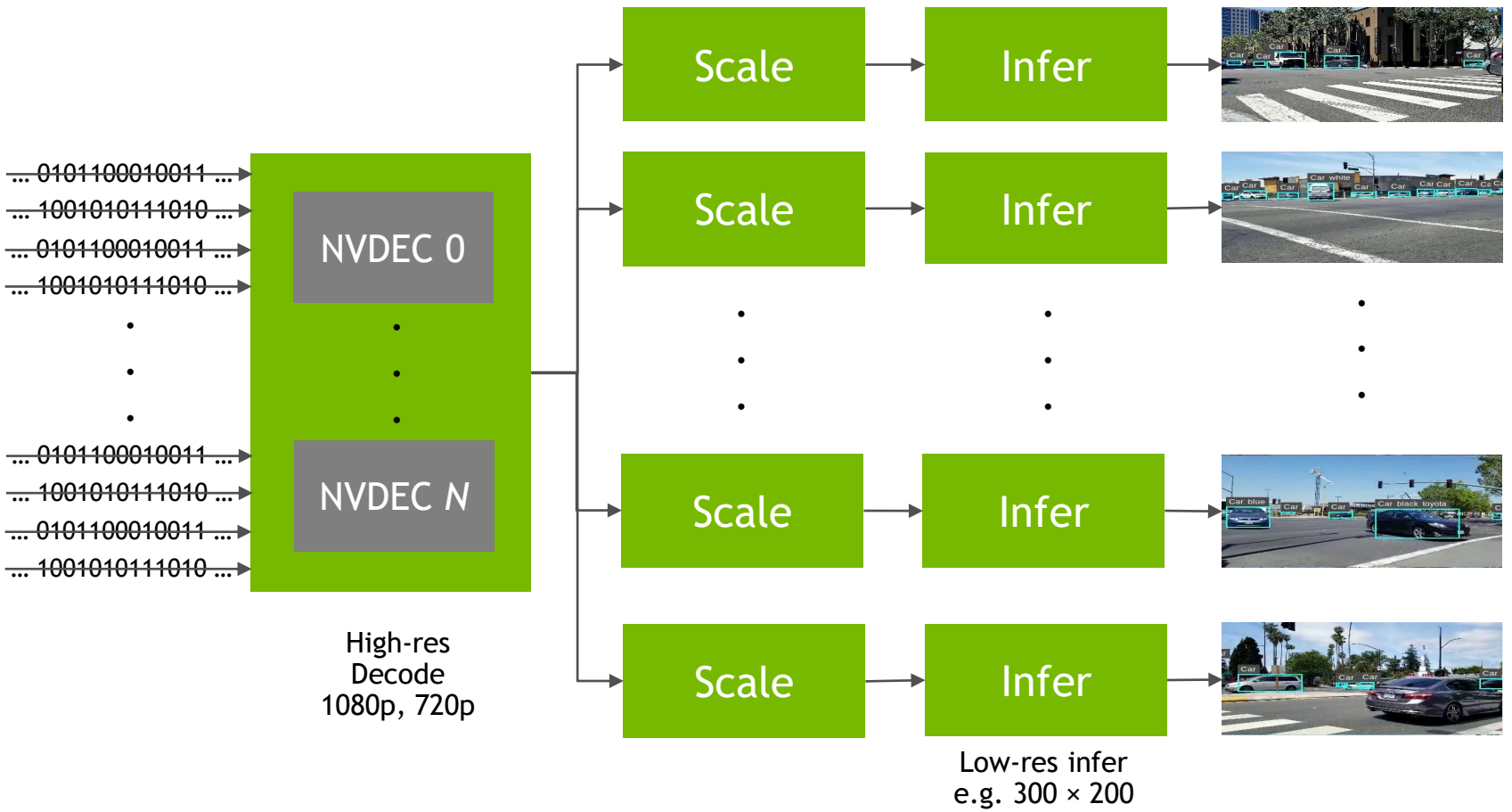
PASCAL & EARLIER

Single NVDEC



TURING

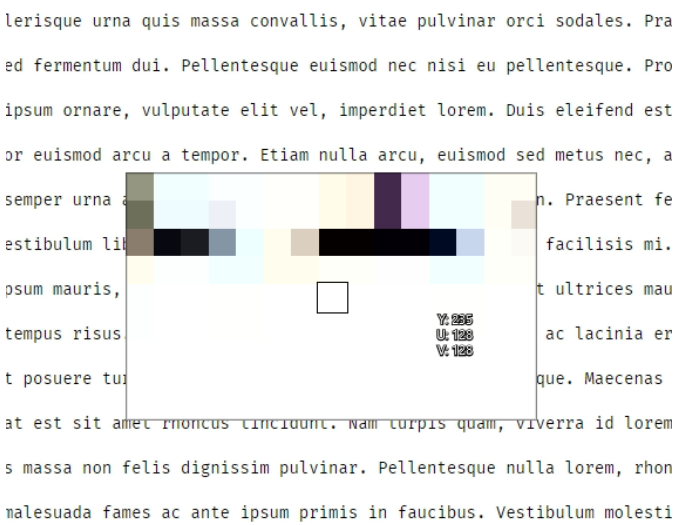
Multiple NVDECs



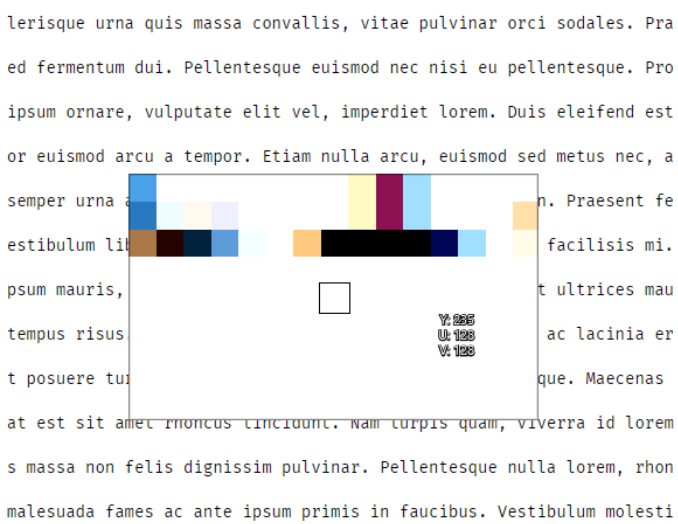
END-TO-END 4:4:4 IN TURING

- Preserves chroma: text and thin lines
- Valuable in desktop streaming

4:2:0



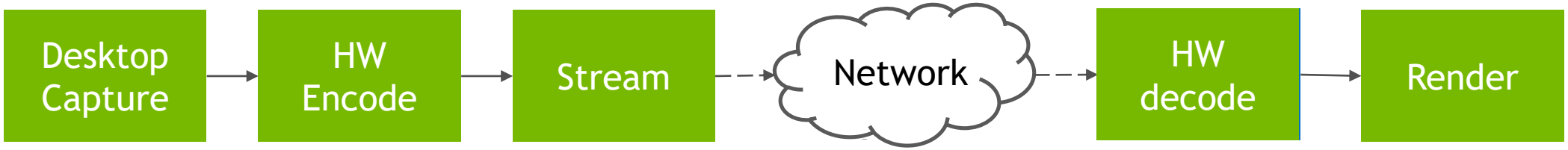
4:4:4



END-TO-END 4:4:4 IN TURING

HEVC 4:4:4 HW encode & 4:4:4 HW decode

Pascal & earlier



TURING NVENC ENHANCEMENTS

NVENC - ENCODING QUALITY

Focus for Turing NVENC

Enhancement	How to use
Rate distortion optimization – RDO	Turing only – always ON
Multiple reference frames	Preset-dependent
HEVC B-frames	NVENCODE API
Others	

