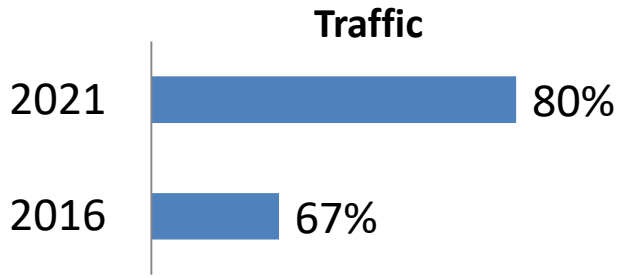
Video analysis using open-source FFmpeg tool and selection of codecs

Ana Gavrovska

PSSOH 2022

1. Introduction

- Video demand on the internet has grown significantly (Cisco)



- Video web services have skyrocketed in the period of Covid-19 pandemic

- ***How to cope with a large amount of media?***
- New media technologies have caused **revising issues of content delivery congestion**

Video production

Visual advertising

Sharing platforms

Video conference

Unified communication and collaboration

Immersive media

...

- Codecs & compression -> **practical implementations**
- Deal with bandwidth **bottleneck and limitations of storage**

- Retaining relevant information without losing media quality
- Different encoded data can be combined by multiplexing and data elements (streams) should coexist within container formats

- Interoperability is essential, and benefits of IP communications provided flexibility and scalability

- Methods for editing, control, quality evaluation, encapsulation, and many other processes are related to **coding and media handling**

- **Free and open-source frameworks and projects** are necessary for providing basics of understanding multimedia technologies that are
  - **becoming more complex and**
  - **of wide-ranging interest**

  among engineers, developers, etc.

https://ffmpeg.org/

- FFmpeg **(Fast Forward MPEG)**
- FFmpeg - one of the leading multimedia frameworks

- Available for implementation of media handling options and manipulations with high portability over different platforms, environments, and machine architectures

2. Codec tools

- A **large number of specific techniques** make a solution satisfying for media delivery

- In video and multimedia technology field **MPEGx and H26x standards** have taken the central place over the years
- **Committees, corporations, communities, and alliances** design variety of new standards and formats

- For example, FFmpeg is found suitable for many users since it supports from **old-fashioned formats to novel solutions**
- As a part of their processing pipeline, e.g. it can be used as backend for recording, media converting and AV streaming

LCEVC
MPEG mp4
HEVC
AV1 webm
VP9 VVC
H.264
ogg …

2.1 FFmpeg tools and libraries

- FFmpeg includes three basic commands:
  - ffmpeg (e.g. for codecs and format conversions)
  - ffprobe (e.g. to parse video content and monitor traffic quantities)
  - ffplay (to reproduce files in order to visualize the content)
- FFmpeg comes with libraries:
  - libavutil (for simplifying programmable approaches and routines),
  - libavcodec (containing encoders and decoders),
  - libavformat (dedicated to multiplexing and demultiplexing),
  - libavdevice (containing devices for grabbing and rendering),
  - libavfilter (holding filters for multimedia),
  - libswscale (for scaling and conversion),
  - libswresample (for resampling, rematrixing and sample format conversions).

The project has technical documentation, testing environment, bug tracker, wiki page

- The main licence is GPL (GNU General Public License) or LGPL (GNU Lesser GPL)

- The commands are easily applied in command line with chosen input and output files, selected actions and parameters
- Extending the framework is possible as well

- The media is kept in containers, like: mpeg, avi (audio video interleaved), m4v that is similar to mp4 (MPEG-4 Part 14), etc.
- Codec examples

