Improving Context-Aware Encoding by Adaptation to "True Resolution" of the Content

Yuriy Reznik, Karl Lillevold, Abhijith Jagannath

Brightcove Inc, Boston, MA, USA.

Nabajeet Barman

Kingston University, London, UK.

Written for presentation at the

SMPTE 2023 Media and Technology Summitt

Abstract. *As well known, when the input video is upscaled, the effectiveness of its transcoding and delivery may suffer. The encoded stream may not look sharp and use more bits than necessary. Then with adaptive streaming, extra streams may be added to reach such a maximum resolution and bitrate. The result is a significant waste of storage, bandwidth, and compute resources. In this paper, we explain the origins of this problem, survey existing methods for addressing it, and then propose our solution. Our proposed design incorporates a novel "true resolution" detection technique and a traditional CAE (context-aware encoding) encoding ladder generator. The CAE generator receives the detected "true resolution" of content as a limit for resolutions to include in the ladder. Such a limit enables all subsequent savings. We describe the details of our proposed resolution detection method, bring examples explaining how it works, and then study the performance of our proposed system in practice. Our study, performed using 500 video assets representing 120 hours of real-world production material, confirms the effectiveness of this technique. It shows that in many practical cases, the incoming content is, in fact, upscaled and that adding a "true resolution" detector to CAE brings very appreciable savings in bandwidth, storage, and compute cost.*

.

Keywords. Video resolution detection, context-aware encoding, CAE, ABR streaming, DASH, HLS.

Introduction

In the modern world, we are witnessing continued evolution and increasingly hybrid operation of traditional/broadcast and OTT/streaming systems. Such co-existence often leads to complex distribution flows with many video transcoding and format conversion operations [1].

For example, consider a hybrid broadcast + OTT distribution system presented in Figure 1.

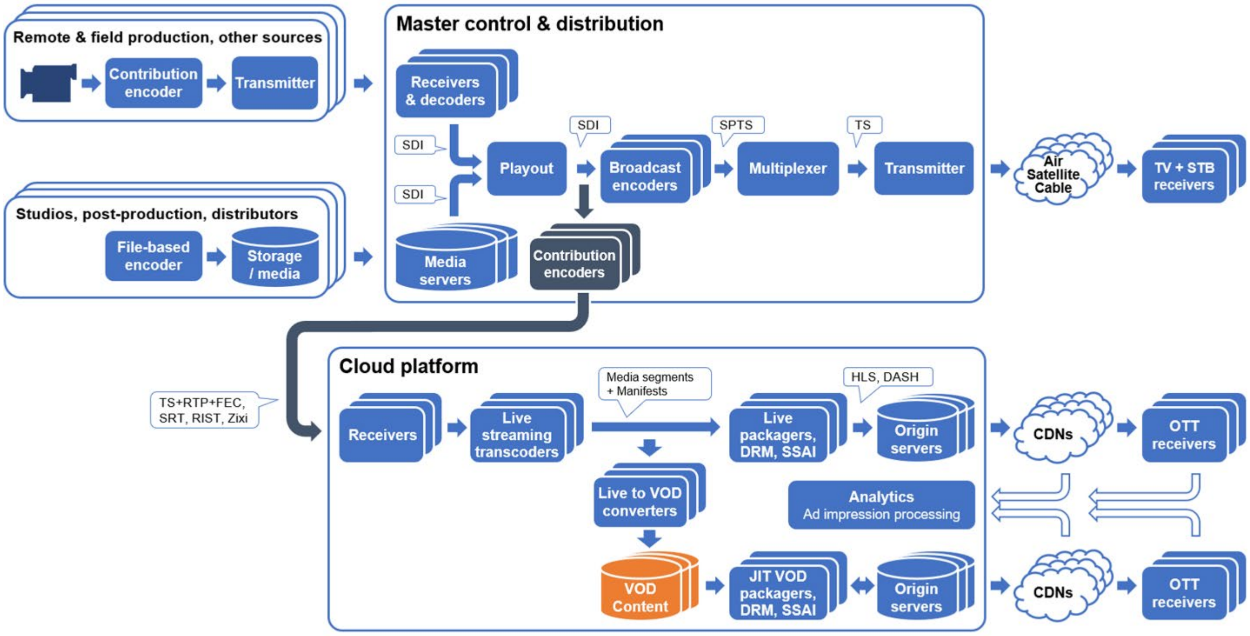

Figure 1. A hybrid video delivery system with multiple transcoding operations.

