On the use of indexing metadata to improve the efficiency of video compression

Javier Ruiz Hidalgo, *Member, IEEE,* and Philippe Salembier, *Member, IEEE,*
Universitat Politècnica de Catalunya, Barcelona, Spain.
{jrh,philippe}@gps.tsc.upc.es

***Abstract*— For the last years, video indexing and video compression have been considered as two separate functionalities. However, multimedia content is growing in such a rate that multimedia services will need to consider both the compression and the indexing aspects of the content in order to efficiently manage this audio-visual content. Therefore, it is interesting to study the synergy between the representations of compression and indexing and in particular to find new schemas that allow the possibility to exploit indexing/compression information in order to increase the efficiency of video compression/indexing capabilities. The principal contribution of this paper is to study and develop new techniques where the compression efficiency of video codecs can be improved by the use of *indexing metadata* where indexing metadata refers to information that has been generated to support indexing capabilities.**

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

I. Introduction

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. The main goal of indexing is also to find an optimum representation but for functionalities such as search, retrieval, filtering, browsing, etc. However, some tools have proved to be useful for both functionalities. This is the case for example of Mosaic representations that have been used to improve the temporal prediction for some codecs (for example in MPEG-4 [1]) as well as to support some indexing functionalities (for example in MPEG-7 [2]).

Moreover, future multimedia services will need to consider the compression as well as indexing aspects of the content. One may wonder if there exist efficient representations that address, at the same time, the compression and the indexing functionalities. In any case, there is a clear interest in studying the potential synergy between compression and indexing.

The principal contribution of this paper is to study and develop new techniques where the compression efficiency of video codecs can be improved by the use of *indexing metadata*. Through this paper, the expression **indexing metadata** refers to information that has been generated to support search, retrieval, filtering or browsing of the content. This is an issue of practical relevance because, in many situations, audio-visual material will be available together with the indexing metadata describing its content [3]. As a consequence, encoders will be able to access this information and use it in order to improve their efficiency or to optimize their strategy. Within this framework, the key point is to know whether metadata created for indexing purposes can help in the encoding process.

Note that indexing metadata can be used in various ways:

- In some situations, the encoder will make use of the indexing metadata to simply optimize its encoding strategy and the resulting bitstream can still be compatible with indexing metadata unaware decoders. That is, the decoder will not need any extra information to decode the received bitstream and will behave as classical decoders.
- In other situations, in order to fully exploit the indexing metadata, the encoder will have to severely modify its encoding strategy and, as a result, the decoding process will have to be modified as well.
- Finally, the decoder will also need the indexing metadata in order to correctly process the video content from the bitstream. In that case, the indexing metadata should be made available to the decoder either by embedding it in the bitstream or by other means.

Fortunately, there are scenarios where 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. Once the user has selected a movie and before the 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. This extra information can then be used by both the encoder and the decoder to improve the coding efficiency and no transmission of indexing metadata has to be done during the transmission of the actual content. In other situations, the indexing metadata will need to be sent together with the content. In that case, a careful bit assignment strategy between content and indexing metadata will have to be designed. Note that the availability of indexing metadata at the decoder side and the necessity to transmit it depends mainly on the application and not so much on the type of indexing metadata nor on the tool that uses this indexing metadata.

This work has been partially supported by the MASCOT project of the European Commission (IST-2000-26467) and the Revise project of the Spanish Government (TIC2001-0996)

To our knowledge, this line of research is rather new and few contributions have been reported. An initial contribution can be found in [4] 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 codecs such as H.264/AVC [5]. The Parametric Motion Descriptor is used to create a super resolution mosaic with the background information of the video sequence. Super resolution mosaics can then be used to improve the prediction step of following frames.

MPEG-7 Texture Descriptors are also being used to improve coding efficiency of current video codecs. In [6], MPEG-7 Texture Descriptors are used to signal the presence of detailed texture within the image. These texture blocks are skipped in the encoder and separately synthesized in the decoder using a synthetic texture generator.