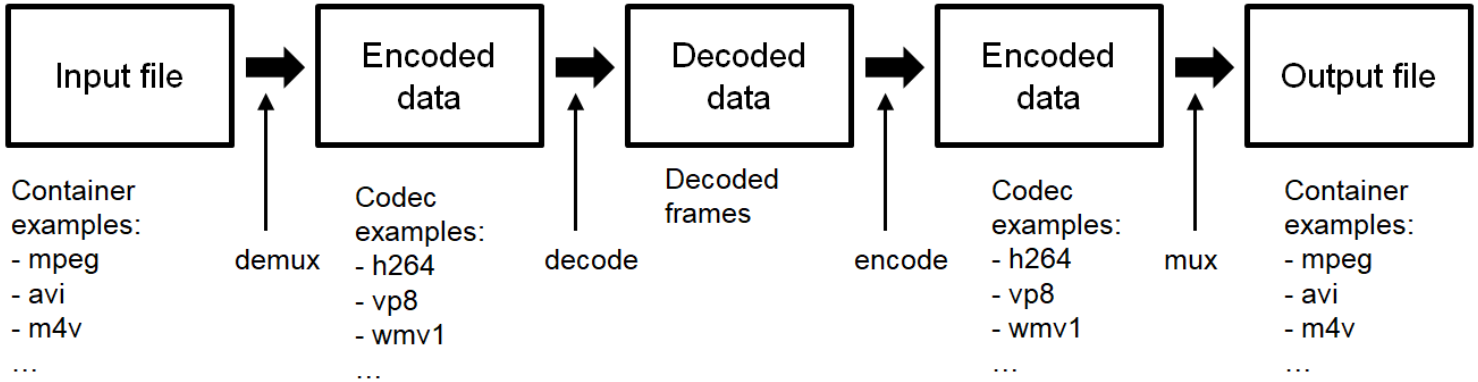


Figure 1. Typical video processing.

2.2 Video coding steps and standards

- Main steps in video coding in Fig. 2
- Each new standard brings novel advancements, like:
  - more partitions and larger blocks,
  - more intra and inter predictions,
  - functions,
  - filters,
  - AI solutions,

  etc.

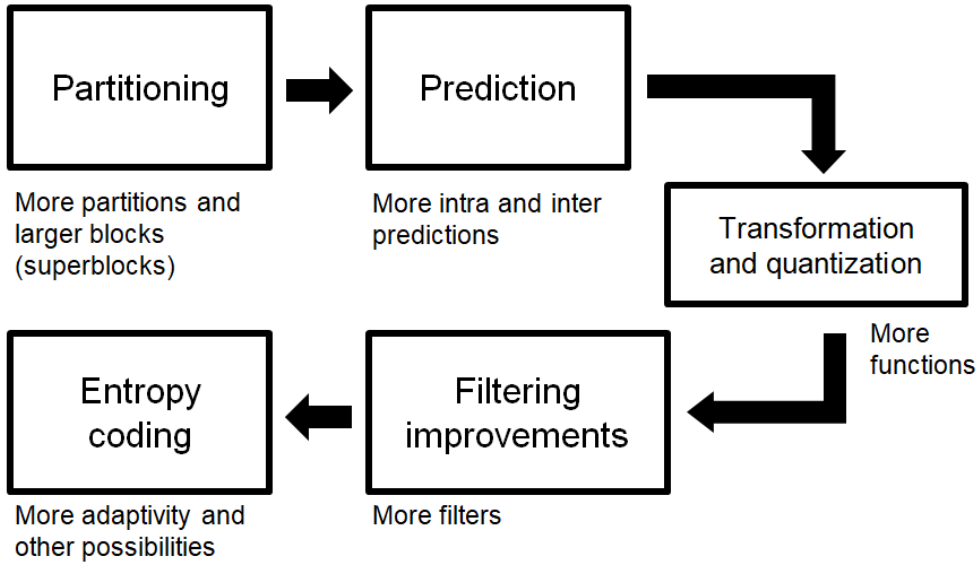


Figure 2. Illustration of main video coding steps and advancement.

- The streaming fundamentals - MPEG-2 and MPEG-4
- **AVC** (MPEG-4 Part 10) - 2003 found by ITU-T and ISO/IEC, **still in force** for generic audiovisual services

AVC

- **HEVC** (MPEG-H Part 2) – 2013, offered bitrate reduction of thirty to fifty percent to achieve comparable quality to AVC
- Unfortunately, it has not proved to be the right solution as its predecessor **despite its technical characteristics**

HEVC

- In 2013 Google released **VP9** based on VP8 (initally On2 technologies) - internet media delivery with similar quality ratings with HEVC

VP9

- In 2018 AOMedia (Alliance for Open Media) developed **AV1** standard, and united top tech leaders for the next generation of media delivery over internet
- The developed AV1 is considered a general-purpose open and royalty-free solution and the successor of VP9

AV1

- For efficient delivery over Internet is the target, the most common web codec formats and corresponding containers listed by Mozilla are presented in Table 1

- Depending on source video format and set configuration parameters different coding results can be obtained.

