
UPDATE ON OPEN OPTIMIZED VVC IMPLEMENTATIONS

VVENC AND VVDEC

JVET-AE0266

A Wieckowski, J. Brandenburg, C. Bartnik, V. George, J. Güther,
G. Hege, C. Helmrich, A. Henkel, T. Hinz, C. Lehmann, C. Stoffers,
B. Bross, H. Schwarz, D. Marpe, T Schierl

Open optimized VVC implementations

Available on GitHub since Sep. 2020

- Fraunhofer Versatile Video Encoder (VVenc)

<https://github.com/fraunhoferhhi/vvenc>

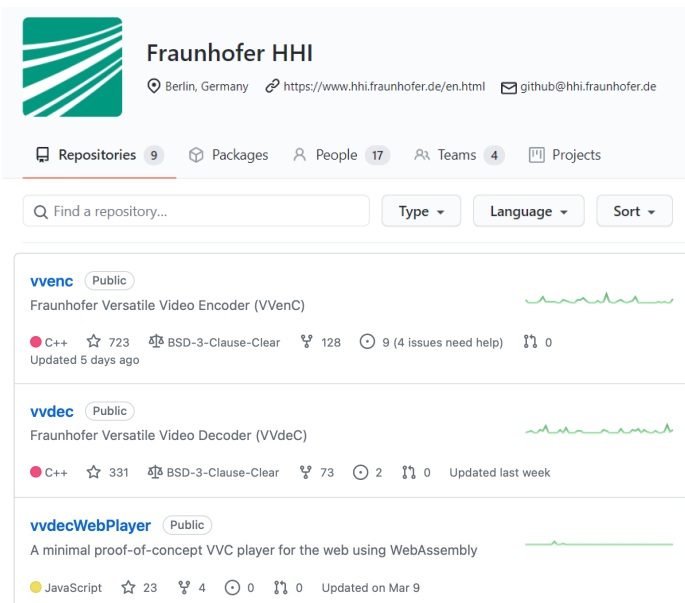
- Fraunhofer Versatile Video Decoder (VVdec)

<https://github.com/fraunhoferhhi/vvdec>

- Fraunhofer VVdec Web Player

<https://github.com/fraunhoferhhi/vvdecWebPlayer>

- Copyright BSD 3-clause clear license (since Mar. 2022)



