Optimal Reference Selection for Random Access in Predictive Coding Schemes

M.- Q. Pham, *Member IEEE,* A. Roumy, *Member IEEE,* T. Maugey, *Member IEEE,*
E. Dupraz, *Member IEEE,* and M. Kieffer, *Senior Member IEEE,*

Abstract

Data acquired over long periods of time like High Definition (HD) videos or records from a sensor over long time intervals, have to be efficiently compressed, to reduce their size. The compression has also to allow efficient access to random parts of the data upon request from the users. Efficient compression is usually achieved with prediction between data points at successive time instants. However, this creates dependencies between the compressed representations, which is contrary to the idea of random access. Prediction methods rely in particular on reference data points, used to predict other data points, and the placement of these references balances compression efficiency and random access. Existing solutions to position the references use ad hoc methods. In this paper, we study this joint problem of compression efficiency and random access. We introduce the storage cost as a measure of the compression efficiency and the transmission cost for the random access ability. We show that the reference placement problem that trades off storage with transmission cost is an integer linear programming problem, that can be solved by standard optimizer. Moreover, we show that the classical periodic placement of the references is optimal, when the encoding costs of each data point are equal and when requests of successive data points are made. In this particular case, a closed form expression of the optimal period is derived. Finally, the optimal proposed placement strategy is compared with an ad hoc method, where the references correspond to sources where the prediction does not help reducing significantly the encoding cost. The optimal proposed algorithm shows a bit saving of -20% with respect to the ad hoc method.

Index Terms

Predictive Coding, Random Access, Integer Linear Programming.

I. INTRODUCTION

Data acquired over a long period of time, for instance a long HD video, or time serie measurements acquired by a sensor, require compression to be stored on a server. Random access (RA) to compressed data is the ability for a user to access any temporal interval, for instance any frame within a video, or any temporal sample acquired by the sensors. On the other hand, efficient compression requires to exploit temporal correlations by jointly processing the data points at successive time instants, see [7, Th. 2.6.6]. This creates dependencies between the successive compressed representations of data points and makes it impossible to directly access any data point at any time instant without sending additional unrequested data points. This paper aims to address the resulting compromise between RA and compression efficiency.

Predictive coding (PC) is one way to achieve efficient compression and is used in many standards for speech, audio, image, and video compression [17]. PC is well adapted to data points that are available sequentially, as in standard video coding [9], [21], [24] and streaming video coding [2]. In PC, there exists two types of sources. First, *reference* sources are coded independently of the others. Second, *predicted* sources are predicted from other sources (either reference or predicted), and only the prediction residual is encoded. PC is very popular since it is known to optimaly exploit the correlation between data points, for a large class of source models. For instance, for lossless compression, if each data point is modeled as the realization of a source, PC achieves the joint entropy of the sources and is therefore optimal [7, Th. 2.5.1] (chain rule). For lossy compression, optimality results exist for Gauss-Markov sources and linear prediction [22, Sec. 6.4.1].

The tradeoff between RA and PC is a longstanding problem in the video coding literature [9], [21], [24]. In video coding, RA is expressed as the ability to start reading a video stream at an arbitrary moment, even if the file has been damaged, or if the users do not start watching at the beginning of the video stream. A classical solution for RA consists in periodically placing references. For instance, in [23], a reference is imposed every eight frames in order to limit the delay to access the data. The video content may also be taken into account to position the references, as in [1]. Or some of the frames may be stored twice, both as a reference and as a prediction, which lowers the storage efficiency [11]. Apart from video coding, RA was also addressed for sensor networks measurements [15], [19]. However, in these works, all the sources are encoded as references. Genome [8] and Internet of Things [20] databases were also considered in the context of RA, as well as some specific sequential lossless compression algorithms, such as Lempel-Ziv [10], or Burrows-Wheeler [18]. What is common to all the above works

is to seek at determining a set of reference positions in order to allow for RA. However, all the proposed solutions are either suboptimal, or specific to a particular coding algorithm.

Therefore, the objective of this paper is to introduce a generic method in order to determine the set of optimal reference positions that will address the tradeoff between RA and compression efficiency. In order to address the RA problem for a wide range of applications, we model the data to be compressed as a sequence of sources indexed by time. Each source of the sequence generates a sample vector which may correspond to a frame (standard video coding), to a set of simultaneous frames with different viewpoints (multiview video coding), or to a set of measurements taken by multiple sensors at the same time instant (sensor networks). Within this model, we define a request as a user-dependent set of sources to be transmitted to the user.

Optimal positioning of references was investigated in the context of tiling for omnidirectional visual content. In this context, the request now concern tiles, where a tile is a subpart of the onmidirectional image and can be seen as a view in multiview coding [12], [13], [25]. Put in our model, [12], [13], [25] have an additional constraint compared to our work, that is that if the request ends between two references, then the whole segment, from previous reference to next reference, is transmitted to the user. Besides, the above works optimize some specific parameters, that are segment size in [12], resolution at which tiles are transmitted in [13], and segment prediction method in [25]. Therefore, these methods do not allow to completely address the tradeoff between RA and compression efficiency.

In order to jointly study the compression and RA problems, we propose to measure compression efficiency with the *storage cost*, *i.e.* the amount of bits needed to encode the whole data. We further propose to measure the RA ability with the *transmission cost*, *i.e.*, the cost to send requests to the users. The introduced transmission cost takes also into account the popularity of the requests modeled as a probability distribution defined over the set of possible requests. In order to take both criterion into account, we then propose to optimize a single metric given by the weighted sum of the storage cost and the transmission cost. In particular, we show that the reference placement problem that trades off storage with transmission cost is an integer linear programming problem, that can be solved by standard optimizer [16]. Moreover, we show that the commonly considered periodic placement of the references [23] is not optimal in general, unless some very restrictive assumptions are considered, that are for instance that the encoding costs of each data point are equal (*i.e.* all data points when compressed as a reference require the same cost, and when compressed as a predicted source require the same cost, but not necessarily equal to the previous one) and that all requests are equally probable.

We demonstrate the performance of the proposed method for: 1) the particular case where the data points have the same encoding cost (*i.e.* the encoding costs of the sources as a reference are equal,

and the encoding costs as predicted are also equal) and 2) the general scenario, where the sources have variable encoding costs. Experiments are conducted on various applications. We first consider the case where a source is a frame of a video encoded with the VVC Test Model (VTM) version 6.2 for VVC Versatile Video Coding (VVC) [5]. We then apply our method to meteorological data and to the occupancy measure of self-service bicycle terminals in Paris. For each case, the transmission cost is evaluated with different popularity distributions: either all sources are equally popular, or with significantly unequal popularity. Depending on the application, we consider different unequal popularity distributions. We use a log-normal distribution for video coding [4], a Gaussian distribution for meteorological data, and an empirical frequency distribution based on real measurements for self-service bicycles.

In order to show the interest of our approach, we compare our optimization results with two other methods. First, we consider a naive approach, in which sources with high predictive encoding cost are chosen to be a reference. Second, we incorporate the tiling constraint of [12], [13], [25] into our method, and derive the corresponding optimal solution. Experimental results are compared in terms of BD-rates for the video coding application, and in terms of rate-saving for the other two applications. Our method shows around -20% of bit-rate saving compared to the naive approach, and around -3% of bit-rate saving compared to the optimized tiling approach.

Notation: Upper case letters denote either a source or a random variable. $\lfloor x \rfloor$ and $\lceil x \rceil$ denotes rounding x to the nearest integer less than, greater than or equal to it, respectively.

II. SYSTEM MODEL AND PROBLEM FORMULATION

In this section, the joint optimization problem of the compression efficiency and random access is derived. First, the source model and the encoding scheme are presented. Then, storage and transmission costs are formally defined. These two quantities form the cost function to be optimized, and this leads to the problem formulation.

A. Source model, coding scheme and source delivery

The data is modeled as the realizations of N correlated sources that can be represented by a N-dimensional vector of random variables denoted $(X_1, \ldots, X_N)$, where each X_n is a multivariate variable of dimension d.

The sources are first compressed according to a predictive scheme, meaning that a source is encoded either independently of the other sources (the source is then said to be a reference or to be *intra-coded*) or by exploiting the knowledge of the previous one, see Fig. 1 (the source is then said to be *predicted* or *inter-coded*). In the following, $\mathcal{I} \subset \{1, ..., N\}$ denotes the index set of the reference sources. To handle
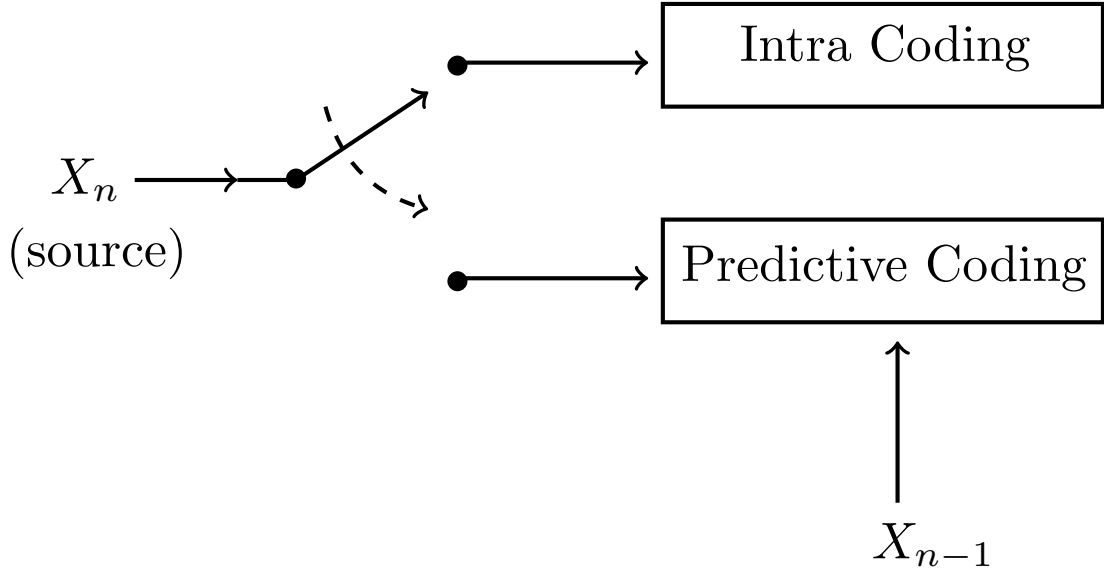
Fig. 1. Predictive coding scheme: each source is either intra coded or predicted.

the case of the first source X_1, we introduce the initialization source X_0, which models the memory state at the encoder when the encoding starts. For instance, X_0 can be the all zero source, or X_N. In the latter case, the encoding is said to be *cyclic*.

For this predictive coding scheme, and given the index set $\mathcal{I}$, we first derive the contribution of the source X_n to the global encoding cost. Encoding of the source X_n leads to a cost r_n (in bits) if the source is intra-coded and $\alpha_n r_n$ if it is predicted. Note that compression can be either lossless, or lossy. α_n corresponds to the reduction that can be achieved with the prediction of the source X_n. Therefore, given the index set of references $\mathcal{I}$, the global cost of encoding the source X_n is

$$cost(X_n|\mathcal{I}) = r_n \mathbb{1}_\mathcal{I}(n) + \alpha_n r_n (1 - \mathbb{1}_\mathcal{I}(n)) \ \forall \ n \in [1, N] \tag{1}$$

where $\mathbb{1}_\mathcal{I}$ is the characteristic function of the set $\mathcal{I}$. For example, if optimal lossless compression is performed, then the rates can be expressed as $r_n = H(X_n)$, and $\alpha_n r_n = H(X_n|X_{n-1})$, where $H(X_n)$ and $H(X_n|X_{n-1})$ denote the entropy and conditional entropy respectively. The case of lossy compression is studied in Sec. V.

