

Master VICO – Lab on Video Coding

Lab: HEVC

Goal:

To use the HEVC reference software (or “HEVC modeling software”, HM) to encode / decode video sequences.

To do:

You can download video sequences (format: YUV 4:2:0) from *Extradoc*. You can display them by using *YUVPlayer.exe*. The HMs is freely available at this URL:

http://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/

The stable versions of the software are in the directory *tags*, we are going to use the *HM16.20* version.

In principle, it requires a SVN client to get the archive. For simplicity reason, the *HM16.20.zip* archive is also directly available on the *Extradoc*.

The documentation is in: `\HM16.20\doc\software-manual.pdf`

In principle, the binaries *TAppEncoder* and *TAppDecoder* are directly in: `\HM16.20\bin\vc2015\Win32\Debug\`

Only if necessary, you can compile the sources by using “Microsoft Visual Studio C++”.

For example, “Microsoft Visual C ++ 20XX Express Edition” is used to compile the project described in: `\HM16.20\build\HM_vc2015.sln`

So after compilation, the executables are in: `\HM16.20\bin\vc2015\Win32\Debug\`

The documentation explains how to use *TAppEncoder* and *TAppDecoder* by the command line.

Examples of encoder configuration files are in : `HM16.20/cfg/`

Example of an *encoding.bat* file:

```
C:\Users\ricordel\Documents\HM16.20\bin\vc2015\Win32\Debug\TAppEncoder.exe
-c C:\Users\ricordel\Documents\ETN5\test-encodage.cfg
-i C:\Users\ricordel\Videos\mobile-cif.yuv
pause
```

Example of a *decoding.bat* file:

```
C:\Users\ricordel\Documents\HM16.20\bin\vc2015\Win32\Debug\TAppDecoder.exe
-b C:\Users\ricordel\Documents\ETN5\test-mobile.hevc
-o C:\Users\ricordel\Documents\ETN5\test-decod-mobile-cif.yuv
pause
```

You have to test different encoding modes, where at each time you:

- encode the video sequence at a given quality level;
- understand the parameters set of the corresponding configuration file;
- see the effects of these parameters on the decoded sequence;
- analyse de HEVC bitstream (for instance by using "Intel Video Pro Analyser" that you can download freely from the site www.softpedia.com).

Exactly, you will test the following modes:

1. "*Intra Only*" (*encoder_intra_main.cfg*) where the frames are encoded successively and using only intra prediction.
2. "*Low delay*" (*encoder_lowdelay_main.cfg* and *encoder_lowdelay_P_main.cfg*) where the frames are encoded successively using the uni & bidirectional predictions respectively, or only the uni-directional prediction.
3. "*Random Access*" (*encoder_randomaccess_main.cfg*), which allows the GOP to be organized with time layers on several levels, this mode is better resilient to the network packet losses (cf. "*error concealment*" approach), because low temporal layers loss do not affect the higher ones.