Universal lossless coding with random user access: the cost of interactivity

Aline Roumy, Thomas Maugey

UNIVERSAL LOSSLESS CODING WITH RANDOM USER ACCESS: THE COST OF INTERACTIVITY

Aline Roumy, Thomas Maugey

INRIA Rennes, Campus de Beaulieu, 35042 Rennes cedex, France

ABSTRACT

We consider the problem of video compression with free viewpoint interactivity. It is well believed that allowing the user to choose its view will incur some loss in terms of compression efficiency. Here we derive the complete rate-storage region for universal lossless coding under the constraint of choosing the view at the receiver. This leads to a counterintuitive result: freely choosing its view at the receiver incurs a loss in terms of storage only and not in the transmission rate. The gain of the optimal scheme with respect to interactive schemes proposed so far is derived and a practical scheme that achieves this gain is proposed.

Index Terms— Free viewpoint, Source coding with side information, Distributed video coding.

1. INTRODUCTION

Free viewpoint television [1] is a new paradigm, in which users can interact with the server and request in real-time a desired viewpoint. The targeted applications are numerous, especially when 3D scenes contain some localized points of interest such as sport, concert or cultural events. Enabling users to interactively navigate through different viewpoints of a static scene is thus a new interesting functionality that however imposes new challenges for 3D streaming systems. In particular, the encoder must prepare a priori a compressed media stream that is flexible enough to enable the free selection of the viewpoint by the users.

A very first approach to offer interactivity, would be to encode each frame of each view independently (in an Intra mode). However, this would be inefficient in terms of compression since the correlation between successive frames and parallel views would not be exploited. On the other hand, non-interactive compression schemes (traditional, scalable or multiview), which do exploit these correlations, can not be used in an interactive scenario. Indeed, all share the property that the encoder knows perfectly the status of the decoder, i.e. which frames have been already decoded. In interactive compression instead, the encoder has to compress one frame by taking into account previous frames of all the views, without knowing which frames will be available at the user side. To work around this problem, [2] re-encodes the frames online, i.e. according to the true user request. However, this scheme fails providing low delay transmission, especially when the number of interacting users is huge. Note that low-delay and large number of users are two features that need to be satisfied in interactive scenarii. Another solution is to encode all possible navigation paths and store them on the server [3, 4, 5]. Upon request, the user directly receives the stream which matches exactly its current navigation. This solution however requires big storage capacity especially when the number of navigation possibilities is large. Note that the number of navigation paths grows exponentially with the number of views and with the GOP size (group of picture). More precisely, a scheme with M views and GOP of G frames needs to store M^{GM} sets of G frames, whereas, for optimality, only GM frames need to be stored and processed.

An interesting solution that saves storage while maintaining a low compression ratio is based on the distributed video coding framework [6]. The authors propose a *worst case* encoding scheme: each frame is encoded assuming that the less correlated frame is available at the decoder. This raises an interesting question: does this worst case scheme achieve the best compression/storage performance?

To answer this question, we introduce two distinct criteria to measure the efficiency of an interactive compression scheme. First, the *storage* corresponds to the average number of bits per source symbol needed to be stored at the server. Second, the *rate* corresponds to the average number of bits per source symbol sent to the user upon request. We derive the whole achievable rate-storage region, and show that a frame has to be stored with the worst case assumption, but can be sent according to the best rate i.e. as if the encoder would know, which frames are available at the decoder. Therefore, it is possible to further improve [6]. In fact, interactivity incurs a loss, with respect to non-interactive schemes, in the storage only but not in the transmission rate, as intuition might suggest.