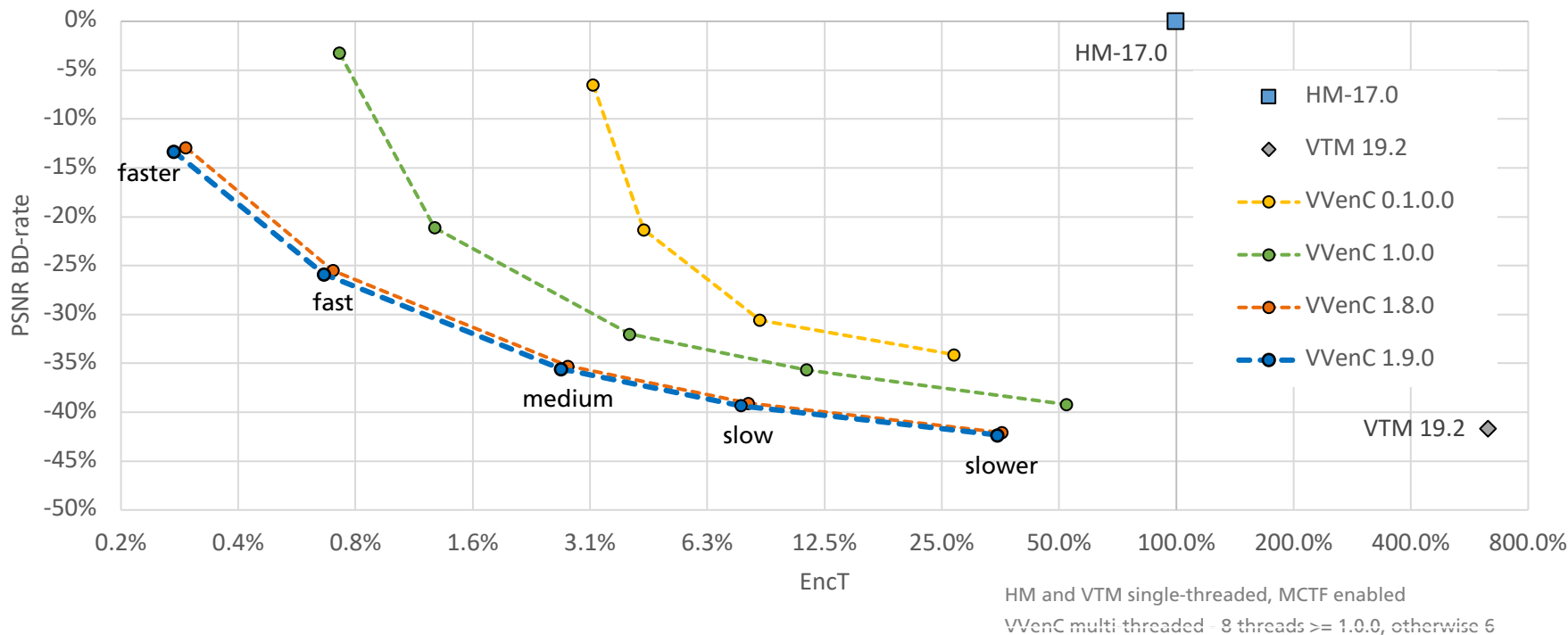
VVenC v1.9.0

Main changes since April 2023 (v1.8.0):

- New features and improvements:
 - **Rate capping** for rate control (RC) encoding (capping at minimum 1.5x target rate),
 - **First pass speedup** for 2-pass RC by temporal subsampling and spatial subsampling (experimental),
 - **Frames-in-flight parallelism support** (experimental)
- Various bugfixes and cleanups

