IBC2024

NOVEL CODING METHODS FOR STORAGE BIT-COST, TRANSCODING COMPLEXITY, AND TRANSMISSION EFFICIENCY TRADE-OFF OPTIMIZATION OF MULTI-PROFILE VIDEO DELIVERY SYSTEM

J. Le Tanou, R. Kaafarani, M. Ropert
MediaKind, France

ABSTRACT

This paper addresses the problem of multi-profile encoding and delivery system optimization for the purpose of standard HTTP-based Adaptive Bitrate (ABR) video streaming. Such delivery systems must process, encode, and usually store the same content in different (bitrate, resolution) pairs, which defines a set of encoding profiles or coded representations (a.k.a. bitrate ladder), to serve and adapt the video content to various end-user bandwidth requirements and device capabilities. The presented research work specifically targets such a system to improve the trade-off between the storage bit-cost of the different representations, the transcoding complexity and transmission efficiency (i.e. bitrate-quality trade-off at transmission) of the requested representation by the end-client while guaranteeing that the delivered output bitstream remains compliant with the legacy decoding system available at the client. For that purpose, a joint multi-profile coding format with corresponding fast transcoding method is proposed and assessed against State-of-the Art methods.

INTRODUCTION

Video streaming services heavily rely on HTTP-based Adaptive Bitrate (ABR) streaming technologies, such as Dynamic Adaptive Streaming over HTTP [1] or HTTP Live Streaming (HLS) [2], to serve video content to varying end-user device capabilities and network conditions. To adapt to the end-client request, ABR delivery system must commonly process, encode, and store the same video content in different resolution and bit-rate pairs, which define a set of encoding profiles or coded representations (a.k.a. bitrate ladder). As a first path of system optimization the bitrate ladder can be optimized per content, using traditional [3-5] or machine learning [6-8] approaches, and with knowledge of the video coding standard or codec efficiency in use [9]. Complementary, and by considering the signal redundancy between the representations, we investigate and introduce novel joint coding formats and transcoding methods to optimize the trade-off between storage cost, transcoding complexity, and transmission cost of those representations in a multi-profile video coding system for standard ABR streaming. The two most common approaches for multi-profile video delivery are Simulcast (SC) and Full Transcoding (FT) which set two extremes in terms of optimization criteria. Simulcast offers the lowest transcoding complexity (i.e. highest scalability to profile requests) and best transmission efficiency but requires the largest amount of storage, while FT has the lowest storage cost but the highest transcoding cost and transmission bitrate overhead.

As a third, in-between approach, we investigate methods that seek to optimize a better trade-off between storage, transcoding, and transmission costs, with similar motivations to a past MPEG initiative formalized by a Call for Evidence (CfE) on Transcoding for Network

Distributed Video Coding (NDVC) [10]. The CfE aimed to propose methods that lower the storage cost of SC while improving transmission efficiency (i.e. reducing bitrate overhead for same quality) of the delivered streams against FT or its transcoding complexity. It includes the additional constraint to keep the final served bitstream compliant with any standard decoding system. In that context, we specifically focus on the best-in-class method from the State-of-the Art (SOTA): the so-called Guided Transcoding using Deflation and Inflation (GTDI) method [11]. The GTDI method is based on predictive residual coding across profiles (or equivalently layers) to reduce storage requirements of low-quality representations. The method preserves the best transmission efficiency of SC and reduces its storage cost but still induces transcoding complexity to serve a dependent representation which requires the full decoding of the representations involved in an inter-layer residual prediction process. As a new trade-off, we propose and discuss a novel predictive residual coding scheme based on partial decoding for standard stream re-generation which significantly lowers the transcoding complexity of the GTDI approach by sacrificing on storage or transmission cost. In addition, we propose novel optimization techniques which improve storage saving of any method based on predictive residual coding. Typically, for the GTDI method, it can further improve the storage saving by -10% on average with a negligible transmission bitrate overhead.

For context completeness, scalable video coding techniques [12-13] are not considered in this work since our motivations (as for the CfE [10]) are to deliver a standard non-scalable stream and to guarantee the lowest bitrate overhead at transmission for the highest quality representation (i.e. nominal or reference bitstream) – representation which is expected to be the most served profile and to impact the CDN costs the most.

OVERVIEW OF AN ABR VIDEO DELIVERY SYSTEM AND ASSOCIATED COSTS

An ABR video streaming service, Live or On-Demand, is composed of 3 main parts: *Encoding*, *Content management* and *Delivery*, for which an overview is given ***Error! Reference source not found.*** with the involved processes (green) and the associated costs (dashed red) which are considered for optimization in the context of this paper.