Once the compression is performed, all sources are stored on a server. Then, *random access* to the sources is allowed in the sense that a client may request any set of sources. The coding and request principles are shown in Fig. 2.
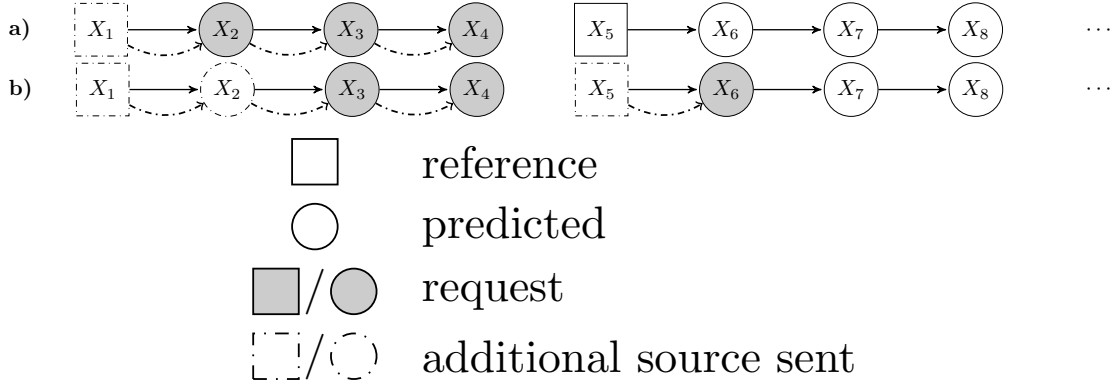
Fig. 2. Serving a request in the predictive coding scheme. Sources selected as a reference are depicted with a square. Predicted sources are depicted with a circle. An arrow from a square/circle to another circle means that the source depicted by the square/circle at the beginning of the arrow is used to predict the other source. A request to a set of sources is depicted as filled gray squares/circles. The set of sources which are effectively sent, includes the request but also additional sources (dotted square/circle) to allow decoding. The decoding process is shown with dotted arrows. Two examples of request are shown: either the requested sources (X_2, X_3, X_4) are included in a single group of source and only one reference needs to be sent (a), or the requested sources (X_3, X_4, X_6) are spread over two groups of source and two references need to be sent (b).

B. Problem description and challenges

The goal of the paper is to optimize the predictive coding scheme by minimizing a cost that depends on two criteria:

- the **storage cost** *i.e.* the averaged encoding cost of the data (1), averaged over all sources,
- the **transmission cost**: the amount of transmitted data per requested source, and averaged over all possible requests.

Classically, in the context of video coding [1], [9], [21], [23], [24], the two criteria are not differentiated. This is a consequence of the fact, that in classical 2D video coding, long consecutive sequences of sources (frames) are requested, such that both criteria are almost equal.

As for the optimization of the predictive coding scheme, the design parameter considered here is the placement of the references, denoted $\mathcal{I}$.

C. Problem formulation

Given a sequence of sources $(X_1, ..., X_N)$ and the predictive coding scheme detailed in Section II-A with source initialization X_0 and reference index set $\mathcal{I}$, the storage is completely determined by the cost

to encode each source (1)

$$S(\mathcal{I}) = \frac{1}{N} \sum_{n=1}^{N} cost(X_n|\mathcal{I}). \tag{2}$$

The users make requests to the database, where a request can be any subset $\mathcal{V} \subset \{1, \ldots, N\}$. Given the index set of the reference sources $\mathcal{I}$, and upon request of the sources with index $\mathcal{V}$, the server delivers a set of compressed sources with index $\mathcal{S}$ such that the requested sources can be recovered (see Fig. 2). More precisely, it is necessary that for each requested source, a reference source and all its successors are sent. Therefore, for a given requested set $\mathcal{V}$, the index set of the sent sources $\mathcal{S}$ must satisfy

$$\forall v \in \mathcal{V}, \exists i \in \mathcal{I} : \{i, i+1, \ldots, v\} \subset \mathcal{S} \tag{3}$$

which expresses the coding dependency chain. Then, the transmission cost corresponds to the cost to send all sources with index in $\mathcal{S}$, where this cost is averaged over the number of requested sources. Finally, the minimum transmission cost is obtained by optimizing over the set of sent sources and is given by (4)

$$R(\mathcal{V}|\mathcal{I}) = \min_{\mathcal{S} \subset \{1, \ldots, N\}} \frac{1}{|\mathcal{V}|} \sum_{n=1}^{N} r_n \big(\mathbb{1}_{\mathcal{I}}(n) + \alpha_n (1 - \mathbb{1}_{\mathcal{I}}(n)) \big) \mathbb{1}_{\mathcal{S}}(n) \tag{4a}$$

$$\text{subject to } \forall v \in \mathcal{V}, \exists i \in \mathcal{I} : \{i, i+1, \ldots, v\} \subset \mathcal{S} \tag{4b}$$

Although the requests from users are unknown in advance and are therefore random, they can be collected and classified into a limited number of typical requests characterized by some popularity distribution. More precisely, we suppose that there exists M typical index sets of requested sources, denoted by $\{\mathcal{V}_1, \ldots, \mathcal{V}_M\}$. This may result from an analysis of the video based on saliency analysis, or change of scene. The cardinality $|\mathcal{V}_m|$ is denoted by ℓ_m, for every $m \in [1, M]$. Vector $\mathbf{p} = (p_1, \ldots, p_M)$ represents the popularity distribution, where p_m stands for the probability that the set $\mathcal{V}_m$ has been requested. In the following, we assume that all sources might be requested with a non zero probability. In other words, we assume that $p_m > 0, \forall m \in \{1, \ldots, M\}$, and $\cup_{m \in \{1, \ldots, M\}} \mathcal{V}_m = \{1, \ldots, N\}$. This leads to the minimum averaged transmission cost (5)

The reference set selection problem consist in finding

$$\mathcal{I}^* \in \underset{\mathcal{I} \subset \{1, \ldots, N\}}{\operatorname{argmin}} \; S(\mathcal{I}) + \lambda R(\mathcal{I}) \tag{6}$$

where the storage $S(\mathcal{I})$ and the transmission $R(\mathcal{I})$ costs are defined in (2) and (5) respectively, $\lambda \in (0, +\infty)$ is a weighting parameter between the storage and the transmission rate. The difficulty of the

$$R(\mathcal{I}) = \sum_{m=1}^{M} \frac{p_m}{|\mathcal{V}_m|} \min_{\mathcal{S}_m \subset \{1,\ldots,N\}} \sum_{n=1}^{N} r_n \big(\mathbb{1}_{\mathcal{I}}(n) + \alpha_n(1 - \mathbb{1}_{\mathcal{I}}(n)) \big) \mathbb{1}_{\mathcal{S}_m}(n) \tag{5a}$$

$$\text{subject to } \forall m \in [1, M], \forall v \in \mathcal{V}_m, \exists i \in \mathcal{I} : \{i, i+1, \ldots, v\} \subset \mathcal{S}_m \tag{5b}$$

problem is twofold. First, a double optimization needs to be performed, one to determine the set of sent sources $\mathcal{S}_m$ for each request m, and the second one to determine the reference index set $\mathcal{I}$. This leads to a quadratic cost function that results from the multiplication of the functions $\mathbb{1}_{\mathcal{I}}$ and $\mathbb{1}_{\mathcal{S}_m}$. Second, the constraint (5b) introduces a complex dependence between the optimization variables: $\mathcal{S}_m, \forall m \in [1, M]$ and $\mathcal{I}$.

Remark: The formulation (6) is quite general. First, it includes the optimization of the weighted metric $\gamma \mathcal{S}(\mathcal{I}) + \mu \mathcal{R}(\mathcal{I})$. This weighted metric can for instance be the time to serve a request (in this case γ represents the time per bit to read a memory, and μ the time to send and receive a bit), or the energy cost to maintain a service, or the price to deploy such a service. Moreover, the solution to the problem (6) allows to solve the two following constrained problems:

$$\min_{\mathcal{I} \subset \{1,\ldots,N\}} S(\mathcal{I}) \text{ subject to } R(\mathcal{I}) \leq R_C$$

$$\min_{\mathcal{I} \subset \{1,\ldots,N\}} R(\mathcal{I}) \text{ subject to } S(\mathcal{I}) \leq S_C$$

where R_C and S_C are global cost constraints. Indeed, we will show in Sec. III that (6) is a linear programming problem. Therefore, all terms $(\mathcal{S}(\mathcal{I}), \mathcal{R}(\mathcal{I}))$ including the constraint (5b) are linear. Therefore, the derivations of Sec. III imply that the two constrained problems above are equivalent to integer linear programming problems.

III. AN EQUIVALENT INTEGER LINEAR PROGRAMMING PROBLEM

In this section, we show that the reference selection problem (6) can be cast into an integer linear programming problem. The first step consists in turning the quadratic cost function into a linear cost function by introducing the optimization vectors $\mathbf{y}, \mathbf{z}^{0,m}, \mathbf{z}^{1,m}$ such that, $\forall n \in [1, N]$

$$y_n = \mathbb{1}_{\mathcal{I}}(n), \tag{7a}$$

$$z_n^{0,m} = \mathbb{1}_{\mathcal{I}}(n) \mathbb{1}_{\mathcal{S}_m}(n), \forall m \in [1, M] \tag{7b}$$

$$z_n^{1,m} = (1 - \mathbb{1}_{\mathcal{I}}(n)) \mathbb{1}_{\mathcal{S}_m}(n), \forall m \in [1, M] \tag{7c}$$

where $y_n, z_n^{0,m}, z_n^{1,m}$ stand for the nth entry of the vectors $\mathbf{y}, \mathbf{z}^{0,m}, \mathbf{z}^{1,m}$. These vectors belong to the set $\{0,1\}^N$. Moreover, to be compatible with the definition of the characteristic functions $\mathbb{1}_{\mathcal{I}}$ and $\mathbb{1}_{\mathcal{S}_m}$, these vectors must satisfy the following constraints

$$z_n^{0,m} \le y_n, \quad \forall n \tag{8a}$$

$$z_n^{1,m} \le 1 - y_n, \quad \forall n \tag{8b}$$

where (8a) follows from (7a) and (7b) and (8b) follows from (7a) and (7c).

With this change of variable, the cost function in Problem (6) becomes (9)

$$\frac{1}{N} \sum_{n=1}^{N} r_n \left(1 - \alpha_n\right) y_n + \lambda \sum_{m=1}^{M} \frac{p_m}{|\mathcal{V}_m|}$$
$$\min_{(\mathbf{z}^{0,m}, \mathbf{z}^{1,m})} \sum_{n=1}^{N} r_n \left(z_n^{0,m} + \alpha_n z_n^{1,m}\right) \tag{9}$$

which is indeed linear, as well as the additional constraints (8).

