INTERNATIONAL ORGANISATION FOR STANDARDISATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC1/SC29/WG4
MPEG VIDEO CODING

ISO/IEC JTC1/SC29/WG4 MPEG/M54897
June 2020, Online

Source **Poznań University of Technology (PUT), Poznań, Poland**
Status **Input**
Title **On the computational time of MIV-related experiments**
Authors Jakub Szekiełda, Jakub Stankowski, Adrian Dziembowski, Dawid Mieloch

1 Introduction

This document provides an analysis and discussion on the time required for performing MIV-related experiments. With an increasing number of test sequences and experiments, the computational load is becoming unreasonable. All the steps require an equivalent of ~56 days of unparalleled work for a single computer. We propose three ideas, which significantly (9 times) reduce computational time, resulting in less than 7 days of computation.

2 TMIV6 anchor – the time of computation

Table 1. The computational time of TMIV6 anchor [M54856].

	Total time of computation [s]			
Sequence	**TMIV encoder**	**HEVC**	**TMIV decoder**	**All steps**
SA	17349	161828	391637	570815
SB	23434	121483	399018	543936
SC	9773	152043	207668	369483
SD	4437	111880	144929	261247
SE	3261	240954	108336	352550
SJ	8398	138855	248624	395877
SL	1827	194272	78863	274962
SN	7231	148876	99950	256057
SO	4876	173025	137634	315535
SP	2153	201327	59209	262688
SQ	7809	139343	277998	425150
SR	6644	144529	203413	354587
ST	1733	163501	54650	219884
SU	2121	168841	56061	227023
All	101045 s	2260759 s	2467990 s	4829795 s
	28.07 h	627.99 h	685.55 h	1341.61 h
	1.17 d	26.17 d	28.56 d	55.90 d

3 Reduction of 'ff' configuration length: 97 -> 65 frames

The main reason for the high computational time of MIV-related experiments is the number of frames needed to be processed. We propose to reduce this number while preserving the main advantages of current settings:

- realistic intra period: 32 frames,
- more than 1 GOP, what allows testing the behavior of a proposal in time.

The idea is to change the length of sequences from 3 to 2 GOPs, thus 97 to 65 frames (2 full GOPs + 1 frame at the end).

Obviously, the results would slightly change because of the characteristics of the sequences (e.g. moving objects or moving camera rig in SA or ST). However, the fair comparison between a proposal and the anchor will still be possible.

Table 2. Sequence length reduction: computational time reduction.

Total time of computation			
TMIV encoder	**HEVC**	**TMIV decoder**	**All steps**
67711 s	1514942 s	1653808 s	3236460 s
18.81 h	420.82 h	459.39 h	899.02 h
0.78 d	17.53 d	19.14 d	37.46 d
% of anchor			
67.0 %	67.0 %	67.0 %	67.0 %

4 HEVC encoder

The technology around TMIV has been designed as codec-agnostic. Therefore, there is a possibility to use arbitrary video codecs to transport TMIV data (i.e. AVC, HEVC, or VVC). Since the choice of HEVC is a reasonable compromise between widespread adoption and novelty, we suggest to keep HEVC-related compression technology but change the encoder implementation.
The HM [HM] is an HEVC reference software. It was developed with easy modification in mind, but when compared to other implementations it is not well optimized. Simple speaking HM encoder is very slow. Moreover, the TMIV related works do not include modifications of a coding technology and TMIV technology is expected to cooperate with production or consumer-grade codecs (including hardware ones).

Consequently, we suggest using another, highly-optimized HEVC encoder in order to reduce computational complexity, reduce computational time, and use a real-world HEVC encoder. We propose to use x265 [X265] encoder. The x265 is a widely adopted, highly optimized, open-source HEVC encoder. It can be used as a standalone application, as a part of FFmpeg [FFMPEG] suite, or as a dynamically linked library. The x265 produces RBSP bitstream files and could be easily integrated with a bitstream multiplexer.