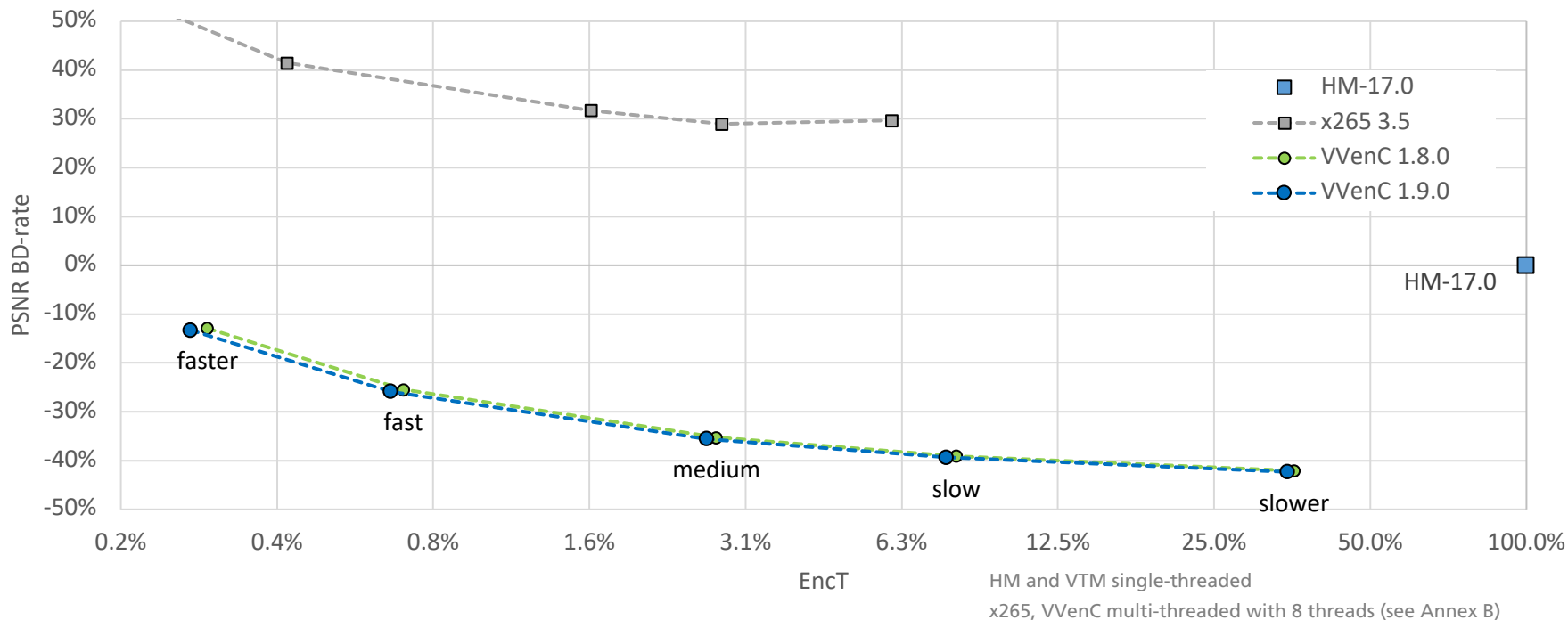
VVenC

JVET CTC Performance – YUV PSNR for HD and UHD (PSNR optimized)



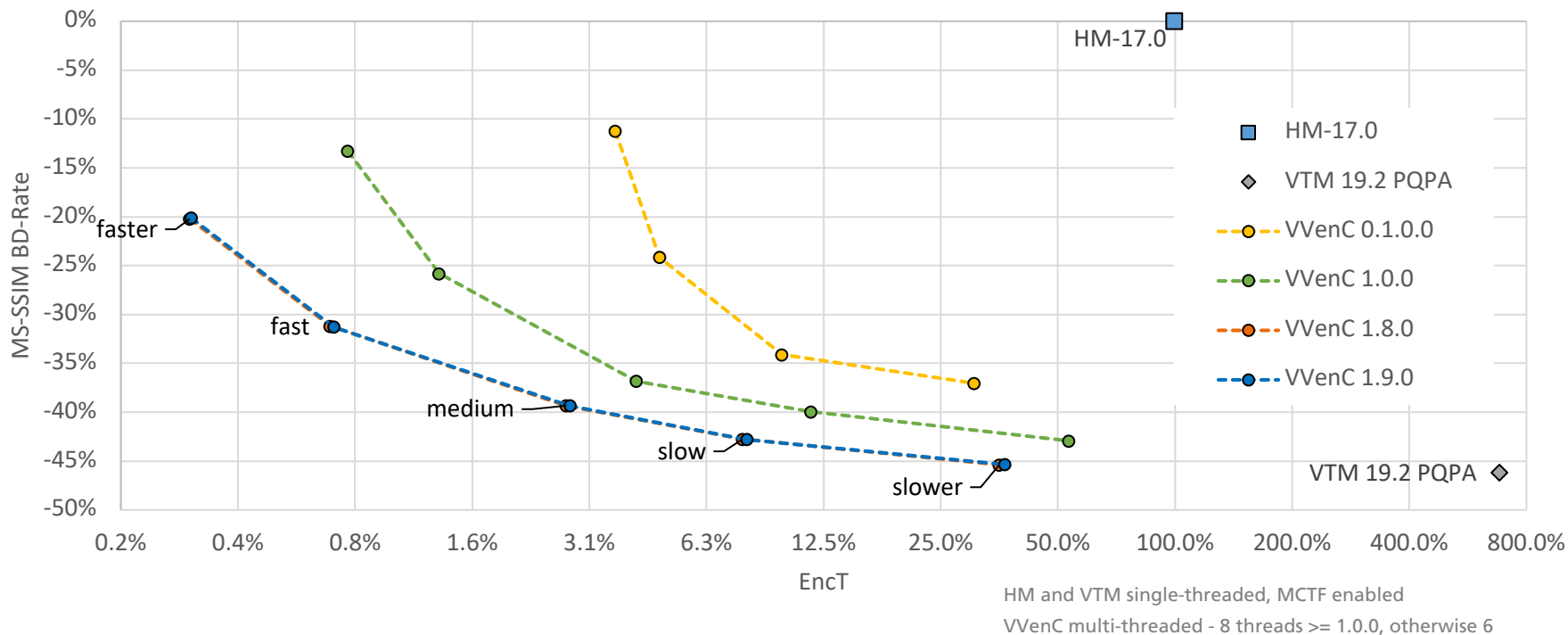
VVenC

JVET CTC Performance – YUV PSNR for HD and UHD (PSNR optimized)



