NVIDIA VIDEO CODEC SDK APPLICATION NOTE - ENCODER

NVENC_Application_Note| Jan 2018

DOCUMENT CHANGE HISTORY

NVENC_Application_Note

Version	Date	Authors	Description of Change
01	Jan 30,2012	AP/CC	Initial release
02	Sept 24, 2012	AP	Update for NVENC SDK 2.0
03	April 10, 2013	AP	Update for Monterey SDK 2.0.0 update
04	Aug 4, 2013	AP	Update for NVENC SDK 3.0
05	June 17, 2014	SM/AP	Update for NVENC SDK 4.0
06	Nov 14, 2014	SM	Update for NVENC SDK 5.0
07	Oct 10, 2015	SM	Update for Video Codec SDK 6.0
08	June 10, 2016	SM	Update for Video Codec SDK 7.0
09	Nov 15, 2016	SM	Update for Video Codec SDK 7.1
10	Apr 11, 2017	SM/AP	Update for Video Codec SDK 8.0
11	Jan 10, 2018	SM	Update for Video Codec SDK 8.1

TABLE OF CONTENTS

LIST OF TABLES

NVIDIA HARDWARE VIDEO ENCODER

1. INTRODUCTION

NVIDIA GPUs - beginning with the Kepler generation - contain a hardware-based encoder (referred to as NVENC in this document) which provides fully-accelerated hardware-based video encoding and is independent of graphics/CUDA cores. With end-to-end encoding offloaded to NVENC, the graphics/CUDA cores and the CPU cores are free for other operations. For example, in a game recording scenario, encoding being completely offloaded to NVENC makes the graphics engine fully available for game rendering. In video transcoding use-case, video encoding/decoding can happen on NVENC/NVDEC in parallel with other video post-/pre-processing on CUDA cores.

The hardware capabilities available in NVENC are exposed through APIs herein referred to as NVENCODE APIs in the document. This document provides information about the capabilities of the hardware encoder and features exposed through NVENCODE APIs.

2. NVENC CAPABILITIES

NVENC can perform end-to-end encoding for H.264, HEVC 8-bit and HEVC 10-bit. This includes, motion estimation and mode decision, motion compensation and residual coding, and entropy coding. It can also be used to generate motion vectors between two frames, which are useful for applications such as depth estimation, frame interpolation or encoding using other codecs not supported by NVENC. These operations are hardware accelerated by a dedicated block on GPU silicon die. NVENCODE APIs provide the necessary knobs to utilize the hardware encoding capabilities.

Table 1 summarizes the capabilities of the NVENC hardware exposed through NVENCODE APIs. Table 2 and Table 3 summarize new NVENCODE API features available in Video SDK 8.0 and Video SDK 8.1 respectively.

Table 1. NVENC hardware capabilities

Feature	Description	Kepler GPUs	1st Gen Maxwell GPUs	2nd Gen Maxwell GPUs	Pascal GPUs	Volta GPUs
H.264 baseline, main and high profiles	Capability to encode YUV 4:2:0 sequence and generate a H.264-bit stream.	✓	✓	✓	✓	✓
H.264 4:4:4 encoding(only CAVLC)	Capability to encode YUV 4:4:4 sequence and generate a H.264-bit stream.	×	✓	✓	✓	✓
H.264 lossless encoding	Lossless encoding.	×	✓	✓	✓	✓
H.264 motion estimation (ME) only mode	Capability to provide macro-block level motion vectors and intra/inter modes.	×	✓	✓	✓	✓
H.264/HEVC weighted prediction	Support for weighted prediction.	×	×	×	✓	✓
Encoding support for H.264 ARGB content	Capability to encode RGB input.	✓	✓	✓	✓	✓
HEVC main profile	Capability to encode YUV 4:2:0 sequence and generate a HEVC bit stream.	×	×	✓	✓	✓
HEVC main10 profile	Support for encoding 10-bit content generate a HEVC bit stream.	×	×	×	✓	✓
HEVC lossless encoding	Lossless encoding.	×	×	×	✓	✓

