Video Signal Compression

Mikołaj Leszczuk, AGH University of Science and Technology

# 1         Purpose of Exercise

The aim of the exercise is to learn from the video sequence compression standards (moving images). As part of the exercise, you will examine the impact of the Compression Ratio () [[1]](#footnote-1) on image quality (distortion size).

Note! Please read the following instructions before performing the exercise!

# 2         Task

Determine the dependence of the degree of distortion on 𝐶𝑅 for a few selected video sequences and several . **Objective** metrics (averaged on the frames) available in a special model (created in MATLAB / Simulink[[2]](#footnote-2)) and **subjective** measures of Mean Opinion Scores (MOS) [[3]](#footnote-3), based on an assessment at the time of viewing should be used as a distortion measure. The student should determine the number of measurement points to reliably cover the whole range of MOS and fit in a meaningful flow rate limit. After an objective and subjective evaluation of **the reconstructed (processed) video sequences**, please apply the assessment chart (MOS as a function of or a function of bit rate) – so it must be done so the number of measurements to allow reliably plotting graphs.

Metrics (only PSNR[[4]](#footnote-4) and AE[[5]](#footnote-5) implemented model):

* ← Remember this formula!
* ← Remember this formula!

In addition to the aggregate charts, please, for these sequences, designate two additional characteristics. One for codecs that use both INTRA frames and INTER frames. The second relates to the coding that uses only INTRA frames.

Please compare obtained in video standards with ZIP (to allow comparison of the graph). Is the ZIP compression of data from a codec video compressed file increasing 𝐶𝑅?

**Exemplary, non-compressed / pristine quality (Uncompressed / Reference AVI) video sequences can be downloaded, for example, from these addresses:**

* <http://www.softage.ru/products/video-codec/demo-page/> (SD),
* <ftp://vqeg.its.bldrdoc.gov/MM/> (SD),
* <ftp://vqeg.its.bldrdoc.gov/SDTV/> (SD),
* <ftp://vqeg.its.bldrdoc.gov/HDTV/NTIA_source/> (HD),
* <http://www.cdvl.org/> (HD),
* <http://live.ece.utexas.edu/> (HD),
* <http://trecvid.nist.gov/> (HD),
* <https://qoe.agh.edu.pl> (HD).

Note: Operating systems may incorrectly indicate that these files are still compressed with “MS H.263” codec – but this is not true. Since some files were prepared for NTSC (30 half-frames per second) and are played in PAL (25 half-frames per second), the playback speed may seem subjectively too low.

Some detailed steps to accomplish the exercises should be done in the following loop:

1. Opening of **an uncompressed video sequence** (**possibly**, **if time permits**, the research should be repeated for different video sequences – until the end of the time available in class). Uncompressed video sequences are available as “Uncompressed AVI” and possibly as YUV (for processing into AVI containers).
2. **Enabling** compression with a choice of AVI or MPEG-4 format, codec and : using ***Convert/Stream*** then ***Choose Profile*** then ***Custom*** then ***Customize*** (VLC media player), using ***Open Source*** then ***Preset*** then ***Save As*** then ***Start*** (HandBrake) or using ***Save as AVI ...*** (VirtualDub) – as far as time permits, you should repeat the test for different .
3. Saving with simultaneous file compression (under a different original name) – please note that not necessarily the resulting video sequence is compressed with a given .
4. Subjective (visual) quality score (MOS) of the moving image.
5. Objective (computer-aimed, quantitative) assessment (according to specified metrics) of the quality of the moving image.

# 3         Codecs

During this exercise, first please test the following video codecs:

* **MPEG-4** (**with H.264 extensions**, also known as **AVC** – Advanced Video Coding, 10 parts of MPEG-4) – may have **problems** due to the limitations of the codec, in this case, you must update the codec – available through **VLC media player**, **HandBrake** and **x264vfw**

If time permits, please also test (although there are some problems with some of them):