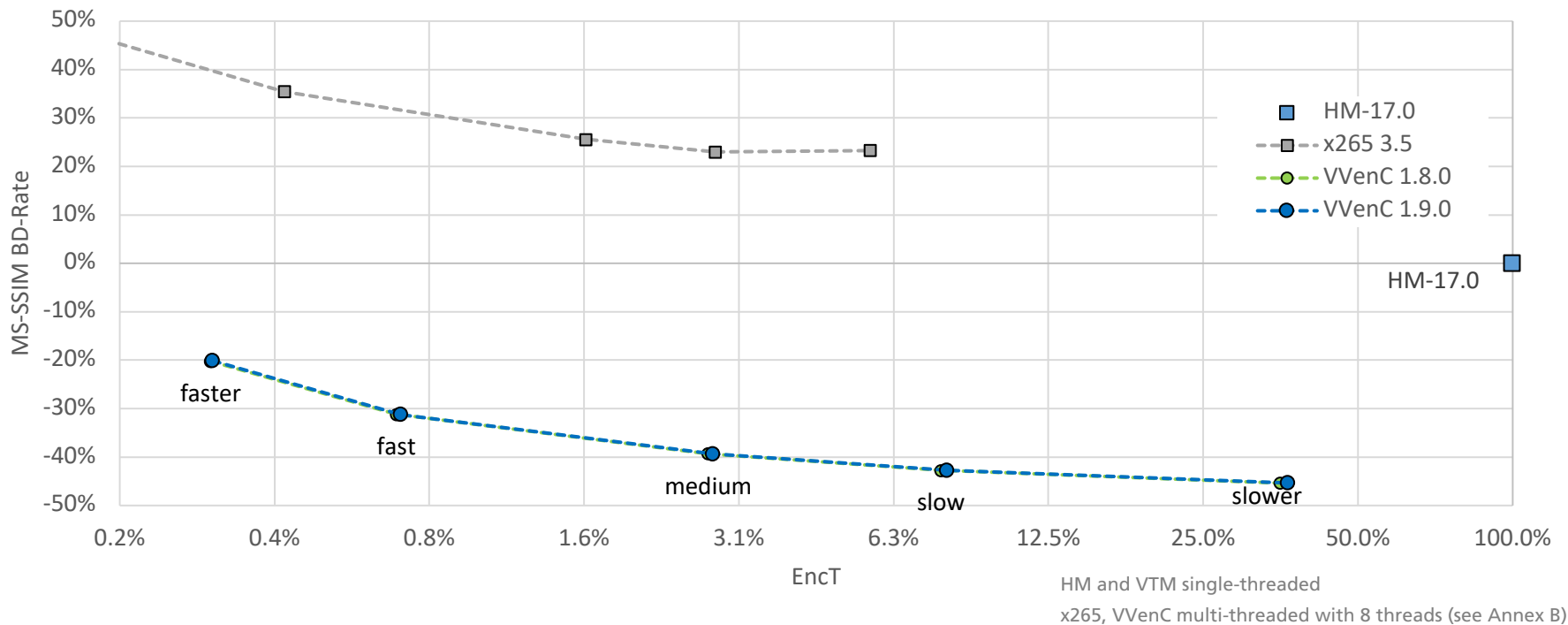
VVenC

JVET CTC Performance – YUV MS-SSIM for HD and UHD (QPA)



VVenC

JVET CTC Performance – YUV MS-SSIM for HD and UHD (QPA)



VVdeC v2.1.0

Main changes since April 2023 (v1.6.1):

- New features and improvements:
 - Error resilience
(initial support)
 - Native ARMv8 support
(optimizing SIMD functions)
- Various speedups, bugfixes and cleanups

Android Playback

- aarch64 build of VVdeC on Android:
 - MPV media player
 - ExoPlayer enabling VVC + MPEG-H demo by Fraunhofer HHI + IIS
- 1080p60 10bit @ 3Mbit/s VVC playback
 - Galaxy 7+, Galaxy 8
- 5h non-stop playback with energy optimized encoder settings as described in the ICIP 2022 paper*