As typical for broadcast systems, the incoming video feeds originate from remote and field production. A contribution encoder is employed at this stage. It converts video from camera-native format to one required on ingest by the broadcast system. Then, once the content reaches the master control/playout system, it undergoes additional transformations. The playout system may add channel bugs, lower thirds, ad avails, etc. It may also mix content from different sources. Then, another encoder is employed to transmit streams from the broadcast system to an OTT delivery workflow. And then, within the OTT delivery system, another encoder produces outputs for DASH/HLS streaming distribution [2,3]. As easily observed, there are several transcoding operations involved.

Each transcoding or editing operation may introduce changes in video formats. Furthermore, in some cases, such conversions may lower the effective "spatial density" of the content. Examples include video upscaling, SAR/DAR conversions, removal of black bars, etc. Conversions between interlaced and progressive formats may also involve upscaling, as with SD to HD format conversions. Table 1 lists several commonly used video formats, along with typical examples of conversion operations increasing the "declared resolutions" of the mezzanines.

Problems Presented by Upscaled Video Content

When the final OTT/streaming transcoder receives the mezzanine content, it is generally unaware of any earlier conversion operations performed. It only sees the resolution and DAR or SAR as declared in the mezzanine metadata. Hence, if input content is upscaled, it becomes transcoded for delivery as is, producing outputs that may be suboptimal from a quality and efficiency standpoint.

For example, if a 1080p asset becomes up-converted to 4K earlier in the workflow, it most likely will be transcoded as 4K content for delivery. Furthermore, with HLS/DASH streaming requirements, this will result in a ladder of 9-12 streams with intermediate resolutions to 4K. Such ABR encoding may easily double or even triple bandwidth and storage costs. But quality-wise, if this was a 1080p stream initially, it won't look much better. The same experience can be delivered by a more compact 1080p ladder using much fewer bits.

The described problem, unfortunately, is quite common in modern practice. As shown in Table 1, there are many standard format conversion operations producing upscaled outputs. With increasingly more complex media delivery workflows and additional encoding and format conversion operations introduced in practical systems, this problem becomes even more significant.

Table 1. Standard video formats and possible up-conversion operations.

Video format	Width	Height	DAR$_1$	SAR$_1$	DAR$_2$	SAR$_2$	May be up-converted to
SD 480i	352	480i	4:3	20:11	16:9	80:33	720p, 1080i, 1080p
	480	480i	4:3	4:3	16:9	16:9	
	528	480i	4:3	40:33	16:9	160:99	
	544*	480i	4:3	40:33	16:9	160:99	
	640	480i	4:3	1:1	16:9	4:3	
	704	480i	4:3	10:11	16:9	40:33	
	720*	480i	4:3	10:11	16:9	40:33	
SD 576i	352	576i	4:3	24:11	16:9	32:11	
	480	576i	4:3	8:5	16:9	32:15	
	544*	576i	4:3	11:12	16:9	64:33	
	704	576i	4:3	12:11	16:9	16:11	
	720*	576i	4:3	12:11	16:9	16:11	
HD 720p HD/1080i HD/1080p	960	720	16:9	4:3			720p, 1080p
	1280	1080i	16:9	3:2			1080i, 1080p
	1280	720	16:9	1:1			1080p, 4K
	1440	1080i	16:9	4:3			
	1920	1080i	16:9	1:1			
	1920	1080	16:9	1:1			4K
Widescreen 1080p, 2K	1920	800	2.4:1	1:1			1080p, 4K
	1920	816	2.35:1	1:1			
	2048	864	2.4:1	1:1			
	2048	1080	17:1	1:1			4K

Related Prior Work

Among related prior work, we must recognize several techniques that may be helpful.

The first category comprises "per-title," "content-aware," and "context-aware" encoding (CAE) techniques [5-10]. Such techniques first analyse each incoming video asset and then decide how many bits to use to encode it most efficiently. In other words, instead of using a fixed ladder, as shown in Table 2, they generate a custom ladder for each input video. If the video is "simple" to encode, it receives fewer bits. If the video is "complex," more bits and possibly more streams may be generated. In the case of upscaled content, it is reasonable to expect at least the top few renditions (the ones with the highest resolutions) to become more compressible. Hence CAE could save some bits. But it won't trim the encoding profile or reduce the maximum resolution automatically. In other words, while CAE could lessen the inefficiency introduced by upscaling, it can't eliminate it

The second category of techniques comprises video encoder-level optimizations, dynamically changing resolutions within the encoded streams. Such functionality is allowed in the latest codecs, such as VVC [11]. With older codecs, such as H.264 [12] and HEVC [13] it is also possible with HLS and appropriate support from HLS clients and decoders [14]. However, dynamic resolution changes are not always safe. For example, they may alter the artistic appearance of film grain, background textures, or other fine details. Dynamic resolution changes may also introduce an inconsistency in video appearance throughout playback. When working with previously upscaled content, such techniques could also help, but there is no guarantee that the resolutions they select dynamically on a segment-to-segment or frame-by-frame basis would match the original resolution of the content. They also will not affect the number of streams in the ABR encoding ladder. In other words, this class of techniques could also help, but only partially.

