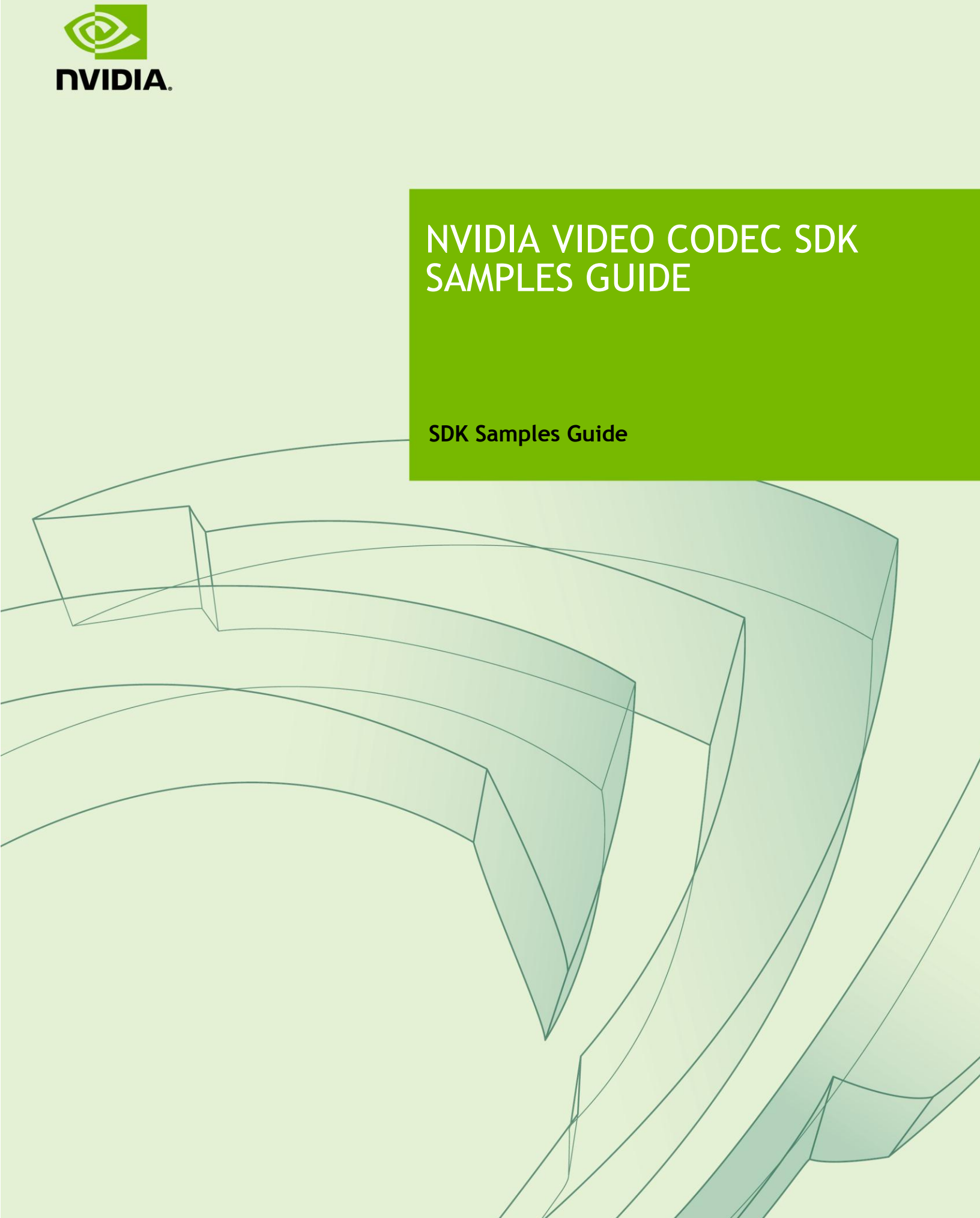

NVIDIA VIDEO CODEC SDK SAMPLES GUIDE

SDK Samples Guide

REVISION HISTORY

Revision	Date	Author	Description
1.0	Nov 14, 2014	YG	Initial release.
2.0	Nov 25, 2015	EY	Update to NVIDIA Video Codec SDK 6.0 Added NVCUVID decode samples
3.0	June 10, 2016	SM/VU/GJ	Update to NVIDIA Video Codec SDK 7.0
4.0	Feb 15, 2017	SM/SKR/GJ	Update to NVIDIA Video Codec SDK 8.0