2. PROBLEM STATEMENT AND SOURCE MODEL

In free viewpoint video coding, M views ($\{X_i\}_{i\in[1,M]}$) are available and can be chosen freely by the user, as shown in Figure 1. The source X_i stands for a frame, while a realization of this source corresponds to a pixel value. At a given time

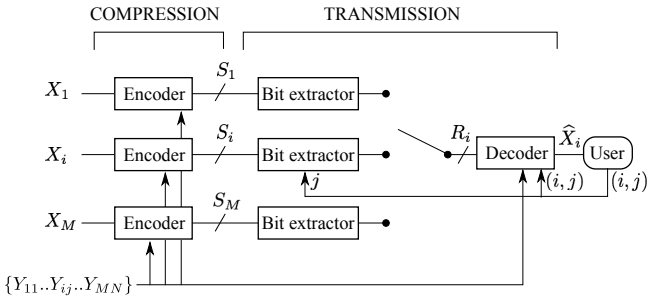

Fig. 1. Free viewpoint video coding.

instant, the user of interest requests X_i. The views previously requested by the same user allows to estimate the current view X_i. This estimate of X_i is modeled as a side information (SI) Y_{ij}, where $ij \in [1, M] \times [1, N]$. This corresponds to the projected view of X_i based on the past navigation path j.

The server consists of two steps: an encoder that performs compression off-line, and a transmitter that adapts the bitstream to the user's request. Compression needs to be independent of the user's request. However, the encoder is not completely blind to the status of the decoder since the set of all "potentially" projected views $\{Y_{ij}\}_{ij\in[1,M]\times[1,N]}$ is available at the encoder. More precisely, upon storage, the realizations of the possible SIs are known but, the choice of the user regarding the previously requested views (j) is unknown. On the contrary, during the transmission phase, once a user requests a view i, its previous requests j are known. Therefore, the server can adapt the transmission to the user request. The adaptation to the user request is made through bitstream extraction. Therefore, two quantities measure the efficiency of the scheme: $(S_i, R_i)_{i\in[1,M]}$, where S_i is the storage for view i (in bits per source symbol) and R_i the transmission rate, when views j and i have been requested. At the receiver, decoding is performed under perfect knowledge of the SI (i.e. its index and its realization).

Without loss of generality and for the sake of clarity, we drop the index i of the current view. The general scheme can be simplified into Figure 2. In order to analyze this scheme, we first define a source, which aims at modeling the uncertainty at the encoder on the decoder status. Then, we define the achievability of a rate-storage pair.

Definition 1. *(*Static source*). A* static source (X, Y) *is a discrete sequence of i.i.d. random variables drawn according to a distribution chosen randomly in the finite set of probability distributions* $\{\mathbb{P}(X, Y_j)\}_{j\in[1,N]}$. *The choice of one distribution, say* $\mathbb{P}(X, Y_j)$*, is made with probability* $\mathbb{P}(J = j) = p_j$. *The source is completely determined by* $\mathcal{P} = \{\mathbb{P}(X, Y_j), p_j\}_{j\in[1,N]}$.

The static source is the static source with prior defined in [7] to model SI uncertainty at the encoder. This source differs from a mixture model, since the probability chosen, will remain the same for the whole sequence. Therefore, the static source is stationary but non ergodic.

Definition 2. *(*Universal lossless source coding with random user access*). Let* (X, Y) *be a static source determined by* $\mathcal{P} = \{\mathbb{P}(X, Y_j), p_j\}_{j\in[1,N]}$.

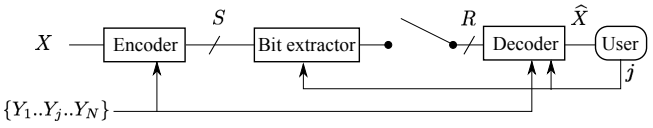

Fig. 2. Universal lossless coding under random user access.