* **Motion JPEG (MJPEG)** – **fixed** , only Intra coding – available through **VLC media player** and **ffdshow**
* **Huffman YUV (HYUV, HuffYUV)** – **fixed** , only Intra coding – available through **ffdshow**
* **MPEG-1** – available through **VLC media player**
* **MPEG-2** – available through **VLC media player**
* **MPEG-4** (**without H.264 extensions**, **in any implementation**, can be **DivX** or **XviD**) – available through **VLC media player**
* **H.261** – you may experience **problems** due to the limitations of the codec, in such a case, update the codec
* **H.263/H.263+** – there may be **problems** due to the limitations of the codec, in such a case, update the codec
* **H.265**, also known as **HEVC** – High Efficiency Video Coding
* **Windows Media Video** (WMV) – available through **VLC media player**
* **Theora** – available through **VLC media player**
* **Dirac** – available through **VLC media player**

Unless otherwise stated, codecs support both intra-and intercoding modes. In preparation for the exercise, students should have basic knowledge of the compression methods used in the MPEG standards.

**Attention! Generally, not all codecs work in MATLAB on every operating system. The MATLAB versions differ in their support for different codecs, depending on the operating system, and what’s more it changes from version to version. Therefore, it is best to first check (empirically) whether MATLAB (or more precisely: Simulink) has support for a given codec. If not, just for the sake of exercise, try just with another. The list of possibilities is quite long anyway.**

# 4         Technical Notes

This section contains information on the technical feasibility of the exercise (i.e., primarily used programs). Special software is required only for compression and objective comparison. **To play back the sequence built-in player is sufficient.**

## 4.1      Compression

The compression step can be performed with the help of two software packages. The first approach (running under any popular operating system) is VLC media player (Section 4.1.1). The second approach (running under any popular operating system) is HandBrake (Section 4.1.2). The third approach (bit dated and works only on Windows) is VirtualDub along with the ffdshow and x264vfw codecs (Section 4.1.3).

No matter which approaches you follow, compression can be made by different settings of the desired stream bit rate. It should be noted that the rate requested is for information purposes only – actual, achieved rate **may vary significantly** from the rate desired, especially for very small and very large bitrates – which is due to the limitations of codecs. The operating system may give the wrong information on the bitrate of compressed streams, it is better if this information is confronted with the size of the file and **calculated by yourself** considering the file size, resolution, color space, duration of sequence and FPS Ratio.

Please think more than twice the number of measurements because probably you cannot do everything. Either explore more levels of quality, or more codecs, but please at least fully analyze one codec.