TABLE OF CONTENTS

NVIDIA VIDEO CODEC SDK SAMPLES GUIDE

INTRODUCTION

NVIDIA Video Codec SDK contains the following samples. The sample applications provided in the package are for demonstration purposes only and may not be fully tuned for Quality and/or Performance. Hence the users are advised to do their independent evaluation for Quality and/or performance.

NvEncoder

This sample demonstrates the usage of basic encoding functionality.

NvEncoderCudaInterop

This sample demonstrates the usage of encoding with CUDA surfaces.

NvEncoderD3DInterop

This sample demonstrates the usage of encoding with D3D9 surfaces.

NvEncoderGLInterop

This sample demonstrates the usage of encoding with OpenGL surfaces.

NvEncoderLowLatency

This sample demonstrates the usage of low latency features such as Intra Refresh and Reference Picture Invalidations.

NvEncoderPerf

This sample demonstrates the maximum achieved encoding performance.

NvTranscoder

This sample demonstrates the transcoding capabilities of NVENC.

NvDecodeD3D9

This sample demonstrates video decode with D3D9 visualization.

NvDecodeD3D11

This sample demonstrates video decode with D3D11 visualization.

NvDecodeGL

This sample demonstrates video decode and OpenGL visualization.

1. BUILDING SAMPLES

Windows

The Windows SDK samples are built using the Visual Studio IDE. Solution files (*.sln) are provided for Visual Studio 2010 and 2013.

Complete samples solution files exist at:

SDK\Samples

Each individual sample has its own set of solution files at:

SDK\Samples\<sample_dir>

To build/examine all the samples at once, the complete solution files should be used. To build/examine a single sample, the individual sample solution files should be used

Linux

The Linux samples are built using makefiles. To use the makefiles, change the current directory to the sample directory you wish to build, and run make:

```
$ cd <sample_dir>
$ make
```

To build/examine all the samples at once, you can use the top-level Makefile in the **Samples** directory.

2. SAMPLES REFERENCE

2.1 NvEncoder

The NvEncoder application demonstrates the code for doing a basic encoding using NVENC. It supports both H.264 and HEVC encoding with different presets. The application allows to configure bitrate, frame rate, number of B frames and allows the user to select from the given Rate Control Modes.

The following are the options that may be specified for NvEncoder Application.

```
-i <string> : Specifies the input YUV File that has to be encoded
-o <string> : Specifies the output bitstream file
-size <integer integer> : Specifies the input resolution width and height

-codec <integer>: Specifies the codec (0) – H264 and (1) - HEVC
-preset <string> :  hq – High Quality Preset
                    hp – High Performance Preset
                    lowLatencyHP – Low Latency High Performance Preset
                    lowLatencyHQ – Low Latency High Quality Preset
                    lossless - Lossless High Quality Preset
-startf <integer> : Specifies the starting frame Index for encoding.
Default value is zero
-endf < integer> : Specifies the end frame Index for encoding. Default
value is zero
-fps <integer> : Specifies the encoding frame rate
-gopLength <integer> : Specifies the GOP (Group of Pictures) Length
-numB <integer> : Specifies the number of B frames
-bitrate <integer> : Specifies the encoding average bitrate
-vbvMaxBitrate <integer> : Specifies the VBV Maximum Bitrate
-vbvSize <integer> : Specifies the Encoding VBV/HRD Buffer Size
-rcmode <integer> : Specifies the Rate Control Mode.
                     0 : Constant QP mode
                     1 : Variable bitrate mode
                     2 : Constant bitrate mode
                     8 : low-delay CBR, high quality
                    16 : CBR, high quality (slower)
                    32 : VBR, high quality (slower)

-qp <integer> : Specifies the qp value for Constant QP Rate Control Mode
-i_qfactor <float> : Specify qscale difference between I-frames and P-
frames
-b_qfactor <float> : Specify qscale difference between P-frames and B-
frames
-i_qoffset <float> : Specify qscale offset between I-frames and P-frames
-b_qoffset <float> : Specify qscale offset between P-frames and B-frames
```