In [3], the MPEG-7 Analytic Transition DS metadata 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. Finally, the MPEG-7 Motion Activity descriptor has also been used to improve the type selection of frames [3]. In this case, descriptors are used to select between a set of predefined GoP structures improving the overall coding efficiency of the sequence.

In this paper two new techniques will be presented that show how indexing metadata (in particular MPEG-7 metadata) can be used in order to improve the efficiency of standard video codecs. A tool for selecting reference frames in the framework of long term prediction will be proposed in section II. A tool for video segment re-ordering will be presented in section III. Both algorithms exploit the same basic concept of grouping similar references together to exploit the temporal redundancy of video sequences. However, the two tools focus on the problem from different starting points. The first tool, long term selection of reference frames, uses a low level descriptor in order to select (in the compression stage) the best possible reference for the frame currently being coded. The second tool uses a high level descriptor, such as the MPEG-7 Video Segment DS descriptor, in order to create a new coding order for the entire video sequence that better exploits the temporal redundancy. For both techniques, the scenario where the indexing metadata is not available at the decoder side is also studied. Finally, conclusions are drawn on section IV.

II. Long Term Selection of Reference Frames

This section reviews the use of indexing metadata for selecting the best possible reference frames for hybrid codecs. The selection is based on a low-level indexing metadata descriptor which selects (among a large number of previous frames) the best possible frames for the one being currently encoded.

A. Motivation

Normal video sequences have a high degree of temporal redundancy. Standard hybrid coders exploit this fact by using past (or future) frames as references for coding the current frame. Once a good candidate has been selected as reference, only the difference between the reference and the current frame is encoded and written in the bitstream. A simple translational motion model is also used to cope with the internal motion of the video sequence. Therefore, for each block to be coded in the current image, the final codec must send the motion vector information plus the difference between the current block and the reference block.

It is natural to think that the closer frame in time will be the most similar to the frame being coded. For that reason current hybrid coders use the closest P (predictive) or I (intra) frame in time as reference for coding the current frame. For each block being coded in the current frame, a motion estimation algorithm finds the best reference block only in the closest reference frame. In this case, all blocks of the current frame being coded share the same reference image (which is always the closest P or I frame in time).

Studies have shown that using more than one reference frame for encoding can increase the coding efficiency [7], [8]. The idea that the most similar frame to the one being coded is the closest in time is generally true for images but not for blocks in an image. The best possible reference for a current block being coded may be situated several frames in the past. 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 has to be added. This implies an increment of the bit-rate. However, results show that the gain in prediction error is greater than the bit-rate needed to encode the reference and so, the final rate-distortion efficiency increases.

Standards like H.263 [9] and the recent H.264/AVC [5] initiative have adopted these ideas into a long term prediction or multi-frame prediction. A group of N reference frames is created and each block to be coded selects the best reference block among these N possible reference frames. In the same sense as before, the N possible references 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: Firstly, the computational complexity of performing a motion estimation algorithm for all reference frames and secondly, the bit-rate increase needed to add the information about the reference frame.

B. Indexing Metadata Based Coding

Indexing metadata can be introduced in long term prediction to improve the overall coding efficiency. Indexing metadata can be used to perform a pre-selection of possible N candidates for reference frames. As indexing metadata has been designed for search and retrieval capabilities, the search space of possible references can be increased without severe penalty in the computational cost. Similarity based on indexing metadata can be used to pre-select N frames (that will be used as reference frames in the same manner as standard multi-frame prediction) among a high number of M possible candidate frames. If the number of reference frames N is the same as the standard long term memory buffer in the H.264/AVC coded, the bit-rate associated for the transmission of the reference frame information will be identical or very similar to the standard codec.

Moreover, indexing metadata can further help the searching of possible reference frames: for example, shot descriptors can be used so that reference frames are only searched in shots that have similar content. The computational complexity of the motion estimation step can be reduced if a lower (but more efficient) number of reference frames (selected by the indexing metadata) is used. Finally, existing indexing metadata about collections of audio-visual documents can point the encoder to similar content of other sequences. For example, a user may watch and store the news everyday at 7pm. Indexing Metadata can inform the encoder that several bitstreams corresponding to past days are already stored and available to the encoder. Although not tested in the experimental results of this paper, it is theoretically possible that the codec can make use of those streams as possible reference for coding the current sequence.

