HEVC and VP9 Codecs: Try them yourself

High efficiency video coding (HEVC) and WebM VP9 are new codec formats designed to support HD and ultra HD streaming and broadcasting. This article explains how to start using HEVC and VP9 software encoders. If you are planning to use these codecs in your application or product, or if you just want to find out more about HEVC and VP9, then this is a good starting point. In this article we provide instructions for getting started with the HM and x265 HEVC encoders and the WebM VP9 encoder.

1 HEVC: HM reference software

The HM software codec is a reference implementation of the HEVC coding standard developed by the Joint Collaborative Team on Video Coding (JCT-VC). It is suitable for experimenting with the various features available in the HEVC coding standard and/or for checking compliance, but is unlikely to be suitable for real-time implementations. Follow these steps to build and run the HM codec:

Step 1: Get the source code

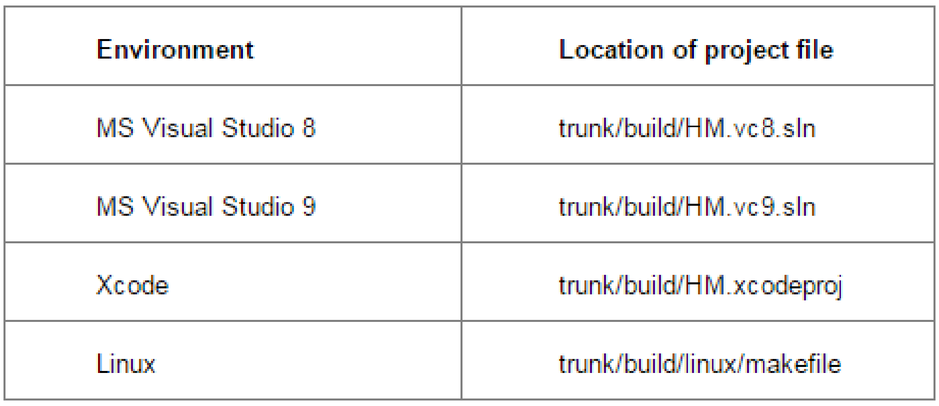
The latest source code can be obtained using SVN as follows:

```
svn checkout https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/trunk/
```

Files will download into a folder called trunk.

Step 2: Build the software

Project files are provided for the following environments:



Environment	Location of project file
MS Visual Studio 8	trunk/build/HM.vc8.sln
MS Visual Studio 9	trunk/build/HM.vc9.sln
Xcode	trunk/build/HM.xcodeproj
Linux	trunk/build/linux/makefile

If you are using Mac or a linux-based machine, we recommend building the release version of the source using the makefile in /trunk/build/linux folder by running these command lines:

```
cd
/trunk/build/llvm
sudo make release
```

For Windows machines, you will need Microsoft Visual Studio to build the solutions.

Depending on the version of the software, choose the relevant project solution in trunk/ build folder and build in release mode.

If everything builds successfully, you should have TAppEncoder and TAppDecoder executable in the /trunk/bin folder.

Step 3: Read the user manual

Detailed documentation is available here:

https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/branches/HM-9.2-dev/doc/software-manual.pdf (https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/branches/HM-9.2-dev/doc/software-manual.pdf)

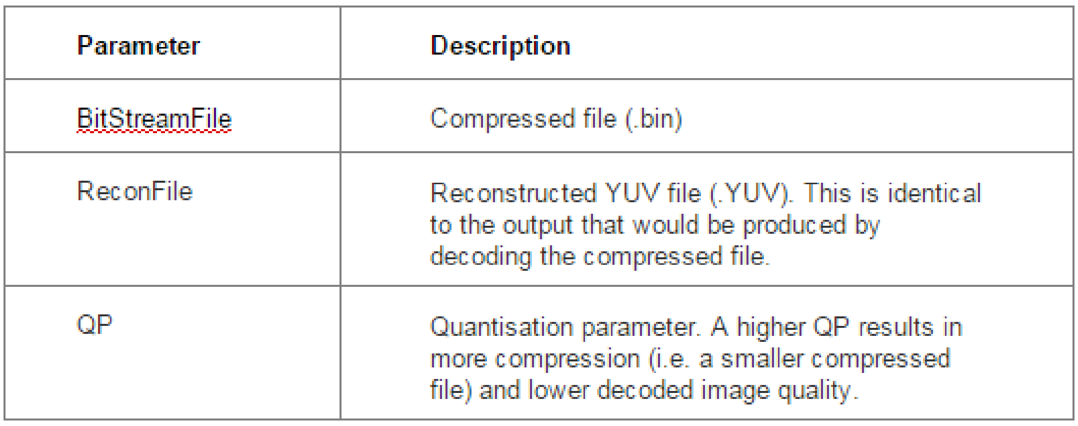
Step 4: Using the encoder

The HEVC encoder requires three inputs:

1. An uncompressed (raw) YUV or Y4M video file,
2. A codec configuration file
3. A sequence-specific configuration file

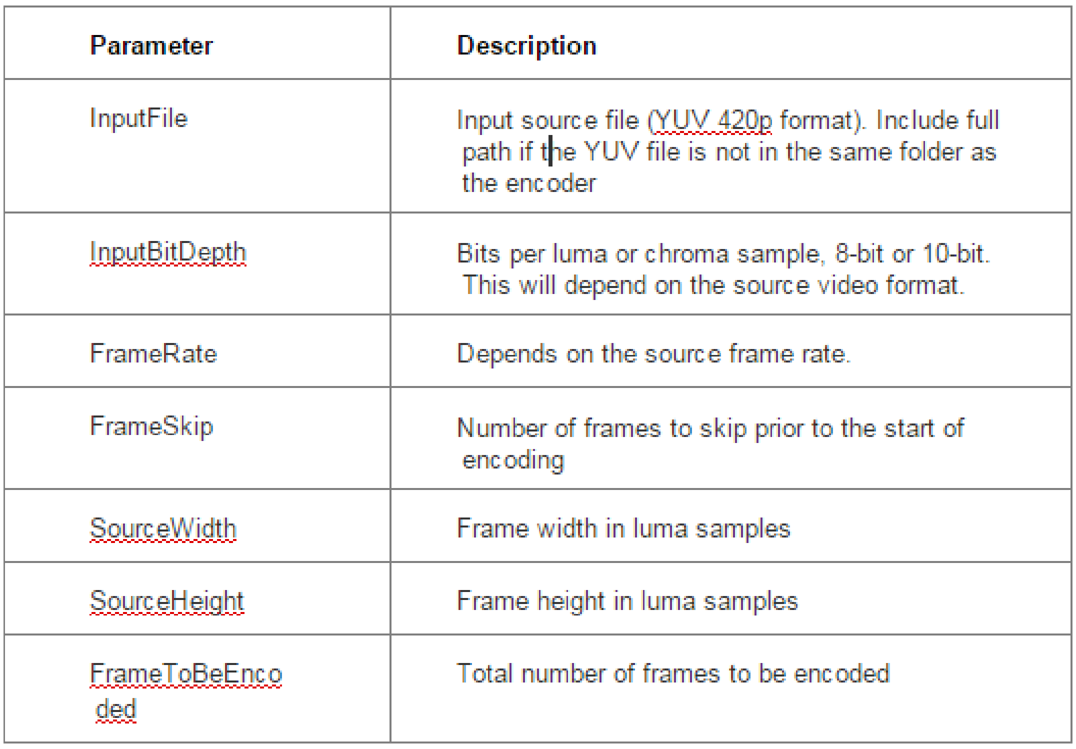
Codec configuration files can be found in the folder trunk/cfg. Each configuration file operates the encoder in a specific mode. For example, encoder_intra_main.cfg operates the encoder using only Intra encoding (no motion compensation), with tools

supported by the Main Profile. Key parameters in the .cfg file include:



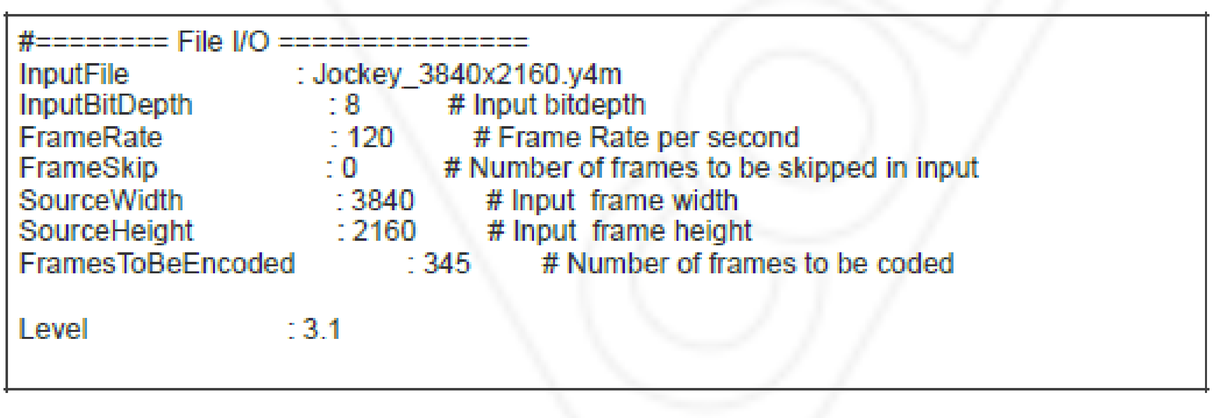
Parameter	Description
BitStreamFile	Compressed file (.bin)
ReconFile	Reconstructed YUV file (.YUV). This is identical to the output that would be produced by decoding the compressed file.
QP	Quantisation parameter. A higher QP results in more compression (i.e. a smaller compressed file) and lower decoded image quality.