-picStruct <integer> : Specify the picture structure
1: Progressive frame
2: Field encoding top field first
3: Field encoding bottom field first
-devicetype <integer> : 0 – DX9 Device Type
1 – DX10 Device Type
2 – DX11 Device Type
3 – CUDA Device Type
-inputFormat <integer> : Specify the input format
0: YUV 420
1: YUV 444
2: YUV 420 10-bit
3: YUV 444 10-bit
-deviceID <integer> : Specify the GPU device on which encoding will take place
-meonly <integer> : Specify Motion estimation only(permissive value 1 and 2 to generates motion vectors and Mode information
1: Motion estimation between startf and endf
2: Motion estimation for all consecutive frames from startf to endf
-preloadedFrameCount <integer> : Specify number of frame to load in memory(default value=240) with min value 2(1 frame for ref, 1 frame for input)
-temporalAQ : Enable TemporalAQ
-generateQpDeltaMap <string> : Demonstrate QP delta map, and use opposite delta values for 1,3 quadrants and 2,4 quadrants for each frame. Also, save the delta map array in the specified file.
-help : Prints Help Information

2.2 NvEncoderCudaInterop

The NvEncoderCudaInterop application demonstrates the interoperability of the NVENC hardware encoder with CUDA surfaces.

The following are the options that may be specified for NvEncoderCudaInterop Application.

-i <string> : Specifies the input YUV File that has to be encoded
-o <string> : Specifies the output bitstream file
-size <integer integer> : Specifies the input resolution width and height

-startf <integer> : Specifies the starting frame Index for encoding. Default value is zero
-endf < integer> : Specifies the end frame Index for encoding. Default value is zero
-codec <integer>: Specifies the codec (0) – H264 and (1) - HEVC

```
-preset <string> :  hq - High Quality Preset
                    hp - High Performance Preset
                    lowLatencyHP - Low Latency High Performance Preset
                    lowLatencyHQ - Low Latency High Quality Preset
                    lossless - Lossless High Quality Preset
-fps <integer> : Specifies the encoding frame rate
-gopLength <integer> : Specifies the GOP (Group of Pictures) Length
-numB <integer> : Specifies the number of B frames
-bitrate <integer> : Specifies the encoding average bitrate
-vbvMaxBitrate <integer> : Specifies the VBV Maximum Bitrate
-vbvSize <integer> : Specifies the Encoding VBV/HRD Buffer Size
-rcmode <integer> : Specifies the Rate Control Mode.
                     0 : Constant QP mode
                     1 : Variable bitrate mode
                     2 : Constant bitrate mode
                     8 : low-delay CBR, high quality
                    16 : CBR, high quality (slower)
                    32 : VBR, high quality (slower)
-qp <integer> : Specifies the qp value for Constant QP Rate Control Mode
-i_qfactor <float> : Specify qscale difference between I-frames and P-
frames
-b_qfactor <float> : Specify qscale difference between P-frames and B-
frames
-i_qoffset <float> : Specify qscale offset between I-frames and P-frames
-b_qoffset <float> : Specify qscale offset between P-frames and B-frames
-deviceID <integer> : Specifies the GPU Device on which encoding will take
place
-help : Prints Help Information
```

2.3 NvEncoderD3DInterop

The NvEncoderD3DInterop application shows the interoperability with DX Surfaces. This application takes a directory of BMP files as an input and generates the output encoded file.