Finally, the last category of relevant techniques includes "original resolution" or "true resolution" detectors [15-18]. These algorithms detect if a given image or video was previously upscaled. Their traditional uses include forensic analysis, restoration, and other applications [17]. But they come with some limitations. For example, the well-known A. C. Gallagher's method [15] only works well for cubic interpolation [16]. The normalized energy density technique [17] is also limited to classic reconstruction filters. The method utilizing the ratio of the low- and high-frequency energy densities, proposed in [18], appears to be more general. It bounds the range of likely original resolutions. However, it does not strongly indicate that a particular sampling frequency is the best candidate. The method of detecting a "sharp decline" in the accumulated log-amplitude spectrum [19] is also more general. The authors in [19] report success in its application to modern super-resolution upscaling techniques [20-24]. However, as we observed in our experiments, none of these methods is perfect. They work in many cases but may also fail in some. Many are sensitive to noise and compression artifacts introduced by prior-generation encoding.

But in principle, we believe that techniques for detecting "original" or "true" resolution provide the right tools for addressing the described problem. We will utilize several of these techniques in our proposed solution.

Proposed Solution

We show the overall processing chain in our system in Figure 2.

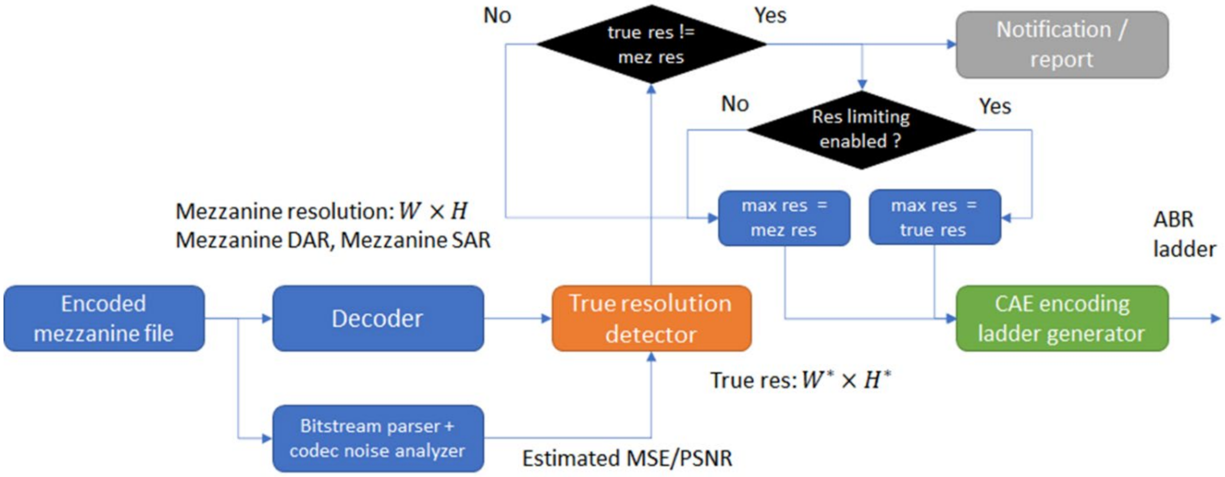

Figure 2. Overall processing chain in the proposed system.

The primary operations are "true resolution" detection and a CAE encoding ladder generator [6,8]. In this work, we use the CAE tool provided by the Brightcove VideoCloud system [25,26]. Our resolution detector is aided by a codec noise analyser [27,28]. This analyser predicts the PSNR of the codec-introduced noise level in the encoded mezzanine we receive as input.

Resolution Detection Algorithm

Figure 3 shows the flow of operations within our detector. The candidate horizontal and vertical resolutions are detected separately, utilizing row and column data in each frame. In both cases, we turn data in the DFT domain [29], extract spectral features, and perform an initial selection of candidate resolutions in both directions. The choice of the best joint (horizontal, vertical) resolution pair follows as a final step. If the system finds no compelling candidate resolutions or cues that the video is upscaled, it reports the mezzanine resolution as "true resolution."