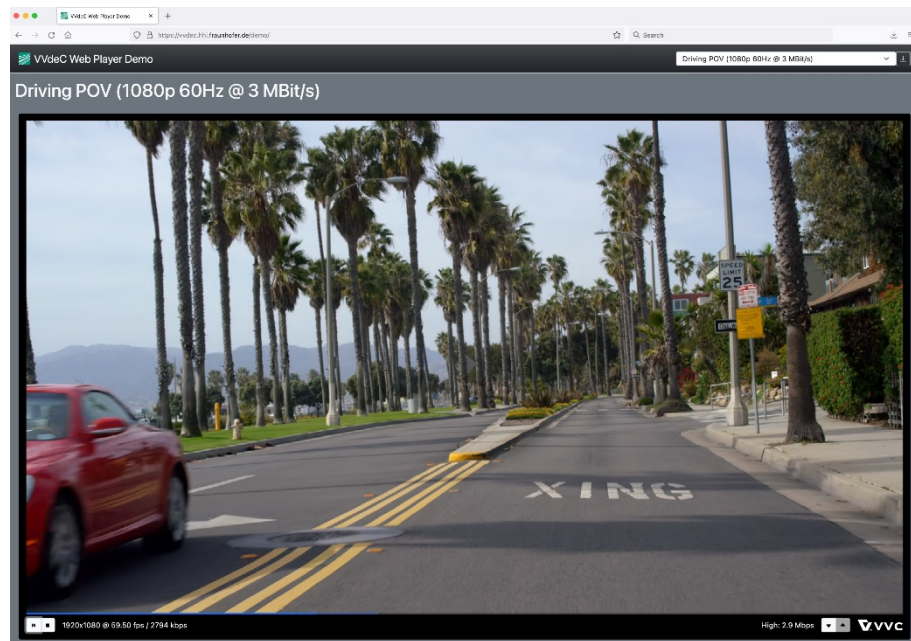
<https://arxiv.org/abs/2206.13483>



VVdeC

Browser Playback with WebAssembly

- Simple web player using
 - WebAssembly (WASM) build of VVdeC
 - WebGL for playback
 - MP4 support using mp4box.js
 - DASH support
- 1080p60 10bit @ 3Mbit/s playback in Edge, Firefox, Chrome on MacBook M1
- Sources available on [GitHub](https://github.com/fraunhoferhhi/vvdecWebPlayer):
<https://github.com/fraunhoferhhi/vvdecWebPlayer>



End-to-End Open-Source VVC Toolchain

Components used in a full VVC open-source toolchain

■ Encoding

- **VVenC** – standalone VVC encoder

■ Muxing – Transport – Demuxing

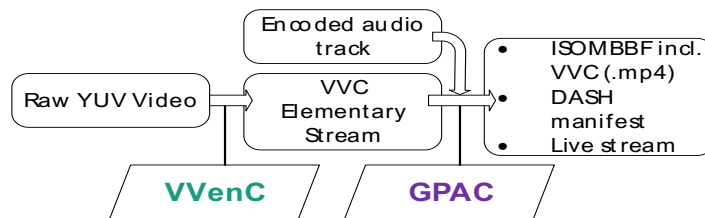
- **GPAC** – integration in upstream repo
- Supports open GOP resolution switching*
- VVC integration dependent on **FFmpeg** (**libavcodec**)

■ Decoding

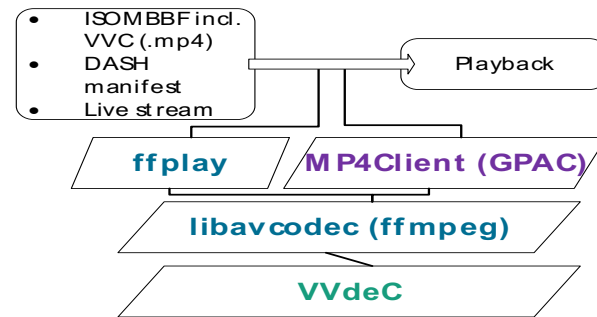
- **VVdeC** – library embedded in FFmpeg