This ordering and pre-selection of previous frames to create the long term buffer can be easily re-created on the decoder side, provided the indexing metadata is also accessible to the decoder. In that case, no extra information is needed. The scenario where indexing metadata is not accessible to the decoder is considered and studied in section II-E.

C. Indexing Metadata Used

There are several indexing metadata candidates that may improve long term prediction. Consider for instance, the Color Layout indexing metadata of MPEG-7. This descriptor specifies a spatial distribution of colors for high-speed retrieval and browsing. This descriptor is simple and easy to compute and it provides a similarity criterion so frames with similar content can be recognized and included in the buffer of reference frames.

The process to extract a color layout descriptor is as follows. The input image is partitioned into 64 blocks. The $(i, j)^{th}$ block is a set of pixels whose size is approximately $\frac{W}{8} \times \frac{H}{8}$ (with W and H the width and height of the image). A thumbnail image of 8×8 pixels is created by extraction of the dominant color for each block. In our case, a simple mean for all pixels in the block has been used to extract the dominant color. The thumbnail image is transformed by a classical DCT. The final DCT coefficients (for the luminance and chrominance images) are quantized and stored. If needed, the quantized luminance and chrominance coefficients of higher frequency can be truncated and discarded. The final coefficients for the luminance and chrominance images are stored as the color layout descriptor for the entire image. Note that a mask can be used to select certain interesting regions of the image during the DCT process.

Similarity between color layout descriptors, and therefore between frames represented by these descriptors, can be easily computed by measuring the distance between DCT coefficients. If Y, Cb, Cr and Y', Cb', Cr' are the DCT coefficients of the first and second layout descriptors respectively and N_Y, N_C are the number of luminance and chrominance coefficients, the distance can be computed as:

$$D^2 = \sum_{i=0}^{N_Y-1} (Y[i] - Y'[i])^2 + \sum_{i=0}^{N_C-1} (Cb[i] - Cb'[i])^2 + \\ + \sum_{i=0}^{N_C-1} (Cr[i] - Cr'[i])^2 \quad (1)$$

D. Results

The test base for the experimental results are composed of 6 different sequences at QCIF and CIF resolutions. Table I shows the properties of the different sequences used. The sequences *Geri*, *Formula1*, *Agora* and *Fr3* correspond to RAW uncompressed sequences of CIF and QCIF resolution. In order to increase the test base, two more sequences, *Jornal da Noite* and *News11*, from the MPEG-7 test set have been used. This two sequences have been extracted from MPEG bitstream at CIF resolution and low-pass filtered and decimated to QCIF resolution in order to create the test sequences with limited coding artifacts. As it will be seen later on the article, these coding artifacts do not change the results obtained by the metadata enabled video codec.

Name	Frames	Resolution	fps	Type of content
Geri	364	QCIF	30	video clip
Jornal da Noite	600	QCIF	25	news
News11	1000	QCIF	25	news
Formula1	2500	QCIF,CIF	25	sports
Fr3	5000	QCIF,CIF	25	news
Agora	14000	QCIF,CIF	25	interview

TABLE I

SEQUENCES USED FOR THE EXPERIMENTAL RESULTS OF THE METADATA BASED VIDEO CODEC

Results on Fig. 1 show rate distortion curves for the sequence *Jornal da Noite* from the MPEG-7 test set. The color layout descriptors are used to pre-select 4 possible reference frames against the last 100 frames (the previous frame is always selected as a possible reference). Dotted line shows the rate-distortion curve for the codec using color layout metadata for long term reference frame selection. On the same figure, the solid line plots the rate-distortion curve for the baseline H.264/AVC codec using the last 5 frames as possible references (typical settings with CABAC modes switched on). Results show gains in bitrate and PSNR quality for all low to high bitrates with bitrate savings from 3% up to 12% for the same visual quality (same Q factor). The best performance in terms of bitrate savings is achieved for low bitrates.