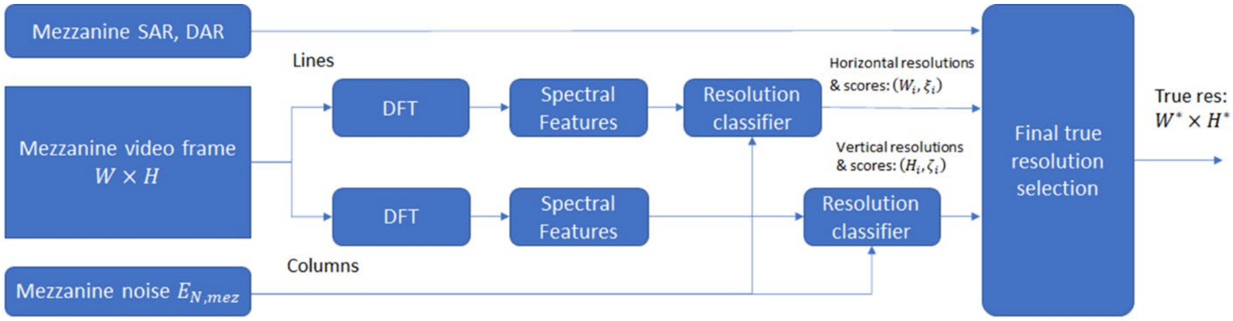
Mezzanine SAR, DAR
Lines
Mezzanine video frame $W \times H$
DFT
Spectral Features
Resolution classifier
Horizontal resolutions & scores: (W_i, ξ_i)
Vertical resolutions & scores: (H_i, ζ_i)
DFT
Spectral Features
Resolution classifier
Columns
Mezzanine noise $E_{N,mez}$
Final true resolution selection
True res: $W^* \times H^*$

Figure 3. Processing chain within the proposed resolution detector.

Frequency-Domain Processing

Figure 4 explains DFT-domain parameters we use for detection. Parameter f_N denotes the Nyquist frequency of the mezzanine sampled data. Parameter f_c represents the "true resolution" frequency under the test. The shaded regions show amplitude spectrum parts before and after f_c. Integrals of squared amplitude spectrum (or "energies") below and after f_c are denoted by $E_{f<f_c}$ and $E_{f\geq f_c}$, respectively.

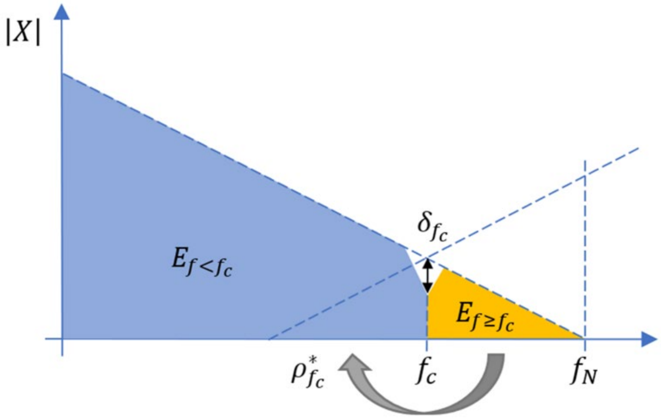

Figure 4. Spectral features used by the resolution detector.

The dotted line moving from the right shows potential overlap with an adjacent spectral image. This overlap is the cause of classic aliasing artifacts [29]. The so-called "post-aliasing" artifacts [30] also relate to the presence of the conjugate-symmetric spectral components coming from the adjacent spectral image.

In theory, the ideal filter designs must eliminate both types of aliasing artifacts from signals. But none of the practical filter designs are achieving this objective. Furthermore, as explained in [30], some minor aliasing and post-aliasing artifacts are quite normal and acceptable in practice. We will use their presence as cues in our detector.

To detect them in the vicinity of f_c, we use correlation metric:

$$\rho^*_{f_c} = \frac{\sum_u x_{im}[f_c - u] \cdot x_{im}[f_c + u]}{\sqrt{\sum_u x_{im}[f_c - u]^2 \cdot \sum_u x_{im}[f_c + u]^2}},$$

where $x_{im}[.]$ are imaginary parts of the DFT spectral components. A Gaussian-smoothed window of +-32 spectral lines around f_c is used for computing these quantities. We first compute such metrics for each line or column in a frame. We then aggregate the results for this frame and for the entire sequence.