■ Playback

- **FFmpeg** (**libavcodec**) – in a fork



a) Encoding/Muxing



b) Decoding/Demuxing/Playback

End-to-End Open Source VVC Toolchain

Where to find the and how to setup the SW

- **VVenC:** <https://github.com/fraunhoferhhi/vvenc>
 - *make install install-prefix=/usr/local*
- **VVdeC:** <https://github.com/fraunhoferhhi/vvdec>
 - *make install install-prefix=/usr/local*
- **FFmpeg:** <https://github.com/tbiat/FFmpeg>
 - *./configure --enable-libvvdec --enable-pic --enable-openssl --enable-libxml2*
 - *make -j && sudo make install*
- **GPAC:** <https://github.com/gpac/gpac>
 - *./configure --extra-ldflags='-Wl,-Bsymbolic' --extra-lldflags='-pie'*
 - *make -j && sudo make install*
- **ACM Open Access Paper:** <https://dl.acm.org/doi/10.1145/3474085.3478320>

End-to-End Open Source VVC Toolchain

FFmpeg patches with VVC support through VVenC and VVdeC

- Instructions how to build FFmpeg using these patches:

<https://github.com/fraunhoferhhi/vvenc/wiki/FFmpeg-Integration>

- Submitted by 3rd party proponent (SpinDigital) und regularly updated (currently v7)

<https://patchwork.ffmpeg.org/project/ffmpeg/list/?series=8577>

Patch	Series	A/L/R/T	S/W/F	▲ Date	Submitter	Delegate	State
[FFmpeg-devel,v7,11/11] avcodec: increase minor version for H266/VVC	Add support for H266/VVC	----	2 - -	2023-03-21	Thomas Siedel		New
[FFmpeg-devel,v7,10/11] avformat: add ts stream types for H266/VVC	Add support for H266/VVC	----	2 - -	2023-03-21	Thomas Siedel		New
[FFmpeg-devel,v7,09/11] avcodec: add external encoder libvvenc for H266/VVC	Add support for H266/VVC	----	2 - -	2023-03-21	Thomas Siedel		New
[FFmpeg-devel,v7,08/11] avcodec: add external decoder libvvddec for H266/VVC	Add support for H266/VVC	----	2 - -	2023-03-21	Thomas Siedel		New
[FFmpeg-devel,v7,07/11] avformat: add muxer support for H266/VVC	Add support for H266/VVC	----	2 - -	2023-03-21	Thomas Siedel		New
[FFmpeg-devel,v7,06/11] avformat: add demuxer and probe support for H266/VVC	Add support for H266/VVC	----	2 - -	2023-03-21	Thomas Siedel		New
[FFmpeg-devel,v7,05/11] avcodec: add MP4 to annexb support for H266/VVC	Add support for H266/VVC	----	2 - -	2023-03-21	Thomas Siedel		New
[FFmpeg-devel,v7,04/11] avcodec: add h266_metadata_bsf support for H266/VVC	Add support for H266/VVC	----	2 - -	2023-03-21	Thomas Siedel		New
[FFmpeg-devel,v7,03/11] avcodec: add bitstream parser for H266/VVC	Add support for H266/VVC	----	2 - -	2023-03-21	Thomas Siedel		New
[FFmpeg-devel,v7,02/11] avcodec: add cbs for H266/VVC	Add support for H266/VVC	----	2 - -	2023-03-21	Thomas Siedel		New
[FFmpeg-devel,v7,01/11] avcodec: add enum types for H266/VVC	Add support for H266/VVC	----	2 - -	2023-03-21	Thomas Siedel		New

Annex A – Additional resources

VVenC and VVdeC wiki pages

Most recent information on:

- How to build?
- How to use?
- Performance
- Publications

<https://github.com/fraunhoferhhi/vvenc/wiki>

<https://github.com/fraunhoferhhi/vvdec/wiki>

Home

Christian Lehmann edited this page 7 days ago · 46 revisions

Build	Usage	Performance
Compiling, installing and developing for VVenC	Using VVenC	Encoder Performance and comparisons

✦ **VVenC: Fraunhofer Versatile Video Encoder**

VVenC is a fast and efficient VVC encoder implementation. After the release of the initial version in September 2020, performance and runtime are improved in each next version. The graph below shows the presets' development over time in multi-threaded operation (0.1-0.2: 6 threads, 8 threads for later versions).