Feature	Description	Kepler GPUs	First generation Maxwell GPUs	Second generation Maxwell GPUs	Pascal GPUs	Volta GPUs
HEVC 4:4:4 encoding	Capability to encode YUV 4:4:4 sequence and generate a HEVC bit stream.	×	×	×	✓	✓
HEVC motion estimation (ME) only mode	Capability to provide CTB level motion vectors and intra/inter modes.	×	×	×	✓	✓
HEVC 8K encoding*	Support for encoding 8192 × 8192 Content.	×	×	×	✓	✓
HEVC sample adaptive offset(SAO)	Improves encoded video quality.	×	×	×	✓	✓

*: Supported in select Pascal generation GPUs and all Volta GPUs

Table 2. What's new in SDK 8.0

Feature	Description
Encoding support for Open GL surfaces	NVENCODE API can accept Open GL surfaces as input directly. The support is availabe only on Linux.
Improved quality for HEVC spatial adaptive quantization	Encoded quality for HEVC spatial adaptive quantization is improved.
Weighted prediction (WP)	WP support is added for H.264, HEVC and HEVC main10. The feature can be enabled through a flag exposed in the NVENCODE API. WP gives significant quality improvement for contents having illumination changes.
External motion hints for H.264 motion estimation (ME) only mode	User can pass external motion hints for H.264 ME only mode.
Fractional constant quality (CQ) support	User can specify fractional values for CQ rate control mode

Table 3. What's new in SDK 8.1

Feature	Description
B frame as reference	This feature enables the user to mark B frame to be used as reference during encoding. This results to improved encoding quality when B frames are used.
Emphasis Map	Using this feature the user can specify regions of the image to be emphasized (by using relatively lower QP in those regions). This emphasis is implemented by reducing QP in emphasized regions. Emphasis map is useful in situations when client has prior knowledge of the image complexity (e.g. NVFBC's Classification Map feature in Capture SDK 7.0) and encoding those high-complexity areas at higher quality (lower QP) is important, even at the possible cost of violating bitrate/VBV buffer size constraints. This feature is currently supported for H.264 only.
Querying residual encoder capacity	This feature enables the user to query the residual encoding capacity of the hardware encoder as a percentage of the overall encoding capacity present on the GPU. This is supported only for windows OS on virtualized GPUs. On bare metal (non-virtualized GPU) and linux platforms, this API always returns 100.
Sample applications developed on re-usable classes.	New sample applications built on classes which abstract the functionalities exposed thorough the NVENCODEAPI have been added in the SDK package. The clients can re-use the classes to develop their own applications. Most of the programming of the encoder is done inside the base classes which makes NVENCODEAPI simple and easy to use.

3. NVENC LICENSING POLICY

There is no change in licensing policy in the current SDK in comparison to the earlier SDKs. The licensing policy is as follows:

As far as NVENC hardware encoding is concerned, NVIDIA GPUs are classified into two categories: "qualified" and "non-qualified". On qualified GPUs, the number of concurrent encode sessions is limited by available system resources (encoder capacity, system memory, video memory etc.). On non-qualified GPUs, the number of concurrent encode sessions is limited to 2 per system. This limit of 2 concurrent sessions per system applies to the combined number of encoding sessions executed on all non-qualified cards present in the system.

For a complete list of qualified and non-qualified GPUs, refer to https://developer.nvidia.com/nvidia-video-codec-sdk.

For example, on a system with one Quadro K4000 card (which is a qualified GPU) and three GeForce cards (which are non-qualified GPUs), the application can run *N* simultaneous encode sessions on Quadro K4000 card (where *N* is defined by the encoder/memory/hardware limitations) and two sessions on all the three GeForce cards combined. Thus, the limit on the number of simultaneous encode sessions for such a system is *N* + 2.

4. NVENC PERFORMANCE

With every generation of NVIDIA GPUs (Kepler, Maxwell 1st/2nd gen, Pascal, Volta), NVENC performance has increased steadily. Table 4 provides *indicative*[1] NVENC performance on Kepler, Maxwell and Pascal for different presets and rate control modes (these two factors play major role in determining the performance and quality). Note that performance numbers in Table 4 are measured on GeForce hardware with assumptions listed under the table. The performance varies across GPU classes (e.g. Quadro, Tesla), and scales (almost) linearly with the clock speeds for each hardware.