We now turn to the constraint and show that the search space of the optimal index set $\mathcal{S}_m$ defined by the constraint (5b) can be reduced without modifying the value of the cost function at the optimum. More precisely, the sent sources (with index in $\mathcal{S}_m$) must contain the information necessary for decoding each requested source. This is the meaning of the constraint (5b). In particular, for each requested source of index $v \in \mathcal{V}_m$, a reference source with index $i \le v$ needs to be sent. However, it is sufficient to send the reference source that is closest to v. All other reference sources with index smaller will lead to an unnecessary extra rate. Therefore, the constraint (5b) can be rewritten as a new *decodability constraint* (10), without modifying the value of the cost function at its optimum

$$\forall m \in [1, M], \ \mathcal{S}_m^*(\mathcal{I}) = \bigcup_{v \in \mathcal{V}_m} \{i_v, i_v + 1, \ldots, v\} \tag{10a}$$

$$\text{where } i_v = \max_{j \in \mathcal{I}, j \le v} j \tag{10b}$$

We now show that the decodability constraint (10) can be expressed in terms of linear equations. The proof consists of several steps.

Step 1. (10)$\Rightarrow$(11) The decodability constraint (10) implies that there must at least one reference source. This can be written as

$$\exists n \text{ s.t. } \mathbb{1}_{\mathcal{I}}(n) = 1 \Leftrightarrow \sum_{n=1}^{N} y_n \ge 1 \tag{11}$$

Step 2. (10)$\Rightarrow$(12)$\Leftrightarrow$(13)$\Leftrightarrow$(14). We now show that the decodability constraint (10) induces a backward recursive construction of the set of sent sources. First, a source is sent if its index n

belongs to the request (12a). Then, if the source of index n is not requested, and if the source $n+1$ is not a reference, then the fact that the source of index $n + 1$ is sent, implies that the previous source of index n must also be sent (12b). This implies a backward recursion, and the recursion stops when a previous source is a reference.

$$\left(n \in \mathcal{V}_m\right) \Rightarrow \left(n \in \mathcal{S}_m\right) \tag{12a}$$

$$(n \notin \mathcal{V}_m) \wedge (n + 1 \in \mathcal{S}_m) \wedge (n + 1 \notin \mathcal{I}) \Rightarrow \left(n \in \mathcal{S}_m\right) \tag{12b}$$

This is equivalent to

$$\mathbb{1}_{\mathcal{S}_m}(n) = 1 \, \forall n \in \mathcal{V}_m \tag{13a}$$

$$\mathbb{1}_{\mathcal{S}_m}(n) = \mathbb{1}_{\mathcal{S}_m}(n + 1)\left(1 - \mathbb{1}_{\mathcal{I}}(n + 1)\right) \, \forall n \notin \mathcal{V}_m \tag{13b}$$

and, thanks to the change of variable (7), this can be rewritten as

$$z_n^{0,m} + z_n^{1,m} = 1 \quad \forall n \in \mathcal{V}_m \tag{14a}$$

$$z_n^{0,m} + z_n^{1,m} = z_{n+1}^{1,m} \quad \forall n \notin \mathcal{V}_m \tag{14b}$$

Step 3. (11)(12)⇒(10) Conversely, from (12a) or equivalently (13a), all requested sources belongs to the set of sent sources. If the source with index v is a reference, then from (13b), the previous and all other previous sources are not sent. If instead, the source with index v is not a reference, then from (13b), the previous and all other previous sources are sent. The recursion stops when a reference source has been found. Moreover, from (11), there exists at least one reference source. Therefore, $\forall v \in \mathcal{V}_m$, the index set $\{i, i + 1, \ldots, v\}$ (where $i \in \mathcal{I}$) belongs to the index set of sent sources $\mathcal{S}_m$. In other words, the sent sources (with index in $\mathcal{S}_m$) are sufficient to decode any requested source.

In conclusion, the new expressions of the cost function (9) and of the constraints (8), (11) and (14), lead to a new formulation of the overall reference selection problem as written in (15), where $\mathbf{z}^0 = \left(z_n^{0,m}\right)_{1 \le n \le N, 1 \le m \le M}$ and $\mathbf{z}^1 = \left(z_n^{1,m}\right)_{1 \le n \le N, 1 \le m \le M}$. We observe that this problem is indeed a linear integer programming problem, that can be solved by standard optimizer as [16]. In this paper, we employ the Mixed-Integer Linear Programming (MILP) and use the ILP Matlab toolbox[1].

IV. A CASE WHERE THE PERIODIC PLACEMENT OF THE REFERENCE SOURCES IS OPTIMAL

A classical approach in video coding [9], [21], [24] is to place references periodically [23]. In this section, we derive sufficient conditions under which this periodic placement is optimal. More precisely, we

---

[1]available at https://fr.mathworks.com/help/optim/ug/intlinprog.html

$$(\mathbf{y}^*, \mathbf{z}^{0,*}, \mathbf{z}^{1,*}) = \underset{(\mathbf{y}, \mathbf{z}^0, \mathbf{z}^1) \in \mathbb{R}^{(2M+1)N}}{\text{argmin}} \frac{1}{N} \sum_{n=1}^{N} r_n (1 - \alpha_n) y_n + \lambda \sum_{m=1}^{M} \beta_m \sum_{n=1}^{N} r_n \left(z_n^{0,m} + \alpha_n z_n^{1,m} \right)$$

$$\text{s.t.} \begin{cases} (y_n, z_n^{0,m}, z_n^{1,m}) \in \{0,1\}^3 & \forall (n,m) \in [1,N] \times [1,M] \\[2mm] \sum_{n=1}^{N} y_n \geq 1 & \\[2mm] -y_n + z_n^{0,m} \leq 0 & \forall (n,m) \in [1,N] \times [1,M] \\[2mm] y_n + z_n^{1,m} \leq 1 & \forall (n,m) \in [1,N] \times [1,M] \\[2mm] z_n^{0,m} + z_n^{1,m} = 1 & \forall (n,m) \in [1,N] \times [1,M] \ \text{and} \ n \in \mathcal{V}_m, \\[2mm] z_n^{0,m} + z_n^{1,m} - z_{n+1}^{1,m} = 0 & \forall (n,m) \in [1,N-1] \times [1,M] \ \text{and} \ n \notin \mathcal{V}_m \end{cases} \tag{15}$$

study the reference selection problem (6) under the hypothesis that all sources have the same distribution and that consecutive requests are done. In particular, we show that periodic placement of the reference sources is optimal, and we derive a closed form expression for the optimal period. More formally, the hypotheses considered here are:

Assumption 1. *The encoding cost of all the sources in* (1) *are constants* i.e.

$$\forall n \in [1,N], \ \alpha_n = \alpha, r_n = r,$$

Assumption 2. *The family of request sets consists of all possible sets of ℓ consecutive indices, each request has the same probability,* i.e.

$$\forall m \in [1,M], \ \ell_m = \ell, \ and \ p_m = \frac{1}{M}.$$

Assumption 3. *The weighting parameter λ between the storage and the transmission rate equals 1.*

Proposition 4. *Consider a set of N sources ($N \to +\infty$) that satisfy the Assumption 1. Further consider that the requests satisfy Assumption 2, and that the cost function satisfies Assumption 3. The optimal positioning of the reference sources that minimizes the optimization problem* (6) *is periodic.*

Proof. The proof for this Proposition can be found in Appendix A. □

In the previous proposition, the reason for the asymptotic study ($N \to \infty$) is to neglect the boundary effect of the last group of sources.

Proposition 5. *Consider a set of N sources ($N \to +\infty$) that satisfy the Assumption 1. Further consider that the requests satisfy Assumption 2, and that the cost function satisfies Assumption 3. The reference index set is periodic of period k. The storage (2) and the transmission cost (5) only depend on this period and are given by*

$$S(k) = \frac{r}{k}[(k-1)\alpha + 1], \tag{16}$$

$$R(k) = \frac{r}{k\ell}\left(k + \ell - 1 + \frac{\alpha}{2}(k-1)(k+2\ell-2)\right) \tag{17}$$

Proof. The proof for (16) directly derives from the definition of the storage (2). The proof for the transmission cost is more involved as it requires to solve an optimization problem. It is given in Appendix B □

Theorem 6. *Consider a set of N sources ($N \to +\infty$) that satisfy the Assumption 1. Further consider that the requests satisfy Assumption 2, and that the cost function satisfies Assumption 3. The optimal positioning of the reference sources that minimizes problem (6) is periodic of period k^**

$$k^* = \underset{k \in \left\{\lfloor \bar{k} \rfloor, \lceil \bar{k} \rceil\right\}}{argmin} F(k) \tag{18}$$

where

$$\bar{k} = \sqrt{\frac{2(1-\alpha)(2\ell-1)}{\alpha}}.$$

Proof. k^* is the value that minimizes the function $F(k) = S(k) + R(k)$. Details of the proof can be found in Appendix C. □

V. Optimal reference placement in the case of lossy compression

A. Equivalent integer linear programing problem with increased search space dimension

In this section, we extend the reference selection problem to the case of lossy compression. We assume that the quantization step size is fixed for all the sources, and, as in the lossless case, we only optimize the reference placement (no rate-distortion allocation). The difference between the two cases lies in the fact that in the lossy case, the rate needed to transmit a source can depend on the distance between this source and its previous reference. Therefore, in this part, we also assume that the distance between two consecutive references is not greater than T. Under these assumptions, we express the cost function in the lossy case, and show that minimizing this cost function is, as before, equivalent to solving a linear programming problem.

For a sequence of sources $(X_1, \ldots, X_N)$, the global cost for the lossy encoding of the source X_n satisfies, for every $n \in [1, N]$,

$$cost(X_n | \mathcal{I}) = r_n \mathbb{1}_{\mathcal{I}}(n) + \alpha_{n,t^*} r_n (1 - \mathbb{1}_{\mathcal{I}}(n)), \tag{19a}$$

$$\text{where} \quad t^* = \arg \max_{t \in \mathcal{I} \cap \mathcal{T}_n} t, \tag{19b}$$

and where $\mathcal{T}_n = [\max\{1, n - T + 1\}, \max\{1, n - 1\}]$ for $n \in [1, N]$. In (19), r_n stands for the rate to encode the reference source X_n, and $\alpha_{n,t} r_n$ is the rate to encode the source X_n from X_{n-1}, given that the source X_t is a reference source (with $t \in \mathcal{T}_n$). The difficulty in (19b) is that the optimization is performed over indices, whereas, in the global optimization problem, the cost function (19a) is homogeneous to a rate. Therefore, we rewrite (19) into (20) assuming that the rate needed to compress a source increases with the distance to the reference.

$$cost(X_n | \mathcal{I}) = r_n \mathbb{1}_{\mathcal{I}}(n) + \left(\min_{t \in \mathcal{I} \cap \mathcal{T}_n} \alpha_{n,t} \right) r_n (1 - \mathbb{1}_{\mathcal{I}}(n)). \tag{20}$$

$$cost^*(X_n | \mathcal{I}) = r_n \mathbb{1}_{\mathcal{I}}(n) + \left(\min_{\mathcal{J}_n \subset \mathcal{I}} \sum_{t \in \mathcal{T}_n} \alpha_{n,t} \mathbb{1}_{\mathcal{J}_n}(t) \right) r_n (1 - \mathbb{1}_{\mathcal{I}}(n)), \tag{21a}$$

$$\text{s.t.} \quad \mathbb{1}_{\mathcal{I}}(n) + \sum_{t \in \mathcal{T}_n} \mathbb{1}_{\mathcal{J}_n}(t) (1 - \mathbb{1}_{\mathcal{I}}(n)) = 1. \tag{21b}$$