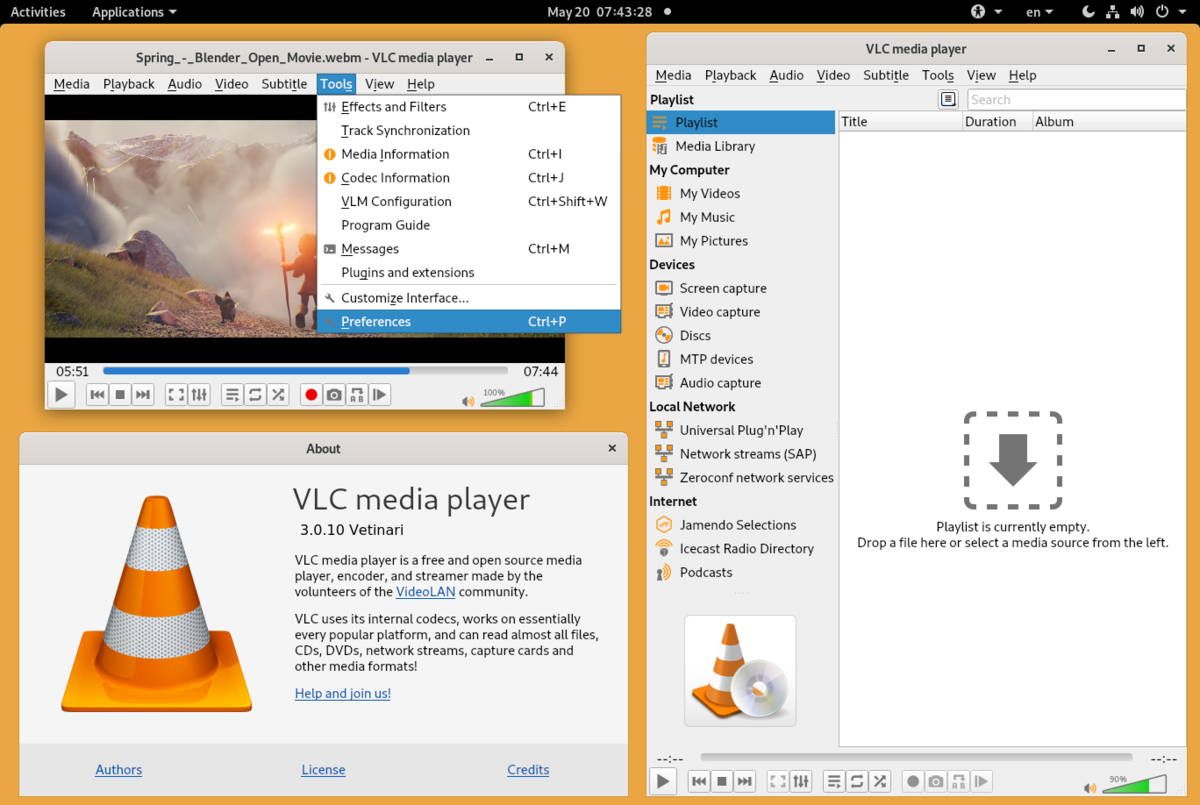
### 4.1.1     VLC media player

**The first approach is running under any popular operating system. To compress, we use VLC media player (downloadable from:** [**https://www.videolan.org/**](https://www.videolan.org/)**).**

**VLC Media player** (previously the **VideoLAN Client** and commonly known as simply **VLC**) is a free and open-source, portable, cross-platform media player software, and streaming media server developed by the VideoLAN project. VLC is available for desktop operating systems, and mobile platforms, such as Android, iOS, iPadOS, Tizen, Windows 10 Mobile, and Windows Phone. VLC is also available on digital distribution platforms such as Apple's App Store, Google Play, and Microsoft Store.

VLC supports many audio compression and video compression methods and file formats, including DVD-Video, video, CD, and streaming protocols. It can stream media over computer networks and to transcode multimedia files.

The default distribution of VLC includes many free decoding and encoding libraries, avoiding the need for finding/calibrating proprietary plugins. The libavcodec library from the FFmpeg project provides many of VLC’s codecs, but the player mainly uses its own muxers and demuxers. It also has its own protocol implementation. It also gained distinction as the first player to support the playback of encrypted DVDs on Linux and macOS by using the libdvdcss DVD decryption library; however, this library is legally controversial and is not included in many software repositories of Linux distributions as a result.



[This photo](https://en.wikipedia.org/wiki/VLC_media_player), author: Author unknown, license: [CC BY-SA](https://creativecommons.org/licenses/by-sa/3.0/)

To take a video file and change it to a different format or bitrate, you will need to follow the process of transcoding, which is described here: <https://wiki.videolan.org/transcode/>.

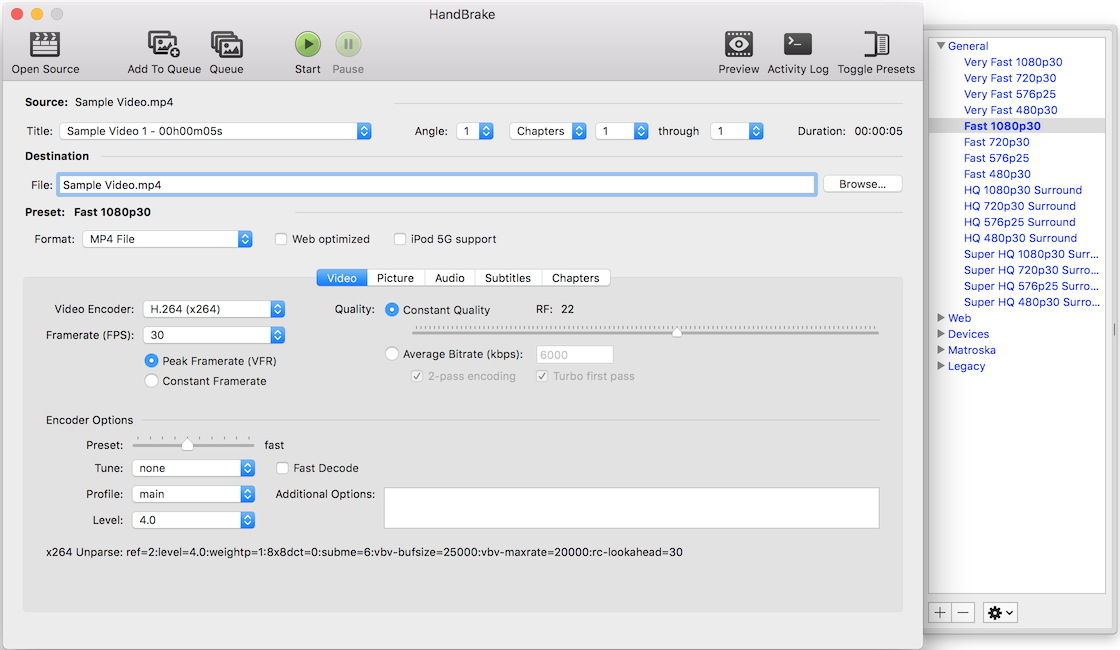
In case of problems with the software, it is recommended that you upgrade it!