Fig. 2 shows the same experiment for 1000 frames of the *News11* sequence. In this case, the color layout metadata was used to pre-select 4 reference frames against the previous 500. Rate-distortion curves and relative bitrate savings show gains up to 7% of bitrate savings for the same quality factor Q.

The following Fig. 3 shows the comparison between the H.264/AVC codec and the metadata modified codec for the sequence *Geri*. The H.264/AVC codec uses again 5 past reference frames while the metadata codec pre-selects 4 frames between the last 100 frames. The Figure shows gains up to 9% of bitrate savings for the same quality factor Q.

Figure 4 shows the results for the sequences *Formula1*, *Agora* and *Fr3* at QCIF resolution. The *Formula1* sequence is a sport like sequence with high internal motion and short shots. In this case the metadata enabled codec is able to select better references than the standard H.264/AVC with the previous 5 frames. In this results gains up to 8% in bitrate savings for the same quality factor Q are obtain. The *Agora* sequence

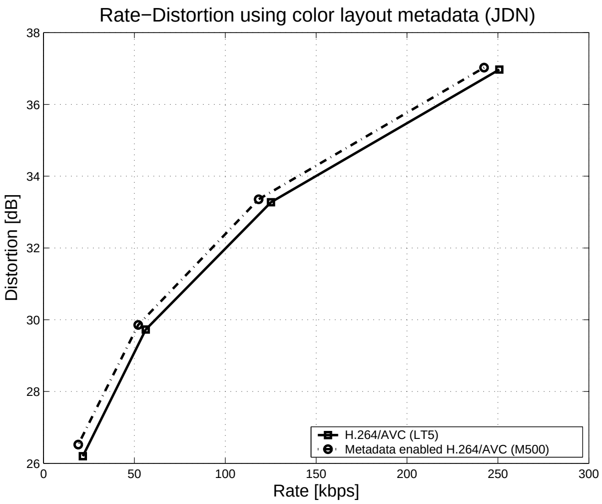

Fig. 1. Rate-Distortion curves comparing the long term selection of reference frames and the baseline H.264/AVC codec for the sequence *Jornal da Noite*. 100 frames are used to pre-select 4 possible reference frames using the color layout metadata.

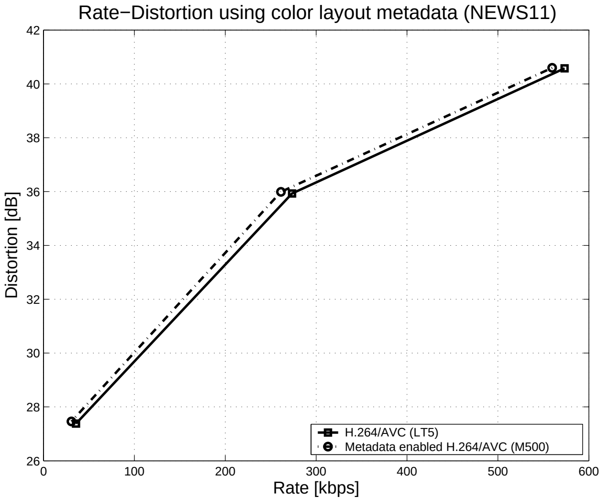

Fig. 2. Rate-Distortion curves comparing the long term selection of reference frames and the baseline H.264/AVC codec for the sequence *News11*. 500 frames are used to pre-select 4 reference frames using the color layout metadata.

corresponds to a low internal motion interview sequence. As shots are similar between the sequence, the metadata is able to locate good reference frames. The Fig. shows up to 10% gains for the *Agora* sequence for the same quality factor Q. However, for the *Fr3* sequence, the use of metadata has no improvements. This is due to the fact that this sequence has long and non-repeated shots and therefore no useful references are found by the Color Layout metadata.

