VVenC & VVdeC

OPEN SOURCE VIDEO ENCODING AND PLAYBACK FOR VVC

H.264/AVC – x264
H.265/HEVC – x265
H.266/VVC – VVenC?

History, current state, and ecosystem around open source VVC implementations.

About VVC…

50% more video compression, same quality

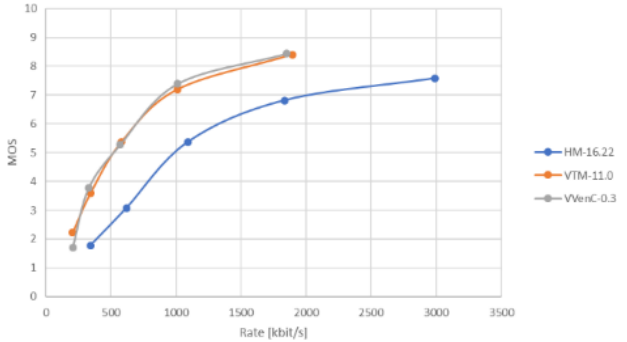
H.266/VVC (Versatile Video Coding)

- Developed by JVET
  (Joint Video Experts Group)
- Finalized in **July 2020**
  - VVC Test Model (**VTM**) version >= 10.0
- Large bit-rate savings
  - **~50% vs HEVC**
- Applicable in Versatile scenarios
  - **Screen content**
  - HDR, Immersive, 8K…
  - **Open GOP** adaptive streaming

UHD

Source: Baroncini, Wien, "JVET-T2020"

HD



Source: Baroncini, Wien, "JVET-V2020"

Fraunhofer HHI

History of VVdeC and VVenC

The team behind VVenC and VVdeC

Video Coding Systems Group at Fraunhofer HHI

- Fraunhofer HHI
  - German **non-profit**
  - Part of biggest European Research Organization
- Video Coding System Group
  - Main developers
- Your presenter… (me)
  - **Adam Wieckowski**
  - HHI since 2016
  - Project Lead since 2019
  - Group Co-Head since 2022

New SW required for compression beyond HEVC

From HM, through NextSoftware, to VTM…

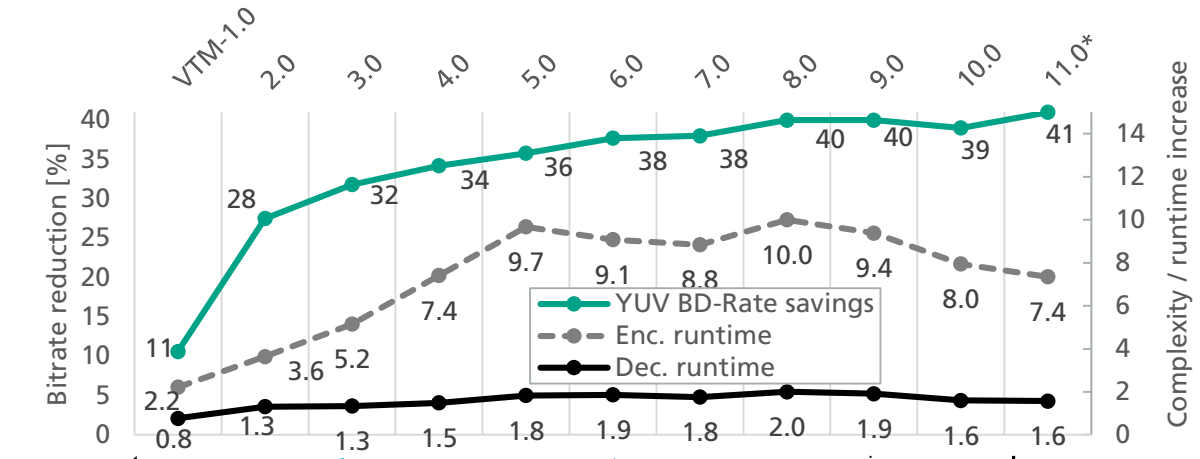
- All modern video codecs work block based
- HEVC used Quadtree -> only **square** partitions
  - HM uses **z-index** for block indexing
  - Amazing idea… for square blocks
- VVC -> rectangular partitions
  - JEM, based on HM, using z-index
  - Lots of code for working around z-index