### 4.1.2     HandBrake

**The second approach is running under any popular operating system. To compress, we use HandBrake (downloadable from:** [**https://handbrake.fr/**](https://handbrake.fr/)**).**

HandBrake is a free and open-source transcoder for digital video files, originally developed in 2003 by Eric Petit to make ripping a film from a DVD to a data storage device easier. HandBrake’s backend contains comparatively little original code; the program is an integration of many third-party, audio and video libraries, both codecs (such as FFmpeg, x264, and x265) and other components such as video deinterlacers (referred to as “filters”). These are collected in such a manner to make their use more effective and accessible (e.g., so that a user does not have to transcode a video’s audio and visual components in separate steps, or with inaccessible command-line utilities).

HandBrake clients are available for Linux, macOS, and Windows.



[This photo](https://en.wikipedia.org/wiki/HandBrake), author: Author unknown, license: [CC BY-SA](https://creativecommons.org/licenses/by-sa/3.0/)

In case of problems with the software, it is recommended that you upgrade it!

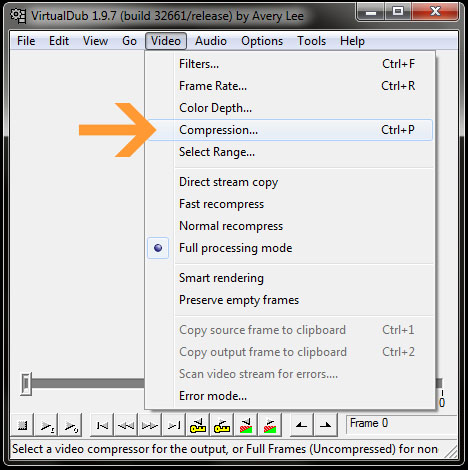
### 4.1.3     VirtualDub, ffdshow and x264vfw

**The third approach is a bit dated and works only on windows. To compress, we use VirtualDub (downloadable from:** [**http://www.virtualdub.org/**](http://www.virtualdub.org/)**), together with a set of ffdshow or x264vfw codecs.**

Enabling the window on the settings of compression parameters in VirtualDub: Video/Compression.../ffdshow…|x264vfw…/Configure.

#### 4.1.3.1     VirtualDub

**VirtualDub** is a free tool under the GNU General Public License. It allows you to edit video files offering the user a number of useful features such as compression (codec switching: “Video / Compression ...”) Along with the saving of the compressed file (function: “Save as AVI ...”), view frame by frame, the discharge of compressed (rather useless in the exercise) and uncompressed (useful in the exercise) frames of a movie into a sequence of images (function: “Save as image sequence ...”), removing and adding some parts of the film, applying filters, as well as pasting the stills into the film (source description and more information: <http://en.wikipedia.org/wiki/VirtualDub>). The program runs in the Windows environment. Note: VirtualDub program itself does not have any codec, but uses the codecs installed on your system (for example, ffdshow or x264vfw that should be used in this exercise).



In case of problems with the software, it is recommended that you upgrade it!

#### 4.1.3.2     ffdshow

**ffdshow** are DirectShow and Video for Windows (VFW) codecs for many video and audio formats, including DivX and XviD using libavcodec, XviD and other open-source libraries (source description and more information: <http://en.wikipedia.org/wiki/Ffdshow> – **there is also a link to the latest ffdshow build**). The package installs the codec in the operating system environment. Note: the ffdshow package itself is not capable of compression, but the codecs are used by programs making compression / decompression (for example, by VirtualDub, Mplayer, or Windows Media Player).