Lossless source coding with random user access *(see Figure 2) is the problem of compressing the source* X *knowing that some SI* Y *will be available at the decoder, storing the compressed bistream with* S *bits per source symbol, and sending a bitstream extracted from the stored one, at a rate of* R *bits per source symbol, once the SI has been revealed to the encoder. The encoder knows the set of possible SIs i.e. for each* j*, the sequence of realizations for the source* Y_j*. This set of sequence realizations is denoted* $\{Y_j\}_{j\in[1,N]}$*. Moreover, the scheme is* **universal** *in the sense that the source statistics* $\mathcal{P}$ *are neither known at the encoder nor at the decoder.*

Definition 3. *(*Achievable rate-storage pair*). Let* (X, Y) *be a static source determined by* $\mathcal{P} = \{\mathbb{P}(X, Y_j), p_j\}_{j\in[1,N]}$. *A* rate-storage pair (R, S) *is said to be* achievable *for universal lossless source coding with random user access, if, for a request* $j \in [1, N]$*, there exists a sequence of encoder, bit extractor and decoder (indexed by the length of the source* (X, Y)*) that can reproduce the source X (i.e.* $\hat{X} = X$*), as the sequence length of the source* (X, Y) *goes to infinity.*

Definition 4. *(*The rate-storage region*). The* rate-storage region *of the universal lossless source coding problem under random user access is the closure of the set of achievable rate-storage pairs* (R, S).

3. COST OF INTERACTIVITY

In this section, we derive the rate storage region as defined in the above section. This allows us to characterize the true cost of interactivity for lossless coding. Then, we compare some existing schemes with the optimal scheme, that achieves the best rate-storage pair.

Theorem 1. *(*The rate-storage region*). Let* (X, Y) *be a static source determined by* $\mathcal{P} = \{\mathbb{P}(X, Y_j), p_j\}_{j\in[1,N]}$. *We consider universal lossless source coding under random user access as in Figure 2. For a request* j*, the region of achievable (rate-storage) pair* (R, S) *is*

$$R \geq H(X|Y_j) \quad (1)$$

$$S \geq \max_{k:\, p_k>0} H(X|Y_k) \quad (2)$$

where $H(X|Y_j)$ *stands for the conditional entropy of the source* X *given* Y_j*. The region of achievable rate-storage is shown in Figure 3 and corresponds to the white area.*

Proof: ***Achievability: inner bound.***

Non universal case. Let us first assume that the statistics $\mathcal{P}$

are known at the encoder and decoder. [8] proposes an incremental coding strategy for broadcasting a source X losslessly to a number of receivers with different qualities of SI, denoted $(Y_1, \ldots, Y_N)$. For each sequence of realizations of the source X, the encoder sends an index (from the least to the most significant bit), whereas the receivers simply "tune out" once they can decode. More precisely, receiver with SI Y_j tunes out when it has received $H(X|Y_j)$ bits per symbol.
This scheme can be used in our setup: the tuning out is performed by the bit extractor. This is possible since the bit extractor knows the previous request j and thus the SI Y_j available at the decoder. It therefore knowns how many bits are required for the decoder to succeed. Therefore, $R = H(X|Y_j)$ is achievable. Moreover, for this code, the total number of bits to be stored corresponds to the worst case i.e. $S = \max_{k:\ p_k>0} H(X|Y_k)$. Therefore $(R, S) = (H(X|Y_j), \max_{k:\ p_k>0} H(X|Y_k))$ is achievable. This point is labeled (1) in the rate-storage region of Fig. 3.
Generalization of the scheme [8] *to the universal case*: The encoder first sends the empirical conditional distribution $\mathbb{P}(X)$ (usually referred to as type). Then, the encoder uses the incremental scheme described above. The type scales logarithmically with the source sequence length, whereas the data grows linearly. Therefore, the universal scheme will asymptotically achieve the same rate-storage pair as the non-universal scheme.

Converse: outer bound. Here, we use the converse of less stringent theorems, where more information is available at the encoder and/or decoder.
• $R \geq H(X|Y_j)$. Proof by contradiction, by invoking the converse of the Slepian-Wolf theorem [9] with perfectly known SI statistics at the encoder.
• $S \geq \max_{k:\ p_k>0} H(X|Y_k)$. Proof by contradiction. If $S < \max_{k:\ p_k>0} H(X|Y_k)$, then with non zero probability, the SI might be $Y_{\bar{k}}$, where $\bar{k} = \arg\max_{k:\ p_k>0} H(X|Y_k)$. From the converse of the Slepian Wolf theorem [9], this source can not be recovered at the decoder. □

It is interesting to compare the optimal scheme for universal lossless coding with random user access (point (1) in Figure 3) to other existing schemes. Let us first consider the scheme without interactivity. This will allow us to derive the cost of interactivity.