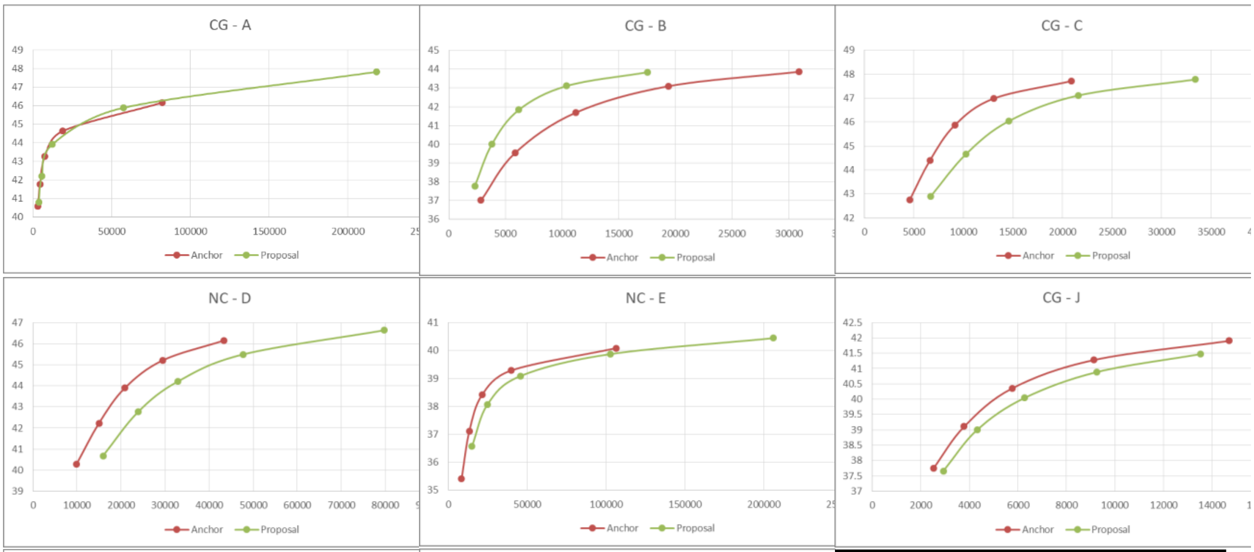
In our experiments we used x265 within ffmpeg, with default setting. Only encoder preset ("slow") and quantization parameter (QP) have been specified. The following listing summarizes x265 encoder configuration used:

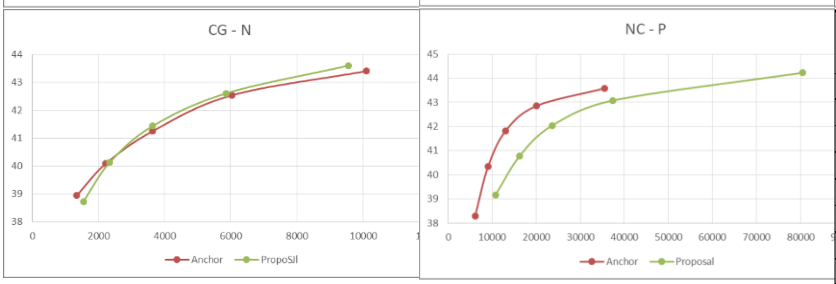
```
x265 [info]: HEVC encoder version 3.4+2-73ca1d7be377
x265 [info]: build info [Windows][GCC 9.3.1][64 bit] 10bit
x265 [info]: using cpu capabilities: MMX2 SSE2Fast SSSE3 SSE4.2 AVX
x265 [info]: Main 10 profile, Level-4 (Main tier)
```

```
x265 [info]: Thread pool created using 8 threads
x265 [info]: Slices                              : 1
x265 [info]: frame threads / pool features       : 3 / wpp(37 rows)
x265 [info]: Coding QT: max CU size, min CU size : 64 / 8
x265 [info]: Residual QT: max TU size, max depth : 32 / 1 inter / 1 intra
x265 [info]: ME / range / subpel / merge         : star / 57 / 3 / 3
x265 [info]: Keyframe min / max / scenecut / bias  : 25 / 250 / 40 / 5.00
x265 [info]: Lookahead / bframes / badapt        : 25 / 4 / 2
x265 [info]: b-pyramid / weightp / weightb       : 1 / 1 / 0
x265 [info]: References / ref-limit  cu / depth  : 4 / on / on
x265 [info]: Rate Control                        : CQP-11
x265 [info]: tools: rect limit-modes rd=4 psy-rd=2.00 rdoq=2 psy-rdoq=1.00
x265 [info]: tools: rskip mode=1 signhide tmvp strong-intra-smoothing lslices=4
x265 [info]: tools: deblock sao
```

When compared to FFmpeg suite, the standalone x265 application offers wider configurability options (https://x265.readthedocs.io/en/master/cli.html) and allows for more precise tuning. This enables changing its behavior (i.e. GOP shape) into HM-like.