Clone this wiki locally
<https://github.com/fraunhoferhhi/vvenc/wiki>

Home

Adam Wackowski edited this page on 20 Apr · 51 revisions

Build	Usage
Compiling, installing and developing for VVdeC	Using VVdeC

✦ **VVdeC: Fraunhofer Versatile Video Decoder**

Introduction

In July 2020 the Joint Video Experts Team (JVET), a collaborative project of the ITU-T Video Coding Experts Group (VCEG) and ISO/IEC Moving Picture Experts Group (MPEG), has finalized a new video coding standard called Versatile Video Coding (VVC)¹². VVC is the successor of the High Efficiency Video Coding (HEVC) standard¹³ and has been published by ITU-T as H.266 and by ISO/IEC as MPEG-H Part 3 (ISO/IEC 23090-3). The new standard targets a 50% bit-rate reduction over HEVC at the same visual quality. In addition, VVC proves to be truly versatile by including tools for efficient coding of video content in emerging applications, e.g. high dynamic range (HDR), adaptive streaming, computer generated content as well as immersive applications like 360-degree video and augmented reality (AR).

The Fraunhofer Versatile Video Decoder (VVdeC) development was initiated to provide a publicly available and fast VVC decoder implementation. The VVdeC software is based on VVC Test Model (VTM), with optimizations including software redesign to mitigate performance bottlenecks, extensive SIMD optimizations and extensive multi-threading support to exploit parallelization.

VVdeC can decode raw bitstreams created by any VVC standard compliant encoder, e.g. the

Clone this wiki locally
<https://github.com/fraunhoferhhi/vvdec/wiki>

Annex A – Additional resources

Publications

- J. Brandenburg et al., "**Towards Fast and Efficient VVC Encoding**", IEEE 22nd Workshop on Multimedia Signal Processing (MMSP 2020), Tampere, Finland, 2020.
- B. Bross, C. Helmrich, A. Wieckowski "**Versatile Video Coding – Open Optimized Implementations**", Workshop on the IEEE Picture Coding Symposium (PCS) 2021, Jul. 2021. <https://youtu.be/IWPBuS2diVg>
- A. Wieckowski et al., "**VVenC: An Open And Optimized VVC Encoder Implementation**" 2021 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), 2021, pp. 1-2. <https://ieeexplore.ieee.org/document/9455944>
- C. R. Helmrich et al., "**A study of the extended perceptually weighted peak signal-to-noise ratio (XPSNR) for video compression with different resolutions and bit depths**", in ITU Journal: ICT Discoveries, vol. 3, no. 1, May 2020. <http://handle.itu.int/11.1002/pub/8153d78b-en>
- R. Skupin, C. Bartnik, A. Wieckowski, Y. Sanchez, B. Bross, C. Hellge, and T. Schierl, "**Open GOP Resolution Switching in HTTP Adaptive Streaming with VVC**," 35th Picture Coding Symposium (PCS), Bristol, US, June-July 2021.
- R. Skupin, C. Bartnik, A. Wieckowski, Y. Sanchez, B. Bross, "**Constrained RASL encoding for bitstream switching**," document JVET-W0133, Joint Video Experts Team (JVET), July 2021.
- M. Kränzler et al., "**Optimized Decoding-Energy-Aware Encoding In Practical VVC Implementations**," 2022 IEEE International Conference on Image Processing (ICIP), 2022, pp. 1536-1540.

Annex B – Encoder comparison settings

Encoding with preset P for quality Q

- HD and UHD sequences from JVET common test conditions JVET-T2010:

https://jvet-experts.org/doc_end_user/documents/20_Teleconference/wg11/JVET-T2010-v2.zip

- Command line options for x265* encoder (no sequence specific parameters) tuned for PSNR (**T** = **psnr**) or MS-SSIM (**T** = **ssim**)

```
-D 10 --preset P --tune T --crf Q --keyint <1s> --min-keyint <1s> --profile main10 --  
output-depth 10
```

* optimal multithreading with x265 is achieved by restricting the number of CPU cores to be used to the desired number of threads (e.g. 8) since x265 determines the best combination of frame level and WPP parallel threads.