While Kepler and first-generation Maxwell GPUs had one NVENC engine per chip, certain variants of the second-generation Maxwell, Pascal and Volta GPUs have two/three NVENC engines per chip. This increases the aggregate encoder performance of the GPU. NVIDIA driver takes care of load balancing among multiple NVENC engines on the chip, so that applications don't require any special code to take advantage of multiple encoders and automatically benefit from higher encoder capacity on higher-end GPU hardware. The encode performance listed in Table 4 is given *per NVENC engine*. Thus, if the GPU has 2 NVENCs (e.g. GP104, GM204), multiply the corresponding number in Table 4 by the number of NVENCs per chip to get aggregate maximum performance (applicable only when running multiple simultaneous encode sessions). Note that performance with single encoding session cannot exceed performance per NVENC, regardless of the number of NVENCs present on the GPU.

NVENC hardware natively supports multiple hardware encoding contexts with negligible context-switching penalty. As a result, subject to the hardware performance limit and available memory, an application can encode multiple videos simultaneously. NVENCODE API exposes several presets, rate control modes and other parameters for

[1] Encoder performance depends on many factors, including but not limited to: Encoder settings, GPU clocks, GPU type, video content type etc.

programming the hardware. A combination of these parameters enables video encoding at varying quality and performance levels. In general, one can trade performance for quality and vice versa.

Table 4. NVENC encoding performance

		H.264 (FPS)			HEVC (FPS)	
Preset	**RC Mode***	**Kepler** 540 MHz	**Second Gen. Maxwell** 1366 MHz	**Pascal** 1911 MHz	**Second Gen. Maxwell** 1366 MHz	**Pascal** 1911 MHz
High Performance	Constant QP	220	485	691	231	421
	Single Pass	217	479	663	223	401
	Dual Pass	112	335	531	174	340
High Quality	Constant QP	80	290	376	174	293
	Single Pass	76	278	370	157	250
	Dual Pass*	min 53	177	236	98	167
		max 53	250	390	132	230
Low latency High Performance	Constant QP	137	410	554	231	421
	Single Pass	134	335	557	223	401
	Dual Pass	78	270	429	175	340
Low latency High Quality	Constant QP	80	288	373	216	422
	Single Pass	77	280	367	217	401
	Dual Pass*	min 54	181	362	121	217
		max 54	259	403	174	340
Lossless			353	478		264

- Resolution/Input Format/Bit depth: 1920 × 1080/YUV 4:2:0/8-bit
- Hardware: Various GeForce GPU hardware with clocks held at P0, Intel Core i7-6700 CPU @ 3.40 GHz,
- GPU Clocks: GPU core clock reported by GPU-Z, as specified in the table
- Software: Windows 10, Video Codec SDK 8.1, NVIDIA display driver: 390.25
- Dual pass performance varies depending upon other settings such as look-ahead, B-frames, VBV buffer size etc. Hence max and min performance is specified
- The encoding performance on Volta GPUs scales up with the performance numbers on Pascal GPUs in proportion to the GPU core clocks.

5. PROGRAMMING NVENC

Video Codec SDK 8.0 and Video Codec SDK 8.1 are supported on R378 and R390 drivers and above respectively. Please refer to the SDK release notes for information regarding the required driver version.

Please refer to the documents and the sample applications included in the SDK package for details on how to program NVENC.

6. FFMPEG AND LIBAV SUPPORT

FFmpeg and Libav are the most popular multimedia transcoding tools used extensively for video and audio transcoding.

The video hardware accelerators in NVIDIA GPUs can be effectively used with FFmpeg and Libav to significantly speed up the video decoding, encoding and end-to-end transcoding at very high performance.

Note that FFmpeg and Libav are open-source projects and their usage is governed by specific licenses and terms and conditions for each of these projects.

Notice

Trademarks

Copyright