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Y\X	0	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60
2	0	0	1	4	5	16	17	20	21	64	65	68	69	80	81	84	85
3	4	2	3	6	7	18	19	22	23	66	67	70	71	82	83	86	87
4	8	8	9	12	13	24	25	28	29	72	73	76	77	88	89	92	93
5	12	10	11	14	15	26	27	30	31	74	75	78	79	90	91	94	95
6	16	32	33	36	37	48	49	52	53	96	97	100	101	112	113	116	117
7	20	34	35	38	39	50	51	54	55	98	99	102	103	114	115	118	119
8	24	40	41	44	45	56	57	60	61	104	105	108	109	120	121	124	125
9	28	42	43	46	47	58	59	62	63	106	107	110	111	122	123	126	127
10	32	128	129	132	133	144	145	148	149	192	193	196	197	208	209	212	213
11	36	130	131	134	135	146	147	150	151	194	195	198	199	210	211	214	215
12	40	136	137	140	141	152	153	156	157	200	201	204	205	216	217	220	221
13	44	138	139	142	143	154	155	158	159	202	203	206	207	218	219	222	223
14	48	160	161	164	165	176	177	180	181	224	225	228	229	240	241	244	245
15	52	162	163	166	167	178	179	182	183	226	227	230	231	242	243	246	247
16	56	168	169	172	173	184	185	188	189	232	233	236	237	248	249	252	253
17	60	170	171	174	175	186	187	190	191	234	235	238	239	250	251	254	255

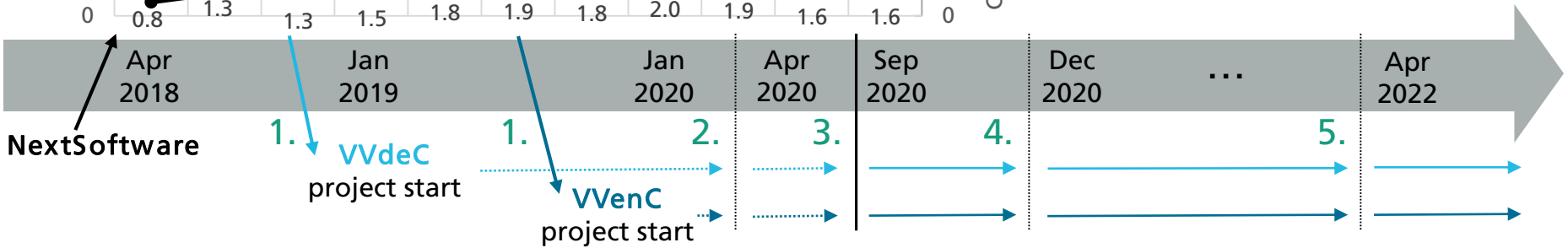
- At HHI, we started work on a **new partitioner** (it didn't make it)
  - z-index utterly unbereable
  - Our own new SW, based on HM: **NextSoftware** (later VTM-1.0)
    - Blocks described by objects, with a global map (position -> CodingUnit)

VVdeC and VVenC early in VVC development

Major project milestones



1. Start of development
2. Ben. Bross as Group Head
   - Open source idea
3. Initial public release
   - Custom license
4. Modified BSD-3
5. Unmodified **Clear BSD-3**