The following are the options that may be specified for NvEncoderD3DInterop Application.

```
-bmpfilePath <string> : Specifies the input RGB BMP file path
-o <string> : Specifies the output bitstream file

-size <integer integer> : Specifies the input resolution width and height
-codec <integer>: Specifies the codec (0) - H264 and (1) - HEVC

-preset <string> :  hq - High Quality Preset
                    hp - High Performance Preset
```

```
                    lowLatencyHP - Low Latency High Performance Preset
                    lowLatencyHQ - Low Latency High Quality Preset
                    lossless - Lossless High Quality Preset

-fps <integer> : Specifies the encoding frame rate
-gopLength <integer> : Specifies the GOP (Group of Pictures) Length
-numB <integer> : Specifies the number of B frames
-bitrate <integer> : Specifies the encoding average bitrate
-vbvMaxBitrate <integer> : Specifies the VBV Maximum Bitrate
-vbvSize <integer> : Specifies the Encoding VBV/HRD Buffer Size
-rcmode <integer> : Specifies the Rate Control Mode.
                     0 : Constant QP mode
                     1 : Variable bitrate mode
                     2 : Constant bitrate mode
                     8 : low-delay CBR, high quality
                    16 : CBR, high quality (slower)
                    32 : VBR, high quality (slower)

-qp <integer> : Specifies the qp value for Constant QP Rate Control Mode
-i_qfactor <float> : Specify qscale difference between I-frames and P-
frames
-b_qfactor <float> : Specify qscale difference between P-frames and B-
frames
-i_qoffset <float> : Specify qscale offset between I-frames and P-frames
-b_qoffset <float> : Specify qscale offset between P-frames and B-frames
-deviceID <integer> : Specifies the GPU Device on which encoding will take
place
-help : Prints Help Information
```

2.4 NvEncoderGLInterop

The NvEncoderGLInterop application demonstrates the interoperability of the NVENC hardware encoder with OpenGL surfaces (textures).

Use of the OpenGL interop is supported only on Linux. The OpenGL interop operation depends on the connection to the X server. Set the DISPLAY environment variable to the X display to be used.

The following are the options that may be specified for NvEncoderGLInterop Application.

```
-i <string> : Specifies the input YUV File that has to be encoded
-o <string> : Specifies the output bitstream file
-size <integer integer> : Specifies the input resolution width and height
```

```
-startf <integer> : Specifies the starting frame Index for encoding.
Default value is zero
-endf < integer> : Specifies the end frame Index for encoding. Default
value is zero
-codec <integer>: Specifies the codec (0) – H264 and (1) - HEVC

-preset <string> :  hq – High Quality Preset
                    hp – High Performance Preset
                    lowLatencyHP – Low Latency High Performance Preset
                    lowLatencyHQ – Low Latency High Quality Preset
                    lossless - Lossless High Quality Preset
-fps <integer> : Specifies the encoding frame rate
-gopLength <integer> : Specifies the GOP (Group of Pictures) Length
-numB <integer> : Specifies the number of B frames
-bitrate <integer> : Specifies the encoding average bitrate
-vbvMaxBitrate <integer> : Specifies the VBV Maximum Bitrate
-vbvSize <integer> : Specifies the Encoding VBV/HRD Buffer Size
-rcmode <integer> : Specifies the Rate Control Mode.
                     0 : Constant QP mode
                     1 : Variable bitrate mode
                     2 : Constant bitrate mode
                     8 : low-delay CBR, high quality
                    16 : CBR, high quality (slower)
                    32 : VBR, high quality (slower)
-qp <integer> : Specifies the qp value for Constant QP Rate Control Mode
-i_qfactor <float> : Specifies the qscale difference between I-frames and
P-frames.
-b_qfactor <float> : Specifies the qscale difference between P-frames and
B-frames.
-i_qoffset <float> : Specifies the qscale offset between I-frames and P-
frames.
-i_qoffset <float> : Specifies the qscale difference between P-frames and
B-frames.
-help : Prints Help Information
```

2.5 NvEncoderLowLatency

The NVEncoderLowLatency application demonstrates the encoding for low latency streaming. The application shows the usage of features such as Intra Refresh and Reference Picture Invalidation, Dynamic Resolution Change and Dynamic Bitrate Change that are extremely useful in error prone streaming environments.