Sequence specific configuration files are available in folder /trunk/cfg/per-sequence. You should customise one of these files based on the input sequence. Description of the parameters are as follows:



Parameter	Description
InputFile	Input source file (YUV 420p format). Include full path if the YUV file is not in the same folder as the encoder
InputBitDepth	Bits per luma or chroma sample, 8-bit or 10-bit. This will depend on the source video format.
FrameRate	Depends on the source frame rate.
FrameSkip	Number of frames to skip prior to the start of encoding
SourceWidth	Frame width in luma samples
SourceHeight	Frame height in luma samples
FrameToBeEncoded	Total number of frames to be encoded

Consider the 4k test sequence Jockey obtained from http://ultravideo.cs.tut.fi/#testsequences (http://ultravideo.cs.tut.fi/#testsequences). The sequence consists of 345 frames, each with luma resolution of 3840x2160 8-bit samples. The sequence specific configuration file, jockey.cfg will be:



```
#======== File I/O ===============
InputFile                : Jockey_3840x2160.y4m
InputBitDepth               : 8         # Input bitdepth
FrameRate                   : 120        # Frame Rate per second
FrameSkip                   : 0        # Number of frames to be skipped in input
SourceWidth                  : 3840       # Input  frame width
SourceHeight                 : 2160       # Input  frame height
FramesToBeEncoded                  : 345        # Number of frames to be coded

Level                   : 3.1
```

The command line to run the encoder in random access mode:

```
./TAppEncoder -c encoder_randomaccess.cfg -c jockey.cfg
```

Note that this will take a considerable length of time. You may wish to first run a test encoding session, for example with FramesToBeEncoded set to 5, to check that everything is working correctly.

Step 5: Decoding / playing the output

The encoder outputs two files: compressed (.bin) file and reconstructed (.yuv) file. The reconstructed file is identical to the output obtained by decoding the .bin file and can be played back using any YUV player, such as GLYUVPlay. The .bin file can be decoded by a compatible HEVC decoder. For example, the GPAC Osmo4 player can decode and play the .bin file if you change the file extension from .bin to .mp4.

2 x265: Standalone encoder for HEVC

x265 is an open source software library for HEVC encoding, developed by a consortium coordinated by MulticoreWare. It is available under the GNU General Public License. The current version (March 2014) provides:

- Full support for Main profile of HEVC including 8-bits per sample, YUV 4:2:0 format
- Limited support for Main 10 profile.