Corollary 1. *(*Cost of interactivity*). The cost of interactivity that allows each user to control the focus of attention rather than receiving the one decided by a director at the server, is in the storage only and is limited to* $\Delta S = \max_{k:\ p_k>0} H(X|Y_k) - H(X|Y_j)$.

Proof: In the case without interactivity, the encoder knows the SI Y_j at the encoder and the source X can be encoded losslessly at rate $H(X|Y_j)$ [9], even in the case where the statistics are neither known at the encoder nor at the decoder [10]. Therefore, the achievable storage-rate pair satisfies $R = S = H(X|Y_j)$. □

The case without interactivity is labeled (3) in Figure 3.

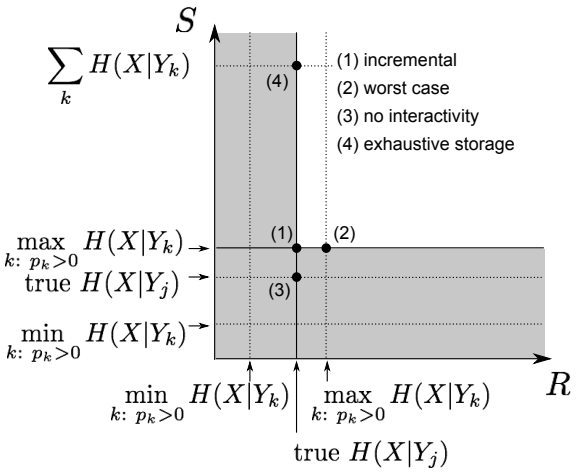

Fig. 3. Achievable (R, S) pair for free viewpoint video.

The statement of Corollary 1 is rather counterintuitive. Indeed, it is well believed that the cost of interactivity is not only in the storage but also in the rate. For instance, [6] proposes an interactive scheme called worst case, where the achieved rate-storage pair (R, S) satisfies $R = S = \max_{k:\ p_k>0} H(X|Y_k)$. (This point is labeled (2) in Figure 3.) Therefore, [6] incurs a loss with respect to non-interactive schemes not only in the storage but also in the rate. Note that the same performance ((2) in Figure 3) is achieved if the sender has no information about the SI available at the decoder [7][11, Theorem 7.3.4.]. In other words, [6] does not use the information related to the previous request j.

The last comparison is made with the exhaustive approach of [3, 4, 5]. The source X is encoded many times. One encoding is performed per SI Y_j and all generated bitstreams are stored. Therefore, the transmission rate is optimal ($R = H(X|Y_j)$) but at the price of a significantly increased storage $S = \sum_k H(X|Y_j)$. This scheme achieves (4) in Figure 3. However, Theorem 1 shows that there is no need to use all that storage to achieve the optimal transmission rate.

4. PRACTICAL SCHEME

We now propose a practical scheme to solve the interactive lossless source coding problem. First note that, to design an optimal interactive lossless source coding scheme, it is necessary to split the encoder into a compression and a transmission phase. If not, the sent bitstream is the stored one, and the transmitted rate equals the storage, which corresponds to the worst case (point (2) in Figure 3). The incremental code proposed in [8] satisfies this splitting. However, it can not be used in practice since it relies on the random generation of the code. But, as noticed in [8], our problem is equivalent to a scheme where the source X is sent unprotected on a broadcast channel, which produces a set of outputs $\{Y_j\}$ (one output per receiver). Then, some additional bits are sent on a perfect channel. Each receiver stops gathering bits once it can reconstruct X. Moreover, there exists codes known as Digital Fountain codes [12] that can solve this problem.