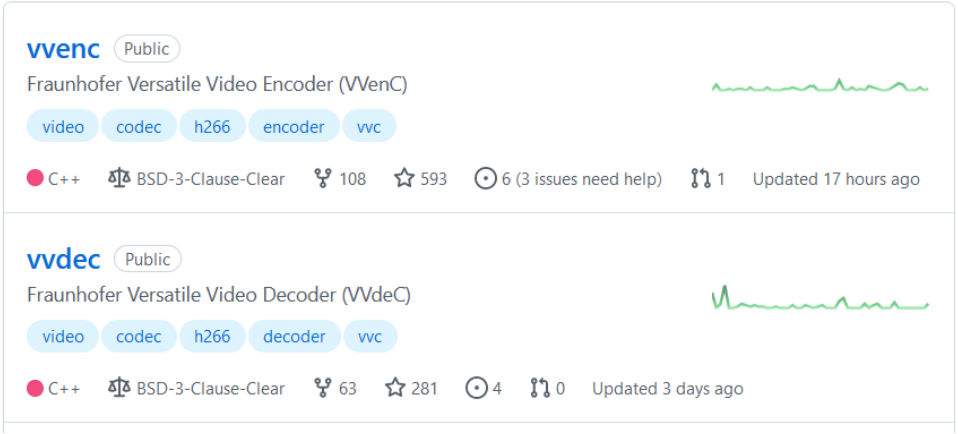


Fraunhofer HHI

VVenC and VVdeC as OSS

Hard facts about VVdeC and VVenC

- Based on VTM
- Fully in C++, with a pure-C interface
- Object based, made simple
- No assembler
  - Vectorization using intrinsics (impl. for C, SSE42/SIMDe, and AVX2)
  - ARM support for free
- Usage as simple as possible
  - Only use-case relevant options
  - Coding options optimal as-is



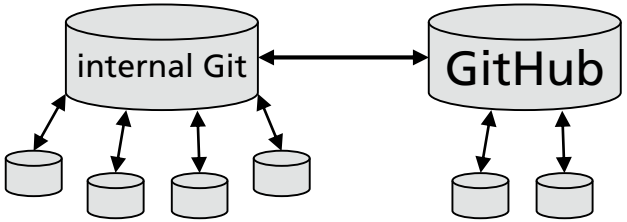
- Available on **GitHub!**
- **BSD-3-Clause-Clear** license

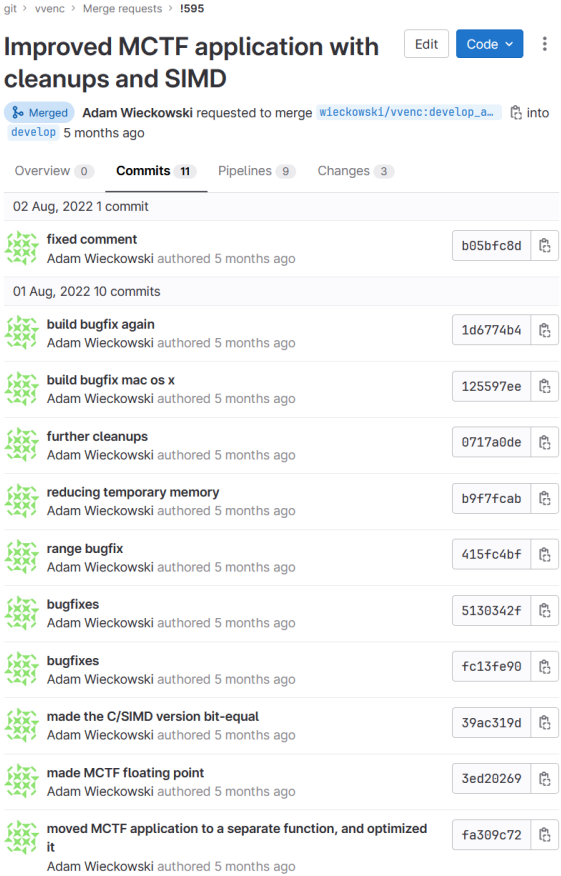
Fraunhofer HHI

Development of VVdeC and VVenC

A look behind the curtains

- Development internal, typical Git workflow
  - Internal, because… (see on the right)
- Periodical, squashed updates to GitHub
  - New versions
  - Bugs or issues
  - Early access to features
  - Rebase large contributions