Now, (20) can be equivalently reformulated as in (21). Indeed, if $n \in \mathcal{I}$, both cost functions are equal, and if $n \in [1, N] \setminus \mathcal{I}$, we have

$$cost^*(X_n | \mathcal{I}) = \left(\min_{\mathcal{J}_n \subset \mathcal{I}} \sum_{t \in \mathcal{T}_n} \alpha_{n,t} \mathbb{1}_{\mathcal{J}_n}(t) \right) r_n \tag{22a}$$

$$\text{s.t.} \quad \sum_{t \in \mathcal{T}_n} \mathbb{1}_{\mathcal{J}_n}(t) = 1. \tag{22b}$$

We now formulate the optimization problem in the lossy case. As for the lossless case, the storage cost is given by

$$S(\mathcal{I}) = \frac{1}{N} \sum_{n=1}^{N} cost^*(X_n | \mathcal{I}) \tag{23}$$

and the minimum averaged transmission cost required to deliver the M sets of requested sources $\{\mathcal{V}_1, \ldots, \mathcal{V}_M\}$ is obtained by

$$R(\mathcal{I}) = \sum_{m=1}^{M} \frac{p_m}{|\mathcal{V}_m|} \min_{\mathcal{S}_m \subset \{1,\ldots,N\}} \sum_{n=1}^{N} cost^*(X_n|\mathcal{I}) \mathbb{1}_{\mathcal{S}_m}(n) \tag{24a}$$

$$\text{subject to } \forall m \in [1, M], \forall v \in \mathcal{V}_m, \exists i \in \mathcal{I} : \{i, i+1, \ldots, v\} \subset \mathcal{S}_m \tag{24b}$$

Therefore, in this case, the reference set selection problem consists of finding an index set $\mathcal{I}^*$ such that

$$\mathcal{I}^* \in \operatorname*{argmin}_{\mathcal{I} \subset \{1,\ldots,N\}} S(\mathcal{I}) + \lambda R(\mathcal{I}) \tag{25}$$

The following proposition shows that, as in the lossless case, finding a solution for (25) is equivalent to solving a linear programming problem.

Proposition 7. *The reference selection problem* (25) *can be cast into an integer linear programming problem.*

Proof. See Appendix D. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\square$

The linear programming problem stated in Section III for the lossless case involves $N(1 + 2M)$ binary variables to optimize. In the lossy case, the linear programming problem of Proposition 7 involves $(N + \sum_{n=1}^{N} |\mathcal{T}_n|)(1 + M)$ binary variables. In the lossy case, the number of variables is thus increased by a factor approaching T, which can be large. In the next section, we consider a source model for which the optimization problem for the lossy case can be simplified into the optimization problem for the lossless case.

B. Reduction of the search space dimension

In the previous section, we showed that the reference selection problem remains an integer linear programming problem in the case of lossy compression. However, the main difference with the lossless case, is that the dimension of the search space can be significantly increased. A way to reduce the dimension of the optimization problem is to break the dependency chain thanks to some assumptions, as it is done in other applications such as bit allocation in a video stream [14], [26] or packet scheduling on a lossy channel [6].

In the following, we show an example of drastic simplification, such that (1) still holds, *i.e.*, the cost function depends on two instead of T parameters as in (20). Consider the first order autoregressive model

$$X_n = AX_{n-1} + Z_n, \tag{26}$$

where X_n models the source, Z_n is an innovation process, A is a fixed $d \times d$ matrix, and all vectors are of dimension d. For ease of presentation, we assume that all processes are centered. Now, the compression scheme consists in first building a linear prediction $\hat{X}_n$, with the $d \times d$ matrix H, then computing a residue E_n. Each component of E_n is scalar quantized and entropy coded. The coded data are sent to the decoder. Finally, the decoder computes the quantized residue $\tilde{E}_n$, and reconstructs the data $\tilde{X}_n$ from the prediction and the quantized residue:

$$\hat{X}_n = H\tilde{X}_{n-1} \tag{27a}$$

$$E_n = X_n - \hat{X}_n \tag{27b}$$

$$\tilde{E}_n = E_n + \Delta_n \tag{27c}$$

$$\tilde{X}_n = \hat{X}_n + \tilde{E}_n. \tag{27d}$$

One has

$$\tilde{X}_n = X_n - E_n + \tilde{E}_n = X_n + \Delta_n. \tag{28}$$

Assume that each component $E_{n,i}$ of the residue E_n is entropy coded, without exploiting the temporal dependencies (along the index n). The rate $R_{n,i}$ required to represent $E_{n,i}$ with a distortion $D_{n,i} = \mathbb{E}[\Delta_{n,i}^2]$ is

$$R_{n,i} = \frac{1}{2} \log \left(\frac{c_{n,i}}{D_{n,i}} \mathbb{E}[E_{n,i}^2] \right), \tag{29}$$

where $c_{n,i}$ is a constant which depends on the quantizer type, and the distribution of the residue. The rate-distortion characterization (29)for a wide class of distributions [22, Sec. 4], but also for all distributions in the high resolution regime [22, Sec. 5.2.3 and Sec. 5.2.6], provided that we apply a scalar quantization with a Variable-Length Code that does not exploit dependencies.

The residue satisfies (30),

$$E_n = Z_n + AX_{n-1} - HX_{n-1} - H\Delta_{n-1} \tag{30a}$$

$$E_n = Z_n - A\Delta_{n-1} \tag{30b}$$

$$\mathbb{E}[E_n E_n^T] = \mathbb{E}[Z_n Z_n^T] - 2\mathbb{E}[Z_n \Delta_{n-1}^T A^T] + A\mathbb{E}[\Delta_{n-1}\Delta_{n-1}^T]A^T \tag{30c}$$

$$= \mathbb{E}[Z_n Z_n^T] + A\mathbb{E}[\Delta_{n-1}\Delta_{n-1}^T]A^T \tag{30d}$$

where (30b) uses the fact that we apply the optimal linear predictor, which minimizes the residue energy

$\mathbb{E}[E_n^T E_n]$, *i.e.*, $H = A$ [17, Chap. 11] [22, Sec. 6], and where (30d) follows from the fact that Z_n is an i.i.d. process (this implies that Z_n is independent of the past quantization error Δ_{n-1}). Therefore, the variance of the ith component of the residue, $\mathbb{E}[E_{n,i}^2]$, only depends on the characteristics of the source (A and $\mathbb{E}[Z_n Z_n^T]$) and on the quantization error statistics $\mathbb{E}[\Delta_{n-1}\Delta_{n-1}^T]$. In other words, given the source model, $\mathbb{E}[E_{n,i}^2]$ only depends on the quantization applied at time $n - 1$, but does not depend on the position of the previous reference, which we summarize as $\mathbb{E}[E_{n,i}^2] = f(D_{n-1})$. Finally, the cost to encode the source X_n, is

$$\sum_{i=1}^{d} R_{n,i} = \frac{1}{2} \sum_{i=1}^{d} \log \left(\frac{c_{n,i}}{D_{n,i}} f(D_{n-1}) \right). \tag{31}$$

To summarize, if the sources follow a first order autoregressive model, if the quantization steps are fixed for all the sources, if the components of the prediction residual are scalar quantized and entropy coded without exploiting the dependencies, then the rate to encode a source does not depend on the position t of the reference, *i.e.*,

$$\alpha_{n,t} = \alpha_n, \forall t \tag{32}$$

and the general cost function (21) boils down to the cost function derived in the lossless case (1).

VI. EXPERIMENTAL VALIDATION

In this section, we aim at validating the performance of the proposed method for either constant or variable source encoding cost α_n in (1). More precisely, the α_n for every $n \in [1, N]$ are chosen in $]0, 1[$ or obtained from real measurements (video, meteorological data, occupancy measure of self service terminal). Each of these experiments is considered with or without taking into account the popularity of the request p_m, ($m \in [1, M]$). Then, for each choice in this input parameter set, different request lengths are considered. The weighting parameter λ is set to 1. In the Fig. 4-8, the labels (ix) in each subfigure indicate:

- i: the test condition
- x: the set of parameters used for this test condition (specified in the Figure's caption) or the quantity that is plotted for this test condition (specified in the Figure's caption)

The four test conditions are:

(1) all requests have the same popularity and the sources have the same encoding cost

(2) all requests have different popularities and all the sources have the same encoding cost

(3) all requests have the same popularity and the sources have different encoding costs

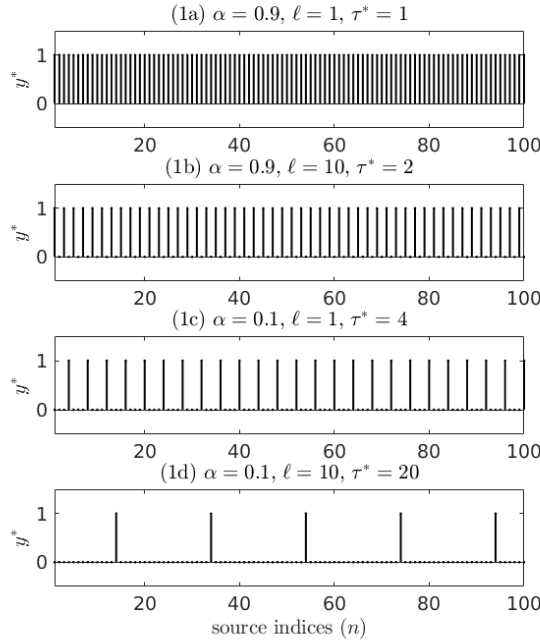(4) all requests have different popularities and the sources have different encoding costs.

Fig. 3. Optimal placement of the reference for constant predictive coding cost α and constant popularity p (test condition 1): y^* solution to Problem (15), where 1 stands for a reference source. Test for four different cases of α and size of request ℓ. (a) $\alpha = 0.9$ and $\ell = 1$; (b) $\alpha = 0.1$ and $\ell = 10$; (c) $\alpha = 0.1$ and $\ell = 1$; (d) $\alpha = 0.9$ and $\ell = 10$.

A. Results on the synthetic dataset

In all the simulations on the synthetic dataset, r_n, the intra coding cost of each source is considered constant.

$\diamond$ *Test condition (1): all requests have the same popularity and the sources have the same encoding cost.* More precisely $\alpha_n = \alpha, \forall n \in [1, N]$ and $p_m = 1/M, m \in [1, M]$. Moreover, N, the total number of sources is set to 100, and cyclic encoding is performed.

Fig. 3 shows the optimal reference placement $\mathbf{y}^*$ obtained by solving the integer linear programming problem (15). First, we observe that, even if the assumption of Proposition 4 is not satisfied (N is finite), the optimal positioning is periodic. We denote τ^* this period. Second, the more the sources are correlated (small α), the larger the period of the Group Of Sources (GOS), as intuition suggests.

Tab I reports the optimal period k^* derived by Theorem 6 *i.e.* obtained in the asymptotic case $N \to \infty$ to neglect the effect of the last GOS. We note that, for all tested cases ($\alpha \in \{0.1, 0.9\}$ and $\ell \in \{1, 10\}$), both optimization solutions (closed form under asymptotic assumption k^* and solution to the optimal τ^*) lead to the same result except when $\ell = 10, \alpha = 0.1$, where $\tau^* = 20$ and $k^* = 19$. This difference

	Request size ℓ									
	1	2	3	4	5	6	7	8	9	10
0.1	4	7	10	11	13	14	15	16	18	19
0.2	3	5	6	8	9	9	10	11	12	12
0.3	2	4	5	6	7	7	8	8	9	9
0.4	2	3	4	5	5	6	6	7	7	8
0.5	2	3	3	4	4	5	5	6	6	6
0.6	1	2	3	3	4	4	4	5	5	5
0.7	1	2	2	3	3	3	3	4	4	4
0.8	1	1	2	2	2	2	3	3	3	3
0.9	1	1	1	1	2	2	2	2	2	2

(Rows labeled by Parameter α.)