Figure 5 shows the same results but for CIF resolution, similar gains are obtained when metadata is used in the H.264/AVC codec. Similar gains are obtained for the sequences *Formula1* and *Agora* than those of Fig. 4 where QCIF resolution was used.

Finally, Fig. 6 shows the same experiments as Fig. 5 but at 5fps (coding 1 every 5 frames). In this case greater gains are achieved for the *Agora* sequence (up to 15% bitrate savings). In this case, when coding at 5fps, the internal motion of the sequence is greater and therefore the standard H.264/AVC with the 5 previous references drops in efficiency. The metadata based coded, on the other hand, is still able to select good references. In the case of the *Formula1* sequence the results are very similar as the internal motion of the sequence was already high.

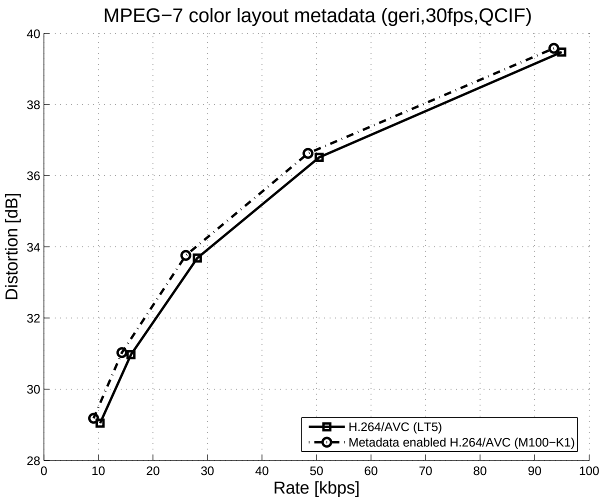

Fig. 3. Rate-Distortion curves comparing the long term selection of reference frames and the baseline H.264/AVC codec for the sequence *Geri*. 100 frames are used to pre-select 4 reference frames using the color layout metadata.

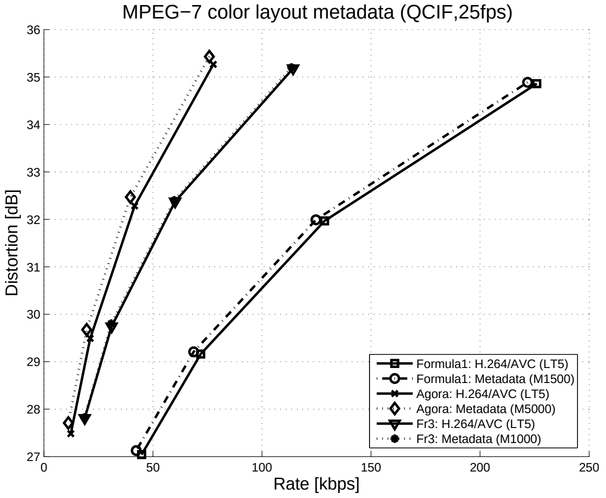

Fig. 4. Rate-Distortion curves comparing the long term selection of reference frames and the baseline H.264/AVC codec for the sequences *Formula1*, *Agora* and *Fr3* at QCIF resolution and 25fps.

As before, the *Fr3* sequence does not contain enough similar shots for the metadata to obtain references and therefore the gains are also slow at 5fps.

E. Data-Metadata Coding

The long term selection of reference frames coding scheme needs to access the indexing metadata information both at the encoder and the decoder ends. In the scenario where indexing metadata is not already available at the decoder it must be

