Multi-Transcoding Sample

Overview

Multi-Transcoding Sample works with **Intel® Media Server Studio 2017 - SDK for Linux* Server** (hereinafter referred to as "**SDK**")

It demonstrates how to use **SDK** API to create a console application that performs the transcoding (decoding and encoding) of a video stream from one compressed video format to another, with optional video processing (resizing) of uncompressed video prior to encoding. The application supports multiple input and output streams meaning it can execute multiple transcoding sessions concurrently.

The main goal of this sample is to demonstrate CPU/GPU balancing in order to get maximum throughput on Intel® hardware-accelerated platforms (with encoding support). This is achieved by running several transcoding pipelines in parallel and fully loading both CPU and GPU.

This sample also demonstrates integration of user-defined functions for video processing (picture rotation plug-in) into **SDK** transcoding pipeline.

This version of sample also demonstrates surface type neutral transcoding (opaque memory usage).

The sample can work together with **Intel® Media Server Studio – HEVC Decoder & Encoder** (hereinafter referred to as "**HEVC**").

Features

Multi-Transcoding Sample supports the following video formats:

Format type	
input (compressed)	H.264 (AVC, MVC – Multi-View Coding), MPEG-2 video, VC-1, JPEG*/Motion JPEG, HEVC (High Efficiency Video Coding), VP8
output (compressed)	H.264 (AVC, MVC – Multi-View Coding), MPEG-2 video, JPEG*/Motion JPEG, HEVC (High Efficiency Video Coding)

Hardware Requirements

See `<install-folder>\Media_Samples_Guide.pdf`.

Software Requirements

See `<install-folder>\Media_Samples_Guide.pdf`.

How to Build the Application

See `<install-folder>\Media_Samples_Guide.pdf`.

Running the Software

See `<install-folder>\Media_Samples_Guide.pdf`.

The executable file requires the following command-line switches to function properly:

-par <par_file>	A parameter file is a configuration file of specific structure. It contains several command lines, each line corresponding to a single transcoding, decoding or encoding **SDK** session.

The following command-line switches are optional:

-p <perf_file>	Instructs sample to write performance statistics to a specified file.
-timeout <seconds>	Timeout for application. The value is set approximately because of using full transcoding cycles.
-greedy	Enable greedy formula for surfaces calculation (in case of 1->N transcoding number of decoder surfaces will be calculated as sum of all surfaces requested by child sessions)
-?	Print help

Below are examples of a command-line to execute **Multi-Transcoding Sample**:

```
$ sample_multi_transcode –par 1.par -p 1.perf
```

For each line in the parameter file, the following command-line arguments are mandatory:

-i::h264|h265|mpeg2|vc1|mvc|jpeg|vp8|rgb4_frame <Input File>	Input video type and (compressed) video file name and path. −i::mvc requires a corresponding −o::mvc and both are supported only with **SDK** API 1.3 and higher. The use of option h265 is possible only if **HEVC** is installed. rgb4_frame input should provide one RGB4 frame only - it is used for creating overlays. Should be used together with -vpp_comp_src_h and -vpp_comp_src_w settings.
or	
-i::source	The session receives the output of the session using the −o::sink option at input (only one session using −o::sink is allowed in a parameter file.).
-o::h264|h265|mpeg2|mvc|jpeg <Output File>	Output video type and (compressed) video file, name and path. −o::mvc requires a corresponding −i::mvc and both are supported only with **SDK** API 1.3 and higher. The use of option h265 is possible only if **HEVC** is installed.
or	
-o::sink	Output of this session serves as input for all sessions using the −i::source (a parameter file can contain several sessions using −i::source).
or	
-o::raw <Output File>	Raw output data is written into the file. No compression will be done. This option works only with nv12, rgb4 and yuy2 color formats

or	
-vpp_comp_only <Sources Number>	Enables composition from several decoding sessions. Result is shown on screen. **You should set -rx11 or -rdrm option along with this option.**

Note 1: You need to have **HEVC** installed to run with h265 codec.