The other cue that we employ is a "sharp decline" effect discovered in [19]. Its presence relates to de-attenuation in transition bands of filters that may have been previously applied. In Figure 4, we illustrate it by gap denoted δ_{f_c}. We compute it as follows:

$$\delta_{f_c} = \frac{\lambda[f_c]}{\mathrm{median}(\lambda[f_c - m], \ldots, \lambda[f_c + m])},$$

where $\lambda[.]$ denote logarithm-domain amplitudes of spectral components averaged across the entire sequence [19]. The median(.) denotes the median filter. A +-32-point window around f_c is used to compute this criterion.

Both aliasing and shape decline criteria typically point to the same candidate frequency. But, in some rare cases, these detectors may disagree, show several candidates, or fail to detect any. To resolve such ambiguities and improve the robustness of our method in general, we must apply several additional constraints. For this purpose, we use signal energies in each band $E_{f<f_c}$ and $E_{f\geq f_c}$, as well as an estimate of codec-introduced noise as present in the mezzanine $E_{N,Mez}$.The first check compares the energy of the signal past f_c to the mezzanine noise level:

$$E_{f\geq f_c} < C_1\left(\frac{f_N - f_c}{f_N}\right)E_{N,Mez}.$$

The second check compares the energy of the signal past f_c to the signal's energy:

$$E_{f\geq f_c} < C_2 \cdot\ E_{f<f_c}.$$

Both tests ensure that the choice of f_c as "true resolution" won't remove any meaningful signal components. The constants C_1 and C_2 are empirically chosen based on the corresponding bounds observed in a dataset of videos with various types of conversions and transcoding operations

Final Checks and Resolution Selection

The last block, shown in Figure 3, performs the final "true resolution" selection based on candidate frequencies supplied by horizontal and vertical detectors. This block applies a few additional rules, disqualifying impossible or highly improbable combinations based on SAR/DAR constraints, and selects a pair of horizontal + vertical resolutions to report. If none of the candidates pass final safety checks, the detector outputs full mezzanine resolution as a default choice.

Example of Operation

To show how the proposed design works, we will use a "Tears of Steel" sequence [31] coming in a wide-screen 1920x800 format, which we convert to 16:9 DAR by taking the midsection anscaling it up to 1080p. Therefore, this video's "true resolution" is only 1422.2 x 800 pixels.

We then encode this video using: (1) a standard HLS ladder for H.264 and 16:9 content [4], (2) a ladder generated by Brightcove CAE [25], treating the input as 1080p, and (3) a ladder generated by Brightcove CAE with "true resolution" detection enabled.

Tables 2-4 show the results. First columns list encoding ladder parameters: codec, profile, and resolution of each stream as encoded. Then we list "true width" and "true height," with resolution parameters clipped by the true resolution limits. Then we list encoding bitrates and SSIM quality values [32]. The final column lists the load probabilities of each rendition, retrieved by the Brightcove analytics system [26] after the playback. In the bottom lines, we report average bitrates, SSIM scores, and average resolution delivered to viewers during the playback. The storage values report the sums of bitrates of all renditions in each profile.

Tables 2 and 3 show improvements achieved by standard CAE vs. HLS reference profile. We observe that average bitrates went down to 3386 kbps from 5705 kbps, storage is now 7912 kbps vs. 25640 kbps, and the number of streams is 6 vs. 9 initially—very significant improvements in all domains.

Table 2 – Encoding and streaming statistics for HLS reference encoding ladder [4].

Rendition	**Codec**	**Profile**	**Width**	**Height**	**True Width**	**True Height**	**Bitrate** [kbps]	**SSIM**	**Pr**
1	h264	High	416	234	416	234	145	0.9390	0.0049
2	h264	High	640	360	640	360	365	0.9604	0.0050
3	h264	High	768	432	768	432	730	0.9739	0.0076
4	h264	High	768	432	768	432	1100	0.9817	0.0336
5	h264	High	960	540	960	540	2000	0.9858	0.0710
6	h264	High	1280	720	1280	720	3000	0.9860	0.1261
7	h264	High	1280	720	1280	720	4500	0.9898	0.1298
8	h264	High	1920	1080	1422	800	6000	0.9874	0.1557
9	h264	High	1920	1080	1422	800	7800	0.9896	0.4586
Average					**1316**	**740**	**5705**	**0.9878**	
Storage							**25640**		

Table 3 – Encoding and streaming statistics for CAE [25]