TABLE I

OPTIMAL REFERENCE PERIOD k^*, SOLUTION DERIVED IN THEOREM 6 FOR A DATASET OF $N = 100$ SOURCES, $\alpha \in \{0.1, 0.2, \ldots, 0.9\}$, AND $\ell = \{1, 2, \ldots, 10\}$.

occurs because the number of GOSs (k^*/N) is very small (< 5) such that the effect of an incomplete GOS does affect the solution. In other words, neglecting the effect of the incomplete last GOS does not affect the placement of the references except when the number of GOSs is small.
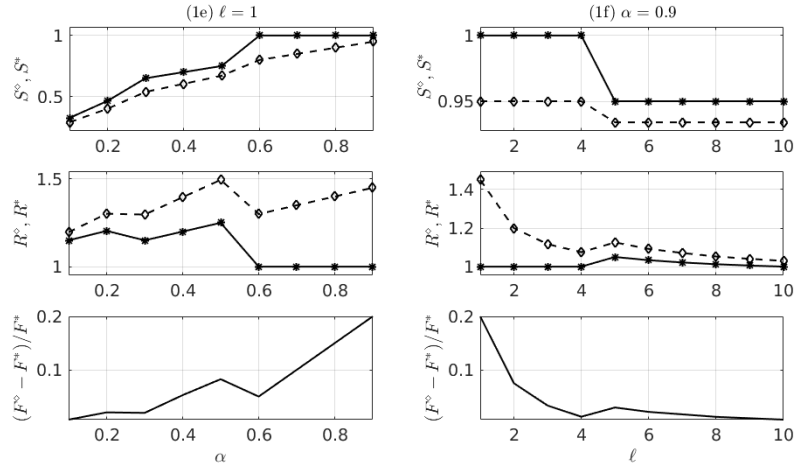

Fig. 4. Comparison of the storage (S), transmission (R) and sum costs ($F = S + R$) obtained either with the optimal reference placement $\mathbf{y}^*$ (line with star) or when the period has been overestimated by only 1 *i.e.* $\mathbf{y}^\diamond$ (dotted line with diamond). (1e): when $\ell = 1$ and $\alpha \in \{0.1, \ldots, 0.9\}$. (1f): when $\alpha = 0.9$ and $\ell \in \{1, \ldots, 10\}$. From top to bottom: storage costs, transmission costs and the ratio $(F^\diamond - F^*)/F^*$ where $F^\diamond$ and F^* correspond to the function values at $\mathbf{y}^*$ and $\mathbf{y}^\diamond$, respectively.

Fig. 4 illustrates the importance to perform the reference placement optimization (15). Indeed, the

storage costs S, (top figures) and the transmission costs R (middle figures) are compared when the period has been overestimated by only 1 with respect to the optimal one k^*. More precisely, we compare these costs for k^* and for $k^\diamond = k^* + 1$. Then, the excess rate of the sum cost $F = S + R$ is computed and the proportional excess rate $((F^\diamond - F^*)/F^*)$ is shown (bottom figures in Fig. 4). Interestingly, an overestimation of the period by only 1 may lead to a significant increase of the sum rate F by up to 20%.
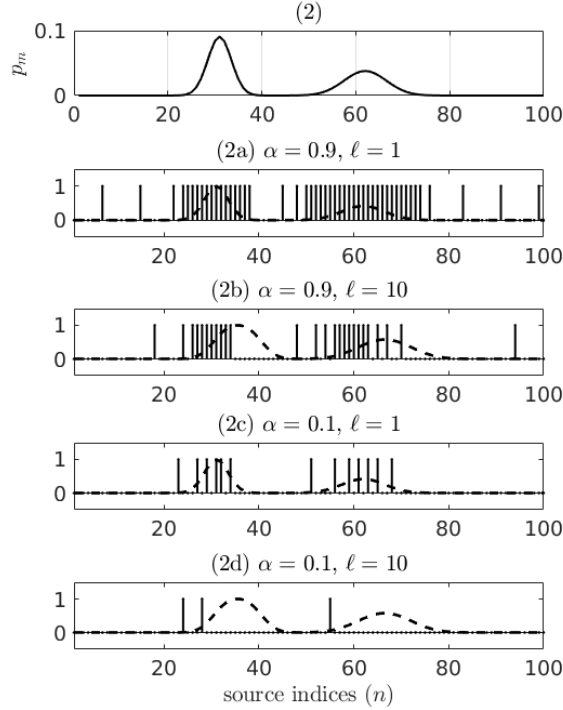

Fig. 5. Optimal placement of the reference for constant predictive coding cost α and variable popularity shown in the top figure (test condition 2): y^* solution to Problem (15), where 1 stands for a reference source. Test for four different cases of α and size of request ℓ: (2a) $\alpha = 0.9$ and $\ell = 1$; (2b) $\alpha = 0.1$ and $\ell = 10$; (2c) $\alpha = 0.1$ and $\ell = 1$; (2d) $\alpha = 0.9$ and $\ell = 10$.

$\diamond$ *Test condition (2): all requests have different popularities and all the sources have the same encoding cost.* More precisely $\alpha_n = \alpha\,(n \in [1, N])$ and the probability of popularity of request V_m is p_m for every $m \in [1, M]$. This popularity probability distribution has be chosen to mimic the case, where a video contains two popular instants, and users access the video around these key events for variable time durations. If every source in the request set has the same probability, we can then compute the probability of request for each source, denoted $(\pi_n)_{1 \leq n \leq N}$, which defines the vector π and is given by the following
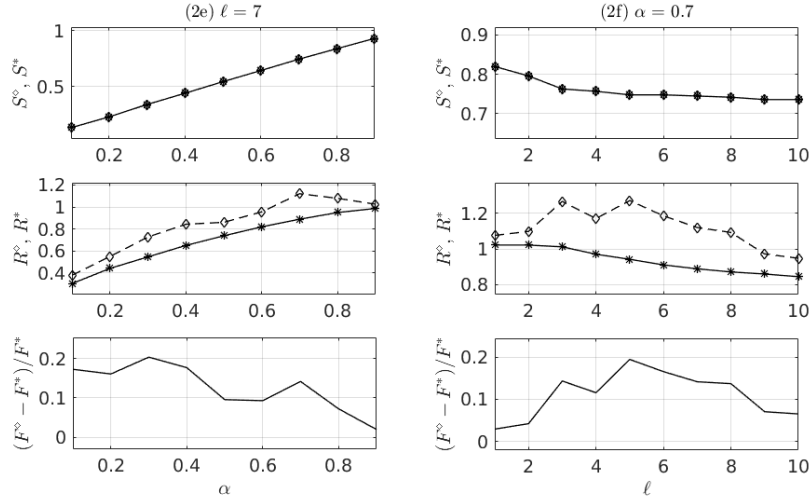
Fig. 6. Comparison of the storage (S), transmission (R) and sum costs ($F = S + R$) obtained either with the optimal reference placement $\mathbf{y}^*$ (line with star) or when the references correspond to the most popular sources *i.e.* $\mathbf{y}^\diamond$ (dotted line with diamond). (2e): when $\ell = 7$ and $\alpha \in \{0.1, \ldots, 0.9\}$. (2f): when $\alpha = 0.7$ and $\ell \in \{1, \ldots, 10\}$. From top to bottom: storage rates, transmission rates and the ratio $(F^\diamond - F^*)/F^*$ where $F^\diamond$ and F^* correspond to the function values at $\mathbf{y}^*$ and $\mathbf{y}^\diamond$, respectively.

formula:

$$\forall n \in [1, N], \ \pi_n = \sum_{\{m \in [1,M] \,:\, v_n \in V_m\}} \frac{p_m}{\ell_m} \tag{33}$$

Moreover, N, the total number of sources is set to 100, and cyclic encoding is performed.

Fig. 5 shows the popularity of each source (top figure) and the resulting optimal reference placement. Interestingly, references correspond to popular sources as intuition suggests. But this is not the only criterion. Indeed, references are also rather positioned at the beginning of a burst of popular sources. Fig. 6 illustrates the importance to perform the reference placement optimization (15). A naive reference placement is performed, where the references correspond to the most popular sources. Then, this naive placement $\mathbf{y}^\diamond$ is compared to the optimal one $\mathbf{y}^\star$ under the hypothesis that both strategies have the same number of references.

Fig. 6 compares the costs (Storage S, Transmission R, and the sum cost $F = S + R$ as proposed in [3]) for both strategies (optimal vs most popular selection). In this experiment, we choose to show two results of (2e) fixed $\ell = 0.7$ with $\alpha \in \{0.1, 0.2, \ldots, 0.9\}$, and (2f) fixed $\alpha = 0.7$ with $\ell \in \{1, 2, \ldots, 10\}$. In both cases, the storage costs of both $\mathbf{y}^*$ and $\mathbf{y}^\diamond$ are equal, however again there are significant differences between the transmission costs (R^* and $R^\diamond$). Finally, the naive placement strategy leads to a significant increase of the sum rate F by up to 20%.
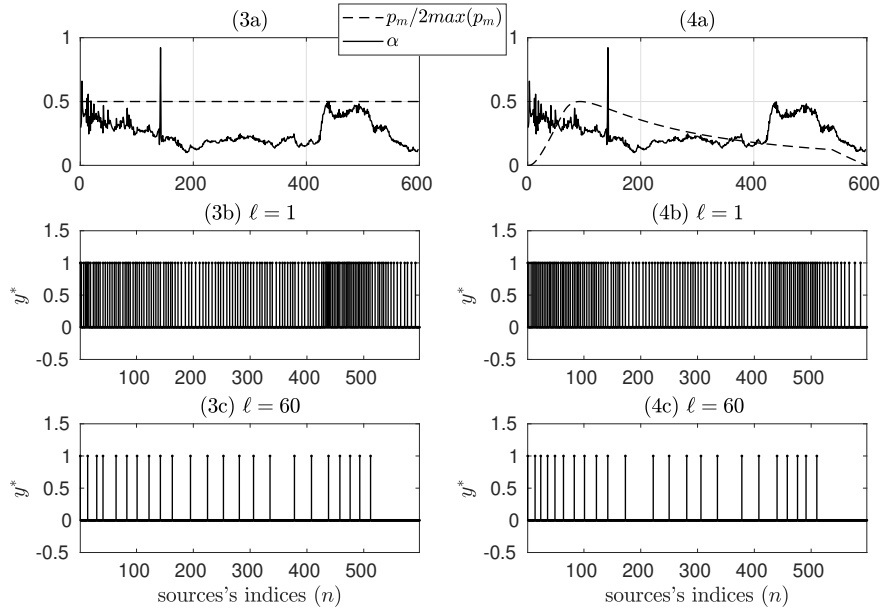
Fig. 7. Optimal placement of the reference sources for variable predictive coding cost α_n (obtained from *RitualDance* sequence) and fixed (test condition 3) or variable (test condition 4) popularity. In the top figures, the dashed curve represents the popularity whereas the solid line curve represents the predictive encoding cost (α_n). The bottom figures plot the solution $\mathbf{y}^*$ to Problem (15), where 1 stands for a reference source. Each test condition are tested for two different cases of size of request ℓ: (3b) and (4a) $\ell = 1$; (3c) and (4c) $\ell = 60$.