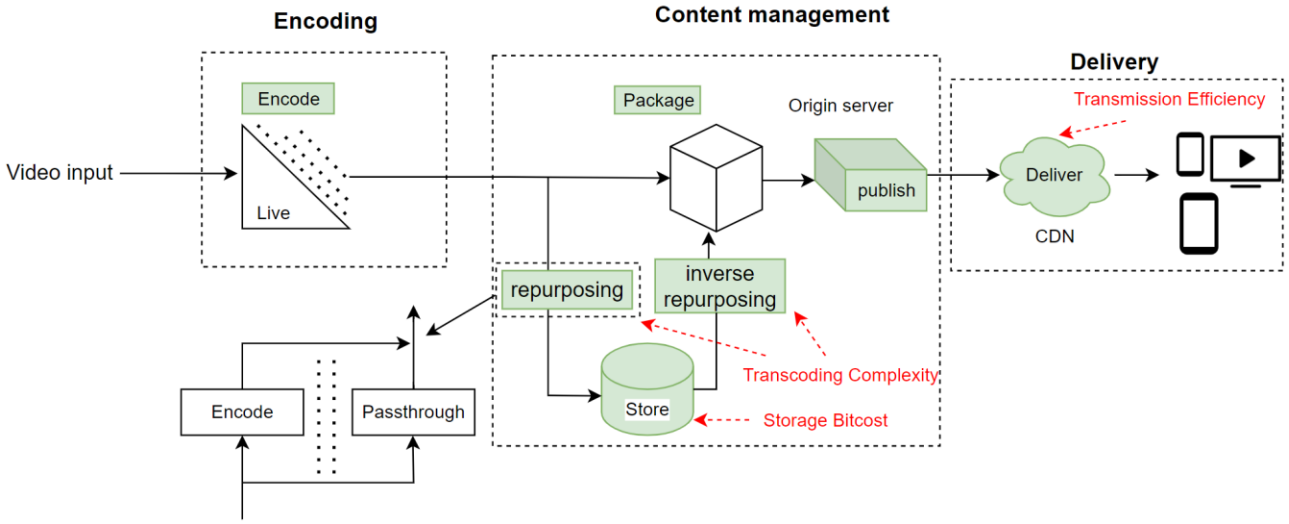

Figure 1 – Overview of an ABR video streaming service – main involved processes (green) and associated costs under consideration in the context of this invention(red)

Encoding – as introduced earlier the first step of the streaming system consists in encoding the input video in one or multiple representations with different (resolution, bitrate) pairs according to a pre-defined video bitrate ladder and coding strategy. Therefore, the cost

associated to the *Encoding* directly depends on the processing complexity of the bitrate ladder and the codec that is in use – it defines the *Encoding Complexity* as a first possible system optimization criterion which is not of focus in this paper.

Content management – at this stage the compressed streams are either packaged for direct publishing through the origin server (e.g. for Live video delivery) or re-purposed (e.g. transcode in a proprietary format for storage optimization) and stored for later playback (e.g. for On-Demand delivery). For the latter case, on a client request the corresponding stored representation must be inverse repurposed to a standard format for delivery. This system part induces two costs or criteria to optimize: the *Storage Bit-cost* and the *Transcoding Complexity*.

Delivery – in the last stage of the system the encoded and packaged videos are received from the origin server and published on the Content Delivery Network (CDN) [14] which optimizes the distribution of the contents to end users from different geographical regions. The CDN costs are directly dependent to the bitrate of the content to deliver which is itself correlated to the Rate-Distortion (R-D) performance achieved by the coding strategy used for delivery. The R-D performance criteria at transmission is defined as *Transmission Efficiency*.

In this paper we investigate solutions for optimizing the trade-off between the three criteria: storage bit-cost, transcoding complexity, and transmission efficiency.

COMPARATIVE ANALYSIS OF STATE OF THE ART METHODS

In this section, we review state of the art methods for multi-profile video coding and delivery and provide a qualitative comparison of their performance.

Simulcast

Simulcast (SC) [10] offers no transcoding complexity and optimal transmission efficiency in return of high storage bit-cost, simply by encoding independently all the bitrate ladder representations of a video upfront. These independent video streams are then stored directly with no re-purposing (which gets reduced to a pass-through as in Figure 1) such that no extra processing is needed once a representation is requested. SC sets the norm for the optimal video quality since there is no modification of the independent encoding process for each profile which leads to no bitrate overhead or quality degradation at transmission.

Full Transcoding

In contrast to SC, the Full Transcoding (FT) [10] technique offers maximal storage savings at the cost of very high transcoding complexity and lowest transmission efficiency. This strategy works by encoding and storing only the highest quality (HQ) representation of the video. By doing so, the storage requirements for this method are heavily reduced. However, once a user requests a representation of the video that is different from the HQ one, a full transcoding process (inverse re-purposing) must be performed in which the HQ representation is decoded (optionally down-sampled), and then re-encoded at the requested bitrate. This process is complex and requires costly computing power. In addition, and since the requested representation is a re-encoding of an already degraded video signal, the transmission efficiency of FT is sub-optimal. Either the quality of the dependent profiles would be degraded, or it would require targeting higher bitrate at encoding for the HQ profile to compensate for the quality degradation resulting in significant transmission bitrate overhead.

Several additional research works in Academic literature investigated alternative strategies to Simulcast and Full transcoding, as developed next sections.

Guided Transcoding

A Guided Transcoding (GT) approach was first proposed in [15] which aims to reduce the transcoding complexity of FT while still maintaining storage savings in comparison to SC. Similar to FT, it encodes a HQ representation and stores it as is. For the lower quality (LQ) encodings, the HQ representation is decoded, downsized, and then encoded at the required resolution/rate. However, the LQ streams are fully stripped from their transform coefficients before storage. Consequently, all the decisions of the encoder for the LQ streams are saved in what is called a Control Stream (CS). When a LQ representation is requested for delivery, the HQ representation is decoded, downsized, and then re-encoded by guiding the encoding process using the corresponding CS. Since the CS contains all the decisions needed for the encoder, complex R-D search operations for mode decisions can be skipped and the re-encoding process is reduced to the regeneration and entropy coding of the transformed coefficients. Another variant of GT was then proposed in [16] to further reduce its transcoding complexity. In this variant, not all the transform coefficients of the LQ streams are omitted but a fraction of them, which belong to pictures assigned to lower temporal layers in a dyadic hierarchical B picture prediction structure. The idea is that transform coefficients of pictures assigned to higher temporal layers usually have lower residual energy than those in lower layers and won't contribute much to the storage savings if omitted. Consequently, keeping them in the stream wouldn't require re-generation of these coefficients and thus, decreases the transcoding complexity for a small storage penalty. The method is flexible allowing the control of the number of layers for which the coefficients of pictures are removed. This offers a trade-off between storage savings and transcoding complexity.