Rendition	**Codec**	**Profile**	**Width**	**Height**	**True Width**	**True Height**	**Bitrate** [kbps]	**SSIM**	**Pr**
1	h264	High	384	216	384	216	145	0.9421	0.0032
2	h264	High	512	288	512	288	267	0.9572	0.0039
3	h264	High	768	432	768	432	534	0.9656	0.0096
4	h264	High	1024	576	1024	576	1068	0.9745	0.0426
5	h264	High	1600	900	1422	800	2136	0.9773	0.1245
6	h264	High	1920	1080	1422	800	3763	0.9823	0.8084
Average					**1392**	**783**	**3386**	**0.9810**	
Storage							**7912**		

Table 4 – Encoding and streaming statistics for CAE with true resolution detection

Rendition	**Codec**	**Profile**	**Width**	**Height**	**True Width**	**True Height**	**Bitrate** [kbps]	**SSIM**	**Pr**
1	h264	High	384	216	384	216	145	0.9421	0.0030
2	h264	High	512	288	512	288	257	0.9556	0.0036
3	h264	High	768	432	768	432	488	0.9627	0.0082
4	h264	High	1024	576	1024	576	977	0.9720	0.0198
5	h264	High	1280	720	1280	720	1667	0.9769	0.0594
6	h264	High	1440	810	1422	800	2625	0.9815	0.8984
Average					**1394**	**784**	**2502**	**0.9806**	
Storage							**6159**		

Table 4 shows additional improvements delivered by CAE with the "true resolution" detection method enabled. We note that the average bitrate is now 2502 kbps vs. 3386 kbps, an extra 26.1% saving in bandwidth, and that overall storage is now 6159 kbps vs. 7912 kbps, a saving of 22.1%. The SSIM statistics are similar to the standard CAE. And yet, we also note that the average effective resolution as delivered is now 1394x784 vs.1392x783 – another slight improvement. The "true resolution" detection has worked – the top rendition resolution is now 1440x810 (a rounded-up 1422x800), enabling all mentioned improvements

Experimental Study

Finally, in this section, we present the results of an experimental study assessing the effects of CAE and CAE with "true resolution" detection on the efficiency of streaming systems.

Table 5 – HLS reference encoding vs. CAE and vs. CAE with true resolution detection.

Content Category	**Renditions**			**Storage [kbps]**			**Bandwidth [kbps]**		
	Ref.	**CAE**	**CAE+TR**	**Ref.**	**CAE**	**CAE+TR**	**Ref.**	**CAE**	**CAE+TR**
Action	9.00	6.08	6.08	24361	8221	7890	5420	3477	3319
Adventure	9.00	6.17	6.17	25803	8964	8602	5741	3648	3516
Baseball	7.00	5.12	5.00	11477	4389	3631	3823	2086	1739
Basketball	8.61	6.06	5.76	23684	8036	5753	5530	3441	2513
Beach Volleyball	9.00	6.91	6.21	25858	12004	8432	5753	4506	3538
Boxing	9.00	6.00	6.00	25588	7590	6388	5693	3225	2622
Cartoon	9.00	5.84	5.70	25256	6690	5791	5619	2923	2545
Comedy	9.00	6.26	6.26	26655	9081	8700	5931	3516	3389
Cricket	7.00	5.00	4.32	12222	3231	2578	4072	1470	1296
Cycling	9.00	6.00	5.96	26272	8036	6805	6146	3289	2757
Documentary	9.00	6.50	6.47	24804	10226	9886	5519	3935	3787
Drama	9.00	6.17	6.17	26560	8492	8188	5910	3498	3362
Field Hockey	9.00	6.92	6.55	26180	11478	9808	5825	4242	3816
Football	7.45	4.67	4.67	14964	4055	3302	4346	1578	1375
Game Show	9.00	6.20	5.89	25137	8615	7870	5593	3566	3374
Gymnastics	9.00	6.00	6.00	24396	7111	6428	5707	2902	2902
Interview	7.07	4.40	4.05	11794	2591	2042	3839	1224	1003
Kids Channel	9.00	6.31	6.24	25292	9521	9141	5628	3829	3650
Late night show	9.00	6.28	5.50	24736	8722	6918	5786	3530	2923
Mixed Sports	9.00	6.84	6.74	24783	12260	11683	5514	4472	4328
News	9.00	6.51	6.23	26893	10038	9088	6291	3869	3666
Reality	9.00	6.50	6.43	25501	10044	9567	5674	3907	3760
Running	9.00	6.37	6.00	24663	9291	7352	5488	3828	3197
Scifi	9.00	6.18	6.12	24370	8933	8457	5422	3670	3519
Sitcom	9.00	5.99	5.98	24381	7284	6773	5425	3061	2863
Soap	9.00	6.14	6.03	25327	8185	7684	5635	3394	3239
Squash	9.00	6.00	6.00	25721	6990	6247	5723	3030	2711
Swimming	9.00	7.00	7.00	25823	13874	12993	5746	4784	4614
Tennis	7.00	5.00	5.00	11961	3553	3047	3984	1711	1450
Weightlifting	9.00	5.90	5.71	25915	6085	5165	5766	2616	2257
Overall	**8.67**	**6.04**	**5.87**	**23212**	**8119**	**7206**	**5418**	**3274**	**2967**
CAE vs Ref [%]		**-30.30**	**-32.25**		**-65.02**	**-68.95**		**-39.57**	**-45.23**
CAE+TR vs CAE [%]			**-2.81**			**-11.25**			**-9.38**