B. Real dataset: video

We now test our optimal placement method on the real datasets. Here, we consider seven CTC sequences which are presented in Tab. II and available at ftp://ftp.tnt.uni-hannover.de. We use the reference software of the Versatile Video Coding (VVC) [5] scheme, known as VVC Test Model (VTM) version 6.2, to estimate the encoding costs and therefore the parameters $(r_n, \alpha_n)_{n \in [1,N]}$. For each video frame of time index n, r_n corresponds to the intra coding cost, and α_n is the ratio between the predictive coding cost and the intra coding cost. The resulting α_n are shown in the top figures of Fig 7 (black line).

⋄ *Test conditions: variable source encoding costs $\alpha_n, n \in [1, N]$ and equal (3) or variable request popularities (4) $p_m, m \in [1, M]$.* The labels (3) and (4) correspond to the label used in Fig. 7 and 8. The two types of popularity are shown as dashed lines in the top figures of Fig. 7: constant request popularity in the left column and variable request popularity in the right column. In [4], the authors show that the popularity in terms of the number of viewers who watched a segment in videos exhibit a log-normal distribution. Therefore, we use a log-normal distribution with parameters $\mu = 0.016$ and $\sigma = 1.35$ to generate the request popularity for the case (4). Here, we observe that most of the references are placed
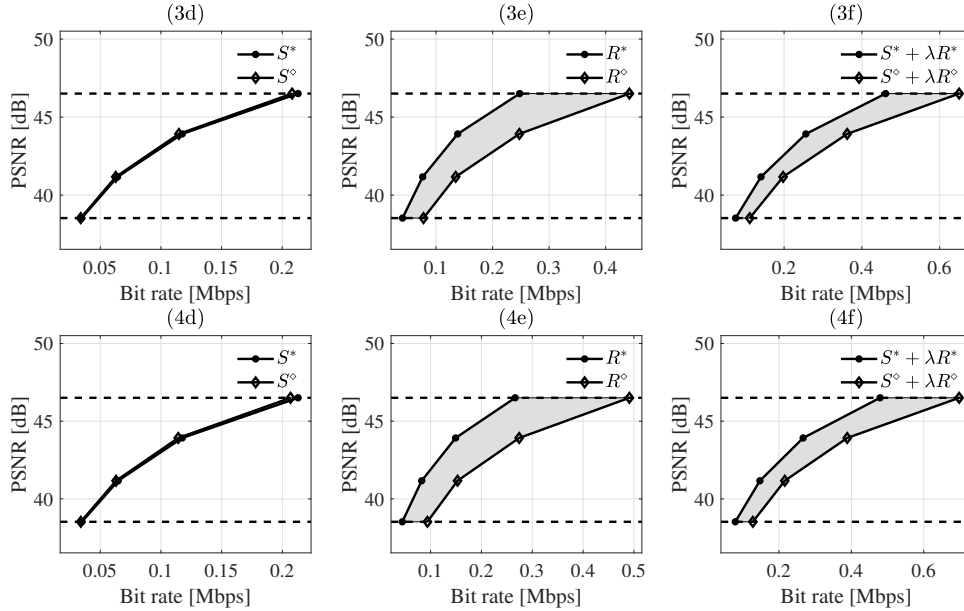
Fig. 8. Comparison of the storage (S), transmission (R) and sum costs ($F = S + \lambda R$) obtained either with the optimal reference placement $\mathbf{y}^*$ (line with star) or with the naive solution i.e. $\mathbf{y}^\diamond$ (line with diamond). Tested on the *RitualDance* sequence over four different values of QP $\in \{22, 27, 32, 37\}$ and with the request length $\ell = 60$. The request popularity is either uniform (top, labeled (3)) or varies (bottom, labeled (4)). From left to right: storage (d), transmission cost (e) and the sum cost (f), respectively.

where α_n is high. This is because, when α_n is close to 1, the costs to encode a source as a reference or as a predicted source are almost the same. Therefore, the source X_n can be encoded as a reference without significantly increasing the total storage and transmission costs. In the right column of Fig. 7, variable popularity is added. Then, many references are concentrated at positions where sources are very popular between 30 and 160.

These observations suggest a naive reference placement strategy $\mathbf{y}^\diamond$, where the references correspond to the higher values of α_n. In Fig. 8, this naive placement $\mathbf{y}^\diamond$ is compared to the optimal one $\mathbf{y}^\star$. However, for the sake of fairness, the number of references kept in the naive approach is optimized in order to take into account both storage and transmission costs. More precisely, the average conditional encoding cost is computed $\bar{\alpha} = \sum_{i=1}^{N} \alpha_n$, and the placement optimization for this average cost is performed (Theorem 6). From this solution, a number of references can be computed, and is used as the number of references to keep in the naive approach. In Fig. 8, we show the results obtained for the *RitualDance* sequence, for four different values of QP $\in \{22, 17, 32, 37\}$ and for the requested length $\ell = 60$. We compute the costs (Storage S, Transmission R, and the sum cost $F = S + \lambda R$ as proposed in [3]) for both strategies (optimal

Class	Sequence name	Requested length $\ell = 60$			Requested length $\ell = 90$		
		BD-rate (S)	BD-rate (R)	BD-rate (F)	BD-rate (S)	BD-rate (R)	BD-rate (F)
B	RitualDance	2.379	-44.226	-29.502	1.973	-36.951	-23.128
D	BasketballPass	1.556	-45.468	-30.853	1.101	-39.942	-26.088
	BlowingBubbles	5.051	-48.988	-34.495	3.482	-37.274	-23.881
	BQSquare	-1.229	-37.800	-27.395	-0.733	-28.861	-19.774
	Flowervase	7.420	-7.005	-1.816	0.622	-0.805	-0.187
	Keiba	0.104	-34.909	-22.973	0.164	-22.072	-13.238
	Mobisode2	4.016	-29.537	-17.798	3.897	-23.195	-12.431
	RaceHorses	1.803	-41.929	-28.513	1.541	-35.248	-22.505
Average class D		2.675	-35.091	-23.406	1.439	-26.771	-16.872
Average		2.638	-36.233	-24.168	1.506	-28.043	-17.654

TABLE II

COMPARISON OF THE BD RATE OF THE STORAGE (S), TRANSMISSION (R) AND SUM COSTS ($F = S + R$) OBTAINED EITHER WITH THE OPTIMAL REFERENCE PLACEMENT $\mathbf{y}^*$ WITH THE NAIVE SOLUTION $\mathbf{y}^\diamond$. THESE RESULTS PRESENTED IN THIS TABLE, FOR EACH SEQUENCE ARE AVERAGED OVER FOUR DIFFERENT VALUES OF QP $\in \{22, 27, 32, 37\}$.

and naive). The naive placement strategy leads to a significant increase of the sum rate F between 0.036 Mbps and 0.189 Mbps for uniform request popularity (test condition 3), and between 0.049 Mbps and 0.219 Mbps for variable request popularity (test condition 4).

We conduct the experiments described above for different sequences from the MPEG common test conditions. We evaluate the rate-distortion performance for the two schemes (naive and proposed) and compare them by computing the Bjontegaard Delta (BD-rate) bit rate saving as classically done in video compression. For each sequence, the performance is averaged over four QP values $\in \{22, 27, 32, 37\}$. Results are reported in Tab. II for two request lengths $\ell \in \{60, 90\}$. We see that our approach leads to impressive gains of about -17% and -24% on average for the global F.

We also compare our approach to a more sophisticated technique [12] that consists in optimizing the partitioning of the video set. The difference between partitioning and reference frame positioning as proposed in this paper is that partitioning cut the video into several segments of optimized size, and at the transmission, it is considered that *the whole segment* is transmitted. Note that these segments are generally coded with a PC scheme as considered here, which implies that if there are not useful, the transmission of the last frames of a segment could be avoided. To evaluate the benefits of taking into account this *partial* transmission in the optimization as we propose in this paper, we optimize the reference frame positioning using [12] and our method and we compare the rate-distortion performance.

Class	Sequence name	Requested length $\ell = 60$			Requested length $\ell = 90$		
		BD-rate (S)	BD-rate (R)	BD-rate (F)	BD-rate (S)	BD-rate (R)	BD-rate (F)
B	RitualDance	-4.439	0.919	-1.612	-3.749	0.159	-1.721
	BasketballPass	-3.967	0.302	-1.694	-3.558	-0.073	-1.719
	BlowingBubbles	-7.901	-0.179	-3.648	-7.818	-0.637	-3.959
	BQSquare	-6.964	0.814	-2.235	-6.136	0.415	-2.259
D	Flowervase	-15.636	-2.963	-8.36	-19.250	-8.098	-13.210
	Keiba	-4.701	-0.048	-2.167	-3.963	-1.037	-2.403
	Mobisode2	-3.819	-1.319	-2.445	-4.286	-1.389	-2.777
	RaceHorses	-4.148	-0.291	-2.033	-3.960	-1.078	-2.412
Average class D		-6.734	-0.526	-3.226	-7.000	-1.700	-4.106
Average		-6.447	-0.346	-3.024	-6.590	-1.467	-3.808

TABLE III

COMPARISON BETWEEN PROPOSED METHOD AND THE PROPOSED METHOD WITH ADDITIONAL CONSTRAINTS IN [12], CALLED [12]$^\diamond$. IN THIS TABLE, THE TRANSMISSION COSTS FOR [12]$^\diamond$ RESULTS ARE RECOMPUTED IN ORDER TO SATISFY ALL THE REQUESTS. THESE RESULTS PRESENTED IN THIS TABLE, FOR EACH SEQUENCE ARE AVERAGED OVER FOUR DIFFERENT VALUES OF QP $\in \{22, 27, 32, 37\}$.

Results are shown in Tab III. We see that the advantage of our method is still significant (around -3% in average for F), which demonstrates that reference frame positioning formulation as we propose in this paper is more accurate than a formulation, which does not take into account the possible partial transmission of a segment.

C. Real dataset: time series measurement

In this experiment, we consider lossless compression of two time series measurements. The first one corresponds to temperature data measured on the European territory by the network MESONET[2]. It consists in 341 temperature sensors recording temperature every 3 hours during 4 months *i.e* from June to September (976 frames of 341 temperature values in the METAR format). The second corresponds to the occupancy measure of a self service bike terminal (Velib) in Paris. It consists in an occupancy value measured every 20 minutes during 10 days on 1188 terminals. In both experiments, the vector X_n models the measurements acquired simultaneously at time n. From the database, we evaluate α_n as the correlation between X_n and X_{n-1}, and we observed that the r_n are constant over time. The request

[2]https://mesonet.agron.iastate.edu

Bit rate Request length	$S^\diamond$	S^*	$R^\diamond$	R^*	$F^\diamond$	F^*
$\ell = 7$	**0.613**	0.622	0.970	**0.801**	1.584	**1.423**
$\ell = 14$	**0.604**	0.610	0.900	**0.736**	1.500	**1.346**
$\ell = 21$	**0.600**	0.605	0.853	**0.708**	1.454	**1.313**
$\ell = 28$	**0.599**	0.603	0.800	**0.691**	1.399	**1.294**

TABLE IV

EVALUATION OF STORAGE (S), TRANSMISSION (R) AND SUM COSTS ($F = S + R$) OBTAINED WITH THE OPTIMAL REFERENCE PLACEMENT y^* AND WITH THE NAIVE SOLUTION $y^\diamond$ FOR METEOROLOGICAL DATA.