The GT techniques achieve a good trade-off between storage savings and transcoding complexity. However, they achieve the same non-optimal transmission efficiency of FT resulting in significant bitrate overhead or quality degradation at transmission. Consequently, in this work, the GT schemes of [15], [16] aren't considered for a full objective comparison.

Guided Transcoding using Deflation and Inflation

Finally, a Guided Transcoding using Deflation and Inflation (GTDI) method has been proposed in [18]. The GTDI strategy specifically targets to reduce storage cost of SC with lower transcoding complexity than FT under the constraint of having the same transmission efficiency of SC. For that purpose, and despite not being formally defined as such, the scheme introduces the concept of Predictive Residual Coding (PRC) with Full decoding using spatial pixel domain reference samples (PRC-Full-PTQ). For the rest of the paper, and to better highlight the commonalities and differences with the new proposed method and optimizations in our work we renamed GTDI as PRC-Full-PTQ. The scheme is depicted in Figure 2 with new functionalities (in comparison to any standard hybrid coding scheme as specified in H.264/AVC, HEVC, VVC or AV1, etc.) to perform the prediction and differential coding of the residual shown in red. It shows the principles of deflation and inflation for an example of two layers (i.e. representations): a reference layer video V_0 and a dependent one V_1 where V_0 is the video representation of highest quality and resolution, and V_1 can be a representation of any quality and/or resolution lower than V_0.

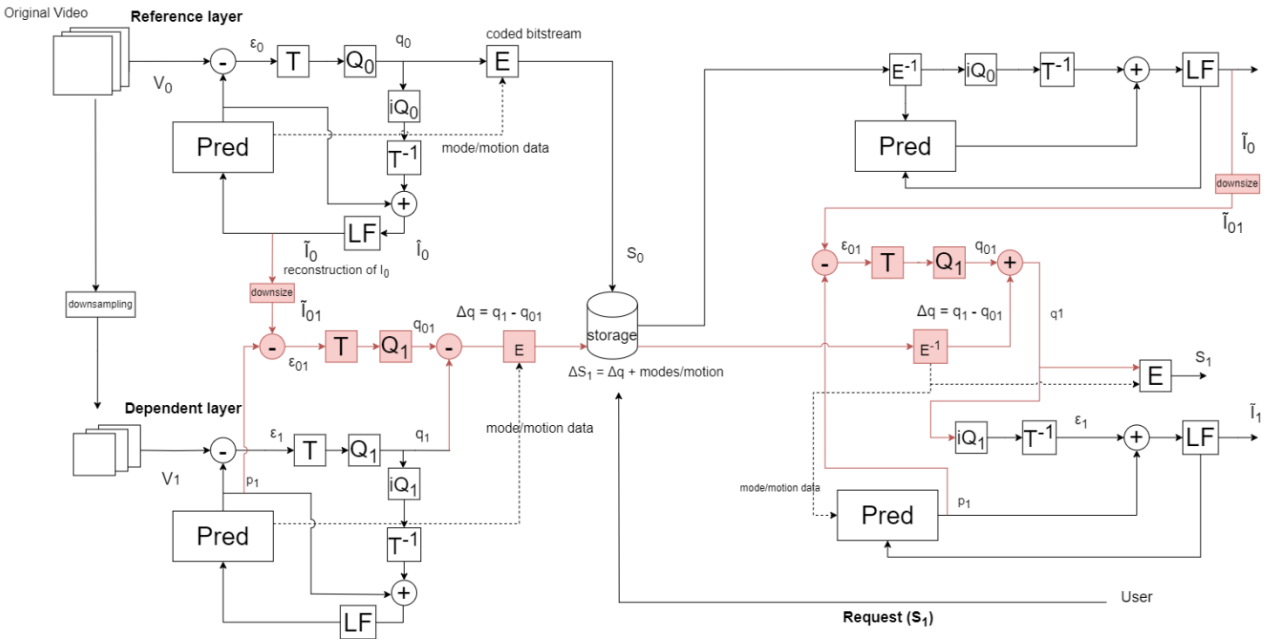

Figure 2 – Guided Transcoding using Deflation and Inflation (or PRC-Full-PTQ) method; showing deflated stream generation (repurposing) on the left and standard stream re-generation (inverse-repurposing) on the right. This approach uses reconstructed pixel samples from a reference high quality video to generate the residual predictor of the dependent lower quality videos.