**By default, in the ffdshow some codecs may be excluded – to play, you must enable them (for ffdshow: Start / All Programs / ffdshow).**

In case of problems with the software, it is recommended that you upgrade it!

#### 4.1.3.3     x264vfw

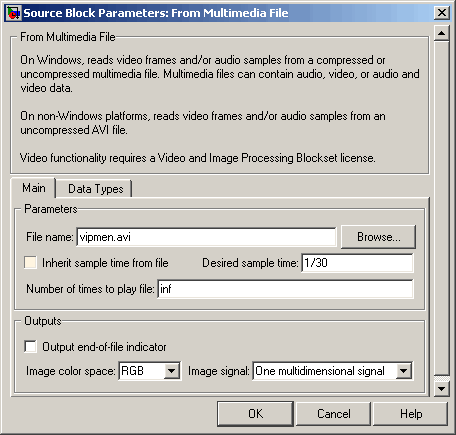
Official version of the x264 codec (<http://www.videolan.org/developers/x264.html>) for VFW framework. As a decoder, the ffh264 code of FFmpeg project is used. More information: <http://sourceforge.net/projects/x264vfw/> – **there is also a link to the latest x264vfw build**).

## 4.2      Determination of Distortion

To determine the objective degree of distortion, the model can be used. The model requires **two files** that contain the original and reconstructed frame video sequence. The model has been placed on the server. In addition, the model visualizes (with a gain specified separately for each channel), the differences between the original and the reconstructed sequence.

Please examine the MATLAB / Simulink program and the model before class.

* **At simulation runtime, please pay attention to match the duration of the simulation (box on the top bar of the simulator) with the duration of the video sequence.**
* The blocks for the choice of input files are “From Multimedia File” blocks (please note that the block has sometimes problems with paths containing spaces and non-ASCII characters). File selection: Main/Parameters/File name:/Browse...