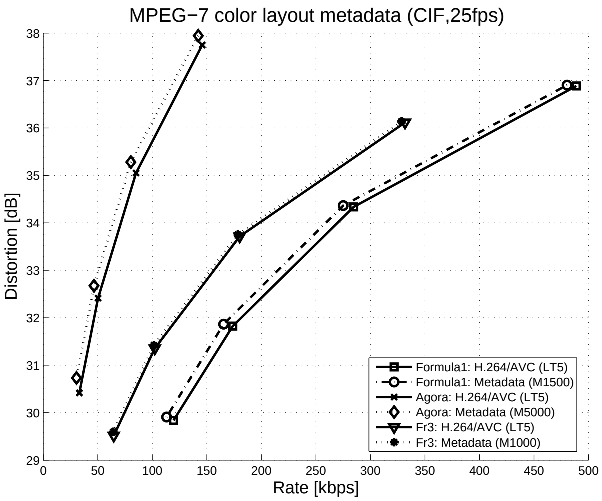

Fig. 5. Rate-Distortion curves comparing the long term selection of reference frames and the baseline H.264/AVC codec for the sequences *Formula1*, *Agora* and *Fr3* at CIF resolution and 25fps.

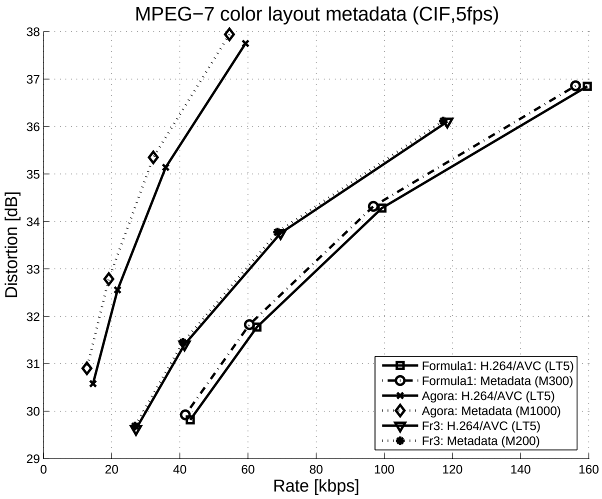

Fig. 6. Rate-Distortion curves comparing the long term selection of reference frames and the baseline H.264/AVC codec for the sequences *Formula1*, *Agora* and *Fr3* at CIF resolution and 5fps (1 every 5 frames).

streamed together with the content. The encoding of the color layout metadata is performed using the MPEG-7 binary coding. One color layout descriptor is extracted for each frame using 6 coefficients for the Y component and 6 coefficients for the Cb and Cr components. As the color layout metadata is independent of the frame size of the input image, the resulting bitrate for sending the indexing metadata information is around $11.87 bytes/frame$ without compression for all kind of sequences. In our tests, color layout metadata is compressed using a standard Lempel-Ziv coding algorithm (LZ77) which results on metadata bitrates up to $2.9 bytes/frame$.

Data (kbps)	Savings (kbps)	Indexing metadata (kbps)
40	4.4	0.58
220	8.8	0.58

TABLE II
DATA AND INDEXING METADATA RATE COMPARISONS

Table II shows the comparison of data and indexing metadata rate when the indexing metadata must be streamed with the content. These tests are performed encoding a QCIF sequence using the standard H.264/AVC video codec at low bitrates ($40kbps$) and high bitrates ($220kbps$). Each line presents the results for the same sequence at different bitrates. First column is the data rate, second column shows the savings when coding the sequence using metadata at the same quality factor as the baseline H.264/AVC codec. Third column shows the data rate needed to send the color layout metadata descriptor for this sequence.

In the case of low bitrates, sending the indexing metadata with the content represents the 1.45% of the data rate while savings the 11% of the data rate. In the case of higher bitrates, indexing metadata only represents the 0.26% of the total data rate. This figure has to be compared to the 4% of bitrate savings when using the indexing metadata in the coding process. The color layout descriptor is independent of frame size so the penalty for sending the metadata together with the content on CIF or TV sequences is even lower. As conclusions, even if the indexing metadata has to be streamed with the content, it is still useful to improve the long term selection of reference frames.

III. VIDEO SEGMENT SHUFFLING

This section describes the use of a high level indexing metadata descriptor (which describes the entire video sequence as a unit) to re-organize the coding order of the complete video in order to better exploit the temporal redundancy.