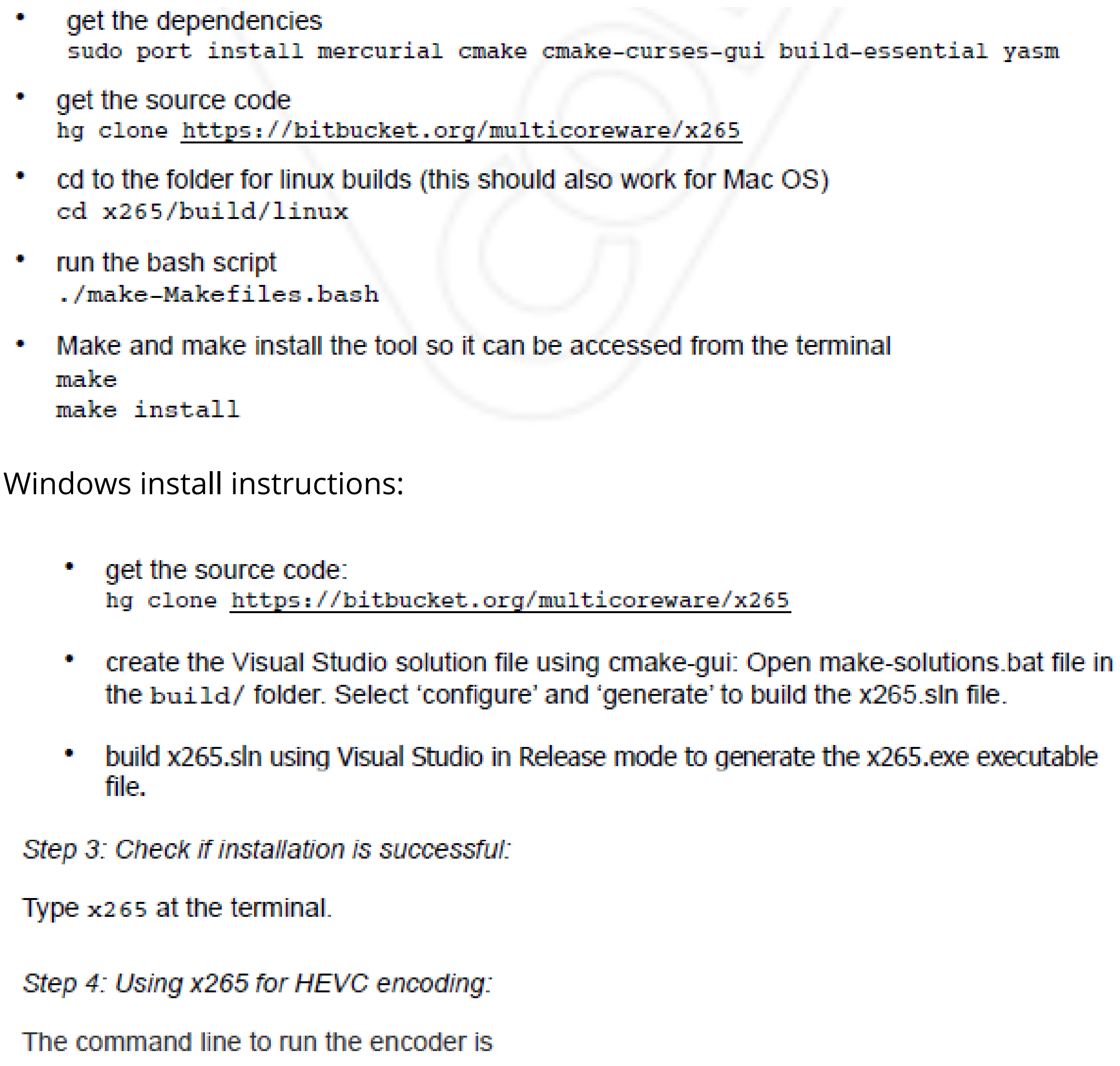
The software can be used as a standalone tool or as a library with FFMPEG. The download and installation instructions are as follows

Step 1: Get the source code and install instructions from:

https://bitbucket.org/multicoreware/x265/wiki/Home (https://bitbucket.org/multicoreware/x265/wiki/Home)

Step 2: Install x265

MacOS / Linux install instructions:



- get the dependencies
  ```
  sudo port install mercurial cmake cmake-curses-gui build-essential yasm
  ```
- get the source code
  ```
  hg clone https://bitbucket.org/multicoreware/x265
  ```
- cd to the folder for linux builds (this should also work for Mac OS)
  ```
  cd x265/build/linux
  ```
- run the bash script
  ```
  ./make-Makefiles.bash
  ```
- Make and make install the tool so it can be accessed from the terminal
  ```
  make
  make install
  ```

Windows install instructions:

- get the source code:
  ```
  hg clone https://bitbucket.org/multicoreware/x265
  ```
- create the Visual Studio solution file using cmake-gui: Open make-solutions.bat file in the `build/` folder. Select 'configure' and 'generate' to build the x265.sln file.
- build x265.sln using Visual Studio in Release mode to generate the x265.exe executable file.

Step 3: Check if installation is successful:

Type `x265` at the terminal.

Step 4: Using x265 for HEVC encoding:

The command line to run the encoder is

```
x265 [options] infile [-o] outfile
```

where “infile” can be YUV or Y4M and “outfile” is a compressed HEVC bitstream.