The method starts by applying a *deflation* (re-purposing) process on LQ dependent videos before storage as follows:

- V_0 is normally encoded to generate a standard stream S_0.
- V_1 is normally encoded up to the point of entropy encoding that would have produced a standard stream S_1.
- The reconstructed images $\widetilde{I_0}$ of V_0 are (optionally) downsized into $\widetilde{I_{01}}$ to match the resolution of V_1.
- The prediction p_1 resulting from the encoding of V_1 is subtracted from $\widetilde{I_{01}}$ (P part for prediction in PTQ acronym) to form an approximate residual $\varepsilon_{01}$of ε_1.
- The approximate residual ε_{01} is then transformed and quantized (TQ part for transform and quantization in PTQ acronym) using the quantization parameter of V_1 to get q_{01}
- A difference between $q_{01}$and the original residual $q_1$is then calculated to get $\Delta q = q_1 - q_{01}$ which is the delta residual to be entropy coded.
- The delta residual and encoder decisions (modes, motion data etc.) of V_1 are entropy encoded to form a non-standard (i.e. deflated) stream that is called ΔS_1.

When a user requests a LQ stream that is represented by a dependent stream ΔS_1 in the scheme, the inverse of the repurposing process (inflation) must be invoked. Such, S_0 is fully decoded to get back the images $\widetilde{I_0}$ which are required to generate back the residual ε_{01} used for prediction. The residual ε_{01} is further transformed and quantized to q_{01}, which is added back to Δq to get back the original residual q_1. A standard Context Adaptive Binary Arithmetic Coding (CABAC) (or any entropy coder as adopted in the considered codec) encoding process of q_1 along with modes and motion data is then carried out to form a compliant stream S_1. Both deflation and inflation operations use the same configurations to ensure that the exact same LQ stream is generated than in the case of Simulcast. Consequently, this

scheme achieves the same transmission efficiency as SC but with lower storage requirements. On the transcoding side, the inflation process to re-generate a LQ stream is coarsely equivalent to the cost of two full decoding loops, which results in this method being faster than Full Transcoding which requires to perform a full decoding followed by a complete encoding with complex RD optimization.

PROPOSAL METHODS FOR SYSTEM COST OPTIMIZATION

The proposed work is twofold by targeting to lower the transcoding complexity, and to improve the trade-offs between storage bit-cost and transmission efficiency of the best-in-class method from SOTA namely the GTDI approach (PRC-Full-PTQ). For that purpose, and as in GTDI (PRC-Full-PTQ), we leverage the redundancy between the residuals of the various video representations by means of predictive residual coding techniques with the main contributions being:

1) To lower the transcoding complexity of the GTDI approach (PRC-Full-PTQ), we propose the idea of Predictive Residual Coding (PRC) with Partial decoding using spatial residual domain reference samples (PRC-Part-TQ)
2) To further improve the coding efficiency of any method based on predictive residual coding, such as PRC-Full-PTQ (GTDI from SOTA) or PRC-Part-TQ (proposal 1), we propose two optimizations:
   a) First optimization conditions the delta residual coding and signalling to ensure lower residual energy,
   b) The second optimization modifies the Rate-Distortion optimization criteria commonly used for coding mode decisions to favour prediction and splitting modes that minimize the final coded delta residual – hence improving the prediction of the residual data.

Partial decoding using spatial residual domain reference samples (PRC-Part-TQ)

PRC-Part-TQ method relies on partial decoding using spatial residual domain reference samples to lower the transcoding complexity of the GTDI / PRC-Full-PTQ approach. The corresponding coding scheme is depicted in Figure 3. In this approach, a residual predictor based on the inverse transformed and inverse quantized residual image of the reference layer video is used (instead of spatial pixel domain reference samples in case of GTDI/PRC-Full-PTQ). A reference layer video V_0 is normally encoded at the highest resolution/quality and saved as is. For the dependent video V_1 the following re-purposing process is invoked:

- The residual image of V_0 is re-scaled to the resolution of V_1 then stored in a buffer.
- The encoding process of V_1 is carried out normally and the encoder is left to make its optimal decisions as for a SC stream.
- Before entropy encoding, and for each coding unit (CU), the corresponding position and area in the residual image of V_0 is transformed and quantized (*TQ* part for *Transform* and *Quantization* in PRC-Part-TQ acronym) to obtain q_{01} (after re-scaling if necessary) before being subtracted from the original residual q_1 of V_1 which leads to $\Delta q = q_1 - q_{01}$.
- The delta residual Δq along with the optimal encoder decisions are entropy encoded to generate ΔS_1 dependent stream.

To re-generate the standard Simulcast version S_1 from the dependent stream ΔS_1, upon user request, the following inverse re-purposing process must be invoked:

- S_0 is entropy decoded and the coefficients are inverse quantized, and inverse transformed to obtain the residual image which is then re-scaled to match the resolution of S_1

- From ΔS_1, the delta coefficients are entropy decoded to obtain Δq. Then, for each CU, the co-located area in the residual image of S_0 is transformed and quantized to get q_{01} which is added to Δq to get back the original residual q_1.

Finally, the original residual q_1 is entropy encoded along with the encoder decisions to obtain the standard simulcast stream S_1.