Fraunhofer HHI

VVdeC

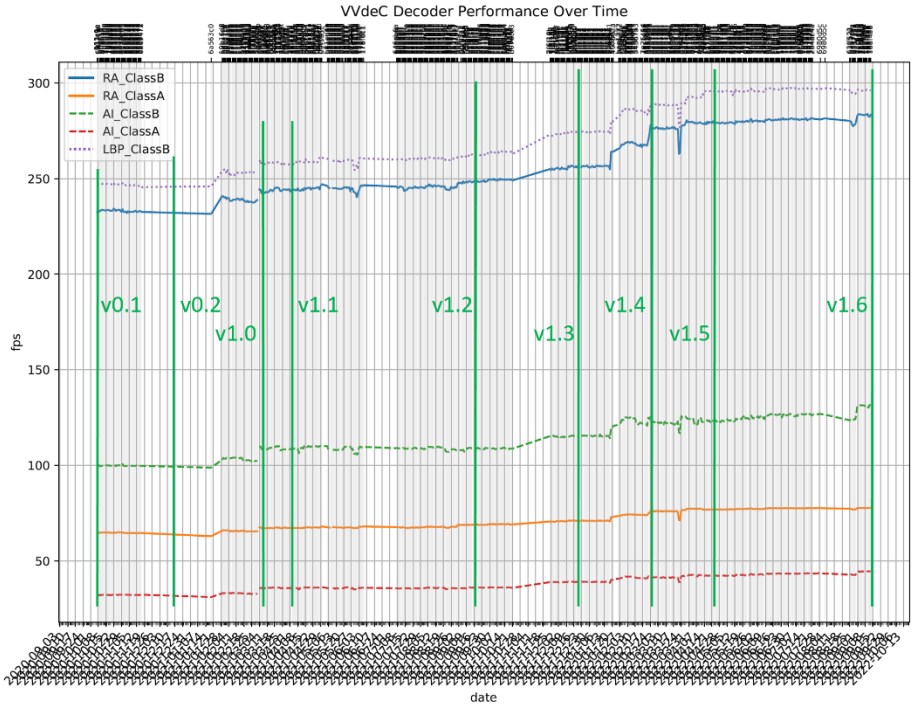
VVdeC highlights

- Fully compliant with **Main10** profile
- **Multi-Platform**
  - Windows
  - Linux (Intel, ARM, RISC-V, …)
  - Mac OS (Intel and ARM)
  - Android
- **Unified thread pool** from version 0.1
  - Can utilize 30+ threads
  - Independent of bitstream features
  - ➢ Today in many other decoders

Fraunhofer HHI

VVdeC performance since 2021

Version history after release, through the eyes of our CI



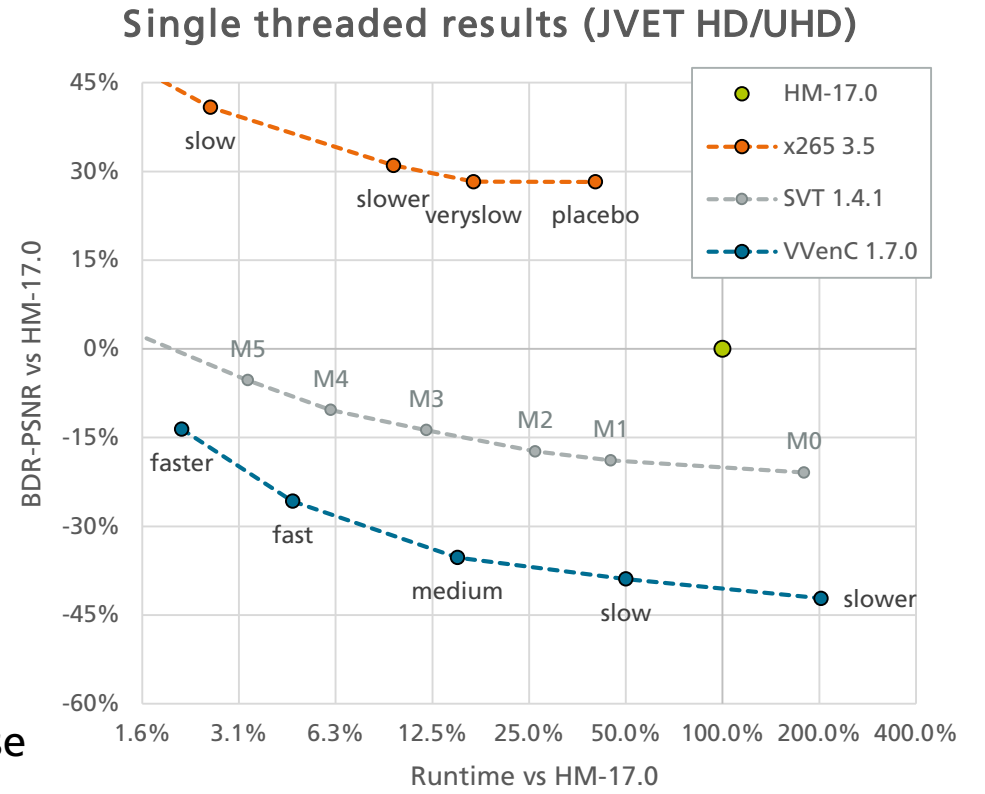
- v1.0
  - Full **compliance**
- v1.3
  - **3x** memory reduction (sic!)
- v1.4
  - External SIMD contributions
- In-between
  - A lot of small improvements