Input options include:

--input	Raw YUV or Y4M input file name
--input-depth	Bit-depth of input file, default 8
--input-res	Source picture size [w x h], auto-detected if Y4M
--input-csp	Source color space parameter, auto-detected if Y4M. 1:i420 3:i444, Default: 1
--fps	Source frame rate (float or num/denom), auto-detected if Y4M
--seek	First frame to encode
-f/--frames	Maximum number of frames to encode. Default all

Executable options:

-h/--h	Show help text and exit
-p/--preset	ultrafast, veryfast, faster, fast, medium, slow, slower, veryslow, or placebo. Generally a slower preset will give better compression and/or quality.
--threads	Number of threads for thread pool (0: detect CPU core count, default)
-o/--output	Bitstream output file name (.265 file)

QP, rate control and rate distortion options:

--bitrate	Target bitrate (kbps), implies ABR. Default 0
-q/--qp	Base QP for CQP mode. Default 32
--crf	Quality-based VBR (0-51). Default 28.000000
--rd	Level of RD in mode decision 0:least....6:full RDO. Default 3

Reconstructed video options:

-r/--recon	Reconstructed raw image YUV or Y4M output file name
--recon-depth	Bit-depth of reconstructed raw image file, defaults to input bit depth

Step 5: Using x265 as a library in the FFMPEG framework

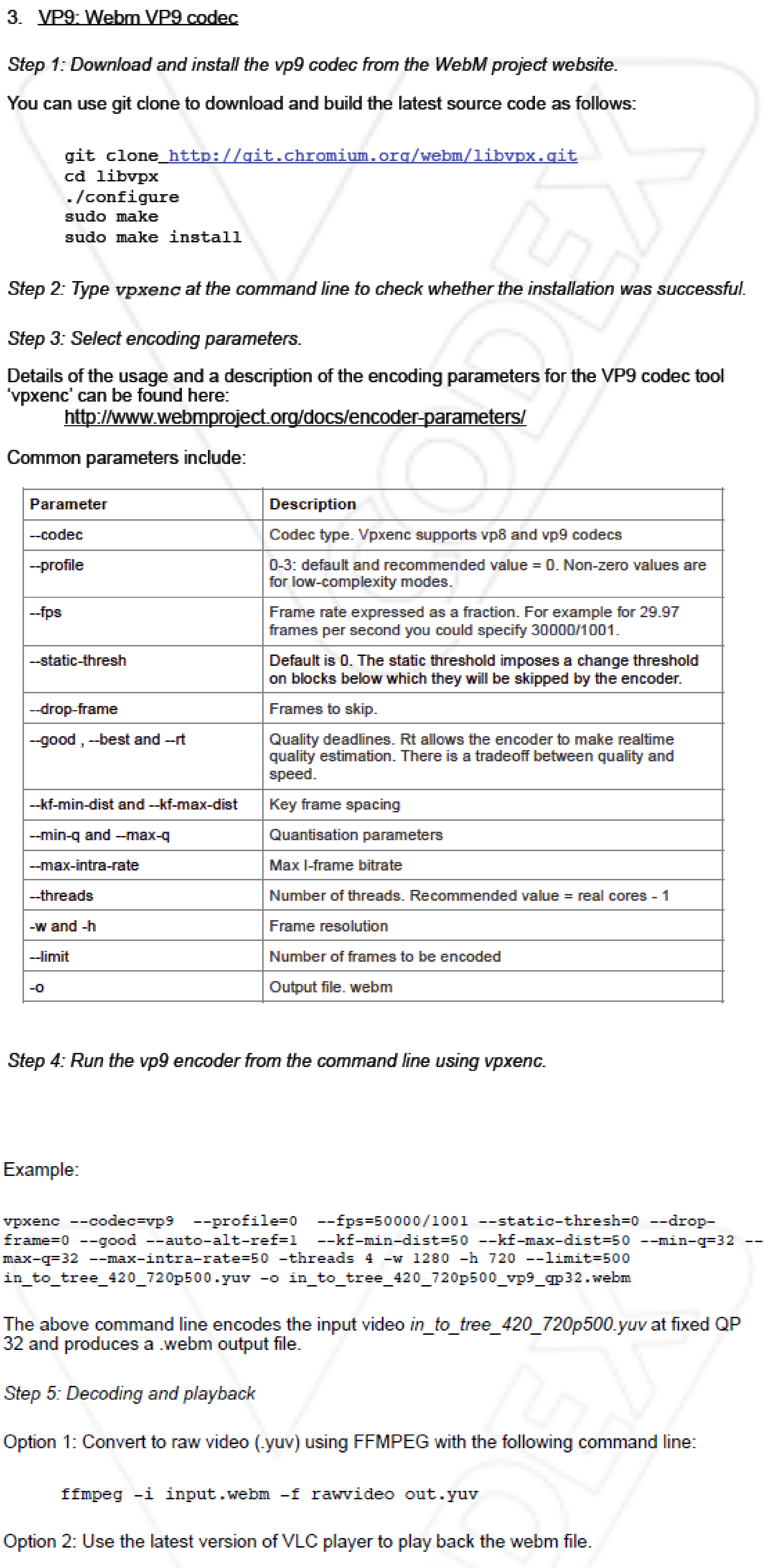
To access the x265 encoder via FFMPEG, install x265 by following steps 1-4 and re-configure FFMPEG with the switch `-enable-libx265`. Install and build instructions for FFMPEG are available at:

http://www.ffmpeg.org/download.html

The command line for using the x265 library via FFMPEG is:

```
ffmpeg -i [input] -c:v libx265 [options] output[.265/.mkv]
```

Note: FFMPEG currently does not support the MP4 file format for HEVC files. You can use the tool MP4Box to convert .265 to mp4 format.



3. VP9: Webm VP9 codec

Step 1: Download and install the vp9 codec from the WebM project website.

You can use git clone to download and build the latest source code as follows:

```
git clone http://git.chromium.org/webm/libvpx.git
cd libvpx
./configure
sudo make
sudo make install
```

Step 2: Type `vpxenc` *at the command line to check whether the installation was successful.*

Step 3: Select encoding parameters.

Details of the usage and a description of the encoding parameters for the VP9 codec tool 'vpxenc' can be found here:
http://www.webmproject.org/docs/encoder-parameters/

Common parameters include:

Parameter	Description
--codec	Codec type. Vpxenc supports vp8 and vp9 codecs
--profile	0-3: default and recommended value = 0. Non-zero values are for low-complexity modes.
--fps	Frame rate expressed as a fraction. For example for 29.97 frames per second you could specify 30000/1001.
--static-thresh	Default is 0. The static threshold imposes a change threshold on blocks below which they will be skipped by the encoder.
--drop-frame	Frames to skip.
--good , --best and --rt	Quality deadlines. Rt allows the encoder to make realtime quality estimation. There is a tradeoff between quality and speed.
--kf-min-dist and --kf-max-dist	Key frame spacing
--min-q and --max-q	Quantisation parameters
--max-intra-rate	Max I-frame bitrate
--threads	Number of threads. Recommended value = real cores - 1
-w and -h	Frame resolution
--limit	Number of frames to be encoded
-o	Output file. webm

Step 4: Run the vp9 encoder from the command line using vpxenc.

Example:

```
vpxenc --codec=vp9  --profile=0  --fps=50000/1001 --static-thresh=0 --drop-
frame=0 --good --auto-alt-ref=1  --kf-min-dist=50 --kf-max-dist=50 --min-q=32 --
max-q=32 --max-intra-rate=50 -threads 4 -w 1280 -h 720 --limit=500
in_to_tree_420_720p500.yuv -o in_to_tree_420_720p500_vp9_qp32.webm
```

The above command line encodes the input video *in_to_tree_420_720p500.yuv* at fixed QP 32 and produces a .webm output file.

Step 5: Decoding and playback

Option 1: Convert to raw video (.yuv) using FFMPEG with the following command line:

```
ffmpeg -i input.webm -f rawvideo out.yuv
```

Option 2: Use the latest version of VLC player to play back the webm file.

About the authors

Abharana Bhat specializes in video compression, media streaming and video quality assessment. Her PhD thesis on quality assessment of compressed video included the development of a novel, fully automated video quality metric.

Iain Richardson is the author of “The H.264 Advanced Video Compression Standard”, a widely cited work in the research literature. He has written three further books and over 50 journal and conference papers on image and video compression. He regularly advises companies on video codec technology, video coding patents and mergers/acquisitions in the video coding industry. Professor Richardson leads an internationally renowned image and video coding research team, contributes to the MPEG industry standards group and is sought after as an expert witness and litigation consultant.

Vcodex Limited, 35 Regent Quay, Aberdeen AB11 5BE Scotland • USA: **+1 803 620 1732 (tel:+18036201732)** • UK: **+44 1224 595939 (tel:+441224595939)** • **info@vcodex.com (mailto:info@vcodex.com)** • **in (http://www.linkedin.com/company/vcodex-limited)** • **(http://twitter.com/vcodexgroup)**