The following are the options that may be specified for NvEncoderLowLatency Application.

```
-i <string> : Specifies the input YUV File that has to be encoded
-o <string> : Specifies the output bitstream file
```

-size <integer integer> : Specifies the input resolution width and height
-maxSize <int int> : Specify maximum resolution <maxWidth maxHeight>
-startf <integer> : Specifies the starting frame Index for encoding. Default value is zero
-endf < integer> : Specifies the end frame Index for encoding. Default value is zero
-codec <integer>: Specifies the codec (0) – H264 and (1) - HEVC
-preset <string> : hq – High Quality Preset
hp – High Performance Preset
lowLatencyHP – Low Latency High Performance Preset
lowLatencyHQ – Low Latency High Quality Preset
lossless - Lossless High Quality Preset
-fps <integer> : Specifies the encoding frame rate
-bitrate <integer> : Specifies the encoding average bitrate
-vbvSize <integer> : Specifies the Encoding VBV/HRD Buffer Size
-rcmode <integer> : Specifies the Rate Control Mode.
0 : Constant QP mode
1 : Variable bitrate mode
2 : Constant bitrate mode
8 : low-delay CBR, high quality
16 : CBR, high quality (slower)
32 : VBR, high quality (slower)
-encCmdFile <string> : Specifies the name of the encode command file. The commands can be given in the following format.
<encode command> <frame number> <param0> < param1>…<param15>
The following commands can be given in the encode command file:
0 : Dynamic Resolution Command <param0 = new Width> <param1 = new Height>
1 : Dynamic Bitrate Change <param0 = new bitrate> <param1 = new vbv size>
2 : Force IDR Frame
3 : Force Intra Refresh <param0 = intra refresh duration>
4. Invalidate Refrence Frame <param 0 = ref frame 0> <param1 = ref frame 1>..<param15 = ref frame 15>
-qpDeltaMapFile <string>: Specify the file containing the external QP delta map
-intraRefresh <boolean> : Specifies if Intra Refresh is used during Encoding
-intraRefreshPeriod <integer> : Specifies the period for cyclic Intra Refresh
-intraRefreshDuration <boolean> : Specifies the number of frames over which intra refresh takes place
-deviceID <integer> : Specifies the GPU Device on which encoding will take place
-help : Prints Help Information

2.6 NvEncoderPerf

The NvEncoderPerf application demonstrates the maximum encoding performance that may be achieved using NVENC. The application buffers a large number of input frames to prevent Disk I/O from being a bottleneck. The execution may be constrained by the Video Memory available on a system. The MAX_FRAMES_TO_PRELOAD compile time variable determines the number of frames that are buffered and may be reduced on systems with constrained Video Memory. The performance numbers give an indication of the Encode Compute Power available on NVENC. Applications need to be pipelined/multi-threaded in order to achieve maximum encode performance in practice.