Fraunhofer HHI

VVenC

VVenC highlights

- Best open-source encoder out there!
  - Optimized for **VOD**, offline
- Five presets
  - *faster*, *fast*, *medium*, *slow*, *slower*
- Efficient multi-threading
- Optimized for **human visual system**
  - Based on the **XPSNR** metric
- Excellent rate-control (1- and 2-pass)
- Simple interface
- Academic, amateur and commercial use

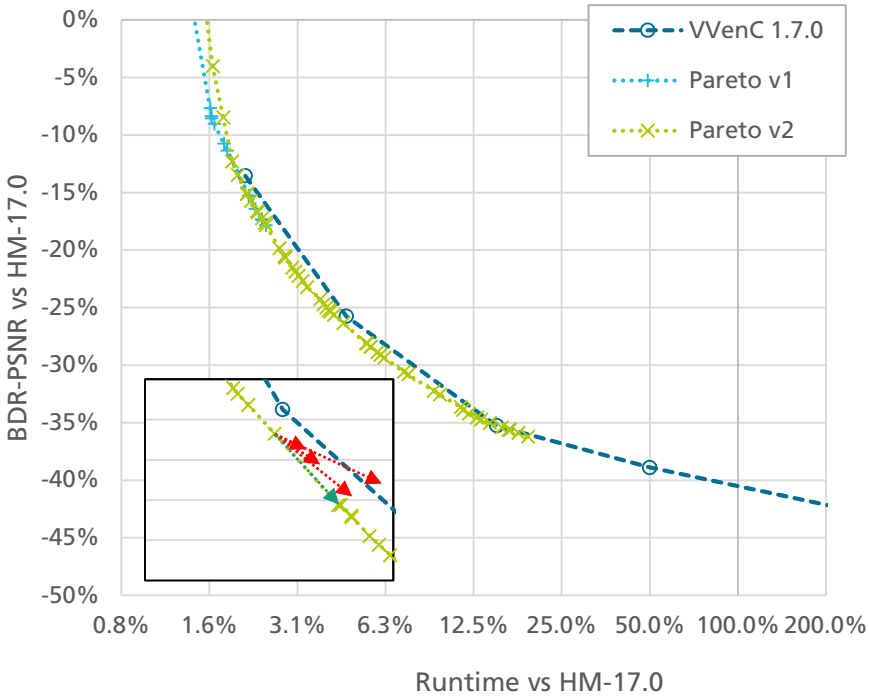


Fraunhofer HHI

VVenC preset derivation

Large scale option space optimization

- All not usecase related options
- Required to put new options in place
- Repeated every 2-3 versions
  - Target: **HD, UHD** natural content
  - Sanity check for lower res, content types
- The curve still too steep
  - SW designed for readability
  - SW efficiency still **work in progress**