To perform this study, we used a corpus of 500 video assets, with a combined duration of over 120 hours, representing 33 different content categories, such as action movies, sports, documentaries, etc.

All these assets were real-world 1080p and 720p mezzanines sampled from existing OTT distribution workflows. Each video was subsequently encoded using three encoding profiles (HLS reference [4], CAE, and CAE with enabled "true resolution" detection (CAE+TR). Then we instrumented players to play the content and collected the playback statistics.

Table 5 presents the results. The top rows present performance statistics as observed for each content category. The last three rows show the overall statistics across all categories, the relative savings delivered by CAE vs. reference profile, and the CAE + true resolution vs. standard CAE.

As can be observed, CAE-delivered savings are very significant. Overall, we note an almost 40% savings in bandwidth and about 65% savings in storage compared to the reference HLS profile over this test set.

However, the savings are even higher with "true resolution" detection. We observe that, on average, "true resolution" detection brings about 9.38% extra savings in bandwidth relative to the standard CAE. In terms of storage, the additional savings are 11.25%. There is also a 2.81% reduction in the number of encoded streams. On a per-category basis, we observe even higher savings. For example, we notice 26.97% savings in bandwidth and 28.41% in storage for basketball content. These are significant savings realized by using a sample of real-world media content.

Conclusions

We have discussed the problems posed by up-converted media content for video streaming applications. We have explained the origins of this problem, surveyed several existing tools and techniques that may be useful for addressing it, and proposed a method integrating them into a practical and easily deployable solution.

The presented experimental results indicate that the proposed solution is effective. Using a dataset with real-world mezzanines, we observed average bandwidth savings of approximately 9.38% and storage savings of 11.25%. Across different content categories, we noted that the savings are approaching 26.97% and 28.41% in bandwidth and storage usage, respectively.

We find these results both encouraging and alarming. On the one hand, they show that our proposed tool works and is effective. But on the other, they also indicate that a significant percentage of videos as distributed OTT today are, in fact, upscaled.

References

[1] Y. Reznik, J. Cenzano, and B. Zhang, "Transitioning Broadcast to Cloud," SMPTE Motion Imaging Journal, vol. 130, no. 9, pp. 18-32, Oct. 2021.

[2] R. Pantos and W. May, "HTTP live streaming, RFC 8216," IETF, 2017.

[3] ISO/IEC 23009-1:2019, "Information technology - Dynamic adaptive streaming over HTTP (DASH) - Part 1: Media presentation description and segment formats," ISO/IEC, Aug. 2019.

[4] Apple, "HTTP Live Streaming (HLS) Authoring Specification for Apple Devices,"https://developer.apple.com/documentation/http_live_streaming/http_live_streaming_hls_a uthoring_specification_for_apple_devices, Apple, November 12, 2021.

[5] Aaron, Z. Li, M. Manohara, J. De Cock, and D. Ronca, “Per-Title Encode Optimization,” 15 Dec. 2015. Online: https://medium.com/netflix-techblog/per-title-encodeoptimization-7e99442b62a2

[6] Y. A. Reznik, K. O. Lillevold, A. Jagannath, J. Greer, and J. Corley, "Optimal Design of Encoding Profiles for ABR Streaming," Proc. 23rd Packet Video Workshop (PV'2018), Amsterdam, The Netherlands, pp. 43–47, 12 Jun. 2018.

[7] Chen, Y. Lin, S. Benting, and A. Kokaram, "Optimized Transcoding for Large Scale Adaptive Streaming Using Playback Statistics," Proc. 25th IEEE Int. Conf. Image Processing (ICIP), Athens, pp. 3269–3273, Oct. 2018.