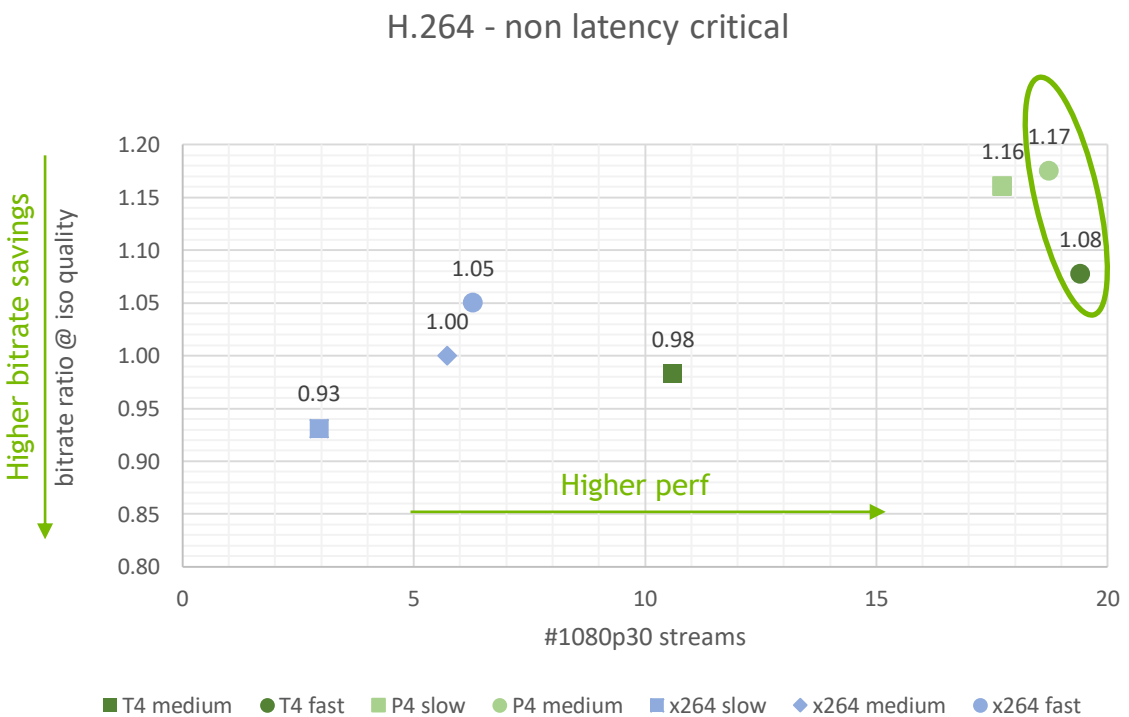
- Higher throughput at same quality as Pascal
- Turing GPUs have single NVENC engine with higher quality

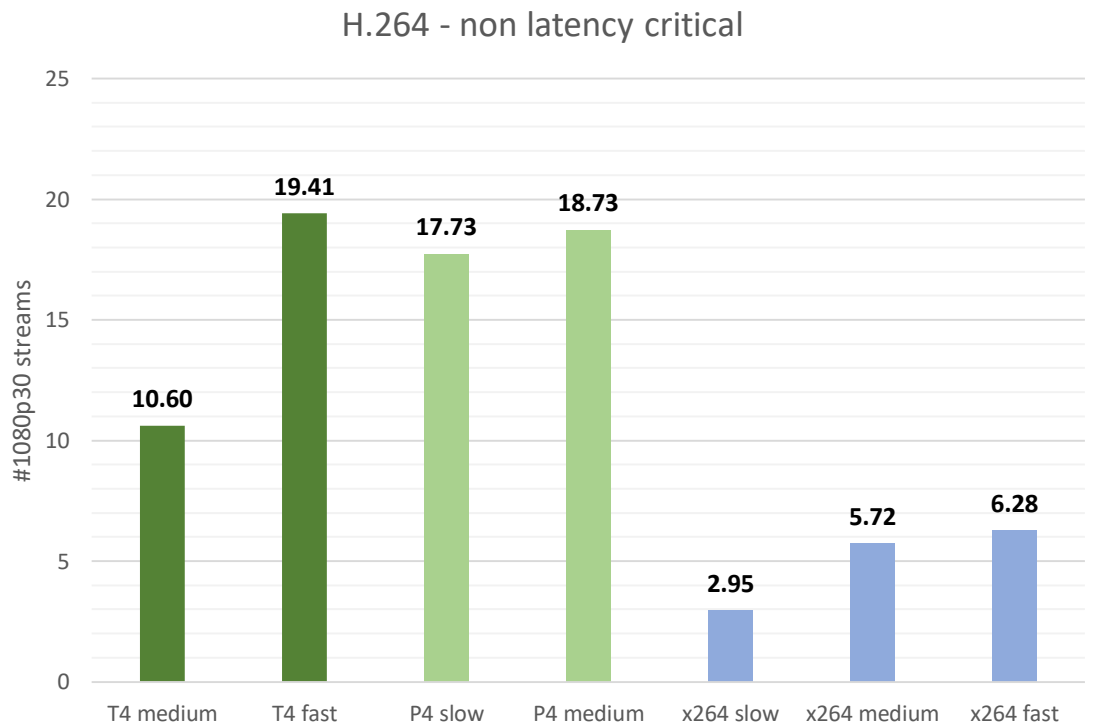
TURING NVENC QUALITY

- Focus on quality – RDO, multi-ref, HEVC B-frames, …
- Quality vs performance trade-off
- Quality is content dependent
- 600+ videos of 10-20 secs each: Natural, animation, gaming, video conference, movies
- 720p, 1080p, 4K, 8K
- Quality: PSNR, SSIM, VMAF, subjective
- Perf: fps, number of 1080p streams per GPU