The following are the options that may be specified for NvEncoderPerf Application.

```
-i <string> : Specifies the input YUV File that has to be encoded
-o <string> : Specifies the output bitstream file
-size <integer integer> : Specifies the input resolution width and height
-codec <integer>: Specifies the codec (0) – H264 and (1) - HEVC
-preset <string> :  hq – High Quality Preset
                    hp – High Performance Preset
                    lowLatencyHP – Low Latency High Performance Preset
                    lowLatencyHQ – Low Latency High Quality Preset
                    lossless - Lossless High Quality Preset
-startf <integer> : Specifies the starting frame Index for encoding.
Default value is zero
-endf < integer> : Specifies the end frame Index for encoding. Default
value is zero
-fps <integer> : Specifies the encoding frame rate
-gopLength <integer> : Specifies the GOP (Group of Pictures) Length
-numB <integer> : Specifies the number of B frames
-bitrate <integer> : Specifies the encoding average bitrate
-vbvMaxBitrate <integer> : Specifies the VBV Maximum Bitrate
-vbvSize <integer> : Specify the encoding vbv/hrd buffer size
-rcmode <integer> : Specifies the Rate Control Mode.
                     0 : Constant QP mode
                     1 : Variable bitrate mode
                     2 : Constant bitrate mode
                     8 : low-delay CBR, high quality
                    16 : CBR, high quality (slower)
                    32 : VBR, high quality (slower)
-qp <integer> : Specifies the qp value for Constant QP Rate Control Mode
-i_qfactor <float> : Specify qscale difference between I-frames and P-
frames
-b_qfactor <float> : Specify qscale difference between P-frames and B-
frames
-i_qoffset <float> : Specify qscale offset between I-frames and P-frames
-b_qoffset <float> : Specify qscale offset between P-frames and B-frames
```

```
-deviceID <integer> : Specifies the GPU Device on which encoding will take
place
-devicetype <integer> : 0 - DX9 Device Type
                                        1 - DX10 Device Type
                                        2 - DX11 Device Type
                                        3 - CUDA Device Type
-inputFormat <integer> : Specify the input format
                                  0: YUV 420
                                  1: YUV 444
                                  2: YUV 420 10-bit
                                  3: YUV 444 10-bit
-temporalAQ                       1: Enable TemporalAQ
-help : Prints Help Information
```

2.7 NvTranscoder

The NvTranscoder application demonstrates transcoding using NVENC. The transcoder application supports 8-bit depth H.264 or HEVC encoded files for input that may be transcoded to H.264 or HEVC files.

The following are the options that may be specified for NvTranscoder Application.

```
-i <string> : Specifies the input file that has to be transcoded
-o <string> : Specifies the output bitstream file
-size <integer integer> : Specifies the input resolution width and height
for encoding. If not specified, it will use the width and height of the
input file.
-codec <integer>: Specifies the codec (0) - H264 and (1) - HEVC
-preset <string> :  hq - High Quality Preset
                    hp - High Performance Preset
                    lowLatencyHP - Low Latency High Performance Preset
                    lowLatencyHQ - Low Latency High Quality Preset
                    lossless - Lossless High Quality Preset
-fps <integer> : Specifies the encoding frame rate. If not specified, it
will use the fps of the input file.
-gopLength <integer> : Specifies the GOP (Group of Pictures) Length
-numB <integer> : Specifies the number of B frames
-bitrate <integer> : Specifies the encoding average bitrate
-vbvMaxBitrate <integer> : Specifies the VBV Maximum Bitrate
-vbvSize <integer> : Specifies the Encoding VBV/HRD Buffer Size
-rcmode <integer> : Specifies the Rate Control Mode.
                    0 : Constant QP mode
                    1 : Variable bitrate mode
                    2 : Constant bitrate mode
                    8 : low-delay CBR, high quality
```

```
                    16 : CBR, high quality (slower)
                    32 : VBR, high quality (slower)
-qp <integer> : Specifies the qp value for Constant QP Rate Control Mode
-i_qfactor <float> : Specify qscale difference between I-frames and P-
frames
-b_qfactor <float> : Specify qscale difference between P-frames and B-
frames
-i_qoffset <float> : Specify qscale offset between I-frames and P-frames
-b_qoffset <float> : Specify qscale offset between P-frames and B-frames
-deviceID <integer> : Specifies the GPU Device on which encoding will take
place
-help : Prints Help Information
```

2.8 NvDecodeD3D9

The NvDecodeD3D9 application demonstrates the code for handling video decode of MPEG-2, VC-1, H.264, HEVC, VP8 and VP9 with NVDEC. The application takes a video file as input and renders the result to D3D9 window for display. The sample will use an included MPEG-2 file if no video file is provided.