ENCODER: ***compression***
Data: $X, Y_1, \ldots, Y_j, \ldots, Y_N$
Result: stored bitstream (S)
order all Y_j s.t. $H(X|Y_1) < \ldots < H(X|Y_N)$;
choose a systematic channel code of rate $r = \frac{1}{1+e_N}$, where e_N is the proportion of erased symbols in the largest occlusion area;
for *each bitplane* B_l *of* X **do**
  encode B_l with rate r and keep only the parity bits;
  for *each SI* Y_j **do**
    determine the number of parity bits b_j s.t. Y_j and $b_1 + \ldots + b_j$ parity bits are sufficient to recover X;
    store b_j bits plus a marker EOS;
  end
end
ENCODER: ***transmission***
Data: stored bitstream (S), j
Result: sent bitstream (R)
for *each bitplane* **do**
  send $b_1 + \ldots + b_j$ parity bits;
end
DECODER:
Data: sent bitstream (R)
Result: X
for *each bitplane* **do**
  estimate X from Y_j and the $b_1 + ... + b_j$ parity bits;
end

Algorithm 1: Practical scheme for lossless source coding with random user access

These Fountain codes were first proposed for the erasure channel. Interestingly, [13] shows that the correlation between the views can be modeled by an m-ary erasure channel. The erased symbols correspond in fact to the disoccluded pixels after view prediction. Therefore, the SI Y_j can be seen as the output of an erasure channel with erasure probability e_j and input, the source X. Let us assume that the source X is i.i.d. uniformly distributed, and can take $m = 2^L$ values. The optimal transmitted rate is $H(X|Y_j) = e_j H(X) = e_j L$ i.e. exactly the amount of erased symbols.

Our scheme (see details in Algorithm 1) encodes the source X with a systematic code and stores the necessary parity symbols. The rate code is chosen in order to deliver the number of parity bits for the maximal erasure probability (worst case). Then, upon receiver request, some parity bits, extracted from the parity bit sequence, are sent. If the number of parity bits extracted, exactly matches the bound $H(X|Y_j)$ for all j, then the code is said to be optimal for the source coding problem with random user access. For the erasure channel, there exist such codes, which are called Maximum Distance Separable codes. However, there decoding complexity is prohibitive. Instead, we choose the standardized LDPC-Staircase codes (RFC 6816) [14], that achieve a very good tradeoff between encoding/decoding complexity and performance. This choice is motivated by the fact that low decoding complexity is a key issue in interactive communication. Moreover, we choose binary codes since [15] shows the equivalence between the m-ary erasure channel and L parallel binary erasure channels, where $m = 2^L$. Therefore, for our problem, codes for the binary erasure channel are sufficient to achieve the optimal rate-storage pair. An implementation is made available by the authors of the RFC at [16]. Furthermore, for complexity reason, and in order to allow parallel decoding of a frame, we split a frame into blocks of size 200x200. For instance, an HD frame (1920x1080) will result into 52 blocks.

	Number of symbols sent			Number
	$e_1 = 1\%$	$e_2 = 5\%$	$e_3 = 10\%$	Stored
theoretical	400	2.000	4.000	4.000
LDPC-Staircase r=.909	411	2.011	4.011	4.011
	Number of symbols sent			Number
	$e_1 = 10\%$	$e_2 = 20\%$	$e_3 = 25\%$	Stored
theoretical	4.000	8.000	10.000	10.000
LDPC-Staircase r=.799	4.023	8.023	10.023	10.023

Table 1. X is a block of $200 \times 200 = 40.000$ symbols; the SI Y_j can predict all X except for a proportion of e_j symbols. In the practical scheme, the number of sent and stored symbols, to recover X from Y_j, is very close to the theoretical bound: $H(X|Y_j)$.

Table 1 compares the theoretical transmission rate and the one obtained with our scheme based on the LDPC-Staircase codes (RFC 6816) [14]. The table shows that the overhead is very limited: only 0.275% (resp. 0.23%) more symbols is needed with respect to the optimal performance, when the occlusion area is up to 10% (25% resp.) of the image size.