H.264 ENCODE BENCHMARK

Non latency critical – Turing vs Pascal vs x264





"iso" quality = x264 medium

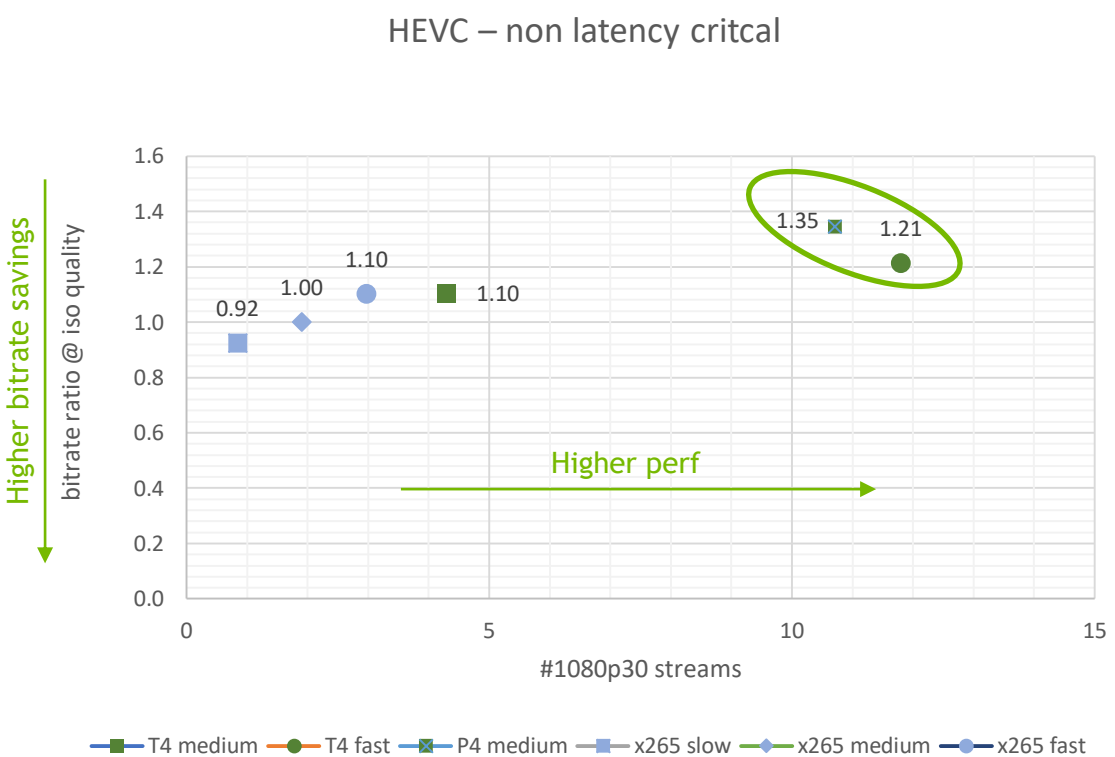
H.264 ENCODE BENCHMARK

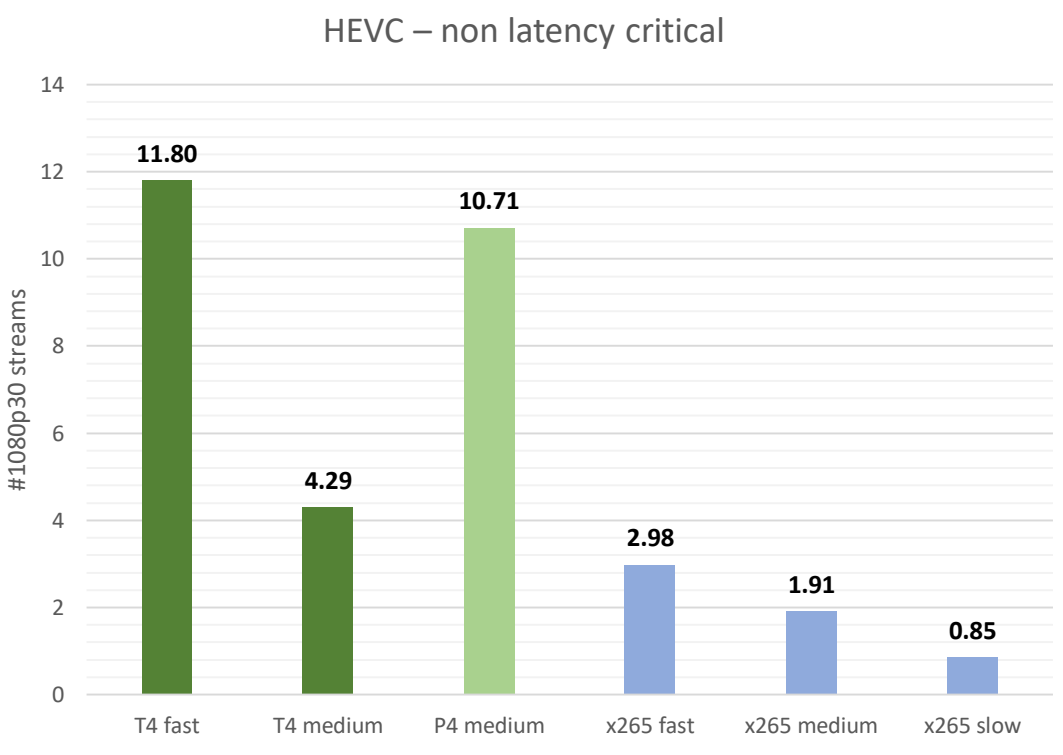
Non latency critical – FFmpeg commands

NVENC slow	-preset slow -bufsize BITRATE*2 -maxrate BITRATE*1.5 -profile:v high -bf 3 -b_ref_mode 2 -temporal-aq 1 -rc-lookahead 20 -vsync 0
x264 slow	-preset slow -tune psnr -vsync 0 -threads 4 -vsync 0
NVENC medium	-preset medium -rc vbr -profile:v high -bf 3 -b_ref_mode 2 -temporal-aq 1 -rc-lookahead 20 -vsync 0
x264 medium	-preset medium -tune psnr -threads 4 -vsync 0
NVENC fast	-preset fast -rc vbr -profile:v high -bf 3 -b_ref_mode 2 -temporal-aq 1 -rc-lookahead 20 -vsync 0
x264 fast	-preset fast -tune psnr -vsync 0 -threads 4 -vsync 0

HEVC ENCODE BENCHMARK