Table 1. Popular codecs and containers

Codec	Full codec name	Container support
AV1	AOMedia Video 1	MP4, WebM
AVC (H.264)	Advanced Video Coding	3GP, MP4
H.263	H.263 Video	3GP
HEVC (H.265)	High Efficiency Video Coding	MP4
MP4V-ES	MPEG-4 Video Elemental Stream	3GP, MP4
MPEG-1	MPEG-1 Part 2 Visual	MPEG, Quick Time
MPEG-2	MPEG-2 Part 2 Visual	MP4, MPEG, Quick Time
Theora	Theora	Ogg
VP8	Video Processor 8	3GP, Ogg, WebM
VP9	Video Processor 9	MP4, Ogg, WebM

Source: https://developer.mozilla.org/en-US/docs/Web/Media/Formats/Video_codecs

- Supported codecs and formats by FFmpeg can be listed
- Explanation of abbreviations is also given

ffmpeg -codecs

```
D.....  = Decoding supported
.E.... = Encoding supported
..V... = Video codec
..A... = Audio codec
..S... = Subtitle codec
..D... = Data codec
..T... = Attachment codec
...I..  = Intra frame-only codec
....L. = Lossy compression
.....S = Lossless compression
```

ffmpeg -encoders or *ffmpeg -decoders*

```
V..... = Video
A..... = Audio
S..... = Subtitle
.F.... = Frame-level multithreading
..S... = Slice-level multithreading
...X.. = Codec is experimental
....B. = Supports draw_horiz_band
.....D = Supports direct rendering method 1
```

ffmpeg -formats

```
D. = Demuxing supported
 .E = Muxing supported
```

Figure 3. Commands for listing supported codecs and formats.

3. Use case for media transcoding and preset options

- Video/audio can be manipulated by (trans-)coding and controlling its quality
- This can be done in many ways like: setting constant factor quality, buffer size, or using constant, constrained or variable bit rate

```
ffmpeg -i av_input -c copy -an  v_file
ffmpeg -i v_file  -c:v libsvtav1 -crf 35 v_output
ffmpeg -i v_file  -c:v libsvtav1 -preset 10 -crf 35 v_output
```

Removing audio

Codec selection

constant rate factor

Preset option

Figure 4. Examples of ffmpeg commands.

SVT-AV1 (Scalable Video Technology for AV1) -> libsvtav1

- **Crf** is selected as a model for controlling output (for popular codecs and keeps the output quality level by rate control method)
- Crf usually from 23 to 63 (VP9, AV1)
- Crf ranges from 19 to 41 (codecs x264 and x265)
- Lower crf values correspond to higher quality

- A preset is a collection of options that can provide certain encoding speed
- **Slower preset** may provide higher quality per filesize with more time needed to encode
- Supported preset options ranges from 0 to 13 with 13 for debugging and higher speed for higher preset value

- Video of about 10 minutes length (19036 frames) and 30fps

- The preset options are focused on speed and codec complexity (no additional tuning applied here)

- Here, default is preset 10
- The lowest filesize is for preset option 8

Table 2. SVT-AV1 results for different coding speed.

No.	Preset option	fps	Lsize	bitrate	speed
1	12	69	151584kB	1957.1kbits/s	2.29x
2	10	40	141200kB	1823.0kbits/s	1.32x
3	8	17	136125kB	1757.5kbits/s	0.583x
4	6	6.3	145011kB	1872.2kbits/s	0.212x
5	4	1.5	147742kB	1907.5kbits/s	0.0485x

- No significant difference after reproduction to a standard viewer here (ten volunteers, LED 23” monitor)
- Transcoding speed is quite different
- Number of fps varies for processing

- Similar preset options are available for AVC and HEVC (from veryslow to ultrafast)
- Here, crf 25 and libx264, without tuning higher speeds