The conducted experiments show that using x265 instead of HM allows for 100x times reduction of encoding computation time (420.82 h → 3.39 h) with an average 20-30% compression efficiency loss. Moreover, the usage of x265 not only improves the "throughput" of experiments but also reduces "latency" i.e. allows to obtain first experiment results in a shorter time.





Sequence		High-BR BD rate IV-PSNR	Low-BR BD rate IV-PSNR	HM coding
ClassroomVideo	SA	9.3%	12.4%	0.8%
Museum	SB	-45.7%	-42.9%	0.7%
Hijack	SC	53.4%	49.6%	1.6%
Kitchen	SJ	25.6%	22.7%	0.4%
Painter	SD	46.7%	45.8%	0.6%
Frog	SE	35.3%	33.7%	0.9%
Chess	SN	-4.8%	2.0%	0.7%
Carpark	SP	64.3%	62.5%	0.9%
MIV		**27.0%**	**26.2%**	0.8%

Fig. TMIV5 anchor: HM encoding (red curves) vs. x265 encoding.

We suggest to set x265 as the default encoder with the following remarks:

- use standalone x265 application,
- fine-tune command line parameters to mimic HM behavior,
- evaluate available encoder presets in order to choose the best complexity-performance tradeoff.

Table 3. HEVC encoder substitution: computational time reduction.

Total time of computation			
TMIV encoder	**HEVC**	**TMIV decoder**	**All steps**
101045 s	12193 s	2467990 s	2587231 s
28.07 h	3.39 h	685.55 h	718.68 h
1.17 d	0.14 d	28.56 d	29.95 d
% of anchor			
100.0 %	0.8 %	100.0 %	53.6 %

5 Reduction of the number of input views being synthesized

While it is valuable to have objective quality metrics, we believe, it is not necessary to synthesize all the input views, especially when the decisions about new proposals are taken based on assessing the posetraces.

We propose to reduce the number of input views which have to be synthesized to just 4 specific views:

- 2 views where anchor achieved the highest objective quality (e.g. QP1 WS-PSNR),
- 2 views where the anchor achieved the lowest objective quality.

Such a limitation would still show, how the proposal behaves for basic and additional views, what will allow a fair comparison with the anchor.

Synthesis of all the views should be mandatory only for the experiments, which change view labeling (basic/additional) or pruning order. For other experiments, reporting of all views should be encouraged, but not mandatory.

Table 4. Reduction of synthesized views number: computational time reduction.

Total time of computation			
TMIV encoder	**HEVC**	**TMIV decoder**	**All steps**
101045 s	2260759 s	697045 s	3058849 s
28.07 h	627.99 h	193.62 h	849.68 h
1.17 d	26.17 d	8.07 d	35.40 d
% of anchor			
100.0 %	100.0 %	28.2 %	63.3 %

6 Combination of all proposed ideas

Table 5. The total reduction of the computational time.

	Total time of computation [days]			
	TMIV encoder	**HEVC**	**TMIV decoder**	**All steps**
Anchor	1.17	26.17	28.56	55.90
Proposed	0.78	0.14	5.41	6.33
	% of anchor			
	67.0 %	0.5 %	18.9 %	11.3 %

7 Acknowledgement

This work was supported by the Ministry of Science and Higher Education.

8 Recommendations

We recommend to change MIV CTC:

- to reduce the number of frames in 'ff' configuration from 97 to 65 (2 full GOPs + 1 frame),
- to use optimized HEVC encoder (e.g. x265) instead of HM,
- to synthesize only a subset of input views (e.g. 2 basic + 2 additional views).

9 References

[N18069] "WS-PSNR Software Manual" ISO/IEC JTC1/SC29/WG11 MPEG/N19495, October 2018, Macao, China.

[N19495] "Software manual of IV-PSNR for Immersive Video" ISO/IEC JTC1/SC29/WG11 MPEG/N19495, July 2020, Online.

[M54856] A. Dziembowski, J. Jung, B. Kroon, G. Lee, B. Salahieh V.K. Malamal Vadakital, "MIV anchors", ISO/IEC JTC1/SC29/WG11 MPEG/M54856, October 2020, Online.

[HM] High Efficiency Video Coding (HEVC) Test Model 16 (HM 16) Improved Encoder Description Update 9, ISO/IEC JTC1/SC29/WG11 MPEG/N17047, July 2017, Torino, Italy.

[FFMPEG] FFmpeg Developers. (2020). ffmpeg tool (Version 3.4+2-73ca1d7be377) [Software]. Available from http://ffmpeg.org/

[X265] MulticoreWare Inc.x265 HEVC Encoder/H.265 Video Codec. (2020). [Online]. Available from http://x265.org