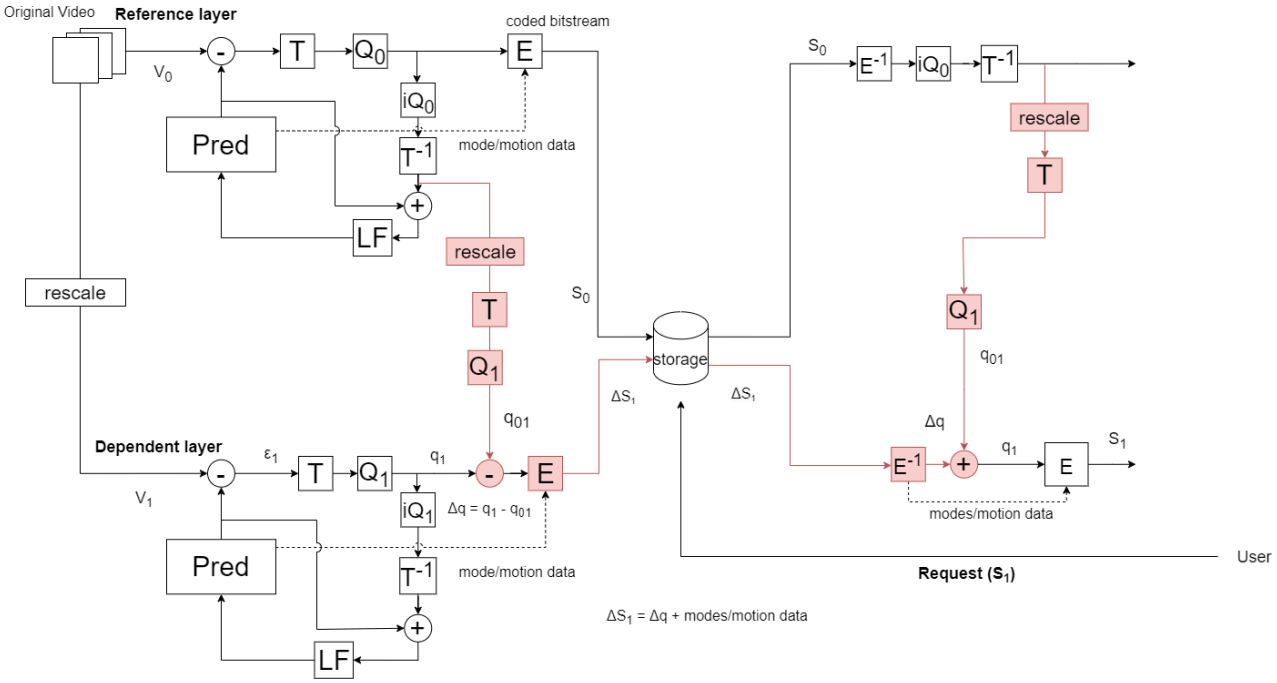

Figure 3 – proposed PRC-Part-TQ method; showing deflated stream generation (re-purposing) on the left and standard stream re-generation (inverse-repurposing) on the right. This approach uses spatial residual samples from a reference high quality video to generate the residual predictor of dependent lower quality videos.

This basis proposal method reduces the transcoding complexity, to generate back a standard stream equivalent to Simulcast, to only 2 partial decoding loops (with re-scaling if necessary) and an entropy encoding operation. It achieves the same optimal transmission efficiency as SC and GTDI/PRC-Full-PTQ while saving on storage bit-cost of dependent streams, by encoding a difference of residuals.

Coding Efficiency Optimizations

To further improve the coding efficiency of the base method PRC-Part-TQ, or any SOTA method based on predictive residual coding such as in GTDI/PRC-Full-PTQ, we propose two complementary optimizations.

Conditional Delta Residual (CDR) coding and signalling

For the base method PRC-Part-TQ or the SOTA method GTDI/PRC-Full-PTQ, a delta residual is calculated and coded for every coding unit (CU) in a group of pictures. However, for some cases, if the residual predictor is not well correlated with the residual blocks to predict then coding the delta-residual can result in a significant bit-cost overhead. To address this issue, we introduce a novel condition for coding the delta-residual of the dependent layer. The idea is to code the inter-layer delta residual only if it lowers the residue energy. More precisely, the differential residual is coded if and only if it satisfies:

Eq. 1. $$\sum_{k=1}^{numComp}\sum_{j=1}^{h/s}\sum_{i=1}^{w/s}\left|\Delta q_{i,j}^{k}\right| < \sum_{k=1}^{numComp}\sum_{j=1}^{h/s}\sum_{i=1}^{w/s}\left|{q_1}_{i,j}^{k}\right|$$

Where $numComp$ is the number of color components (e.g. 3 for YCbCr), w and h are the width and height of the current coding block (or unit) respectively, Δq and q_1 are the delta residual and original residual, s a scale factor according to the color component and chroma sub-sampling (e.g. for YCbCr 4:2:0, $s = 2$ for Cb/Cr and $s = 1$ for Y). If this condition is not satisfied, then the original residual q_1 of the block is coded.

To control this condition and to be able to have a decodable stream, a flag called *InterLayerResidualPrediction* is added and coded for each CU which indicates if it is inter-layer predicted (true) or not (false). The flag can be entropy coded using CABAC (or any other entropy coder as per the considered codec for implementation) using either the bin probability initialization states of the root Coded Block Flag if available (root CBF as standardized in H264/AVC, HEVC or VVC) or any custom bin probability model that can be typically contextualized according to neighbouring flag values (e.g. top or left coding block neighbours).

CDR coding and signalling demonstrates to improve the storage savings on dependent streams with no impact on the quality at transmission or on the transcoding complexity.

Rate-Distortion Optimization based on Delta Residuals (RDODR)

The idea is to update the Rate-Distortion Optimization (RDO) process commonly used for coding mode search and decision by using delta residual bit-cost for the rate estimations, to favour prediction and splitting modes that will minimize the delta residual to code for the dependent streams.