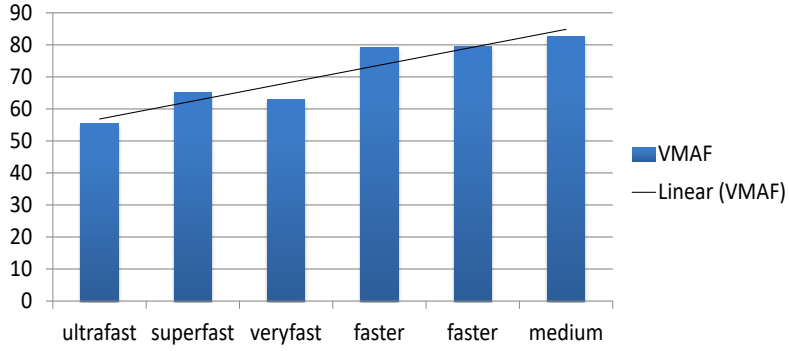
- *Medium* is default preset option
- The lowest filesize and bitrate is for preset option *veryfast*

Table 3. H.264 results for different coding speed.

No.	Preset option	fps	Lsize	bitrate	speed
1	ultrafast	199	645603kB	8335.3kbits/s	6.63x
2	superfast	116	249509kB	3221.7kbits/s	3.87x
3	veryfast	96	143810kB	1856.9kbits/s	3.21x
4	faster	58	169341kB	2186.6kbits/s	1.92x
5	fast	50	179782kB	2321.4kbits/s	1.68x
6	medium	43	178491kB	2304.7kbits/s	1.44x

- Still no significant difference to a standard viewer here
- Transcoding speed is quite different
- Number of fps varies for processing

- The quality difference exists, meaning **often higher quality for slower coding**
- **Preset default option** responds to higher speed at the price of slightly decrease of quality
- **Example** of VMAF (Video Multimethod Assessment Fusion) quality evaluation (still no significant difference to a standard viewer in this 1080p case)



VMAF scores using six presets

- Lower VMAF results with the highest speed
- Best quality for *veryslow*
- *Medium* - default

Average Quality	Ultrafast	Superfast	Veryfast	Faster	Fast	Medium	Slow	Slower	Veryslow	Placebo	Total Delta
Tears of Steel	89.20	92.00	93.29	95.45	95.59	96.22	96.43	96.56	96.67	96.65	8.38%
Sintel	88.29	92.66	93.85	95.84	95.99	96.38	96.56	96.68	96.83	96.75	9.68%
Big Buck Bunny	87.26	91.26	92.68	95.03	95.29	95.53	95.75	95.87	96.05	96.01	10.08%

Ozer J. Introduction to ABR Production & Delivery, https://www.streamingmedia.com Streaming Media West 2019. [25]

- **Example** of preset details [26] from ultrafast to veryslow

Legend

- Presets compared against medium:
  - ***** means differing values from medium;
  - **[ ]** values are either set by default, or by another argument;
  - **+** means the x264 argument is used;
  - **-** means the x264 argument is not used.

Option	ultrafast	superfast	veryfast	faster	fast	medium	slow	slower	veryslow	placebo
aq-mode	0*	1	1	1	1	1	1	1	1	1
aq-strength	[0]*	[1.0]	[1.0]	[1.0]	[1.0]	[1.0]	[1.0]	[1.0]	[1.0]	[1.0]
b-adapt	0*	1	1	1	1	1	1	2*	2*	2*
bframes	0*	3	3	3	3	3	3	3	8*	16*
chroma-qp-offset	[0]*	[0]*	[0]*	[0]*	[-2]	[-2]	[-2]	[-2]	[-2]	[-2]
deblock	[0:0]*	[1:0]	[1:0]	[1:0]	[1:0]	[1:0]	[1:0]	[1:0]	[1:0]	[1:0]
direct	spatial	spatial	spatial	spatial	spatial	spatial	auto*	auto*	auto*	auto*
me	dia*	dia*	hex	hex	hex	hex	hex	umh*	umh*	tesa*
merange	16	16	16	16	16	16	16	16	24*	24*
no-8x8dct	+*	-	-	-	-	-	-	-	-	-
no-cabac	+*	-	-	-	-	-	-	-	-	-
no-deblock	+*	-	-	-	-	-	-	-	-	-
no-fast-pskip	-	-	-	-	-	-	-	-	-	+*
no-mbtree	+*	+*	-	-	-	-	-	-	-	-
no-mixed-refs	+*	+*	+*	+*	-	-	-	-	-	-
no-weightb	+*	-	-	-	-	-	-	-	-	-
partitions	none*	i8x8,i4x4*	p8x8,b8x8,i8x8,i4x4	p8x8,b8x8,i8x8,i4x4	p8x8,b8x8,i8x8,i4x4	p8x8,b8x8,i8x8,i4x4	p8x8,b8x8,i8x8,i4x4	all*	all*	all*
rc-lookahead	0*	0*	10*	20*	30*	40	50*	60*	60*	60*
ref	1*	1*	1*	2*	2*	3	5*	8*	16*	16*
scenecut	0*	40	40	40	40	40	40	40	40	40
slow-firstpass	-	-	-	-	-	-	-	-	-	+*
subme	0*	1*	2*	4*	6*	7	8*	9*	10*	11*
trellis	0*	0*	0*	1	1	1	2*	2*	2*	2*
weightp	0*	1*	1*	1*	1*	2	2	2	2	2