The following command-line switches are optional:

-rx11	Enables on-screen rendering support under X Server (requires X Server running)
-rdrm	Enables on-screen rendering support thru the framebuffer from the Linux Console mode (requires any graphical server, like X server, to be switched off)
-rdrm-<connector>	Same as -rdrm, but enables on-screen rendering support to the monitor connected thru the specified <conector>. Some possible connectors are: DisplayPort, HDMIA, HDMIB, VGA, DVII, DVID, DVIA, eDP and others
-b <bitrate>	Bitrate of the encoded stream in Kbits/second, supported for all encoders except JPEG*/Motion JPEG
-f <framerate>	Video frame rate for the FRC and Deinterlace options
-override_decoder_framerate <framerate>	Forces decoder output framerate to be set to provided value (overwriting actual framerate from decoder)
-override_encoder_framerate <framerate>	Overwrites framerate of stream going into encoder input with provided value (this option does not enable FRC, it just ovewrites framerate value)
-u <quality, speed, balanced>	Target usage, 4 (balanced) by default. This parameter specifies a trade-off between quality and speed. Supported for all encoders except JPEG*/Motion JPEG
-q	Quality parameter for JPEG*/Motion JPEG encoder. In range [1,100], 100 is the best quality
-w <width>	Width of destination video frame. If specified and the value is different from the source frame width, video preprocessing (VPP) for scaling (resizing) is invoked.
-h <height>	Height of destination video frame. If specified and the value is different from the source frame height, video preprocessing (VPP) for scaling (resizing) is invoked.
-hw|-sw	Implementation of **SDK**: platform-specific or software. If not specified, hardware is used. Software implementation should not be set since there is no software library on Linux platforms.
-ext_allocator	Force usage of external allocators
-sys	Force usage of external system allocator
-join	Join the session to another session. If a parfile contains several lines with a –join switch, the first session in the list acts as parent and all other sessions are joined to the parent. Option is effective only with **SDK** API 1.1 and higher.
-priority <0,1,2>	Set priority for the session (available only if session is joined): 0 = Low

	1 = Normal (default)
	2 = High
-threads <num>	Number of session internal threads to create
-gpucopy::<on,off>	Enable or disable GPU copy mode
-n <num>	The number of frames to process
-async <depth>	Depth of asynchronous pipeline. 1 by default.
-angle 180	Invokes sample plug-in for 180 degrees picture rotation. CPU implementation is used by default. Rotate plugin module `sample_rotate_plugin.so` must be available when running the application with this option.
-opencl	Invokes Intel®OpenCL™ implementation of 180 degrees picture rotation. Rotate plugin module `libsample_plugin_opencl.so` must be available. File `ocl_rotate.cl` must exist in the local folder when running the application with this option.
-deinterlace	Enables deinterlacing for an interlaced input stream.
-deinterlace::ADI	Forces VPP to deinterlace input stream using ADI algorithm
-deinterlace::ADI_SCD	Forces VPP to deinterlace input stream using ADI_SCD algorithm
-deinterlace::ADI_NO_REF	Forces VPP to deinterlace input stream using ADI no ref algorithm
-deinterlace::BOB	Forces VPP to deinterlace input stream using BOB algorithm
-detail <level>	Enables detail (edge enhancement) filter with provided level(0..100)
-denoise <level>	Enables denoise filter with provided level (0..100)
-FRC::PT	Enables FRC filter with Preserve Timestamp algorithm
-FRC::DT	Enables FRC filter with Distributed Timestamp algorithm
-FRC::INTERP	Enables FRC filter with Frame Interpolation algorithm
-field_processing t2t|t2b|b2t|b2b|fr2fr	Copy fields separately
-dc::nv12|rgb4|yuy2	Forces decoder output to use provided chroma mode
-ec::nv12|rgb4|yuy2|nv16|p010|p210	Forces encoder input to use provided chroma mode
-vpp_comp <Sources Number>	Enables composition from several decoding sessions. Result is written to the file.
-vpp_comp_dst_x	X position of this stream in composed stream (should be used in decoder session)
-vpp_comp_dst_y	Y position of this stream in composed stream (should be used in decoder session)
-vpp_comp_dst_w	Height of this stream in composed stream (should be used in decoder session)
-vpp_comp_dst_h	Width of this stream in composed stream (should be used in decoder session)
-vpp_comp_src_h	Width of this stream in composed stream (should be used in decoder session)
-vpp_comp_src_w	Width of this stream in composed stream (should be used in decoder session)

-vpp_comp_num_tiles <Num>	Quantity of tiles for composition. if equal to 0 tiles processing ignored
-vpp_comp_dump	null_render Disabling rendering after VPP Composition. This is for performance measurements
-vpp_comp_tile_id <TileNumber>	Tile_id for current channel of composition (should be used in decoder session)
-vpp_comp_dump <File_Name>	Dump of VPP Composition's output into file. Valid with -vpp_comp* options
-vpp_comp_dump null_render	Disabling rendering after VPP Composition. This is for performance measurements.
-l <num_slice>	Number of slices in each video frame. 0 by default. If num_slice equals zero, the encoder may choose any slice partitioning allowed by the codec standard.
-mss <slice_size>	Maximum slice size in bytes. Supported only with harware library (-hw) and H.264 encoder. This option is not compatible with -l.
-la	Use the look ahead bitrate control algorithm (LA BRC) for H.264 encoder. Supported only with -hw option on 4th Generation (or later) Intel Core processors.
-lad <depth>	Depth parameter for the LA BRC, the number of frames to be analyzed before encoding. In range [10,100].
-la_ext	Use external LA plugin (compatible with h264 & hevc encoders)
-hrd <size>	Maximum possible size of any compressed frames in KB
-wb	Maximum bitrate for sliding window in Kbps
-ws	Sliding window size in frames
-gop_size <size>	GOP size in frames
-vbr	Use variable bitrate control. By default Constant Bitrate Control is used
-CodecProfile <profile>	Specifies codec profile
-CodecLevel <level>	Specifies codec level
-GopOptFlag:closed	Encoder generates closed GOP. Frames in this GOP do not use frames in previous GOP as reference.
-GopOptFlag:strict	Encoder strictly follows given GOP structure as defined by parameter GopPicSize, GopRefDist etc.
-InitialDelayInKB <size>	The HRD decoder starts decoding after the buffer reaches the initial size InitialDelayInKB, which is equivalent to reaching an initial delay of InitialDelayInKB*8000/TargetKbps ms
-MaxKbps <size>	For variable bitrate control, specifies the maximum bitrate at which the encoded data enters the Video Buffering Verifier buffer
-cqp	Constant quantization parameter (CQP BRC) bitrate control method (by default constant bitrate control method is used), should be used along with -qpi, -qpp, -qpb.
-qpi	Constant quantizer for I frames (if bitrace control method is CQP). In range [1,51]. 0 by default, i.e.no limitations on QP.
-qpp	Constant quantizer for P frames (if bitrace control method is CQP). In range [1,51]. 0 by default, i.e.no limitations on QP.

-qpb	Constant quantizer for B frames (if bitrace control method is CQP). In range [1,51]. 0 by default, i.e.no limitations on QP.
-qsv-ff	Enable QSV-FF mode
set	This option is used to specify codec plugin to be used in multiple sessions, supported with –i::h265 and –o::h265 only. This plugin will be used for all sessions defined below. You can specify plugin as 32-character hexadecimal guid string or path to dll.
-stat <N>	Output statistic every N transcoding cycles
-stat-log <name>	Output statistic to the specified file (opened in append mode)
-fps <frames per second>	Transcoding frame rate limit.
-timeout	Transcode in a loop not less than specific time in seconds. Performs complete input stream transcoding on every iteration. Output file frames amount can be bigger than in input due to buffered frames in pipeline. Output file is rewrote every iteration.
-pd <plugin ID or path>	Set decoding plugin for this particular session. This setting overrides plugin settings defined by SET clause.
-pe <plugin ID or path>	Set encoding plugin for this particular session. This setting overrides plugin settings defined by SET clause.
-dist	Distance between I- or P- key frames
-num_ref	Number of reference frames
-bref	Arrange B frames in B pyramid reference structure
-repartitioncheck::<on,off>	Enable or disable RepartitionCheckEnable mode

The command-line interface allows 2 usage models (which can be mixed within one parameter file):

1. Multiple intra-session transcoding: several transcoding sessions, any number of sessions can be joined. Each session includes decoding, preprocessing (optional), and encoding.
2. Multiple inter-session transcoding: output of a single decoding session serves as input for several encoding sessions. Either all or none of the sessions are joined. Any of the encoding sessions can optionally include preprocessing (resizing).

Below are several examples of parameter file contents.

Single intra-session transcoding:

```
-i::vc1 input.vc1 -async 10 -o::h264 output.h264 –n 100 –w 320 –h 240 –f 30
–b 2000 –u speed
```

Multiple intra-session transcoding, several sessions joined:

```
-i::vc1 input1.vc1 -async 10 -o::mpeg2 output1.mpeg2
-i::h264 input2.h264 -o::mpeg2 output2.mpeg2 –join
-i::h264 input3.h264 -o::mpeg2 output3.mpeg2 –join
```

Multiple inter-session transcoding, all sessions joined:

```
-i::h264 input.h264 -o::sink –join
-o::mpeg2 output1.mpeg2 -i::source –join –w 640 –h 480
-o::mpeg2 output2.mpeg2 -async 2 -u 3 -i::source –join
```

Mixed model:

```
-i::vc1 input.vc1 -async 10 -o::h264 output.h264 -n 100 -w 320 -h 240 -f 30
 -b 2000 -u speed
-hw -i::h264 input.h264 -o::sink -join
-o::mpeg2 output1.mpeg2 -i::source -join -w 640 -h 480
-o::mpeg2 output2.mpeg2 -async 2 -u 3 -i::source -join
```

Single intra-session MVC transcoding:

```
-i::mvc input.mvc -async 10 -o::mvc output.mvc -n 100 -w 320 -h 240 -f 30 -b
 2000 -u speed
```

Multiple intra-session transcoding, same **HEVC** is used in both cases:

```
set -i::h265 15dd936825ad475ea34e35f3f54217a6
-i::h265 input1.265 -o::h264 output1.264
-i::h265 input2.265 -o::mpeg2 output2.mpeg2
```

Multiple intra-session transcoding with **HEVC**:

```
set -i::h265 /path/to/so/decoder_plugin.so
set -o::h265 /path/to/so/encoder_plugin.so
-i::h265 input1.265 -o::h264 output1.264
-i::mpeg2 input2.mpeg2 -o::h265 output2.265
```

Please, also pay attention on "Running the Software" section of `<install-folder>`/ `Media_Samples_Guide.pdf` document where you will find important notes on backend specific usage (drm and x11).

ROI file format description

Tips

1. To achieve maximum throughput use `-async` >= 5 and the –join option when running several transcoding pipelines.
2. If you need only one transcoding session you can avoid creating a par file and pass the arguments of this session to the application using command line. E.g.:

```
sample_multi_transcode -i::mpeg2 input.mpeg2 -async 10 -o::h264 output.h264
-n 100 -w 320 -h 240 -f 30 -b 2000 -u speed -p 1.perf -hw
```

Known Limitations

- Configurations <multiple joined inter-session transcoding where one of the encoders is MPEG2> are not supported when sample application uses platform-specific **SDK** implementation on systems with Intel® HD Graphics 3000/2000 and 4000/2500. Application can exit with error or hang. An example of a corresponding par file is given below:

```
-i::h264 input.h264 -o::sink -join
-o::mpeg2 output1.mpeg2 -i::source -join
-o::h264 output2.h264 -i::source -join
```

Systems with Intel® Iris™ Pro Graphics, Intel® Iris™ Graphics and Intel® HD Graphics 4200+ Series are free from this limitation.

- Picture rotation sample plug-ins do not swap view order in the pipeline with MVC encoder. This should be considered if viewing of the output video is involved.
- **Multi-Transcoding Sample** if run with `-opencl` option requires input video frame width to be aligned by 4.
- **Multi-Transcoding Sample** if run with `-f` option without -FRC or -deinterlace options does not perform frame rate conversion. Specified frame rate simply overrides frame rate value of the original stream and is used as decoder setting.
- In case of using HEVC plugin (h265 video type), plugin type (hardware or software) used by default is set depending on -sw or -hw sample options. However, hardware HEVC plugins work on specific platforms only. To force usage of specific HEVC plugin implementation, please use -pe and -pd options with proper plugin GUID.
- SW HEVC plugin in 10bit mode cannot be used together with HW library VPP. Although library allows that, this is bad practice because additional per-pixel data shift is required. Please use HW HEVC + HW library or SW HEVC + SW library instead.
- Sample may crash if composition filter is used with more than 10 sources (because of limitations in MSDK library).
- -timeout option set in command line together with par file name may work incorrectly. Please use -timeout option set inside par file instead.
- Sample may not function properly on systems that have a non-Intel VGA controller as the first (primary) because Intel device is not first in the list.

    To workaround this issue, swap names of DRI device files:

    ```
    $ cd /dev && mv card0 tmp && mv card1 card0 && mv tmp card1
    ```

    and do the same for the files `control64/65` and `renderD128/129`

Legal Information

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

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 SSE3 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