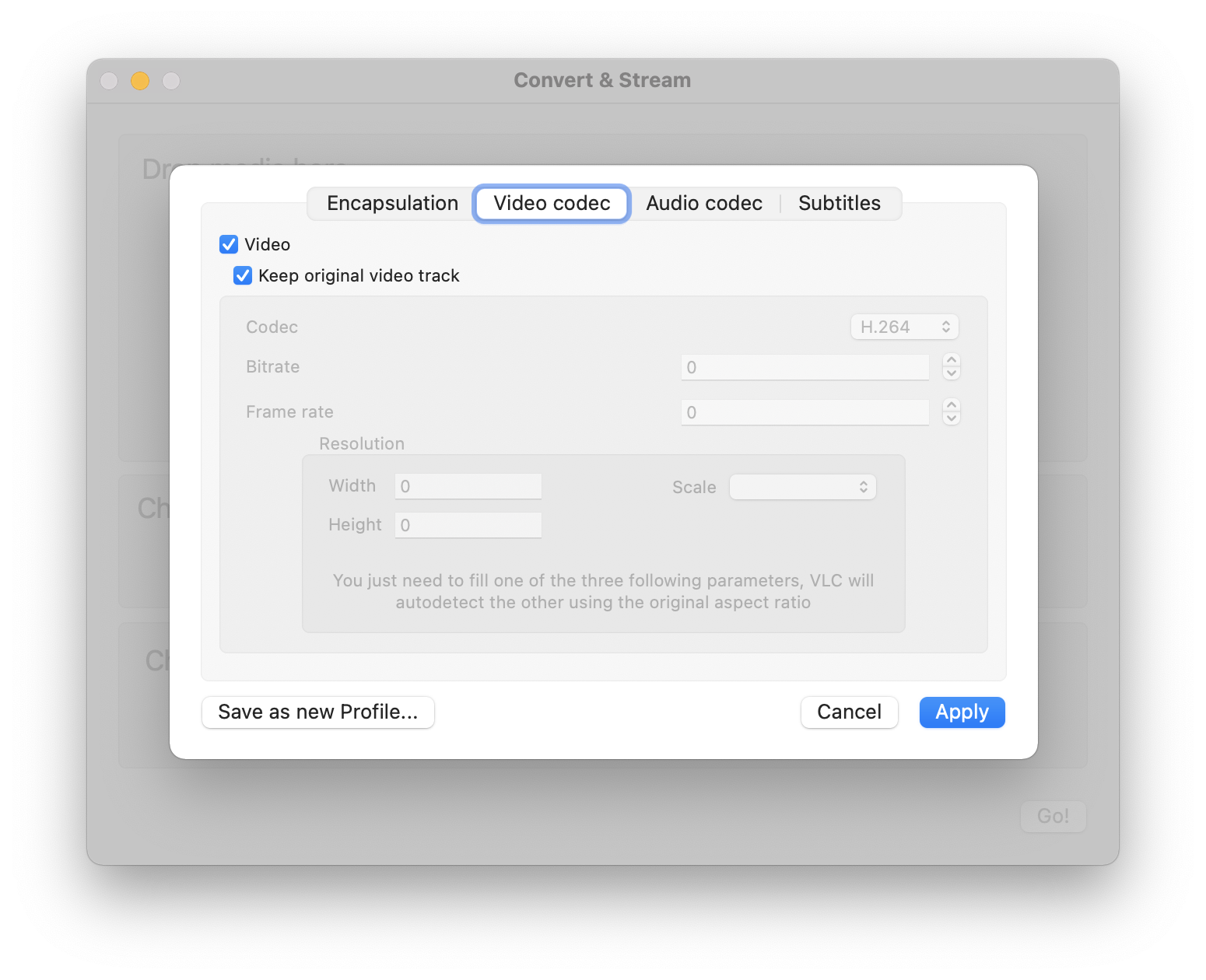


**The “From Multimedia File” the block reads multimedia files containing audio, video, or audio and video data.**

**For Windows platforms, this block reads compressed or uncompressed multimedia files.**

**For non- Windows platforms, this block reads uncompressed video and audio AVI files, and the video is only compressed or uncompressed files.**

If MATLAB running under your operating system does not allow to open AVI files, please try with MPEG-4 format. You may want to use VLC media player to change the encapsulation from AVI to MP4/MOV without reencoding, by selecting “Keep the original video track” in the “Video codec” tab of the “Convert & Stream” dialog:



* “Start the simulation” a (triangular) button is used to run the simulation.



* **“Display” blocks show the values of moving averages (objective assessment). The most important is the final value of the moving average, as is the average for the entire sequence.**

****

* “Gain” block defines the strengthening difference signal components.



* “Scope” a block displays graphs of a function of time (graphs can be maintained by a screenshot, [http://en.wikipedia.org/wiki/screenshot](http://en.wikipedia.org/wiki/Screenshot), or output to a file – the latter, depending on the configuration of your computer in the lab, may require installing the virtual printer). To open a “Scope” chart, you need to click on it (on the block). **“Auto-Scale”** allows you to automatically adjust the boundaries of the axis of charts.
* All values in Simulink must be provided in a fractional notation (0.10), not the percentage notation (10%).

# 5         References

* Austerberry D.: *The Technology of Video and Audio Streaming*. Focal Press, Oxford 2002
* Halsall F.: *Multimedia Communications, Applications, Networks, Protocols and Standards*. Addison-Wesley, Essex 2001
* Griwodz C., Halvorsen P.:” Media and User Behavior”. *INF 5070 – Media Servers and Distribution Systems*, 2005
* Thomas Wiegand: „New Techniques for Improved Video Coding”

1. http://en.wikipedia.org/wiki/Data\_compression\_ratio [↑](#footnote-ref-1)
2. **You are kindly asked for not keeping an open MATLAB / Simulink, because it blocks the licenses to other users.** Before you begin, make sure that the MATLAB and Simulink programs have been installed on the computer. Otherwise, while in the lab: labreinstall. [↑](#footnote-ref-2)
3. <http://en.wikipedia.org/wiki/Mean_opinion_score> [↑](#footnote-ref-3)
4. More information, including the borders: <http://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio> [↑](#footnote-ref-4)
5. <http://en.wikipedia.org/wiki/Mean_absolute_error> [↑](#footnote-ref-5)