OBS, Streaming with x264, https://obsproject.com/blog/streaming-with-x264#presets 2017.

- **Example** of preset details from 0 to 12 (svt-av1)
- What is ON or OFF among the features?

Category	Feature	0	1	2	3	4	5	6	7	8	9	10	11	12
Prediction structure & RC	Hierarchical levels	5L	5L	5L	5L	5L	5L	5L	5L	5L	5L	5L	5L	5L
	aq-mode	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON
	max reference frame count	7	7	7	7	7	7	4	4	4	4	4	4	4
Motion Estimation	Full pel Motion Estimation	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON
	Hierarchical ME	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON
	subpel	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON
Block Partitioning	sb size	128	128	128	64	64	64	64	64	64	64	64	64	64
	min block size	4	4	4	4	4	4	8	8	8	8	8	8	8
	Non-square partitions	ON	ON	ON	ON	ON	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF
AV1 mode decision features	DC	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON
	Smooth, Smooth_V, Smooth_H	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON
	Directional Angular modes	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON
	Paeth	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	OFF	OFF	OFF
	Chroma from Luma (CfL)	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON
	Filter intra	ON	ON	ON	ON	ON	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF
	Intra block copy (IBC) (SC)	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	OFF	OFF
	Palette prediction (SC)	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON
	Single-reference prediction	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON
	Compound-reference prediction	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON
	Eighth-pel f(resolution, qindex)	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON
	Interpolation Filter Search	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON
	Warped motion compensation	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON
	Global motion compensation	ON	ON	ON	ON	ON	ON	ON	OFF	OFF	OFF	OFF	OFF	OFF
	Motion Field Motion Vector (MFMV)	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	OFF
	Overlapped Block Motion Compensation (OBMC)	ON	ON	ON	ON	ON	ON	ON	OFF	OFF	OFF	OFF	OFF	OFF
	Inter-Intra prediction	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	OFF
	Wedge prediction	ON	ON	ON	ON	ON	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF
	Difference-weighted prediction	ON	ON	ON	ON	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
	Distance-weighted prediction	ON	ON	ON	ON	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
Transform	Transform type search	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON
	Transform Size search	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON
AV1 inloop filters	Deblocking Filter	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON
	CDEF	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON
	Restoration Filter - Wiener Filter	ON	ON	ON	ON	ON	ON	ON	ON	ON	ON	OFF	OFF	OFF
	Restoration Filter - SG Filter	ON	ON	ON	ON	ON	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF

Source: https://gitlab.com/AOMediaCodec/SVT-AV1/-/blob/master/Docs/CommonQuestions.md#what-presets-do

4. Towards new trends in video codecs

- JVET (Joint Video Exploration Team; ISO/IEC(MPEG) and ITU-T (VCEG)) released new standard in 2020, where UHD 4k/8k, HDR, VR, 360 degrees are taken into account
- **Versatile Video Codec (VVC)** as the successor of HEVC with up to 50% quality improvement

(JVET)
AVC,
HEVC,
VVC

- Example: Coding-independent code points (CICP) and Versatile Supplemental Enhancement Information (VSEI) for improved interpretation

- **EVC (Essential Video Coding)** addresses cases where coding standards have not be adopted despite technical characteristics.
- **LCEVC (Low Complexity Enhancement Video Coding)** is developed, where enhancement layer is specified for streaming when combined with a base encoded video

(MPEG)
EVC
LCEVC

- H.264 is still dominant? (Bitmovin's Industry Report in 2020)
- Compression efficiency or real-time applications?