In a common RDO process, the encoder exhaustively tests different prediction and splitting modes or options ($\forall p \in P$), then decides which mode to use for a given block or unit based on the minimization of a rate-distortion cost function defined as $J(R, D) = D + \lambda . R$ where R is the bit-cost, D is the distortion and λ is the Lagrange multiplier that balances the importance of bit-cost and distortion.

For each coding unit, and candidate coding mode $\forall p \in P$, the distortion D is typically estimated by performing the prediction, transform, quantization and inverse processes plus optional in-loop filtering and measured the distance (e.g. L2 based on MSE) of the reconstructed samples with the source samples. The rate or bit-cost is usually estimated by invoking the pseudo-coding of the prediction mode and transformed quantized residuals (i.e. q_1) using a CABAC (or any other entropy coder as per the considered codec) estimation process, as formalized Eq. 2.

Eq. 2. $$p^* = argmin_p\big(D(p) + \lambda . R(q_1|p)\big)$$

In the context of any predictive residual coding scheme, we propose to update the bitrate estimations in the RDO process, such that the delta residual bit-cost (i.e. Δq) is calculated for each block instead of the default residuals q_1, as formalized Eq. 3. Such optimization can be combined with the prior proposal of CDR coding and signalling such that the appropriate bit-cost of delta-residuals or residuals is estimated according to the condition defined Eq. 1.

Eq. 3. $$p^* = argmin_p\big(D(p) + \lambda . R(\Delta q|p)\big)$$

Such optimization enables further storage bit-cost saving for no impact on the transcoding complexity. However, it can slightly lower the transmission efficiency (e.g. vs SC) but still being negligible with respect to storage saving benefits.

PERFORMANCE ASSESSMENT

The different proposals, as well as methods from SOTA namely Simulcast, Full Transcoding, GTDI/PRC-Full-PQT, have been implemented and compared in the context of VVC codec [17]. They were all implemented on top of VTM version 19.0 [18]. For techniques based on Predictive Residual Coding, including the different proposal variants and GDTI/PRC-Full-PTQ, the VVC multi-layer coding structure was leveraged on using VTM Multi-Layer Main 10 profile. With the Layer 0 set as the reference video layer and Layer 1 set as the dependent video layer. For the re-scaling of the reconstructed and residual reference samples between layers, the Reference Picture Resampling (RPR) filter (as specified in VVC standard) was used (but any resampling filter can be used).
The performance of the different predictive residual coding schemes, including the proposed method, optimizations, and GDTI/PRC-Full-PTQ, are assessed and compared to SC and FT. The performances at different stages of the video delivery scheme are considered: storage bit-cost, transmission efficiency and transcoding complexity for two *Multi-Rate* and *Multi-Resolution* video delivery scenarios defined as follows:

1) A *Multi-Rate* scenario: in this scenario, we consider a fixed resolution bitrate ladder where the representations vary in bitrate only according to the chosen QP value. All the streams are encoded using the native resolution of the test sequence. The reference stream is encoded with a QP value $QP_0 \in \{22, 27\}$. The dependent streams are then encoded using the following QP values: $QP_1=QP_0+\text{offset}$ where $\text{offset} \in \{2,4,6,8\}$ which yields $QP_1 \in \{24,26,28,30\}$ for $QP_0=22$ and $QP_1 \in \{29,31,33,35\}$ for $QP_0=27$.
2) A *Multi-Resolution* scenario: in this scenario, we consider a bitrate ladder where the dependent streams can be of resolutions different from the native one with varying bitrates for the same resolution. The reference layer is fixed at the native resolution L_0 of the test sequence which is 2160p for classA and 1080p for classB sequences. The QP value of the reference layer is $QP_0 \in \{22,27\}$. As for the dependent streams, the resolution called L_1 is defined as $L_1 \in \{1440p,1080p,720p,540p,360p\}$ for classA sequences, and $L_1 \in \{720p,540p,360p\}$ for classB sequences, as per the Common Test Conditions (CTC) of the MPEG CfE on NDVC [10]. The down-scaled versions of each of the sequences are generated with FFmpeg using its bi-cubic filter. In addition, for each sequence and each resolution, dependent streams were encoded using same QP values QP_1 than in the previous *Multi-Rate* scenario.

The Table 1 and Table 2 summarize the performance results for the different SOTA methods (marked by an asterisk (*) and framed in dashed orange) and proposals (framed in dashed green). The results for storage bit-cost savings are shown for two cases: when considering all streams (“All” column) and when only considering dependent streams (“Dependent” column). For transmission efficiency and transcoding complexity, the results can only be shown for dependent streams and are averaged over the different sequences and QP values QP_1. The transmission efficiency results were compared to those of the SC encodings on a similar quality basis. For that purpose, 3rd order R-D polynomial functions were estimated using bitrates and PSNRs of each of the SC sequences. Then, for each PSNR of a sequence in the tested methods, the corresponding SC bitrate is interpolated using the polynomial

function. Hence, the resulting bitrate is the SC bitrate at the same quality of the tested approach. The methodology to calculate the different savings at the different stages were taken from the CfE on NDVC [10] and are as follows:

- For storage bit-cost:

$$Diff(storage)_{SC} = 100 \times \frac{\sum_{n=0}^{N-1} \tilde{r}_n - \sum_{n=0}^{N-1} r_n}{\sum_{n=0}^{N-1} r_n}$$

  where $\tilde{r}_n$ is the bitrate of stream n for the method under test, r_n is the SC bitrate of stream n and N is the total number of streams (representations) for a specific sequence. To note that for "*dependent*" streams only storage saving measurements, the reference stream (i.e. index 0) is omitted in the sums.
- For transmission efficiency:

$$Diff(transmission)_{SC} = 100 \times \frac{\tilde{r}_n - \hat{r}_n}{\hat{r}_n}$$

  where $\hat{r}_n$ is the SC bitrate of stream n interpolated to match the PSNR of $\tilde{r}_n$ for fair comparison.
- For transcoding complexity:

$$Diff(complexity)_{ref} = 100 \times \frac{t_{method_n} - t_{ref_n}}{t_{ref_n}}$$

  where t_{method_n} is the transcoding time of representation n for the method under test, t_{ref_n} is the transcoding time of representation n for the reference method (FT or GTDI/PRC-Full-PTQ)

Table 1 – Performance of the different approaches and their variants in a Multi-Rate Scenario

			Storage Bitcost vs SC				Transmission efficiency vs SC		Transcoding complexity vs FT	Transcoding complexity vs PRC-Full-PTQ
		Streams:	All		Dependent		Dependent		Dependent	Dependent
	Approach	Variant	QP_0 22	QP_0 27	QP_0 22	QP_0 27	QP_0 22	QP_0 27	QP_0 22, 27	QP_0 22, 27
SOTA	Full Transcoding* [10]	NA	-60.5%	-66.1%	-100%	-100%	14.2%	17.5%	0%	2079.8%
SOTA	PRC-Full-PTQ	Base* [11]	-25.3%	-25.5%	-41.4%	-38.5%	0%	0%	-95.2%	0%
Proposals		CDR	-24.8%	-24.4%	-40.5%	-36.8%	0%	0%	-95.2%	0%
Proposals		RDODR	-28.8%	-30.3%	-47.1%	-45.6%	0.8%	-0.2%	-95.2%	0%
Proposals		CDR+RDODR	-29.0%	-30.3%	-47.3%	-45.7%	0.4%	-0.9%	-95.2%	0%
Proposals	PRC-Part-TQ	Base	-6.4%	-6.9%	-10.5%	-10.5%	0%	0%	-98.5%	-67.5%
Proposals		CDR	-7.9%	-7.8%	-13.0%	-11.8%	0%	0%	-98.5%	-67.5%
Proposals		RDODR	-10.1%	-10.5%	-16.5%	-15.9%	3.4%	1.7%	-98.5%	-67.5%
Proposals		CDR+RDODR	-10.7%	-10.8%	-17.6%	-16.4%	1.8%	0.8%	-98.5%	-67.5%

Table 2 – Performance of the different approaches and their variants in a Multi-Resolution Scenario

			Storage Bitcost vs SC				Transmission efficiency vs SC		Transcoding complexity vs FT	Transcoding complexity vs PRC-Full-PTQ
		Streams:	All		Dependent		Dependent		Dependent	Dependent
	Approach	Variant	QP_0 22	QP_0 27	QP_0 22	QP_0 27	QP_0 22	QP_0 27	QP_0 22, 27	QP_0 22, 27
SOTA	Full Transcoding* [10]	NA	-76.8%	-82.1%	-100%	-100%	6.1%	6.4%	0%	910.3%
SOTA	PRC-Full-PTQ	Base* [11]	-22.5%	-25.7%	-28.9%	-31.4%	0%	0%	-79.9%	0%
Proposals	PRC-Part-TQ	CDR	-25.6%	-25.7%	-33.2%	-31.5%	0%	0%	-79.9%	0%
Proposals	PRC-Part-TQ	RDODR	-29.3%	-31.6%	-38.0%	-38.7%	7.2%	3.0%	-79.9%	0%
Proposals	PRC-Part-TQ	CDR+RDODR	-30.6%	-32.5%	-39.6%	-39.7%	0.7%	-0.9%	-79.9%	0%
Proposals	PRC-Part-TQ	Base	-2.5%	-3.3%	-3.5%	-4.2%	0%	0%	-89.9%	-48.2%
Proposals	PRC-Part-TQ	CDR	-5.7%	-5.4%	-7.6%	-6.6%	0%	0%	-89.9%	-48.2%
Proposals	PRC-Part-TQ	RDODR	-5.9%	-6.5%	-7.9%	-8.0%	1.9%	1.2%	-89.9%	-48.2%
Proposals	PRC-Part-TQ	CDR+RDODR	-8.3%	-8.2%	-11.0%	-10.1%	0.7%	0.4%	-89.9%	-48.2%

The proposed PRC-Part-TQ method based on the concept of partial decoding using spatial residual domain reference samples enables significant reduction of the GTDI transcoding complexity, with about -68% and -48% transcoding run-time acceleration in Multi-Rate and Multi-Resolution scenarios, respectively. The transcoding complexity reduction comes at the expense of loss in storage savings which can be mitigated by the proposed optimizations CDR+RDODR – resulting in about -17% and -11% storage bit-costs on dependent streams in comparison to SC for the Multi-Rate and Multi-Resolution scenarios, respectively, and negligible impacts on transmission efficiency (i.e. < 1%) – which can set an interesting trade-off if transcoding complexity is of concern for the targeted application case.

If transcoding complexity is less of concern – the GTDI/PRC-Full-PTQ approach combined with the two proposed optimizations CDR+RDODR can significantly improve the base method in terms of storage savings – with -9.5% additional savings on average across the tested scenarios – while keeping the same initial method benefits in comparison to FT: i.e. a much lower transcoding complexity (about -95% and -80% across the tested scenarios) than FT and near equivalent transmission efficiency than SC; while 11% bitrate overhead is observed on average for FT at transmission for the tested conditions.

