COMPARISON OF MPEG-7 DESCRIPTORS FOR LONG TERM SELECTION OF REFERENCE FRAMES

Javier Ruiz-Hidalgo and Philippe Salembier

Universitat Politècnica de Catalunya, Barcelona, Spain.
{j.ruiz,philippe.salembier}@upc.edu

ABSTRACT

During the last years, the amount of multimedia content has greatly increased. This has multiplied the need of efficient compression of the content but also the ability to search, retrieve, browse, or filter it. Generally, video compression and indexing have been investigated separately. However, as the amount of multimedia content grows, it will be very interesting to study representations that, at the same time, provide good compression and indexing functionalities. Moreover, even if the *indexing metadata* is created for functionalities such as search, retrieval, browsing, etc., it can also be employed to increase the efficiency of current video codecs. Here, we use it to improve the long term prediction step of the H.264/AVC video codec. This paper focuses on the comparison between four different MPEG-7 descriptors when used in the proposed scheme.

Index Terms— Indexing, Metadata, MPEG-7, Video Coding, H.264, AVC

1. INTRODUCTION

For many years, video compression and indexing have been considered as two independent issues and, therefore, they have been mostly studied separately. This is due to the fact that the main goal of video compression is to find an optimum representation, in a rate-distortion sense, for visualization. On the other hand, the main goal of indexing is to find an optimum representation to provide functionalities such as search, retrieval, navigation, browsing, filtering, etc. However, even though the functionalities provided by both video compression and indexing are different, the exponential growth of digital multimedia content will force future multimedia services to consider the compression and indexing aspects of the content simultaneously. In that sense, it is important to explore new strategies which are able, at the same time, to provide functionalities of compression, search, retrieval, navigation, etc.

An initial contribution of representations sharing compression and indexing functionalities can be found in [1] and [2]. In these papers, the MPEG-7 *Parametric Motion* descriptor is used to improve the motion estimation and compensation step in advanced prediction schemes (Global Motion Compensation) for the H.264/AVC standard [3]. In [4], MPEG-7 texture descriptors are used to signal the presence of detailed texture within the image. A texture analyzer identifies the texture regions of the image with no important subjective details. These texture blocks are skipped in the encoder and separately synthesized in the decoder using a texture generator.

In [5], the MPEG-7 *Analytic Transition* is used to improve the coding inside transitions. The descriptor is used to improve the prediction of the interpolative mode of B frames within the transition. The MPEG-7 *Motion Activity* descriptor has also been used to improve the frame type selection [5]. In this case, descriptors are used to select between a set of predefined GoP structures improving the overall coding efficiency. A tool for video segment re-ordering is presented in [6] where a high level descriptor, the MPEG-7 *Video Segment* descriptor, is employed to create a new coding order for the entire video sequence that better exploits the temporal redundancy. In [7], the MPEG-7 *Color Layout* descriptor is employed to improve the long term prediction step of the H.264/AVC video encoder.

The main contribution of this paper is to further develop and study the technique presented in [7]. Here, various MPEG-7 indexing descriptors are studied and their ability to improve the coding performance is compared against the MPEG-7 *Color Layout*.

The structure of the paper is as follows. The following section gives an overview of the technique used to select reference frame sub-blocks within H.264/AVC. Section 3 reviews all the indexing metadata used to select those reference frames sub-blocks. Section 4 presents several experimental results of the proposed technique. Final conclusions are given in Section 5.

2. LONG TERM SELECTION OF REFERENCE FRAME SUB-BLOCKS

The technique used to select long term reference frame sub-blocks is an extension of the one proposed in [6] and it has been first presented in [7]. The technique improves the selection of frames in the long term prediction buffer (LTPB) of current H:264/AVC codecs. In H.264/AVC codecs, the LTPB is usually filled, for each frame to be coded, with the N closest P or I frames in the video sequence [8]. In practice, the number N of possible reference frames in the LTPB is limited due to two factors: First, the computational complexity of the motion estimation and second, the bitrate increase needed to add the information about the reference frame.

Indexing metadata can be introduced in the long term temporal prediction scheme by performing a pre-selection of N candidates reference frames to be included in the LTPB among a very large number M of possible frames. This strategy formulates the prediction of the current frames in three steps: 1) divide the frame into sub-blocks (typically 1x1 or 2x2 sub-blocks per frame), 2) search and retrieve the sub-blocks to create the LTPB and 3) motion compensate the sub-blocks of the LTPB. As indexing metadata have been designed for search and retrieval capabilities, the search space M of possible reference frames can be increased without a severe penalty in the computational cost. Also, the LTPB can be filled with frames the content of which is similar to the frame being coded as they are selected by the indexing metadata and not chosen only based on their temporal proximity.

The process to fill the LTPB with possible reference frames can be described as follows. Each frame in the LTPB is sub-divided into $K \times L$ uniform sub-blocks. Each sub-block of the LTPB is filled

978-1-4244-2354-5/09/$25.00 ©2009 IEEE

with the sub-block (from M past encoded frames) with minimum distance to the corresponding sub-block of current frame $\mathbf{I}(t)$. The distance between sub-blocks in the current and the previous images is computed using the indexing metadata. As the frames are subdivided, each frame has $K \times L$ indexing metadata associated with it. The comparison is made between the indexing metadata of the sub-block of the original image with all the sub-blocks of previously encoded images. If there are N possible reference frames in the LTPB, the N sub-blocks with closest distance are chosen to fill the corresponding possible reference frame sub-blocks.

Figure 1 shows an example of filling the LTPB with $K \times L = 2 \times 2$ and $N = 4$. As an example, the top-left sub-block of the LTPB (marked as i in the current frame) is filled by using indexing metadata to measure the distance between the sub-block i of the current frame and all other sub-blocks of the previous M encoded frames. The numbers in the sub-blocks in the previous M encoded frames represent the order of the distance measured with the indexing metadata. So, for instance, the sub-block marked as *1* has the minimum distance between the indexing metadata of the i block of the current image and all other sub-blocks. The block marked as *2* has the second minimum distance and so on. In this case, with $N = 4$, the 4 sub-blocks with minimum distance are selected to fill the corresponding sub-block in the LTPB. The process to fill the entire LTPB follows the same strategy for the remaining sub-blocks.

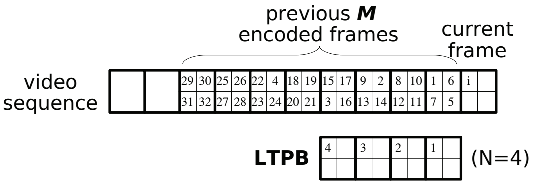

Fig. 1. Long term selection of reference frame sub-blocks using indexing metadata.

3. SELECTED INDEXING METADATA

In this paper, four different MPEG-7 descriptors (*Color Layout*, *Color Structure*, *Edge Histogram* and *Texture Homogeneous*) are compared when used as indexing metadata in the presented long term selection of reference frames sub-blocks. In our scheme, the descriptors are created for all $K \times L$ sub-block of each frame to be encoded. Also, a distance $\mathcal{D}_d$ is defined for each selected descriptor d. The resulting distance $\mathcal{D}_d$ measures the similarity between two descriptors. For instance, if $\mathcal{D}_d = 0$, both descriptors are identical and the sub-blocks represented by these indexing metadata are expected to be very similar. However, greater values of $\mathcal{D}_d$ represent very different indexing metadata and, therefore, sub-blocks with no similar content between them.

The *Color Layout* descriptor (CLD) specifies a spatial distribution of colors for high-speed retrieval and browsing [9]. The process of extracting a CLD descriptor is as follows. Firstly, a thumbnail image of 8×8 pixels is created. Secondly, the thumbnail is transformed by a classical DCT and the DCT coefficients (for luminance and chrominance) are quantized and stored using a simple zig-zag scan. Finally the DCT coefficients are truncated so only 6 luminance coefficients and 6 chrominance coefficients (3 for each color component) are used. The distance between CLD descriptors is computed by measuring the Euclidean distance between those 12 DCT coefficients. Let us denote by vectors $\mathbf{y}, \mathbf{cb}, \mathbf{cr}$ the luminance and chrominance coefficients of one sub-block of the input image $\mathbf{I}(t)$ and $\mathbf{y}', \mathbf{cb}', \mathbf{cr}'$ the CLD descriptor coefficients corresponding to a sub-block of a previously encoded frame. The distance $\mathcal{D}_{CLD}$ between CLD descriptors can be measured as:

$$\mathcal{D}_{CLD} = \sqrt{\sum_{i=0}^{5} (\mathbf{y}_i - \mathbf{y'}_i)^2} + \sqrt{\sum_{i=0}^{2} (\mathbf{cb}_i - \mathbf{cb'}_i)^2} + \sqrt{\sum_{i=0}^{2} (\mathbf{cr}_i - \mathbf{cr'}_i)^2}$$

The *Color Structure* descriptor (CSD) represents an image by both its color distribution and the local spatial structure of color. It corresponds to a histogram of 32 values (quantized to eight bits). The histogram is first computed using 256 bins in the HMMD color space and then non-linearly quantized into 32 bins. The distance between CSD descriptors is defined by measuring the mean absolute difference between each vector $\mathbf{s}$ and $\mathbf{s}'$ corresponding to the histogram of two CSD descriptors respectively.

$$\mathcal{D}_{CSD} = \frac{\sum_{i=0}^{31} |\mathbf{s}_i - \mathbf{s'}_i|}{32}$$

The *Homogeneous Texture* descriptor (HTD) characterizes the texture of a region or image using the mean energy and the energy deviation from a set of frequency channels. The descriptor corresponds to a feature vector $\mathbf{h}$ of 62 non-linearly scaled and quantized (to eight bits) values. The first two values are the mean and standard deviation of the entire sub-block. The remaining values correspond to the mean energy and energy deviation of each channel in a partition of the 2D frequency plane. The frequency plane partitioning is uniform along the angular direction (with a step size of 30 degrees) and non-uniform in the radial direction. The distance between two HTD descriptors is measured by summing the absolute difference between the corresponding feature vectors $\mathbf{h}$ and $\mathbf{h}'$:

$$\mathcal{D}_{HTD} = \sum_{i=0}^{61} |\mathbf{h}_i - \mathbf{h'}_i|$$

The *Edge Histogram* descriptor (EHD) represents the local edge distribution in the image [10]. It corresponds to a histogram $\mathbf{e}$ of non-linearly quantized values. The process to extract an EHD descriptor starts by dividing the sub-block into 4×4 sub-images. The histogram $\mathbf{e}$ is then created categorizing edges in each sub-image in five types: vertical, horizontal, $45°$ diagonal, $135°$ diagonal and non-directional edges. Since there are 16 sub-images, a total of $16 \times 5 = 80$ histogram bins are needed. Each of the bin measures the appearance of a specific edge type in the corresponding sub-image. Finally, the histogram values are normalized and quantized to three bits. The distance between EHD descriptors is computed as follows:

$$\mathcal{D}_{EHD} = \sum_{i=0}^{79} |\mathbf{e}_i - \mathbf{e'}_i|$$

It should be noted that the proposed technique needs to access the indexing metadata at the decoder end to be able to re-create the same LTPB used in the encoding process. This means that if the indexing metadata is not available at the decoder or cannot be recreated, then it must be streamed together with the content. Fortunately, there are scenarios where the indexing metadata will be available at both the encoder and the decoder sides at no cost. For instance, a scenario where a user is browsing a database of movies from a video content provider. In this scenario, the user, in an initial search phase, is able to get a local copy of the indexing metadata to be able to search the content. However, in scenarios where no indexing metadata is available, the indexing metadata is streamed using the binary representation proposed by MPEG-7 [11]. Table 1 shows the amount of bits needed to stream the four MPEG-7 descriptors for different $K \times L$ configurations. As indexing metadata is very similar between sub-blocks, it is compressed using a standard Lempel-Ziv algorithm [12] (in parentheses in the Table). As all the MPEG-7 descriptors are independent of the sub-bock size, the resulting number of bits is the same for all kind of sequences and resolutions. Thus, the penalty of having to stream the indexing metadata is greater for low resolutions or low coding quality.

Sub-divisions	CLD	CSD	THD	EHD
1 × 1	55 (9.2)	256 (23.1)	496 (37.1)	240 (17.3)
2 × 2	220 (33.8)	1024 (84.2)	1984 (124.9)	960 (56.8)

Table 1. Number of bits/frame needed to store the MPEG-7 descriptors for various sub-divisions of the frames. In parentheses the number of bits when compressed with a Lempel-Ziv algorithm.

4. EXPERIMENTAL RESULTS

All results in this section are based on the H.264/AVC video codec reference software version JM-8.1a. The conditions and settings of the H.264/AVC video encoder are set to standard values such as Hadamard and CABAC on, ± 16 motion vector range, $1/8$ pixel accuracy, Intra period equal to 0, all variable size motion modes and no B frames. The LTPB size has been fixed to $N = 5$ for both the standard H.264/AVC and the proposed technique. Therefore, the standard H.264/AVC uses the 5 previously encoded frames to fill the LTPB while the proposed technique selects, among all sub-blocks of the M previously encoded frames, the 5 ones with smaller distance using one of the four MPEG-7 descriptors. Also, the scenario where indexing metadata must be streamed together with the content is considered so the bitrate shown in Table 1 is included in the rate-distortion plots.

The first tests compare the rate-distortion of the proposed technique and of the standard H.264/AVC encoder for the *Geri* sequence [1] using all four MPEG-7 descriptors. In this test, the indexing metadata is used to compare each sub-block of the frame being coded against all the sub-blocks of $M = 100$ previously coded frames. Figure 2 shows the rate-distortion curves for the $K \times L = 1 \times 1$ configuration while Figure 3 shows the same curves but for the $K \times L = 2 \times 2$ configuration.

It can be seen how using the four MPEG-7 descriptors in the proposed technique outperforms the standard H.264/AVC in the 1×1 configuration. However, in the 2×2 configuration only the CLD and the EHD descriptors are able to outperform the standard H.264/AVC. This situation is expected as the CLD and EHD descriptors are the ones with lower bitrate penalty when having to send them together with the video. This difference is accentuated in the 2×2 configuration as the bitrate is increased in almost 4 times the bitrate needed for the 1×1 configuration. Note that overall, the best results are obtained with CLD descriptor in the 2×2 configuration.

Figure 4 shows the rate-distortion curves resulting for the *Jornal da Noite* sequence [2] for the 1×1 configuration and using $M = 200$ previously coded frames to fill the LTPB using indexing metadata.

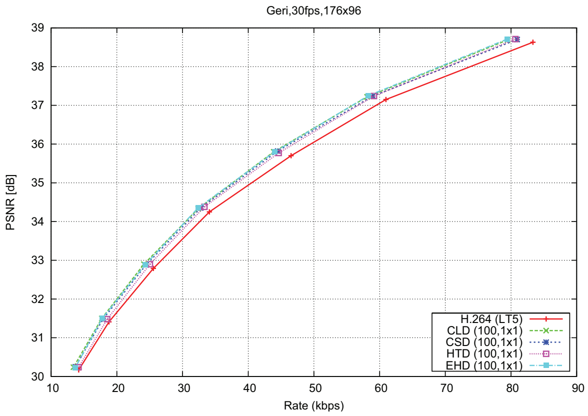

Fig. 2. Rate-distortion curves comparing the standard H.264/AVC encoder and the proposed technique (1×1) for sequence *Geri*.

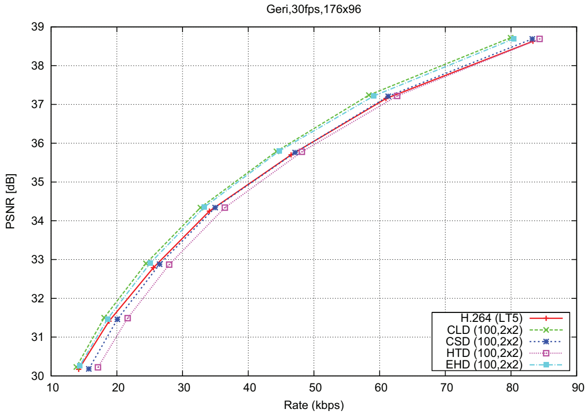

Fig. 3. Rate-distortion curves comparing the standard H.264/AVC encoder and the proposed technique (2×2) for sequence *Geri*.

Again, all four descriptors outperform the standard H.264/AVC and, this time, the EHD descriptor is the descriptor with the best rate-distortion efficiency among the four with up to 0.8db gains with respect to standard H.264/AVC.

Finally, Fig. 5 shows the experimental results for the *Telediario* sequence [2] with the 2×2 configuration using $M = 200$ previously coded frames to select the best sub-blocks. In this case, the descriptor with the best rate-distortion efficiency is the CLD descriptor. However, as can be seen in all figures, as the bitrate increases, the penalty of having to send the indexing metadata is lower and the difference of efficiency of the different descriptors is also lower.

5. CONCLUSIONS

This paper further exploits indexing metadata to improve coding. Up to four different MPEG-7 descriptors have been analyzed and employed to improve the long term prediction step of the H.264/AVC video encoder. The basic strategy formulates the prediction of current frames in two steps: 1) search and retrieval of candidates for the Long Term Prediction Buffer (LTPB) and 2) motion compensation of the data of the LTPB. As the proposed technique exploits the long term temporal redundancy between frames of the video, it should

[1] The *Geri* sequence corresponds to a video clip composed of 364 frames of RAW uncompressed frames at a resolution of 176 × 92 pixels and 30fps.

[2] The *Jornal da noite* and *Telediario* sequences are extracted from the MPEG-7 test set, decompressed, low-pass filtered and decimated from CIF to QCIF resolution to reduce the coding artifacts. Both sequences correspond to a news program and contain a mixture of low, medium and high motion shots with a duration of 600 frames at 5fps.

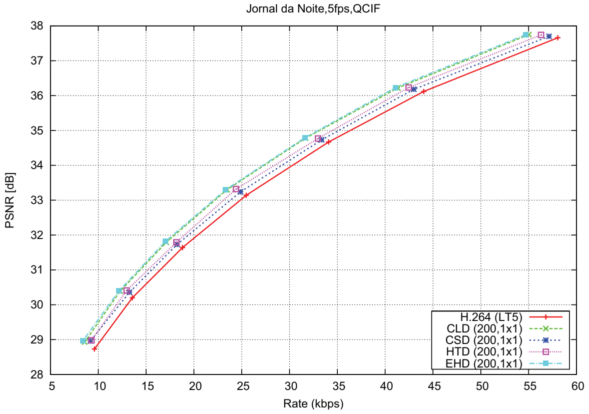

Fig. 4. Rate-distortion curves comparing the standard H.264/AVC encoder and the proposed technique (1 × 1) for *Jornal da Noite* sequence.

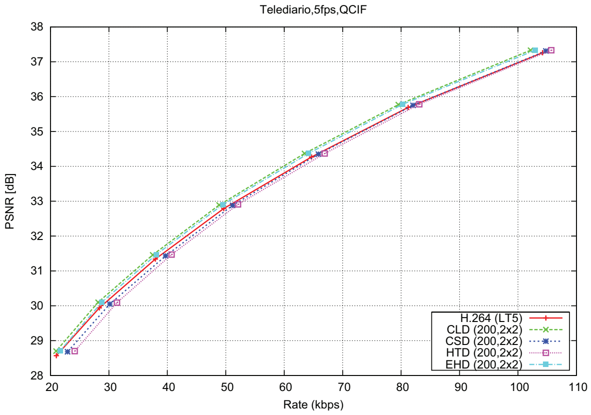

Fig. 5. Rate-distortion curves comparing the standard H.264/AVC encoder and the proposed technique (2 × 2) for *Telediario* sequence.

perform better on sequences with repeating shots or sequences where the same objects disappear and reappear at a later time. Results have shown promising gains in these kind of sequences for all four descriptors. However, if scenarios where the indexing metadata is not available in the decoder are considered, then the MPEG-7 CLD and EHD descriptors are better candidates as the amount of bits needed to store their representation is lower.

Even if up to 200 previously coded frames have been used to fill the LTPB in the experimental results, the computational complexity increase in the encoder is almost negligible. The process of filling the LTPB uses indexing metadata, which has been created with that functionality in mind, and thus, the selection of reference frame sub-blocks is fast and efficient. However, the proposed technique increase the memory requirements of the decoder, as frames in the LTPB are needed to compensate the motion estimation. If memory size in the decoder is limited, such as for mobile devices, the number of possible candidates must be reduced.

Our future research will focus on performing additional experiments with the proposed technique focusing in the combination of descriptors. As experimental results have shown, all four descriptors are good candidates to be used. However, the CLD and EHD descriptors provide greater gains in scenarios where the indexing metadata must be streamed together with the content. In any case, as the descriptors rely on different characteristics of the image (texture, color or edges), it might be possible to combine them to obtain even greater gains. New techniques to automatically select between descriptors (depending on the block being coded) will be investigated. Finally, new ways of compressing the indexing metadata will be analyzed in order to further reduce the penalty of using the proposed technique when indexing metadata must be streamed together with the content.

6. REFERENCES

[1] A. Smolic, Y. Vatis, H. Schwarz, and T. Wiegand, “Improved H.264/AVC coding using long-term global motion compensation,” in *Proceedings of Visual Communication and Image Processing*, San Jose, USA, January 2004, vol. 5308, pp. 343–354.

[2] A. Smolic, Y. Vatis, and T. Wiegand, “Long-term global motion compensation applying super-resolution mosaics,” in *IEEE Proceedings of the International Symposium on Consumer Electronics*, Erfurt, Germany, September 24-26 2002.

[3] ISO/IEC International Standard 14496-10:2003, “Information technology - coding of audio-visual objects - part 10: Advanced video coding,” 2003.

[4] P. Ndjiki-Nya, B. Makai, A. Smolic, H. Schwarz, and T. Wiegand, “Improved H.264/AVC coding using texture analysis and synthesis,” in *Proceedings of International Conference on Image Processing*, Barcelona, Spain, September 2003, vol. 3, pp. 849–852.

[5] J. Ruiz-Hidalgo and P. Salembier, “Metadata based coding tools for hybrid video codecs,” in *Proceedings of Picture Coding Symposium*, St. Malo, France, April 23-25 2003, pp. 473–477.

[6] J. Ruiz-Hidalgo and P. Salembier, “On the use of indexing metadata to improve the efficiency of video compression,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 3, pp. 410–419, March 2006.

[7] J. Ruiz-Hidalgo and P. Salembier, “Long term selection of reference frame sub-blocks using MPEG-7 indexing metadata,” in *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, Honolulu, USA, April 16-10 2007, pp. 669–672.

[8] T. Wiegand and B. Girod, *Multi-frame motion-compensated prediction for video transmission*, Kluwer Academic Plublisher, 2001.

[9] E. Kasutani and A. Yamada, “The MPEG-7 color layout descriptor: A compact image feature description for high-speed image/video retrieval,” in *Proceedings of International Conference on Image Processing*, Thessaloniki, Greece, October 2001, vol. 1, pp. 674–677.

[10] S.J. Yoon, D.K. Park, C.S. Won, and S.J. Park, “Image retrieval using a novel relevance feedback for ehd of MPEG-7,” in *Proceedings of IEEE International Conference on Consumer Electronics*, Los Angeles, USA, 2001, pp. 354–355.

[11] ISO/IEC/JTC1/SC29/WG11, “MPEG-7 overview (version 10),” N6828, 2004.

[12] J. Ziv and A. Lempel, “A universal algorithm for sequential data compression,” *IEEE Transactions on Information Theory*, vol. 23, no. 3, pp. 337–343, May 1977.