- AV1 and VVC seem to incur long coding and compression times -> **optimization, presets?**

- **Patents** do have affects on standards and applications
- Different **patent holders and patent pools** have been related to standards implementation

- **AI – trend in coding**
- MPAI community - licensing formats

MPEG- LA
Access Advance
MPAI community
…

- **Information of legal status** of some of the solutions is difficult to answer whether one is a lawyer or not

- Affecting the speed of **general acceptance of the new standards**
- The novel standards are results of united work -> conflicts & impartiality challenges

- **FRAND** means **fair, reasonable and non-discriminatory**
- Licenses may be cosidered FRAND or non-FRAND

- The licensing for new solutions are of great interest
- Fair, adaptive, but balanced approaches are expected for
  - future media codec market,
  - Industry
  - research needs

5. Conclusion

- Open-source and free video and multimedia technology frameworks, tools and projects play **important role in education and research**
- The necessity of such projects should be recognized at a larger scale due to necessity of **future video and multimedia experts**
- The impression is that such tools are extremely **useful to the professionals for further research and making extensions**
- New codec solutions should have in mind:
  - **different presets and optimizations**, as well as
  - **practical licensing approaches**

Thank You For Your Attention!

Contact:anaga777@etf.rs

References (1)

[1] Chellappa R, Theodoridis S. Academic Press Library in Signal Processing, Volume 7: Array, Radar and Communications Engineering. Academic Press; 2017 Dec 1.

[2] Zhang T, Mao S. An overview of emerging video coding standards. GetMobile: Mobile Computing and Communications. 2019 May 2;22(4):13-20.

[3] Ma N. Distributed video coding scheme of multimedia data compression algorithm for wireless sensor networks. EURASIP Journal on Wireless Communications and Networking. 2019 Dec;2019(1):1-9.

[4] Hossain K, Roy S. A data compression and storage optimization framework for iot sensor data in cloud storage. In2018 21st International Conference of Computer and Information Technology (ICCIT) 2018 Dec 21 (pp. 1-6). IEEE.

[5] Cisco VNI Complete Forecast Highlights. Global - 2021 Forecast Highlights.https://www.cisco.com/c/dam/m/en_us/solutions/service-provider/vni-forecast-highlights/pdf/Global_2021_Forecast_Highlights.pdf (last accessed 01.07.2022.)

[6] Pelurson S, Cozanet J, Guionnet T, Abdoli M, Biatek T. AI-Based Saliency-Aware Video Coding. SMPTE Motion Imaging Journal. 2022 May 10;131(4):21-9.

[7] Gavrovska A. Uvod u savremene video tehnologije i sisteme. Akademska misao, 2021.

[8] Ozer J. What is Codec? https://www.streamingmedia.com/Articles/ReadArticle.aspx?ArticleID=74487 (last accessed 01.07.2022.)

[9] Ulas D. Digital transformation process and SMEs. Procedia Computer Science. 2019 Jan 1;158:662-71.

[10] Kale V. Digital transformation of enterprise architecture. CRC Press; 2019 Jul 8.

[11] FFmpeg. https://ffmpeg.org/, https://trac.ffmpeg.org/ (last accessed 01.07.2022.)

[12] Ferrando N. "FFmpeg - From Zero to Hero". https://ffmpegfromzerotohero.com/ (last accessed 01.07.2022.)

References (2)

[13] MPEG. https://www.mpeg.org/ (last accessed 15.07.2022.)

[14] OpenCV - Open source Coputer Vision, Video I/O with OpenCV Overview, https://docs.opencv.org/4.x/d0/da7/videoio_overview.html (last accessed 15.07.2022.)