Non latency critical – Turing vs Pascal vs x265





"iso" quality = x265 medium

HEVC ENCODE BENCHMARK

Non latency critical – FFmpeg commands

NVENC slow	-preset slow -rc vbr_hq -b:v BITRATE -profile:v 4 -bf 2 -rc-lookahead 20 -g 250 -vsync 0
x265 slow	-preset slow -b:v BITRATE -bf 2 -tune psnr -threads 4 -vsync 0
NVENC medium	-preset medium -rc vbr_hq -b:v BITRATE -profile:v 4 -bf 2 -rc-lookahead 20 -g 250 -vsync 0
x265 medium	-preset medium -b:v BITRATE -bf 2 -tune psnr -threads 4 -vsync 0
NVENC fast	-preset fast -rc vbr_hq -b:v BITRATE -profile:v 4 -bf 2 -temporal-aq 1 -rc-lookahead 20 -g 250 -vsync 0
x265 fast	-preset fast -b:v BITRATE -bf 2 -tune psnr -threads 4 -vsync 0

SOFTWARE UPDATES

RECONFIGURE DECODER

Video Codec SDK 8.2

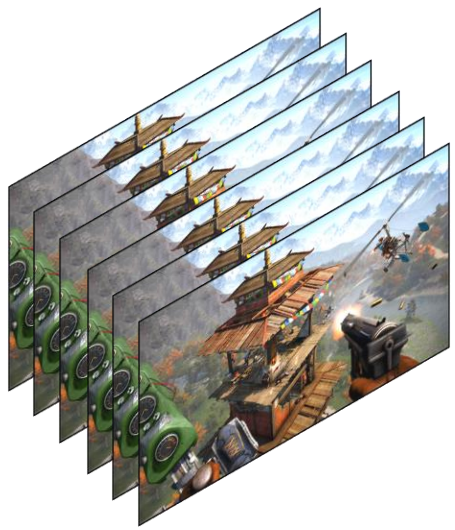
No init time, reuse context, lowers memory fragmentation