5. CONCLUSION

In this paper, we studied video compression with free viewpoint interactivity. For lossless compression, we derived the optimal performance under the assumption that the statistics of the source are neither known at the encoder nor at the decoder. We showed that the optimal scheme needs to store the data according to the worst case (as if the transmitter would not use the previous user requests) but sends the data at the same rate as a non-interactive scheme. Therefore, interactivity incurs a loss in terms of storage only. The gain with respect to interactive schemes proposed so far was derived. Finally, a practical scheme that achieves this gain was proposed in the case of i.i.d. sources. Future work will include the design of codes for multiview videos.

6. REFERENCES

[1] M. Tanimoto, "FTV: Free-viewpoint television," *IEEE Signal Processing Magazine*, vol. 27, no. 6, pp. 555–570, Jul. 2012.

[2] JG. Lou, H. Cai, and J. Li, "A real-time interactive multi-view video system," in *Proc. ACM Int. Conf. on Multimedia*, Singapore, 2005, pp. 161–170.

[3] Y. Liu, Q. Huang, S. Ma, D. Zhao, and W. Gao, "RD-optimized interactive streaming of multiview video with multiple encodings," *Journal on Visual Commun. and Image Repr.*, vol. 21, no. 5-6, pp. 1–10, Jul. 2010.

[4] H. Kimata, M. Kitahara, K. Kamikura, and Y. Yashima, "Free-viewpoint video communication using multi-view video coding," *NTT Technical Review*, vol. 2, no. 8, pp. 21–26, Aug. 2004.

[5] S. Shimizu, M. Kitahara, H. Kimata, K. Kamikura, and Y. Yashima, "View scalable multiview video coding using 3-d warping with depth map," *IEEE Trans. on Circ. and Syst. for Video Technology*, vol. 17, no. 11, pp. 1485–1495, Nov. 2007.

[6] G. Cheung, A. Ortega, and NM. Cheung, "Interactive streaming of stored multiview video using redundant frame structures," *IEEE Trans. on Image Proc.*, vol. 3, no. 3, pp. 744–761, Mar. 2011.

[7] E. Dupraz, A. Roumy, and M. Kieffer, "Source Coding with Side Information at the Decoder and Uncertain Knowledge of the Correlation," *IEEE Transactions on Communications*, vol. 62, no. 1, pp. 269 – 279, Jan. 2014.

[8] M. Feder and N. Shulman, "Source broadcasting with unknown amount of receiver side information," in *Information Theory Workshop (ITW), Proceedings.*, march 2002, pp. 302 – 311.

[9] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Transactions on Information Theory*, vol. 19, no. 4, pp. 471–480, July 1973.

[10] E. Dupraz, *Codage de sources avec information adjacente et connaissance des corrélations*, Ph.D. thesis, Univ. Paris Sud, 2013.

[11] T.S. Han, *Information-spectrum methods in information theory*, Springer, 2003.

[12] D. MacKay, "Fountain Codes," *IEE Proc.-Commun.*, vol. 152, no. 6, pp. 1062–1068, Dec. 2005.

[13] L. Toni, T. Maugey, and P. Frossard, "Correlation-aware packet scheduling in multi-camera networks," *IEEE Trans. on Multimedia*, vol. 16, no. 2, pp. 496–509, 2014.

[14] V. Roca, M. Cunche, and J. Lacan, "LDPC-Staircase Forward Error Correction (FEC) Schemes for FECFRAME," Dec. 2012.

[15] J.L. Massey, "Capacity, cut-off rate and coding for a direct-detection optical channel," *IEEE Trans. on Comm.*, pp. 1615 – 1621, Nov 1981.

[16] M. Cunche, J. Detchart, J. Lacan, and V. Roca, "OpenFEC.org: because open, free, AL-FEC codes and codecs matter," http://openfec.org.