The following are the options that may be specified for NvDecodeD3D9 Application.

```
NVDecodeD3D9 [parameters]

        -i=source      - Input file for decoding
        -o=output.yuv  - Ouput YUV file
        -psnr=ref.yuv   - compare PSNR against reference YUV
        -decodecuda    - Use CUDA for MJPEG (Available with 64+ CUDA cores)
        -decodedxva    - Use NVDEC for decode.
        -decodecuvid   - Use NVDEC for decode (optimized)
        -vsync         - Enable vertical sync.
        -novsync       - Disable vertical sync.
        -repeatframe   - Enable frame repeats.
        -repeatfactor=n - Force repeat every frame n times
        -updateall     - always update CSC matrices.
        -displayvideo  - display video frames on the window
        -nointerop     - create the CUDA context w/o using graphics interop
        -readback      - enable readback of frames to system memory
        -device=n      - choose a specific GPU device to decode video with
        -nframestart=n - set the start frame number
        -nframeend=n   - set the end frame number
        -bitdepth=n     - specify bit depth of video input
```

2.9 NvDecodeD3D11

The NvDecodeD3D9 application demonstrates the code for handling video decode of MPEG-2, VC-1, H.264, HEVC, VP8 and VP9 with NVDEC. The application takes a video file as input and renders the result to D3D11 window for display. The sample will use an included MPEG-2 file if no video file is provided.

The following are the options that may be specified for NvDecodeD3D11 Application.

```
NVDecodeD3D9 [parameters]

        -i=source       - Input file for decoding
        -o=output.yuv   - Ouput YUV file
        -psnr=ref.yuv    - compare PSNR against reference YUV
        -pass=<threshold> - PSNR threshold for PASS/FAIL test
         -decodecuda     - Use CUDA for MJPEG (Available with 64+ CUDA cores)
        -decodedxva     - Use NVDEC for decode.
        -decodecuvid    - Use NVDEC for decode (optimized)
        -vsync          - Enable vertical sync.
        -novsync        - Disable vertical sync.
        -repeatframe    - Enable frame repeats.
        -repeatfactor=n - Force repeat every frame n times
        -updateall      - always update CSC matrices.
        -nodisplay       - do not open a window for display
        -displayvideo   - display video frames on the window
        -nointerop      - create the CUDA context w/o using graphics interop
        -readback       - enable readback of frames to system memory
        -device=n       - choose a specific GPU device to decode video with
        -nframestart=n  - set the start frame number
        -nframeend=n    - set the end frame number
        -bitdepth=n      - specify bit depth of video input
```

2.10 NvDecodeGL

The NvDecodeGL application demonstrates the code for handling video decode of MPEG-2, VC-1, H.264, HEVC, VP8 and VP9 with NVDEC. The application takes a video file as input and renders the result to an OpenGL window for display. The sample will use an included MPEG-2 file if no video file is provided.

The following are the options that may be specified for NvDecodeGL Application.

```
NVDecodeGL [parameters]

      -i=source      - Input file for decoding
      -o=output.yuv  – Ouput YUV file
      -psnr=ref.yuv   - compare PSNR against reference YUV
      -decodecuda    - Use CUDA for MJPEG (Available with 64+ CUDA cores)
      -decodecuvid   - Use NVDEC for decode (optimized)
      -vsync         - Enable vertical sync.
      -novsync       - Disable vertical sync.
      -repeatframe   - Enable frame repeats.
      -repeatfactor=n - Force repeat every frame n times.
      -updateall     - always update CSC matrices.
      -displayvideo  - display video frames on the window
      -nointerop     - create the CUDA context w/o using graphics interop
      -readback      - enable readback of frames to system memory
      -device=n      - choose a specific GPU device to decode video with
      -nframestart=n – set the start frame number
      -nframeend=n   - set the end frame number
      -bitdepth=n     - specify bit depth of video input
```

Notice

Trademarks

Copyright