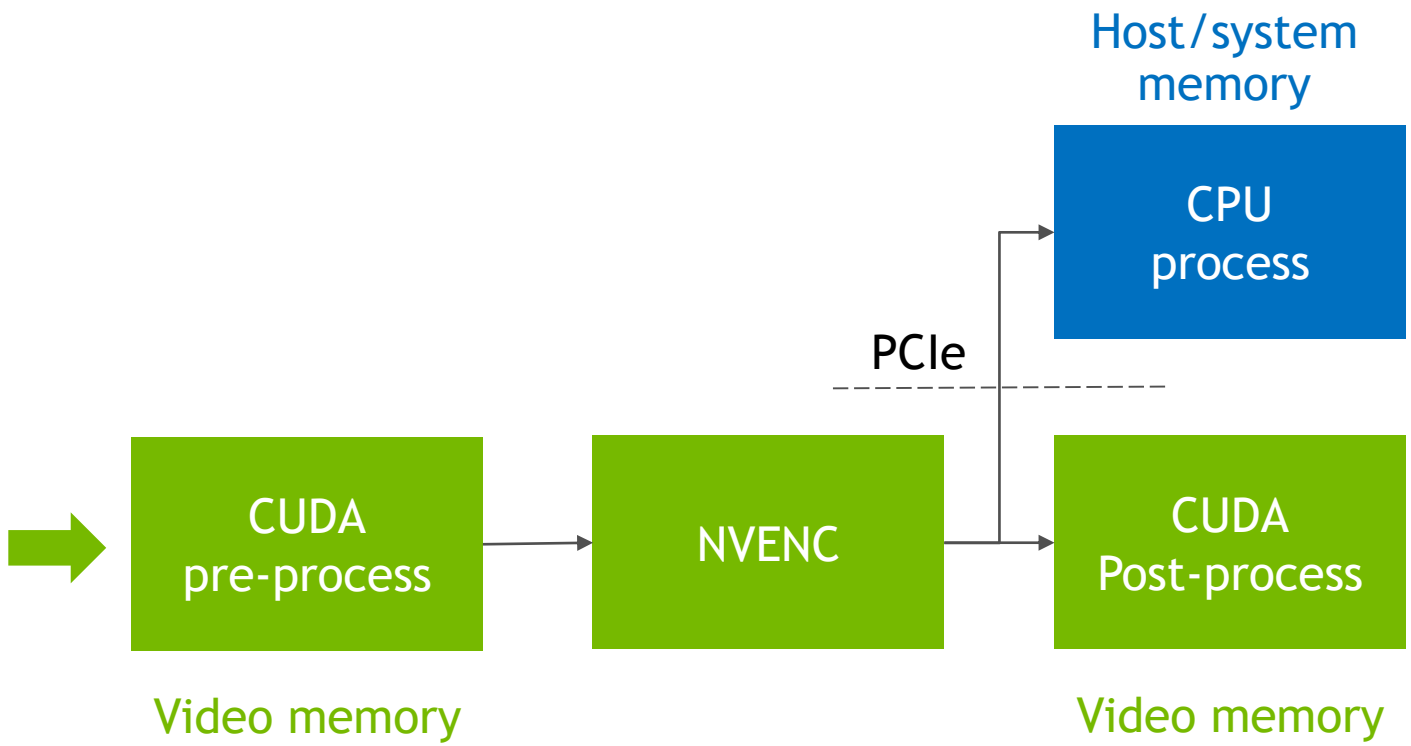
- ✓ Input resolution
- ✓ Scaling resolution
- ✓ Cropping rectangle
- ✗ Codecs
- ✗ Bit-depth and chroma format
- ✗ Deinterlace mode
- ✗ Input resolution beyond max width or max height

DIRECT OUTPUT TO VIDMEM

Video Codec SDK 9.0

SDK 8.0 & earlier





OTHER UPDATES

- Video Codec SDK now supported on Power 9 + Tesla V100 SXM2
- High-level NVDEC error status

OPTICAL FLOW

New HW Functionality

- 4 × 4 optical flow vector, up to 4K × 4K
- Close to true motion
- Robust to intensity changes
- 10x faster than CPU; same quality
- New Optical Flow SDK
- Action recognition, object tracking, video inter/extrapolation, frame-rate upconversion
- Legacy ME-only mode support

More information: http://developer.nvidia.com/opticalflow-sdk

TIPS FOR NVENC OPTIMIZATION

OPTIMIZATION STRATEGIES

General Guidelines

- Minimize PCIe transfers
  - Eliminate, if possible
  - Use CUDA for video pre-/post-processing
- Multiple threads/processes to balance enc/dec utilization
  - Monitor using nvidia-smi: `nvidia-smi dmon -s uc -i <GPU_index>`
  - Analyze using GPUView on Windows
- Minimize disk I/O
- Optimize encoder settings for quality/perf balance

FFMPEG VIDEO TRANSCODING

Tips

- Look at FFmpeg users' guide in NVIDIA Video Codec SDK package
- Use `-hwaccel` keyword to keep entire transcode pipeline on GPU
- Run multiple 1:*N* transcode sessions to achieve *M*:*N* transcode at high perf

LOW LATENCY STREAMING (1/3)

Optimization tips

- Low latency ≠ Low *encoding* time
- Latency determined by
  - B-frames
  - Look-ahead
  - VBV buffer size & avlbl bandwidth