Bit rate Request length	$S^\diamond$	S^*	$R^\diamond$	R^*	$F^\diamond$	F^*
$\ell = 30$	**0.873**	0.885	1.221	**0.930**	2.094	**1.815**
$\ell = 60$	**0.867**	0.876	1.119	**0.908**	1.986	**1.784**
$\ell = 90$	**0.865**	0.872	1.060	**0.896**	1.925	**1.768**

TABLE V

EVALUATION OF STORAGE (S), TRANSMISSION (R) AND SUM COSTS ($F = S + R$) OBTAINED WITH THE OPTIMAL REFERENCE PLACEMENT y^* AND WITH THE NAIVE SOLUTION $y^\diamond$ FOR SELF-SERVICE BIKE TERMINAL OCCUPANCY DATA.

popularity for both data is generated using a Gaussian mixture distribution. For temperature data, it simulates the evolution of, for example, tourists' interest to meteorological data over time (*e.g.,* higher in the summer). For the Velib data, it simulates the evolution of consumers' interest in bicycle renting over time (*e.g.,* higher before/after working hours). We compare our method with the naive approach, where the references correspond to the most popular sources. Results are shown in Tab. IV (for meteorological data) and in Tab. V (for Velib data). We can see that the global cost function F is significantly lower with our approach than with the naive one (about -10% to -20% saving). The naive approach favors the storage cost S, whereas our approach optimize the global cost F.

VII. CONCLUSION

In this paper, the tradeoff between compression efficiency and random access to sequentially processed data has been studied. More precisely, the data were processed with a predictive coding scheme. First an optimization problem has been formulated to solve this tradeoff. In particular, it was shown that solving this trade off is a question of determining the placement of the references. Second, this problem has been shown to be equivalent to an integer linear programming problem, for which classical solvers

exist. Then, a classical and heuristic approach has been studied: the case of periodic placement of the references. Sufficient conditions under which this periodic placement is optimal, have been derived. Finally, experiments performed on synthetic but also real datasets (video sequences and the time series measurement: meteorological and self-service bicycles data) showed the benefits ot the proposed method.

APPENDIX A
PROOF OF PROPOSITION 4

Proof. Let $\mathcal{I}^*$ be the optimal reference index set of Problem (6). First we prove the following statement:

a) For every three consecutive references in $\mathcal{I}^*$, denoted $(i_1, i_2, i_3) \in \mathcal{I}^*$, then i_2 is in the middle between i_1 and i_3 i.e. $i_2 \in \{\lceil \frac{i_1+i_3}{2} \rceil, \lfloor \frac{i_1+i_3}{2} \rfloor\}$. Proof by contradiction: assume that the above statement is not satisfied i.e. $i_2 \notin \{\lceil \frac{i_1+i_3}{2} \rceil, \lfloor \frac{i_1+i_3}{2} \rfloor\}$, and let $\mathcal{I}^\diamond$ be the set $\mathcal{I}^*$, where i_2 is replaced by the index in the middle. More precisely, $\mathcal{I}^\diamond = \mathcal{I}^* \setminus \{i_2\} \cup \{\tilde{i}_2\}$ where $\tilde{i}_2 \in \{\lceil \frac{i_1+i_3}{2} \rceil, \lfloor \frac{i_1+i_3}{2} \rfloor\}$. This implies that

$$\tilde{i}_2 + i_2 \neq i_1 + i_3 \tag{34}$$

We now aim to show that $R(\mathcal{I}^\diamond) < R(\mathcal{I}^*)$.

Let $\mathcal{V}_m = \{m, \ldots, m+\ell-1\}$ be an arbitrary request set. To ensure the reconstruction of the requested sources at the decoder (10), the indices of the sent sources must be either $\mathcal{S}_m^* = \{i_m^*, \ldots, m + \ell - 1\}$ or $\mathcal{S}_m^\diamond = \{i_m^\diamond, \ldots, m + \ell - 1\}$ for the reference index set $\mathcal{I}^*$ or $\mathcal{I}^\diamond$, respectively, where $i_m^* = \max_{j \leq m, j \in \mathcal{I}^*} j$ and $i_m^\diamond = \max_{j \leq m, j \in \mathcal{I}^\diamond} j$. Let $\underline{i}_2 = \min(i_2, \tilde{i}_2)$ and $\bar{i}_2 = \max(i_2, \tilde{i}_2)$. Moreover, from the definition of $\mathcal{I}^*$ and $\mathcal{I}^\diamond$, we deduce that $i_m^* = i_m^\diamond, \ \forall m \in [\underline{i}_2, i_3 - 1]$. The computations in (35) show

$$
\begin{aligned}
&R(\mathcal{I}^*) - R(\mathcal{I}^\diamond) \\
&= \frac{r}{M\ell} \sum_{m=\underline{i}_2}^{i_3-1} \sum_{n=1}^{N} \left(\mathbb{1}_{\mathcal{I}^*}(n) \mathbb{1}_{\mathcal{S}_m^\diamond}(n) - \mathbb{1}_{\mathcal{I}^\diamond}(n) \mathbb{1}_{\mathcal{S}_m^*}(n) \right) (1-\alpha) + \left(\mathbb{1}_{\mathcal{S}_m^*}(n) - \mathbb{1}_{\mathcal{S}_m^\diamond}(n) \right) \alpha \\
&= \frac{r}{M\ell} \sum_{m=\underline{i}_2}^{i_3-1} \left(\left(\mathbb{1}_{\mathcal{S}_m^\diamond}(i_2) - \mathbb{1}_{\mathcal{S}_m^*}(\tilde{i}_2) \right) (1-\alpha) + \sum_{n=1}^{N} \left(\mathbb{1}_{\mathcal{S}_m^*}(n) - \mathbb{1}_{\mathcal{S}_m^\diamond}(n) \right) \alpha \right) \\
&= \frac{r}{M\ell} \alpha \left(\sum_{m=\underline{i}_2}^{\bar{i}_2-1} \sum_{n=1}^{N} \left(\mathbb{1}_{\mathcal{S}_m^*}(n) - \mathbb{1}_{\mathcal{S}_m^\diamond}(n) \right) + \sum_{m=\bar{i}_2}^{i_3-1} \sum_{n=1}^{N} \left(\mathbb{1}_{\mathcal{S}_m^*}(n) - \mathbb{1}_{\mathcal{S}_m^\diamond}(n) \right) \right) \\
&= \frac{r}{M\ell} \alpha \left(\bar{i}_2 - \underline{i}_2 \right) \left(i_1 + i_3 - (\underline{i}_2 + \bar{i}_2) \right) \geq 0 \ \ \forall i_2 \in \{i_1 + 1, \ldots, i_3 - 1\}
\end{aligned}
\tag{35}
$$

that $R(\mathcal{I}^*) \geq R(\mathcal{I}^\diamond)$ with equality if and only if $i_2 \in \{\lceil \frac{i_1+i_3}{2} \rceil, \lfloor \frac{i_1+i_3}{2} \rfloor\}$ (from (34)), which contradicts the initial assumption. This proves statement *(a)*.

We now prove that *b) for a fixed number of references K and if $N = Kq$ with $q \in \mathbb{N}$, then the optimal positioning for these K references is periodic.* Let us denote a group of sources (GOS) a sequence of consecutive sources between two intra coded sources. More specifically, the GOS contains all sources between index n and m such that n and $m+1$ are intra coded, and all other sources with indices between $n + 1$ and m are predicted. Note that the difference of GOS sizes between two arbitrary consecutive GOSs is zero or one. This is a consequence of assumption *(a)*. Moreover, if the difference of GOSs sizes is one i.e. $\frac{i_1 + i_3}{2} \notin \mathbb{N}$, there are two possible positions for $i_2 \in \{\lceil \frac{i_1 + i_3}{2} \rceil, \lfloor \frac{i_1 + i_3}{2} \rfloor\}$, and both give the same rate (from the equality case in (35)). It implies that for any three consecutive GOSs, at least two of them have the same size. Without changing the rate of the encoding scheme, we can shift the references such that all the GOSs with the same size are placed at the beginning of the list of the sources. With the latter positioning of the references, there are at most two values of GOSs sizes, denoted by k and $k + 1$. Let $q > 0$ be the number of references that starts a GOS of size k. We have that the total number of sources satisfies

$$N = qk + (K - q)(k + 1) = Kk + K - q$$

Since by assumption N is a multiple of K, it follows that $K - q = 0$ i.e. there is no GOS of size $k + 1$ i.e. all GOS have the same size. Hence the proof of statement *(b)*.

Finally, if N is not a multiple of K, there exists $r \in \mathbb{N}$ such that $N = Kk + r$ and $1 \leq r \leq K - 1$. However, when $N \to +\infty$, we have $r \to 0$. Hence the proof of this proposition.

$\square$

APPENDIX B
PROOF OF PROPOSITION 5

Proof. As we consider only the requests of ℓ consecutive sources in a predictive encoding scheme which has periodic reference positions, then the per-source transmission rate over all requests can computed from k first requests. We can simplify (5) as follows:

$$R(k) = K \frac{r}{M\ell} \left((1 - \alpha) \sum_{m=1}^{k} \mathbb{1}_{\mathcal{S}_m^* \cap \mathcal{I}}(n) \right.$$

$$\left. + \alpha \sum_{m=1}^{k} \sum_{n=1}^{N} \mathbb{1}_{\mathcal{S}_m^*}(n) \right) \quad (36)$$

where $\mathcal{S}_m^* = \{i_m, \ldots, m + \ell - 1\}$, and $i_m = \max_{j \leq m, j \in \mathcal{I}} j$. Then

$$\sum_{m=1}^{k} \sum_{n=1}^{N} \mathbb{1}_{\mathcal{S}_m^*}(n) = \sum_{m=1}^{k} (m + \ell - i_1) = \frac{k(k+1)}{2} + k(\ell - 1) \quad (37)$$

Let $q \in \mathbb{N}$ and $r \in \mathbb{N}$ be such that $\ell = qk + r$ with $q \geq 0$ and $1 \leq r \leq k$. One can observe that depending on the request set, it may require the transmission of more than one GOS. More precisely, when $m \leq k - r + 1$ the requested sources $\mathcal{V}_m$ are spread over either $q + 1$ or $q + 2$ GOSs. This means that

$$
\sum_{m=1}^{k} \mathbb{1}_{\mathcal{S}_m^* \cap \mathcal{I}}(n) = \sum_{m=1}^{k-r+1} \mathbb{1}_{\mathcal{S}_m^* \cap \mathcal{I}}(n) + \sum_{m=k-r+2}^{k} \mathbb{1}_{\mathcal{S}_m^* \cap \mathcal{I}}(n)
$$
$$
= \sum_{m=1}^{k-r+1} q + \sum_{m=k-r+2}^{k} (q+1)
$$
$$
= (k - r + 1)(q + 1) + (r - 1)(q + 2) \tag{38}
$$

Using (37) and (38), the formula (36) becomes

$$
R(k) = \frac{K}{M} \frac{r}{\ell} \Bigg((1 - \alpha)\,(k + kq + r - 1)
$$
$$
+ \alpha \left(\frac{k(k+1)}{2} + k(\ell - 1) \right) \Bigg)
$$
$$
= \frac{r}{k\ell} \left(k + \ell - 1 + \frac{\alpha}{2}\,(k - 1)\,(k + 2\ell - 2) \right)
$$

where the last expression follows from the fact that $N \to +\infty$. Indeed, since the request size ℓ is finite, we have that $M \to +\infty$ and this implies that $K/M \to 1/k$ when $N \to +\infty$. This concludes the proof. $\qquad\square$

APPENDIX C

PROOF OF THEOREM 6

Proof. We aim to find the minimizer of the following cost function

$$
F(k) = \frac{r}{k}\,[(k - 1)\alpha + 1]
$$
$$
+ \frac{r}{k\ell} \left(k + \ell - 1 + \frac{\alpha}{2}\,(k - 1)\,(k + 2\ell - 2) \right) \tag{39}
$$