A. *Motivation*

Standard hybrid codecs exploit the temporal redundancy of video sequences by performing motion estimation between the current image being coded and a previous reference (or various previous references such as H.263 or H.264/AVC codecs). B frame types can use two reference frames in order to estimated the current frame. This references may also be situated in the future. The use of B frames has been proved to generally increase the coding efficiency, either because of the linear combination of two motion compensated references or due to the exploitation of future references where regions may not be occluded [10].

The use of B frames with previous and past references implies that the coding frame order and the display order need to be changed. Generally, the ordering is changed only by a small amount of frames (the distance between P frames). This distance is usually small for two reasons. The internal motion of the sequence makes inefficient to have references too far in time and, for real time applications, the delay introduced in the system cannot be very high.

B. *Indexing Metadata Based Coding*

In the case of non real time applications, such as stored video, it is then reasonable to think that the display order of video frames does not imply that it is the best possible order

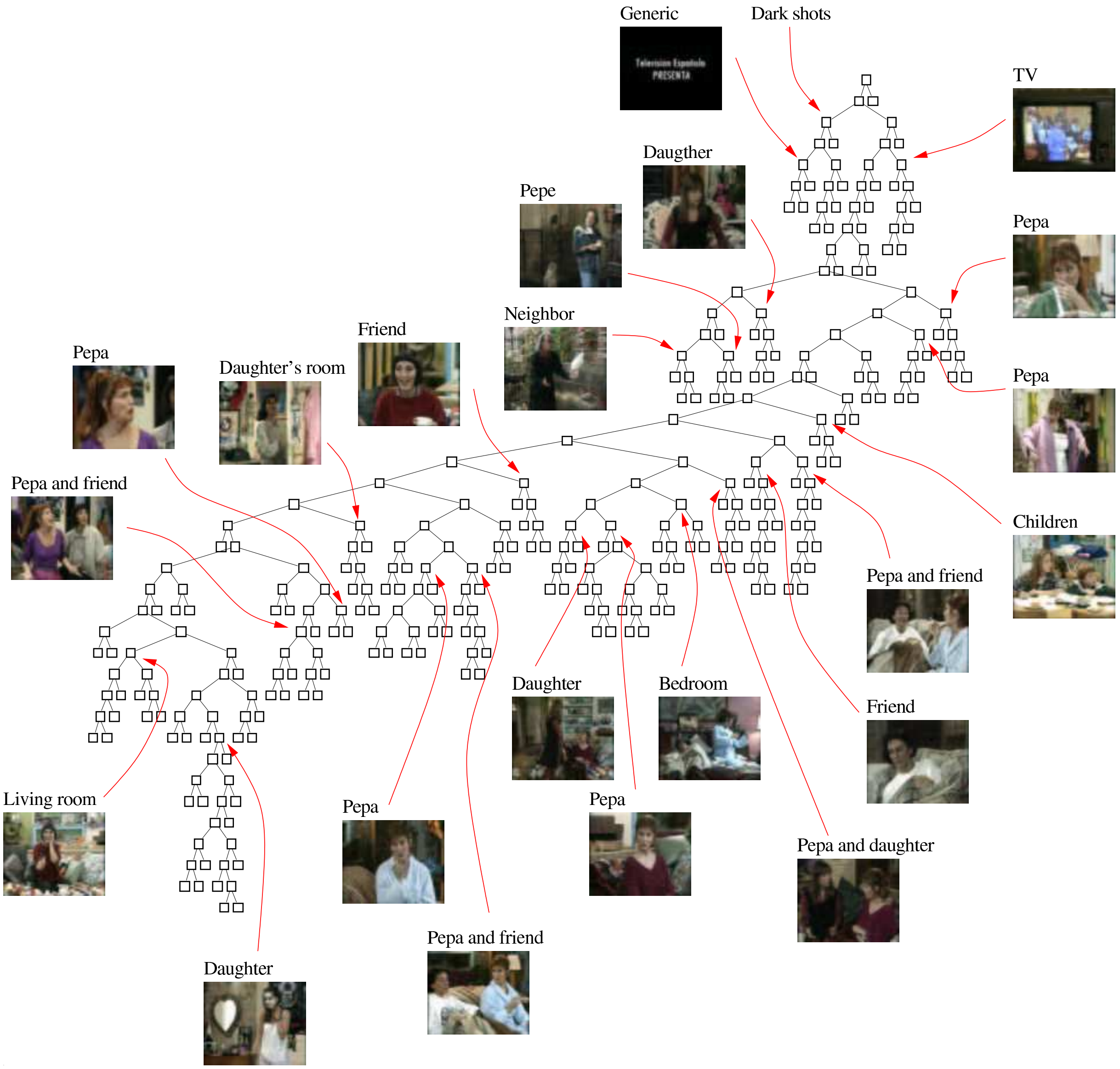

Fig. 7. Segment DS tree example of a sequence.

for coding. Therefore, the main question that this indexing metadata encoding tool addresses is: will the re-ordering of video frames prior the encoding process help to improve the coding efficiency? Obviously, finding the best possible re-ordering is a difficult task [11]. Again, existing indexing meta-data can help the decision of the coding order. Some indexing metadata, such as the MPEG-7 Segment Description Scheme (DS), makes a relationship between different video segments (usually shots of the sequence). This tree structure gives a relation between segments (which may be non-connected in time) that can be used to create a new coding order better suited to encode the entire video sequence.

C. Indexing Metadata Used

An indexing metadata useful for this purpose is the MPEG-7 Segment DS. The Segment DS is an abstract descriptor from which the specialized segment description tools are derived. The Segment DS describes the properties of segments, such as creation information, media information, usage information, semantic information, text annotation, matching hints, point of view descriptions, and so forth. In the case of the shuffling tool, a specific type of Segment DS, the Video Segment DS, is used. The Video Segment DS describes a video or a temporal segment of a video. In the case of digital video, a video segment can correspond to a single frame, an arbitrary group of frames, or the full video sequence. The video segment does not need to be connected in time. A segment can also be decomposed into several segments creating hierarchical structures to describe the content.