[8] Y. Reznik, X. Li, K. Lillevold, R. Peck, T. Shutt, and P. Howard, "Optimizing Mass-Scale Multiscreen Video Delivery," SMPTE Motion Imaging Journal, vol. 129, no. 3, pp. 26 - 38, 2020.

[9] Y. Reznik, "Average Performance of Adaptive Streaming," Proc. Data Compression Conference (DCC'21), Snowbird, UT, Mar. 2021

[10] M. Manohara, A. Moorthy, J. De Cock, I. Katsavounidis, and A. Aaron, "Optimized shot-based encodes: Now Streaming!," Mar 9, 2018. Online: https://netflixtechblog.com/optimized-shot-based-encodes-now-streaming-4b9464204830

[11] ISO/IEC 23090-3:2022, "Information technology — Coded representation of immersive media — Part 3: Versatile video coding," ISO/IEC, 2022.

[12] ISO/IEC 14496-10:2003, "Information technology – Coding of audio-visual objects – Part 10: Advanced Video Coding," ISO/IEC, December 2003.

[13] ISO/IEC 23008-2:2013, "Information technology – High efficiency coding and media delivery in heterogeneous environments – Part 2: High efficiency video coding," ISO/IEC, December 2013.

[14] X. Ducloux, J.-L. Diascorn, and T. Fautier, "Exploring the benefits of dynamic resolution encoding and support in DVB standards," IBC, Amsterdam, NL, 15-18 Sept. 2022.

[15] A. C. Gallagher, "Detection of linear and cubic interpolation in JPEG compressed images," Proc. The 2nd Canadian Conference on Computer and Robot Vision, 2005.

[16] R. G. Keys, "Cubic Convolution Interpolation for Digital Image Processing," IEEE Trans. Acoustics, Speech, Signal Proc, Vol. ASSP-29, No. 6, 1981, pp. 1153-60.

[17] X. Feng, I. J. Cox, and G. Doerr, "Normalized Energy Density-Based Forensic Detection of Resampled Image," IEEE Transactions on Multimedia, vol. 14, no. 3, 2012, pp. 536-545.

[18] Katsavounidis, A. Aaron, and D. Ronca, "Native Resolution Detection of Video Sequences," SMPTE Technical Conference, 2015.

[19] Z. Yang, Y. Dong, L. Song, R. Xie, L. Li and Y. Feng, "Native Resolution Detection for 4K-UHD Videos," IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), Paris, France, 2020, pp. 1-5.

[20] C. Dong, et al., "Image Super-Resolution Using Deep Convolutional Networks," IEEE Transactions on Pattern Analysis and Machine Intelligence 38.2(2014).

[21] C. Ledig, et al., "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) IEEE, 2017.

[22] B. Lim, et al., "Enhanced Deep Residual Networks for Single Image Super-Resolution," 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) IEEE, 2017.

[23] J. Kim, J. K. Lee, and K. M. Lee, "Accurate Image Super-Resolution Using Very Deep Convolutional Networks." (2015).

[24] M. Haris, G. Shakhnarovich, and N. Ukita, "Recurrent Back-Projection Network for Video Super-Resolution," Proc. Of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[25] Brightcove Context-Aware Encoding, https://apis.support.brightcove.com/general/overviewcontext-aware-encoding.html

[26] Brightcove VideoCloud platform, https://www.brightcove.com/en/onlinevideo-platform

[27] A. Eden, "No-reference estimation of the coding PSNR for H. 264-coded sequences," in IEEE transactions on consumer electronics, 667 –674 (2007).

[28] R. Vanam, K. Lillevold, Y. Reznik, "Assessing objective video quality in systems with multi-generation transcoding," Proc. SPIE Applications of Digital Image Processing, San Diego, CA, August 24-27, 2020.

[29] A.V. Oppenheim and R.W. Schafer, "Digital signal processing," Englewood Cliffs, N. J., Prentice-Hall, Inc., 1975. 598 p.

[30] D. P. Mitchell and A. N. Netravali, "Reconstruction filters in computer graphics," Computer Graphics, vol. 22, no. 4, pp. 221–228, Aug. 1988.

[31] "Tears of steel" video, Blender project. https://mango.blender.org/download/

[32] Z. Wang, A. Bovik, H. Sheikh, E. Simoncelli, "Image quality assessment: from error visibility to structural similarity". IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600–612, 2004