CONCLUSION

In this paper, we studied the problem of multi-profile video coding and delivery system optimization for ABR video streaming. We looked at optimizing the system globally for the three criteria: storage bit-cost, transmission efficiency, and transcoding complexity. We first proposed the idea of partial decoding in a differential predictive coding scheme, such as the Guided Transcoding Using Deflation and Inflation (GTDI) method. This allows transcoding complexity reduction at the expense of loss in storage savings. Across the different test conditions and scenarios, the best proposed variant achieves -48.2% transcoding complexity reduction over GTDI and 89.9% over Full Transcoding (FT) for about -9.5% additional storage savings on average and negligible transmission bitrate increase (<1%) in comparison to Simulcast (SC); while 11% bitrate overhead can be observed on average for FT. Moreover, the two R-D optimizations introduced show to further improve the storage savings of GTDI by -11% on average for near equivalent transmission efficiency than SC and same transcoding complexity reduction benefit (i.e. 85+% faster than FT). This later proposal shows an excellent trade-off for common ABR delivery applications.

REFERENCES

1. MPEG, “MPEG, “Dynamic Adaptive Streaming Over HTTP,” ISO/IEC 23009. Available: https://mpeg.chiariglione.org/standards/mpeg-dash

2. Apple, “HLS Authoring Specification for Apple Devices.” [Online]. Available: https://developer.apple.com/documentation/httplivestreaming/hlsauthoringspecificationforappledevices
3. J. De Cock, Z. Li, M. Manohara, and A. Aaron, “Complexity-based consistent-quality encoding in the cloud,” in IEEE International Conference on Image Processing (ICIP), 2016, pp. 1484–1488.
4. Y. A. Reznik, X. Li, K. O. Lillevold, A. Jagannath, and J. Greer, “Optimal multi-codec adaptive bitrate streaming,” in IEEE International Conference on Multimedia and Expo Workshops (ICMEW), 2019, pp. 348–353.
5. Y. A. Reznik, K. O. Lillevold, A. Jagannath, J. Greer, and J. heng, Corley, “Optimal design of encoding profiles for abr streaming,” in Proceedings of the 23rd Packet Video Workshop, ser. PV ’18. New York, NY, USA: Association for Computing Machinery, 2018, p. 43–47
6. A. V. Katsenou, J. Sole, and D. R. Bull, “Efficient bitrate ladder construction for content-optimized adaptive video streaming,” IEEE Open Journal of Signal Processing, vol. 2, pp. 496–511, 2021.
7. M. Takeuchi, S. Saika, Y. Sakamoto, T. Nagashima, Z. C K. Kanai, J. Katto, K. Wei, J. Zengwei, and X. Wei, “Perceptual quality driven adaptive video coding using JND estimation,” in Picture Coding Symposium (PCS), 2018, pp. 179–183.
8. Y. A. Reznik, K. O. Lillevold, A. Jagannath, J. Greer, and J. Corley, “Optimal design of encoding profiles for ABR streaming,” in Proceedings of the 23rd Packet Video Workshop, ser. PV ’18. New York, NY, USA: Association for Computing Machinery, 2018, p. 43–47.
9. R. Kaafarani, M. Blestel, T. Maugey, M. Ropert, and A. Roumy, “Evaluation of bitrate ladders for versatile video coder,” in International Conference on Visual Communications and Image Processing (VCIP), 2021, pp. 1–5.
10. MPEG, “Call for Evidence on Transcoding for Network Distributed Video Coding,” July 2017. [Online]. Available: https://mpeg.chiariglione.org/standards/exploration/network-distributed-media-coding/call-evidence-transcoding-network-distributed
11. C. Hollmann and R. Sjöberg, “Guided transcoding using deflation and inflation,” in Proceedings of the 23rd Packet Video Workshop, ser. PV’18. New York, NY, USA: Association for Computing Machinery, 2018, p. 19–24.
12. H. Schwarz, D. Marpe, and T. Wiegand, “Overview of the scalable video coding extension of the h.264/avc standard,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 17, no. 9, pp. 1103– 1120, 2007.
13. J. M. Boyce, Y. Ye, J. Chen, and A. K. Ramasubramonian, “Overview of shvc: Scalable extensions of the high efficiency video coding standard,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 26, no. 1, pp. 20–34, 2016.
14. Akamai Technologies, “What is a CDN?” 2024. [Online]. Available: https://www.akamai.com/glossary/what-is-a-cdn
15. G. Van Wallendael, J. De Cock, and R. Van de Walle, “Fast transcoding for video delivery bym eans of a control stream,” in 19th IEEE International Conference on Image Processing, 2012, pp. 733–736.
16. T. Rusert, K. Andersson, R. Yu, and H. Nordgren, “Guided just-in-time Transcoding for cloud-based video platforms,” in IEEE International Conference on Image Processing (ICIP), 2016, pp. 1489–1493.
17. B. Bross, Y.-K. Wang, Y. Ye, S. Liu, J. Chen, G. J. Sullivan, and J.R. Ohm, “Overview of the versatile video coding (VVC) standard and its applications,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 31, no. 10, pp. 3736–3764, 2021.
18. MPEG ITU-T, “Test model of Versatile Video Coding (VTM)”. [Online]. Available: https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftwareVTM