Video Conferencing Sample

Overview

Video Conferencing Sample works with **Intel® Media Server Studio 2015 for Linux**.

It demonstrates how to use the **Intel® Media Server Studio – SDK** (hereinafter referred to as "**SDK**") API features, intended for generic video conferencing tasks.

- How to configure **SDK** Encode for generating low latency bit stream
- How to recover from situation when receiving part couldn't decode some frame due to network packets loss
- How to change target bitrate of **SDK** Encode dynamically
- How to force **SDK** Encode to insert a key frame at particular position out of predefined GOP structure
- How to force **SDK** Encode to create long term reference from particular frame
- How to use per frame mode QP (quantization parameter)
- How to enable Microsoft* Lync* Features (temporal scalability)
- How to enable Reference Picture Marking Repetition SEI
- How to change encoding resolution dynamically
- How to measure latency of **SDK** Encode
- How to enable **SDK** Encode rolling intra refresh feature

Features

Video Conferencing Sample supports the following video formats:

Format type	
input (uncompressed)	YUV420
output (compressed)	H.264 (AVC)

Note: For format YUV420, **Video Conferencing Sample** assumes the order Y, U, V in the input file.

Hardware Requirements

See `<install-folder>\Media Samples Guide.pdf`.

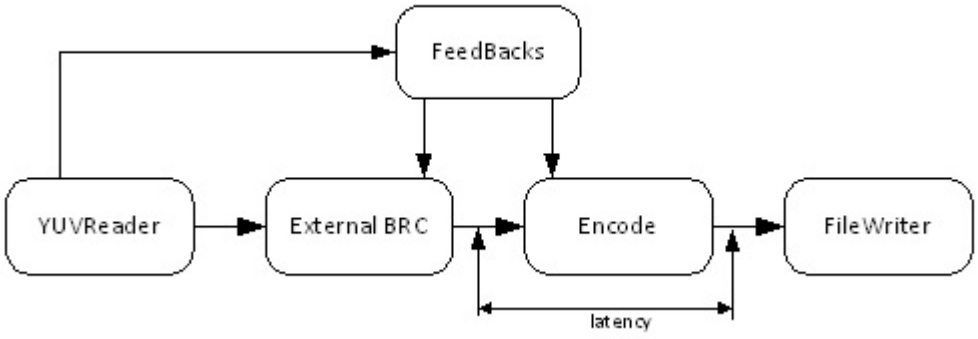
Software Requirements

See `<install-folder>\Media Samples Guide.pdf`.

Application data flow

In **Video Conferencing Sample** feedbacks on the quality of encoding are imitated and created during initialization stage based on command line parameters, while in a real conferencing application they are to be created dynamically based on the receiver reports.

The diagram below describes the flow of the data in the application:



1. YUV Frame is read from a file
2. Feedbacks buffer is examined for available feedbacks. If some feedbacks are ready for the current frame, they are applied to change Encode or External BRC settings
3. Latency timer started
4. YUV frame is encoded
5. Latency timer stopped
6. Encoded frame is written to a file

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:

-i <InputFile>	Input (uncompressed) video file, name and path
-o <Output>	Output (compressed) video file, name and path
-w <width>	Width of input video frame
-h <height>	Height of input video frame

The following command-line switches are optional:

-hw	Use platform-specific implementation of **SDK** (default)
-sw	Use software implementation of **SDK** (platform-specific implementation is used by default)
-b <bitrate>	Bitrate of the encoded stream in Kbits/second, supported for all encoders except JPEG*/ Motion JPEG
-f <framerate>	Frame rate of the encoded stream (30 by default)

-bs <frame, scale>	Calculate new target bitrate as current target bitrate * scale. If external bitrate control enabled(`-brc`), it will be reset with new target value prior to encoding of frame, no changes to **SDK** encode will be done. If external bitrate control disabled, encoder itself is reset with new target bitrate prior to encoding of frame. **Note:** the behavior of **SDK** encode with regards to bitrate change is described in Appendix C of `mediasdk-man.pdf`
-bf <frame, broken_frame>	Simulates the situation when video conferencing receiver can not decode brokenFrame and this information reached video conferencing sender prior to encoding of frame. **Note:** the behavior of **SDK** encode with regards to rejected reference list selection is described in Appendix C of `mediasdk-man.pdf`.
-gkf <frame_num>	Generates IDR frame at specific position.
-ltr <frame_num>	Marks specific `frame_num` as longterm reference. **Note:** the behavior of **SDK** encode with regards to preferred and long-term references lists selection is described in Appendix C of `mediasdk-man.pdf`.
-ts <num_layers>	Will create up to 4 temporal layers. **Note:** the behavior of **SDK** encode with regards to temporal layer creation is described in `mfxExtAvcTemporalLayers` structure description in `mediasdk-man.pdf`.
-brc	Enables external bitrate control based on perframe QP.
-l0 <frame> <l0_len>	Specifies number of reference frames in L0 array for encoding frame
-latency	Calculates per frame latency of **SDK** Encode and writes to the standard output along with respective frame types (I, P). **Note:** to estimate latency of generic videoconferencing scenario (encode + decode), per frame latency values for **SDK** Decode can be obtained using **Decoding Sample** with input stream encoded by **Video Conferencing Sample**.
-ir <cycle_size, qp_delta>	Specifies that intra frame will be completely refreshed in `cycle_size` number of pictures. This value is taken from the interval [3; 29] `qp_delta` defines the quantization parameter difference for the inserted intra macroblocks. The valid values for this parameter belong to the interval [-51; 51]
-par <parameters_file>	Specifies parameters file that may contain any of supported options.
-?	Print help

Below are examples of a command-line to execute **Video Conferencing Sample**:

```
sample_videoconf -i input.yuv -o output.h264 -w 720 -h 480 -b 10000
-hw -bs 10 2
sample_videoconf -i input.yuv -o output.h264 -w 720 -h 480 -b 10000
-hw -bf 10 8
sample_videoconf -i input.yuv -o output.h264 -w 720 -h 480 -b 10000
-hw -ltr 10
```

Known Limitations

- Frames might be not fully restored after encoding with intra refresh (`-ir`).

Legal Information