[15] Wu X, Qu P, Wang S, Xie L, Dong J. Extend the FFmpeg Framework to Analyze Media Content. arXiv preprint arXiv:2103.03539. 2021 Mar 5.

[16] ITU-T H.264 : Advanced video coding for generic audiovisual services. https://www.itu.int/rec/T-REC-H.264-202108-I/en (last accessed 15.07.2022.)

[17] Bitmovin to Bitmovin's Video Developer Report. https://go.bitmovin.com/video-developer-report-2020 (last accessed 15.07.2022.)

[18] Alliance for Open Media. https://aomedia.org/ (last accessed 15.07.2022.)

[19] Chen Y, Murherjee D, Han J, Grange A, Xu Y, Liu Z, Parker S, Chen C, Su H, Joshi U, Chiang CH. An overview of core coding tools in the AV1 video codec. In2018 Picture Coding Symposium (PCS) 2018 Jun 24 (pp. 41-45). IEEE.

[20] Gavrovska AM, Milivojevic MS, Zajic G. Analysis of SVT-AV1 format for 4k video delivery. In 2020 28th Telecommunications Forum (TELFOR) 2020 Nov 24 (pp. 1-4). IEEE.

[21] Web video codec guide, https://developer.mozilla.org/en-US/docs/Web/Media/Formats/Video_codecs (last accessed 15.07.2022.)

[22] Wu PH, Katsavounidis I, Lei Z, Ronca D, Tmar H, Abdelkafi O, Cheung C, Amara FB, Kossentini F. Towards much better SVT-AV1 quality-cycles tradeoffs for VOD applications. InApplications of Digital Image Processing XLIV 2021 Aug 1 (Vol. 11842, pp. 236-256). SPIE.

[23] SVT-AV1. https://gitlab.com/AOMediaCodec/SVT-AV1 (last accessed 15.07.2022.)

[24] Big Buck Bunny, h HYPERLINK "https://peach.blender.org/"ttps://peach.blender.org (last accessed 01.07.2022.)

[25] Ozer J. Introduction to ABR Production & Delivery, https://www.streamingmedia.com Streaming Media West 2019 (last accessed 01.07.2022.)

References (3)

[26] OBS, Streaming with x264, https://obsproject.com/blog/streaming-with-x264#presets 2017 (last accessed 01.07.2022.)

[27] Bross B, Wang YK, Ye Y, Liu S, Chen J, Sullivan GJ, Ohm JR. Overview of the versatile video coding (VVC) standard and its applications. IEEE Transactions on Circuits and Systems for Video Technology. 2021 Aug 2;31(10):3736-64.

[28] Minopoulos G, Memos VA, Psannis KE, Ishibashi Y. Comparison of video codecs performance for real-time transmission. In2020 2nd International Conference on Computer Communication and the Internet (ICCCI) 2020 Jun 26 (pp. 110-114). IEEE.

[29]] Fujihashi, T., Koike-Akino, T. and Watanabe, T., 2021. Soft Delivery: Survey on A New Paradigm for Wireless and Mobile Multimedia Streaming. *arXiv preprint arXiv:2111.08189*.

[30] Bitmovin to Bitmovin's Video Developer Report. https://go.bitmovin.com/video-developer-report-2020 (last accessed 01.07.2022.)

[31] Panayides, A. S., Pattichis, M. S., Pantziaris, M., Constantinides, A. G., & Pattichis, C. S. (2020). The battle of the video codecs in the healthcare domain-a comparative performance evaluation study leveraging VVC and AV1. *IEEE Access*, *8*, 11469-11481.

[32] Pfeiffer, S., 2009. Patents and their effect on Standards: Open video codecs for HTML5. *International Free and Open Source Software Law Review*, *1*(2), pp.131-138

[33] MPEG-LA. https://www.mpegla.com/ (last accessed 01.07.2022.)

[34] Access Advance. https://accessadvance.com/ last accessed 01.07.2022.)

[35] MPAI community. https://mpai.community/standards/mpai-evc/about-mpai-evc/ last accessed 01.07.2022.)

[36] Jones SL, Leiponen A, Vasudeva G. The evolution of cooperation in the face of conflict: Evidence from the innovation ecosystem for mobile telecom standards development. Strategic Management Journal. 2021 Apr;42(4):710-40.