LONG TERM SELECTION OF REFERENCE FRAME SUB-BLOCKS USING MPEG-7 INDEXING METADATA

Javier R. Hidalgo and Philippe Salembier

Universitat Politècnica de Catalunya, Barcelona, Spain.
{jrh,philippe}@gps.tsc.upc.edu

ABSTRACT

Traditionally, video indexing and compression have been considered as two separate functionalities. However, the high amount of available multimedia content creates the need for multimedia services to consider both the compression and the indexing aspects of the content in order to efficiently manage it. Therefore, it is interesting to find new techniques that efficiently exploit the indexing/compression information in order to improve the compression/indexing capabilities of the content. This paper focusses on the development of one technique where the compression efficiency of the H.264 encoder is increased by the use of standard indexing information, called *indexing metadata*. This indexing metadata, even if extracted or generated to support indexing capabilities, can be exploited to enhance current standard video codecs such as H.264.

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

1. INTRODUCTION

During the last years, visual content compression and indexing have generally been considered as two separate issues. This is partially due to the fact that they support different functionalities: the main focus of video compression is to find an optimum representation in the rate-distortion sense for visualization. On the other hand, the main goal of indexing is also to find an optimum representation, called indexing metadata, but for functionalities such as search, retrieval, filtering, browsing, etc. However, future multimedia services will need to consider the compression as well as indexing aspects of the content. For that reason, exploring new representations that can, at the same time, provide functionalities of browsing and indexing may be very interesting for future applications.

In general, this line of research is rather new and few contributions have been reported in the literature. An initial contribution can be found in [1] and [2] where 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 standard [3]. A rate-distortion optimization method is used to identify macroblocks that are only affected by the global motion of the scene. For such macroblocks, no prediction error is transmitted and macroblocks are reconstructed using only the MPEG-7 *Parametric Motion* descriptor at the decoder end.

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 synthetic texture generator.

In [5], the MPEG-7 *Analytic Transition* is used to improve the coding inside transitions. The descriptor is used in order 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 type selection of frames [5]. In this case, descriptors are used to select between a set of predefined GoP structures improving the overall coding efficiency of the sequence. A tool for video segment re-ordering is presented in [6] where a high level descriptor, such as the MPEG-7 *Video Segment* descriptor, is employed in order to create a new coding order for the entire video sequence that better exploits the temporal redundancy.

The principal contribution of this paper is to study a technique where the compression efficiency of video codecs can be improved by the use of the MPEG-7 *Color Layout* descriptor. The indexing metadata, extracted for indexing functionality, is used to pre-select reference frame sub-blocks in the H.264 video encoder.

The structure of the paper is as follows. Next Section overviews the technique used to select reference frame sub-blocks. Section 3 reviews the indexing metadata used to select reference frames. Section 4 is devoted to the experimental results of the proposed technique. Finally, conclusions are drawn in Section 5.

2. LONG TERM SELECTION OF REFERENCE FRAME SUB-BLOCKS

The approach motivation is based on the high degree of temporal redundancy in real video sequences. Standard hybrid coders such as MPEG-4 or H.264 exploit this fact by using past (or future) frames as reference frames when coding the current frame. It is natural to think that the closest frame in time will be the most similar to the frame being coded. For that reason, most hybrid codecs use the closest P or I frame in time as the reference frame when coding the current frame. However, several studies have shown that using more than one reference frame can increase the coding efficiency [7]. In that case, all encoded blocks in a single frame do not share the same reference frame. For each block, information about the frame that has been used as reference frame has to be added. This implies an increment of the bitrate. However, results show that the gain in prediction error is greater than the bitrate needed to encode the reference frame and, hence, the final rate-distortion efficiency increases.

Standards such as H.263 [8] and H.264 [3] have adopted these ideas into a long term temporal prediction or multi-frame prediction scheme. In this scheme, a group of N reference frames is created for each frame to be coded and stored in a long term prediction buffer (LTPB). For each block of the current frame, the motion estimation selects the best reference block among the N possible reference frames in the LTPB. In the same sense as before, the N possible reference frames are the N closest P or I frames in the video sequence. In practice, the number N of possible reference frames 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 possible frames.. As indexing metadata have been designed for search and retrieval capabilities, the search space 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.

The LTPB is computed for each input frame being coded $\mathbf{I}(t)$. 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 with the sub-block (from past encoded frames) with minimum distance to the corresponding sub-block of frame $\mathbf{I}(t)$. In the case where $K = L = 1$ then the unique sub-block correspond to the entire image and the scheme is the same proposed in [6].

The distance between sub-blocks in the current and the previous images is computed using the indexing metadata. As the frames are sub-divided, 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. As can be seen, this strategy formulates the prediction of the current frames in two distinct steps: 1) search and retrieval of sub-blocks to create the LTPB and 2) motion compensation of sub-blocks of the LTPB.

The indexing metadata of previous frames can be extracted from the original non-coded frames or from the coded frames. If the original images are used, the indexing metadata can be previously computed and there is no need to extract new descriptions. However, if the indexing metadata is computed from the coded version of previous frames, the comparison between indexing metadata can be computed against frames that will actually be used in the decoder itself. This second approach has the advantage that sub-blocks in the LTPB are selected taking into account the coded versions. For instance, at low bitrates, where coded frames differ significantly from the original, computing the indexing metadata from coded images selects better reference frames sub-blocks. If the chosen indexing metadata is easy and fast to compute (as it will be seen in the next section), it is desirable to compute it from already coded frames.

3. SELECTED INDEXING METADATA

An indexing metadata useful for the proposed scheme is the MPEG-7 *Color Layout* descriptor. This descriptor specifies a spatial distribution of colors for high-speed retrieval and browsing. The descriptor is simple, easy to compute and a distance criterion can be used to perform similarity matches. The process to extract an MPEG-7 *Color Layout* descriptor is described in [9] and it can be summarized as follows. The input sub-block of the image is partitioned into 8×8 uniform blocks. A thumbnail image of 8×8 pixels is created by extraction of one dominant color for each of the 64 blocks. The thumbnail is transformed by a classical DCT. The final DCT coefficients (for the luminance and chrominance images) are quantized and stored. A simple zig-zag scan of the DCT coefficients is performed to create three vectors with luminance and chrominance quantized coefficients. At the last step, the coefficients are truncated so only 6 luminance coefficients and 6 chrominance coefficients (3 for each color component) are used. The final coefficients for the luminance and chrominance images are stored as the indexing metadata, in this case a *Color Layout* descriptor, of the sub-block.

The MPEG-7 standard recommends a distance measure so similarity between *Color Layout* descriptors, and therefore between sub-blocks represented by the descriptor, can be computed. The distance is computed by measuring the Euclidean distance between 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 *Color Layout* coefficients corresponding to a sub-block of a previously encoded frame. The distance $\mathcal{D}$ between sub-blocks can be measured as:

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

Even though the *Color Layout* descriptor allows storing the 64 coefficients for each Y, Cb and Cr components, the quantized coefficients are truncated and only the 12 corresponding to the lowest frequencies are stored in the representation. This ensures that the *Color Layout* descriptor is kept simple and that the distance criterion can be computed very fast without any significant penalty on similarity accuracy. The resulting distance $\mathcal{D}$ measures the similarity between two *Color Layout* descriptors. For instance, if $\mathcal{D} = 0$ then both *Color Layout* descriptors are identical and the sub-blocks represented by these indexing metadata are expected to be very similar. However, greater values of $\mathcal{D}$ represent very different indexing metadata and, therefore, sub-blocks with no similar content between them.

4. EXPERIMENTAL RESULTS

When using indexing metadata to improve the efficiency of video codecs, it is important to study if the indexing metadata will be also needed at the decoder end. In the case of the proposed technique, as the LTPB must be recreated at the decoder, the indexing metadata is also needed in the decoder end to re-fill the LTPB. This means that if the metadata is not available at the decoder or cannot be recreated, then the indexing metadata 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. Consider for instance a scenario where a user is browsing a database of movies from a video content provider. The user is able to browse the archive, search and query for different movies. During this initial search phase, the user has the opportunity to get a local copy of the indexing metadata, hence, before the actual downloading starts, both encoder (at the server side) and decoder (at the user side) have the common knowledge of the indexing metadata used in the browsing, query and searching steps. In other situations, however, the decoder will not have the indexing metadata available beforehand and it will need to be transmitted.

For the experimental results of this section, the scenario where no indexing metadata is accessible to the decoder (and cannot be recreated) is assumed. Therefore, the indexing metadata is streamed together with the content. Note that streaming the indexing metadata has the disadvantage that more bits are needed in the transmission but, on the other hand, functionalities of search and retrieval are added to the bitstream.

In order to encode the *Color Layout*, the binary representation proposed by the MPEG-7 standard is used [10]. If 12 quantized

DCT coefficients are extracted selected for each sub-block, the binary representation needs 55 bits to be stored. To exploit the similarity of the indexing metadata across frames in the same shot, all the indexing metadata of the sequence are compressed using a standard Lempel-Ziv algorithm [11] and sent to the decoder before the actual transmission of the content itself. Table 1 shows the amount of bits needed to send the indexing metadata for different sub-divisions configurations.

Sub-divisions	Size (bits/frame)	Compressed Size (bits/frame)
1×1	55	9.2
2×2	220	33.8
3×3	495	75.1

Table 1. Number of bits needed to store the 12 coefficients of the MPEG-7 *Color Layout* descriptor for a different number of sub-divisions of the frames.

As the *Color Layout* descriptor is independent of the sub-bock size, the resulting number of bits is the same for all kind of sequences and resolutions. For that reason, the penalty (in bitrate) of having to send the indexing metadata together with the content is greater for low resolutions where data bitrate is lower.

Three different configurations are studied in the experimental results. Sub-divisions of 1×1, 2×2 and 3×3 are studied to see the effect of the proposed technique compared with the standard H.264 video encoder. All results are based on the H.264 video codec reference software version JM-6.0a. The conditions and settings of the H.264 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.

Experimental tests are performed on three different sequences, *Geri*, *Telediario* and *Jornal da Noite*. As the proposed technique exploits the long term temporal redundancy between frames of the video, it should perform better on sequences with repeating shots or sequences where the same objects disappear and reappear in the scene. The sequence *Geri* corresponds to a video clip composed of 364 frames of RAW uncompressed frames at a resolution of 176×92 pixels and 30 fps. The sequences *Jornal da noite* and *Telediario* correspond to two different news programs from the MPEG-7 test set. The MPEG-7 test set is originally encoded in MPEG at CIF resolution, thus, both sequences are extracted from the MPEG bitstream and low-pass filtered and decimated to QCIF resolution in order to create test sequences with limited coding artifacts. Both sequences contain a mixture of low, medium and high motion shots with a duration of 600 frames at 5 fps.

In all tests, the LTPB size has been fixed to $N = 5$ for both the standard H.264 (legend *H.264 (LT5)*) and the proposed technique (legend *Sub-Block Selection*). Therefore, the standard H.264 fills the LTPB with the 5 previous coded frames while our proposed technique fills the 5 frames with the most similar (in terms of *Color Layout* distance) sub-blocks to the frame being coded. Fig. 1 compares the rate-distortion curves of the proposed technique and the standard H.264 encoder for the sequence *Geri*. For this test, the indexing metadata is used to compare each sub-block of the current frame being coded against all the sub-blocks of 100 previously coded frames. It can be seen how the proposed technique outperforms the standard H.264 at both the 1×1 and 2×2 configurations. Using 2×2 sub-divisions to fill the LTPB is more efficient than using 1×1 (no sub-divisions), even if the number of bits needed to send the indexing metadata is nearly four times greater. However, for the sequence *Geri*, the 3×3 sub-divisions configuration is not as efficient as the other configurations. This is due to the fact that, for the 3×3 sub-division configuration, the number of bits needed to send the indexing metadata together with the content is comparable to the amount of bits saved by using the indexing metadata to select reference frame sub-blocks. The 2×2 sub-division configuration results in PSNR gains of up to 0.5 dB compared to the standard H.264 encoder (at the same bitrate).

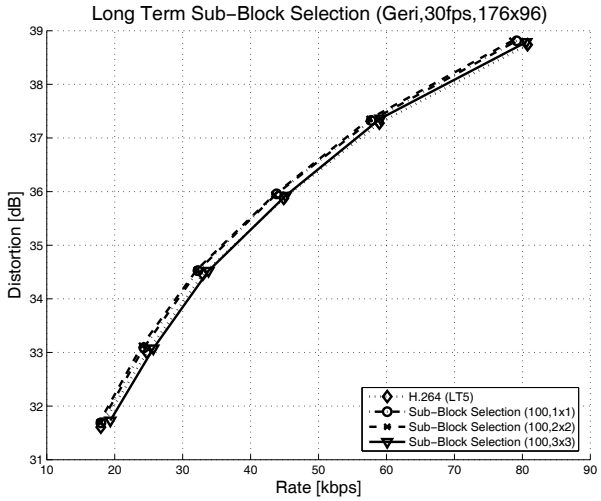

Fig. 1. Rate-distortion curves comparing the standard H.264 encoder and the proposed technique for sequence *Geri*.

Fig. 2 shows the same experimental tests but for the sequence *Jornal da Noite*. This time, the proposed technique explores the 200 previously coded frames to select the best sub-blocks using the *Color Layout* indexing metadata. For this sequence, the best sub-division configuration is, again, 2×2 with gains up to 0.7 dB compared with the standard H.264.

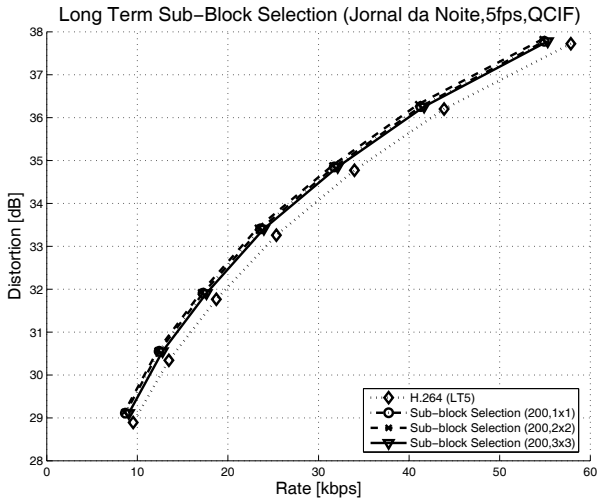

Fig. 2. Rate-distortion curves comparing the standard H.264 encoder and the proposed technique for sequence *Jornal da Noite*.

Finally, Fig. 3 shows the experimental results for sequence *Tele-*

diario using 200 previously coded frames to select the best sub-blocks. Again, the best sub-division configuration, in terms of rate-distortion efficiency, results the 2×2 one. In this case, gains up to 0.4 dB are obtained when using the proposed technique.

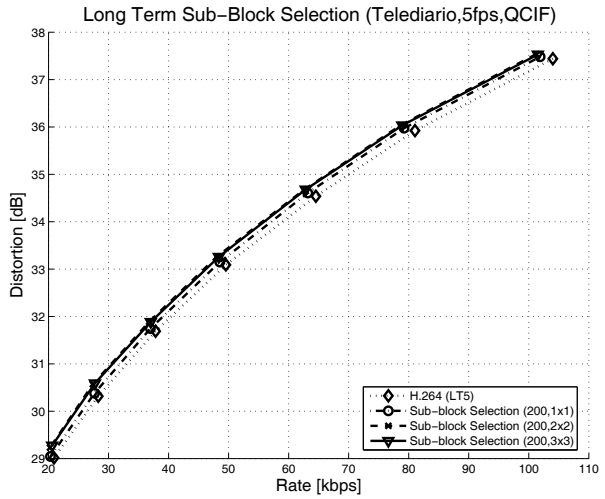

Fig. 3. Rate-distortion curves comparing the standard H.264 encoder and the proposed technique for sequence *Telediario*.

Even though up to 200 previously coded frames are used to select the reference frame sub-blocks, the increase in computational complexity in the encoder is very low. The similarity is computed using the indexing metadata and, therefore, the selection of reference frame sub-blocks is fast and efficient. Experimental results have shown that the search and retrieval step of the proposed technique is less that 1% of the computational time needed for the motion estimation itself.

However, using up to 200 frames as possible candidates in the LTPB also increases the memory requirements of the decoder, as frames in the LTPB are needed in the decoder to compensate the motion estimation. Therefore, the decoder needs to have access to all encoded frames that were used in the encoding process. If memory size in the decoder is limited, then the number of possible candidates needs to be reduced. Another possibility for memory limited decoders is to add a mark in the bitstream. The mark can signal, for each coded frame, the encoded frames that the encoder has selected to fill the reference frame sub-blocks using the *Color Layout* descriptor. The decoder could inspect those marks and, therefore, only the needed encoded frames are stored in memory. Unfortunately, this last option would require a two pass encoding strategy and thus, only useful for non-real time applications.

5. CONCLUSIONS

This paper focuses on exploiting indexing metadata to improve coding. It is important to note that the indexing metadata has been extracted to provide functionalities such as search, retrieval, browsing, filtering, etc. but not to provide better compression of the content. This point, together with the fact that indexing metadata may be already present in the video content to allow possible searches by the user makes indexing metadata a very good candidate to be exploited by current video coding standards. In this article, the MPEG-7 *Color Layout* descriptor has been employed to improve the long term prediction step of the H.264 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. The results reported show promising gains when exploiting the *Color Layout* descriptor in current standards video codecs.

Our future research will focus on performing additional experiments with different sequences and resolutions. However, as the *Color Layout* descriptor has a fixed bitrate independent of the resolution, it is expected that similar or better results will be obtained for sequences at CIF or TV resolutions. As this technique exploits the long term temporal redundancy of the sequence, only sequences with repeated shots or with objects or areas that disappear and reappear at a later time will really benefit from the proposed technique.

Finally, the authors propose to incorporate more MPEG-7 descriptors, such as for instance, texture descriptors, in the technique presented in this paper in order to evaluate its efficiency for different classes of indexing metadata.

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.

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

[8] ITU-T, "Video coding for low bitrate communication: Recommendation H.263," Version 2, 1998.

[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.

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

[11] 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.