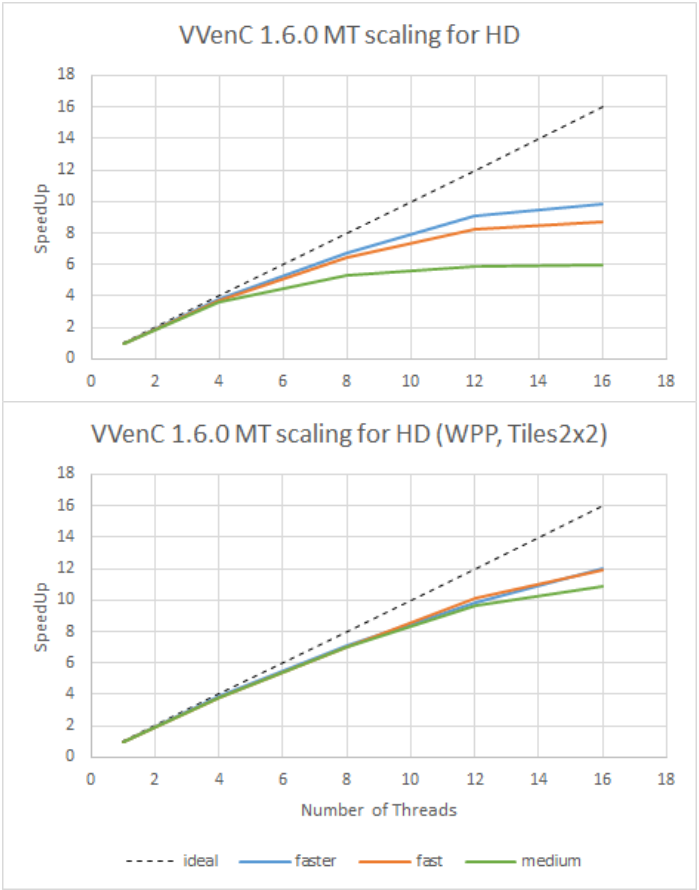
Brandenburg et al., MMSP, 2021.

Fraunhofer HHI

VVenC multi-threading

Dependency on preset, resolution, options

- Multi-threading
  - CTU- and CTU-line (simulated wavefront)
  - Independent frame parallelization
- CTU: 64 for *faster* and *fast*, 128 otherwise
  - Higher res -> more CTUs
- Better MT efficiency
  - Wavefront: no above-right dependency
  - Tiles: processed independently
  - 5% loss for *faster*, *fast*; 3% for *medium*
- Not yet ready for >= 32 threads (e.g. live)



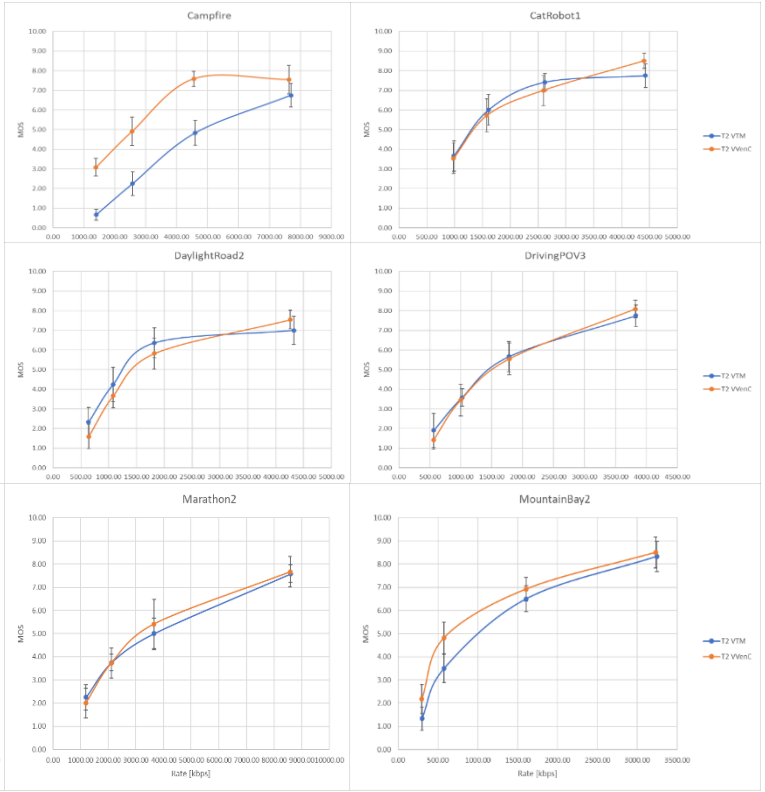
George et. al., ISM, 2022.

Fraunhofer HHI

VVenC optimization for the human visual system

Based on the extended PSNR (XPSNR)

- VVenC *slower* comparable to VTM (but faster)
- VVenC tested alongside VTM-18.2 in Oct 2022
  - *slow*, 2pRC, with **visual optimization**
  - Results comparable or better than VTM!
  - ~5% objective BDR gap compensated!
- XPSNR
  - High correlation to MOS based on multiple datasets
  - Contributed to FFmpeg



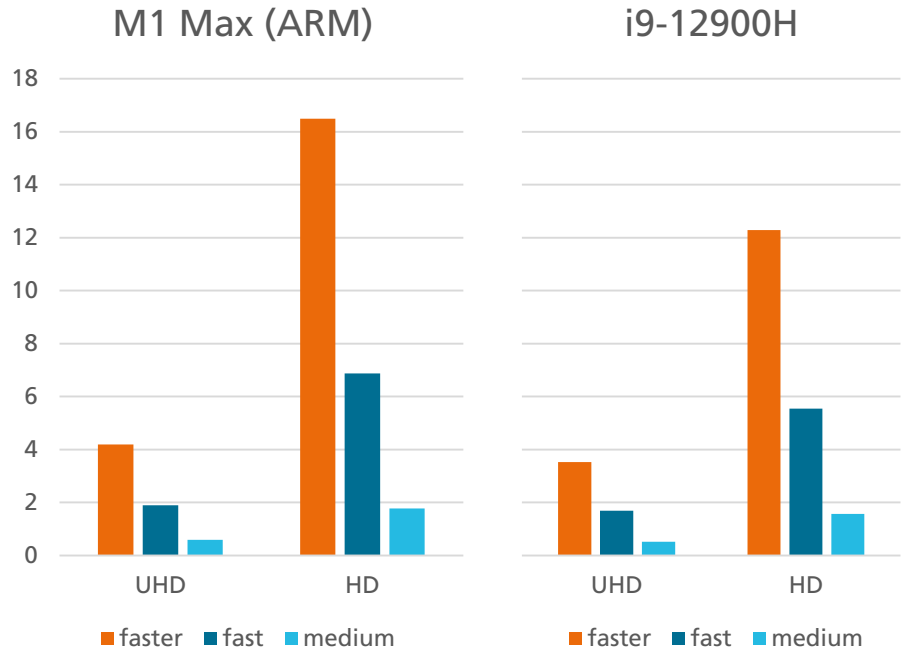
Plots: Wien et al., 2023, "JVET-AC0267".
XPNSR: Helmrich et al., ITU Journal, 2020.

Fraunhofer HHI

VVenC in practice

FPS numbers for typical mobile encodes using 8 threads

- Encodes using all-defaults
- ARM using SIMDe
  - Outperforms comparable class Intel mobile CPU
- HD encodes vs live with 8 threads
  - *faster*: x4, *fast*: x10, *medium*: x30
- UHD
  - *faster*: x15, *fast*: x30
    medium: for large scale VOD
- FPS also depends on
  - Bitrate, content, and more…

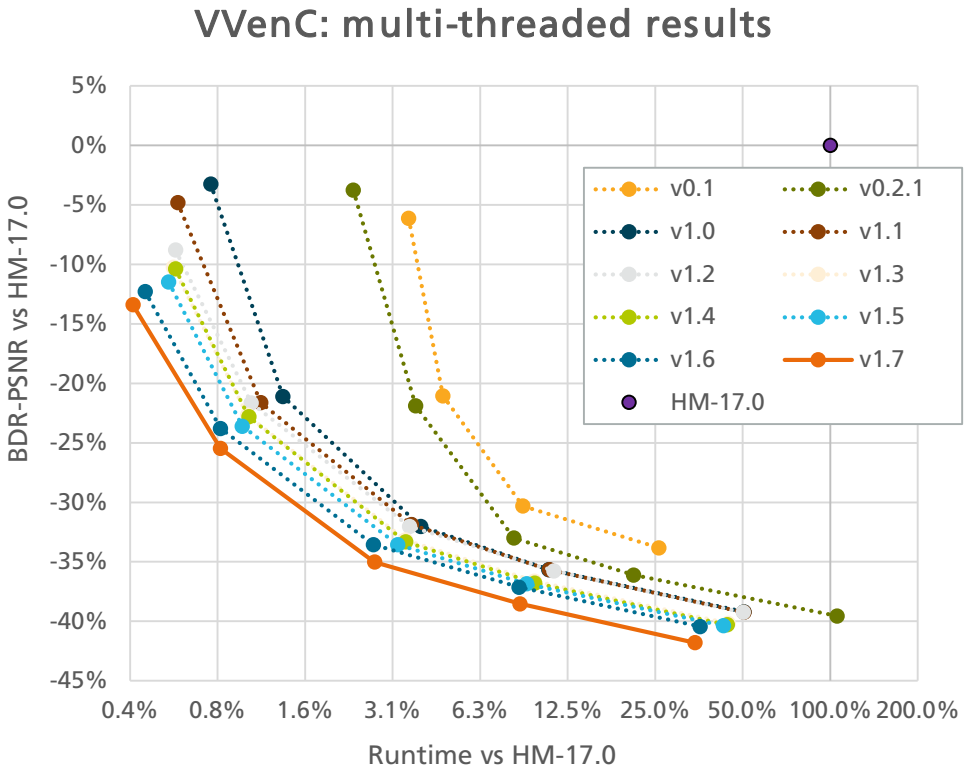


Fraunhofer HHI

VVenC improves with every release

Major milestones and future

- v0.3: **frame-threading**
- v1.0: pure-C interface
- v1.4: **scene-cut** detection
- v1.5: - arbitrary intra period
  - fast-decode preset
- v1.7: **ARM** support
- Every version
  - Improved **rate-control** thanks to community feedback!
- Future: even faster and better



Fraunhofer HHI

Ecosystem and community

Ecosystem: how to integrate it into your workflows?

Raw video en- and decoders only of academic interest

- **FFmpeg**: based on 3rd party patches, have a look at
  - Wieckowski at. al., ACMM, 2021. (Open-Access)
    https://dl.acm.org/doi/10.1145/3474085.3478320
  - Step-by-step: https://github.com/fraunhoferhhi/vvenc/wiki/FFmpeg-Integration
- Playback
  - **mpv**: link to FFmpeg with VVC
  - **VLC**: same, plus `-demux ffmpeg` (untested)
  - **ExoPlayer**: some versions are going around, ping us
  - Web-Browser: https://github.com/fraunhoferhhi/vvdecWebPlayer
- Muxing, DASHign, etc…
  - **GPAC**: link to FFmpeg with VVC

Community: why and how to be part of it?

Projects open for contribution

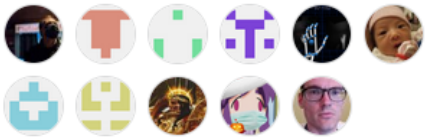
- Why contribute?
  - Best open source encoder out there
  - Interesting novel approach to video codecs
  - Authors retain copyright
- What to contribute?
  - Efficiency, speed-ups, improved implementation
  - Features…
- What to expect?
  - Throughout review, testing, generating results
  - Feedback and merge…

License

Please see LICENSE.txt file for the terms of use of the contents of this repository.

For more information, please contact:
vvc@hhi.fraunhofer.de

Contributors 11



Fraunhofer HHI

Conclusion

Recap of VVenC and VVdeC

If you just entered the room…

- Open-Source implementations of **VVC**
- Available on GitHub under the **BSD-3-Clause-Clear** License
  - https://github.com/fraunhoferhhi/vvenc
  - https://github.com/fraunhoferhhi/vvdec
- Decoder: Enabling **playback** on **mobile** devices, laptops and PCs
- Encoder: Optimized for **VOD**, **simple**, and very **efficient**
- 3rd-party integrated into FFmpeg, and other frameworks, for simple usage
  - Upstream/alternative integrations pending
- Looking forward to **contributions**, **results**, **tests**, **feedback**!

VVenC & VVdeC
OPEN SOURCE VIDEO ENCODING AND PLAYBACK FOR VVC

Thank you for your attention!

Questions?

Or: adam.wieckowski@hhi.fraunhofer.de