Minimizing $F(k)$ is equivalent to minimizing the function $\tilde{F}(k)$:

$$
\tilde{F}(k) = (1 - \alpha)\frac{(2\ell - 1)}{\ell}\frac{1}{k} + \frac{\alpha}{2\ell}k + 2\alpha + \frac{1}{\ell}\left(1 - \frac{3}{2}\alpha\right).
$$

Moreover,

$$
(1 - \alpha)\frac{(2\ell - 1)}{\ell}\frac{1}{k} + \frac{\alpha}{2\ell}k \geq \sqrt{(1 - \alpha)\frac{(2\ell - 1)}{\ell}\frac{\alpha}{2\ell}}
$$

with equality if and only if

$$(1 - \alpha)\frac{(2\ell - 1)}{\ell}\frac{1}{k} = \frac{\alpha}{2\ell}k. \tag{40}$$

Therefore

$$\bar{k} = \sqrt{\frac{2(1 - \alpha)(2\ell - 1)}{\alpha}}. \tag{41}$$

Finally, the minimizer of $F(k)$ belongs to $\{\lfloor \bar{k} \rfloor, \lceil \bar{k} \rceil\}$. Hence, the proof of Theorem 6. $\qquad \square$

APPENDIX D

PROOF OF THEOREM 7

Proof. Let us define the binary vectors $\mathbf{y}^0, \mathbf{y}^{1,t}, \mathbf{z}^{0,m}$, and $\mathbf{z}^{1,m,t}$ which n-th components are

$$y_n^0 = \mathbb{1}_{\mathcal{I}}(n) \tag{42a}$$

$$y_n^{1,t} = \mathbb{1}_{\mathcal{J}_n}(t)(1 - \mathbb{1}_{\mathcal{I}}(n)), \ \forall t \in \mathcal{T}_n \tag{42b}$$

$$z_n^{0,m} = \mathbb{1}_{\mathcal{I}}(n)\mathbb{1}_{\mathcal{S}_m}(n), \ \forall m \in [1, M] \tag{42c}$$

$$z_n^{1,m,t} = \mathbb{1}_{\mathcal{J}_n}(t)\left(1 - \mathbb{1}_{\mathcal{I}}(n)\right)\mathbb{1}_{\mathcal{S}_m}(n), \ \forall t \in \mathcal{T}_n, m \in [1, M] \tag{42d}$$

where $n = 1, 2, \ldots, N$. We aim to show that the reference selection problem (25) can be cast into an integer linear programming problem with respect to the binary vectors $\mathbf{y}^0, \mathbf{y}^{1,t}, \mathbf{z}^{0,m}$, and $\mathbf{z}^{1,m,t}$.

First, with the change of variables introduced in (42), the quadratic cost function in Problem (25) can be rewritten as (43) and becomes linear in the optimization variables.

$$\frac{1}{N}\sum_{n=1}^{N} r_n \left(y_n^0 + \min_{\mathbf{y}^{1,t}} \sum_{t \in \mathcal{T}_n} \alpha_{n,t} y_n^{1,t} \right) + \lambda \sum_{m=1}^{M} \frac{p_m}{|\mathcal{V}_m|} \min_{(\mathbf{z}^{0,m}, \mathbf{z}^{1,m,t})} \sum_{n=1}^{N} r_n \left(z_n^{0,m} + \sum_{t \in \mathcal{T}_n} \alpha_{n,t} z_n^{1,m,t} \right) \tag{43}$$

Second, the vectors $\mathbf{y}^0, \mathbf{y}^{1,t}, \mathbf{z}^{0,m}, \mathbf{z}^{1,m,t}$, introduced in (42), must satisfy the following additional linear constraints to be compliant with the definition of the characteristic functions $\mathbb{1}_{\mathcal{I}}, \mathbb{1}_{\mathcal{J}_n}$ and $\mathbb{1}_{\mathcal{S}_m}$.

$$y_n^{1,t} \leq y_t^0, \qquad\qquad\qquad \forall n \in [1, N], t \in \mathcal{T}_n \tag{44a}$$

$$y_n^0 + \sum_{t \in \mathcal{T}_n} y_n^{1,t} = 1, \qquad\qquad\qquad \forall n \in [1, N] \tag{44b}$$

$$z_n^{0,m} \leq y_n^0, \qquad\qquad\qquad \forall n \in [1, N], m \in [1, M] \tag{44c}$$

$$z_n^{1,m,t} \leq y_n^{1,t}, \qquad\qquad\qquad \forall n \in [1, N], m \in [1, M], t \in \mathcal{T}_n. \tag{44d}$$

where (44a) follows from (42a) and (42b), (44b) follows from (21b), and (44c) and (44d) follow from (42a-d).

Finally, we show that the decodability constraint for lossy compression (24b) can be expressed in terms of linear equations. The proof follows similar steps as in the lossless compression case. Indeed, the constraint in the lossless case (5b) and lossy case (24b) are the same. Therefore, as in the lossless case, (10) and (24b) are equivalent. We now show that the decodability constraint (10) can be expressed in terms of linear equations of the new variables introduced in (42). The proof consists of several steps.

Step 1. (10)$\Rightarrow$(45). There must be at least one reference source, *i.e.*

$$\sum_{n=1}^{N} y_n^0 \geq 1 \tag{45}$$

Step 2. (10)$\Rightarrow$(46). If an index source n belongs to a request then it must be sent, *i.e.*

$$z_n^{0,m} + \sum_{t \in \mathcal{T}_n} z_n^{1,m,t} = 1 \quad \forall n \in \mathcal{V}_m \tag{46}$$

(10)$\Rightarrow$(47). If the source of index n is not requested, and if the source $n+1$ is not a reference, then the fact that the source of index $n+1$ is sent, implies that the previous source of index n must also be sent, *i.e.*

$$z_n^{0,m} + \sum_{t \in \mathcal{T}_n} z_n^{1,m,t} = \sum_{t \in \mathcal{T}_n} z_{n+1}^{1,m,t} \quad \forall n \notin \mathcal{V}_m \tag{47}$$

Step 3. (45), (46), (47)$\Rightarrow$(10). This converse is the same as Step 3 derived in Sec. III for the lossless case.

In conclusion, the new expressions of the cost function (43) and of the constraints (44), (45), (46) and (47), lead to a new formulation of the overall reference selection problem which is indeed a linear integer programming problem.

□

REFERENCES

[1] D. Alfonso, B. Biffi, and L. Pezzoni. Adaptive gop size control in h.264/avc encoding based on scene change detection. In *in Signal Processing Symposium NORSIG*, Rejkjavik, Iceland,, Jun. 2006.

[2] B. Banerjee and J. K. Dutta. A predictive coding framework for learning to predict changes in streaming data. In *2013 IEEE 13th International Conference on Data Mining Workshops*, pages 497–504, Dec 2013.

[3] N. M. Bidgoli, T. Maugey, and A. Roumy. Evaluation framework for 360-degree visual content compression with user view-dependent transmission. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 3756–3760, Sep. 2019.

[4] A. Brampton, A. MacQuire, M. Fry, I. A. Rai, N. J. P. Race, and L. Mathy. Characterising and exploiting workloads of highly interactive video-on-demand. *Multimedia Systems*, 15(1):3–17, Feb. 2009.

[5] Jianle Chen, Marta Karczewicz, Yu-Wen Huang, Kiho Choi, Jens-Rainer Ohm, and Gary J Sullivan. The Joint Exploration Model (JEM) for Video Compression with Capability beyond HEVC. *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

[6] Philip A Chou and Zhourong Miao. Rate-distortion optimized streaming of packetized media. *IEEE Transactions on Multimedia*, 8(2):390–404, 2006.

[7] T.M. Cover and J.A. Thomas. *Elements of information theory, second Edition*. Wiley, 2006.

[8] Sebastian Deorowicz and Szymon Grabowski. Robust relative compression of genomes with random access. *Bioinformatics*, 27(21):2979–2986, 2011.

[9] H. O. Dominguez and K. R. Rao. *Versatile Video Coding*. River, 2019.

[10] Akashnil Dutta, Reut Levi, Dana Ron, and Ronitt Rubinfeld. A simple online competitive adaptation of lempel-ziv compression with efficient random access support. In *2013 Data Compression Conference*, pages 113–122. IEEE, 2013.

[11] M. Karczewicz and R. Kurceren. The SP- and SI-frames design for H.264/AVC. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):637–644, July 2003.

[12] R. Ma, T. Maugey, and P. Frossard. Optimized data representation for interactive multiview navigation. *IEEE Transactions on Multimedia*, 20(7):1595–1609, July 2018.

[13] F. Qian, L. Ji, B. Han, and V. Gopalakrishnan. Optimizing 360 video delivery over cellular networks. In *Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges*, pages 1–6, 2016.

[14] Kannan Ramchandran, Antonio Ortega, and Martin Vetterli. Bit allocation for dependent quantization with applications to multiresolution and mpeg video coders. *IEEE transactions on image processing*, 3(5):533–545, 1994.

[15] F. F. Rodler. Wavelet based 3d compression with fast random access for very large volume data. In *Proceedings. Seventh Pacific Conference on Computer Graphics and Applications (Cat. No.PR00293)*, pages 108–117, Oct 1999.

[16] M. W. P. Savelsbergh. Preprocessing and probing techniques for mixed integer programming problems. *ORSA J. Computing*, 6(4):445–454, 1994.

[17] Khalid Sayood. *Introduction to Data Compression, Third Edition (Morgan Kaufmann Series in Multimedia Information and Systems)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 5th edition, 2018.

[18] Sandip Sinha and Omri Weinstein. Local decodability of the burrows-wheeler transform. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 744–755. ACM, 2019.

[19] P. Suei, C. Kuo, R. Luoh, T. Kuo, C. Shih, and M. Liang. Data compression and query for large scale sensor data on COTS DBMS. In *2010 IEEE 15th Conference on Emerging Technologies Factory Automation (ETFA 2010)*, pages 1–8, Sep. 2010.

[20] A. Ukil, S. Bandyopadhyay, and A. Pal. IoT data compression: Sensor-agnostic approach. In *Data Compression Conference*, March 2015.

[21] T. Wiegand and H. Schwarz. *Video Coding: Part II of Fundamentals of Source and Video Coding*. Number ptie. 2 in Foundations and Trends in Signal Processing Series. Now Publishers, 2016.

[22] Thomas Wiegand and Heiko Schwarz. Source coding: Part i of fundamentals of source and video coding. *Foundations and Trends® in Signal Processing*, 4(1–2):1–222, 2011.

[23] M. Wien, R. Cazoulat, A. Graffunder, A. Hutter, and P. Amon. Real-Time System for Adaptive Video Streaming Based on SVC. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(9):1227–1237, Sept 2007.

[24] Mathias Wien. *High Efficiency Video Coding: Coding Tools and Specification*. Springer, 2015.

[25] Alireza Zare, Alireza Aminlou, Miska M. Hannuksela, and Moncef Gabbouj. HEVC-compliant Tile-based Streaming of Panoramic Video for Virtual Reality Applications. In *Proceedings of the 24th ACM International Conference on Multimedia*, pages 601–605, 2016.

[26] Tiesong Zhao, Zhou Wang, and Chang Wen Chen. Adaptive quantization parameter cascading in hevc hierarchical coding. *IEEE Transactions on Image Processing*, 25(7):2997–3009, 2016.