LOW LATENCY STREAMING (2/3)

Optimization tips

- For 1-2 frame latency (e.g. cloud gaming), use
  - RC_CBR_LOWDELAY_HQ & Low VBV buffer size
    - Minimizes frame-to-frame variations
  - Any preset (Default, HQ, HP preferred)
    - LL presets have resolution-dependent behavior
  - No look-ahead
  - No B-frames

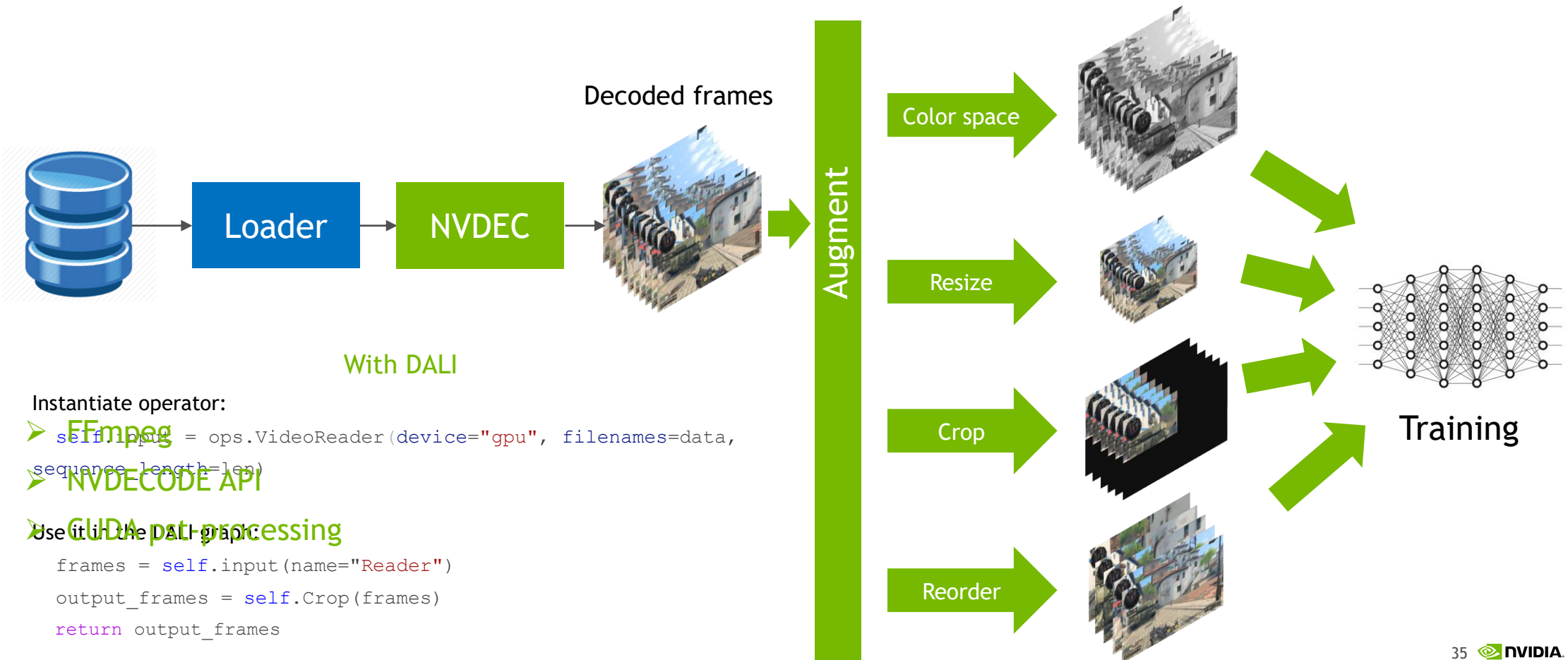
LOW LATENCY STREAMING (3/3)

Optimization tips

- Similar to HQ (non latency critical) encoding
- For higher (8-10 frames) latency (e.g. OTT, broadcast), use
  - Any RC mode
  - Any preset (default, HQ, HP preferred)
  - VBV buffer size as per channel bandwidth constraints
  - Look-ahead depth < tolerable latency
  - B-frames as needed

VIDEO DL TRAINING

Typical Workflow



With DALI

Instantiate operator:

```
self.input = ops.VideoReader(device="gpu", filenames=data,
sequence_length=len)
```

Use it in the DALI graph:

```
frames = self.input(name="Reader")
output_frames = self.Crop(frames)
return output_frames
```

- FFmpeg
- NVDECODE API
- CUDA post-processing

ROADMAP

ROADMAP

Video Codec SDK 9.1

- Q3 2018
- Error handling – Retrieve last error
- Perf/quality tuning
- Support for CUStream

RESOURCES

Video Codec SDK: https://developer.nvidia.com/nvidia-video-codec-sdk

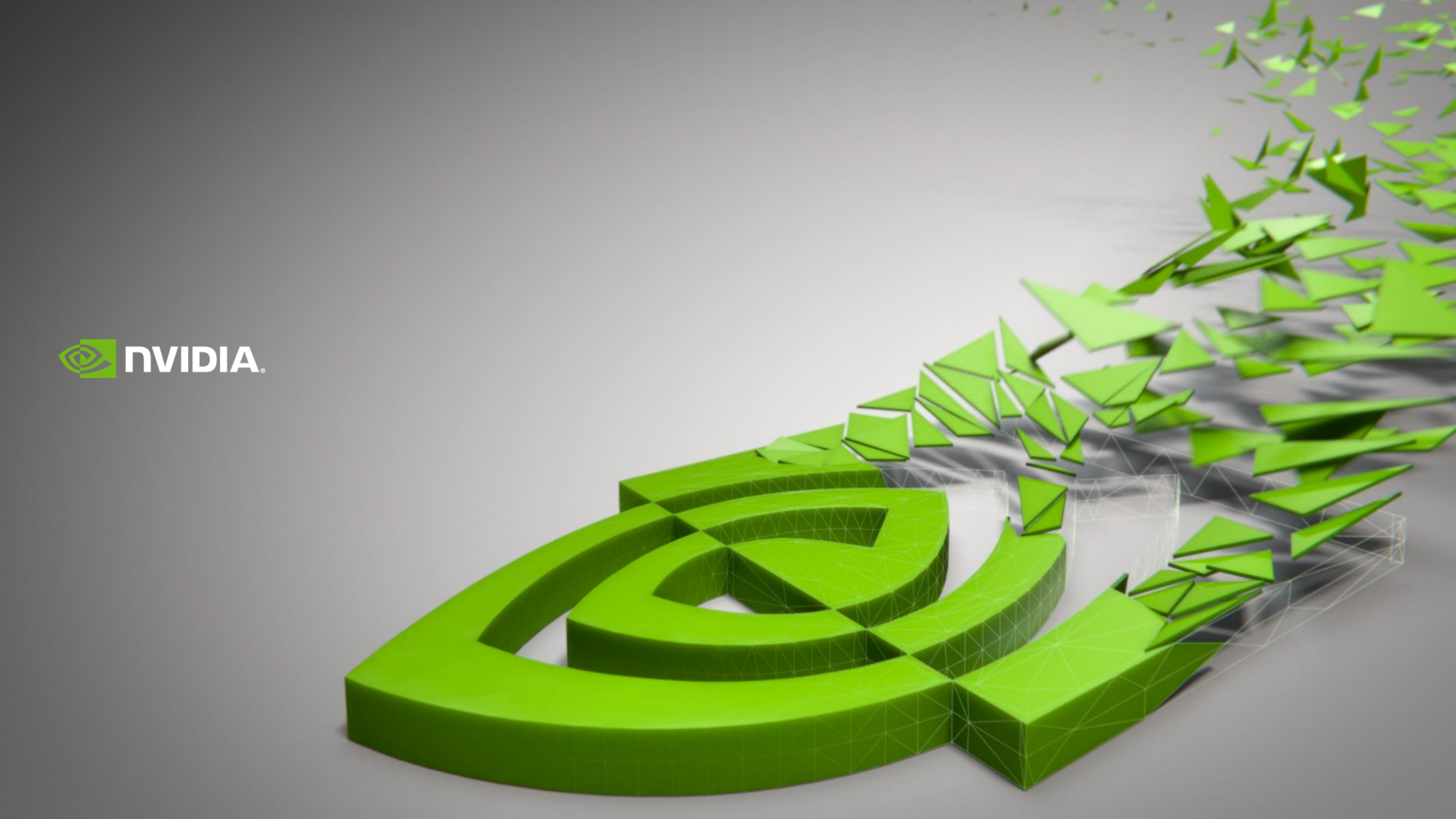
FFmpeg GIT: https://git.ffmpeg.org/ffmpeg.git

FFmpeg builds with hardware acceleration: http://ffmpeg.zeranoe.com/builds/

Video SDK support: video-devtech-support@nvidia.com

Video SDK forums: https://devtalk.nvidia.com/default/board/175/video-technologies/

Connect with Experts (CE9103): Wednesday, March 20, 2019, 3:00 pm


NVIDIA