Using this DS, hierarchical structures describing relationships between segments of an initial temporal partition of the video can be constructed. This hierarchical tree representation describes the video content in a similar way the Table of Contents and the Index describe the contents of a book [12].

This indexing metadata is obtained by analyzing the entire video sequence and creating a hierarchical structure from an initial partition of the video sequence. This initial partition can be composed of single frames, of groups of N frames or even of video shots. Initially, all segments belonging to the initial partition are considered and a merging criterion is used to group segments that match a certain similarity criterion. Note that segments do not have to be contiguous to be merged (as in an index [12]). One by one, all segments are merged with the most similar one (in terms of the chosen criterion). The final hierarchical tree structure keeps information of all merging steps that have been done until the entire video sequence is merged.

Fig. 7 shows an example of the resulting tree when computing the Segment DS for a video sequence. In this case, the initial partition is composed of all individual shots of the sequence. The similarity criterion used for the merging algorithm is composed of a weighted distance between color similarity (using shot histograms) and motion similarity (using frame displaced difference) of the different shots that compose the initial partition (please refer to [13] for more details). As the merging criterion only merges two different regions at a time, the resulting structure is a binary tree called BPT (binary partition tree). It can be seen that the final tree is a structured representation of the input sequence. The leaves of the tree represent the different shots while the similarity between shots is represented by the tree structure.

The resulting tree that has been constructed to represent and index the video sequence can also be useful to re-order the video frames in order to improve the final coding efficiency. In the previous example, similar shots are grouped in the same branches of the tree. For instance, all dark frames are grouped into the top left branch. If a new sequence is constructed by reversing the tree from the root node, the resulting sequence contains the same number of frames as the initial one. In this case, however, similar frames (in terms of color and motion similarity) will be closer in display order. Therefore, standard hybrid codecs will then be able to code directly the modified video sequences in order to exploit the entire redundancy of the sequence instead of the local redundancy of video frames.

D. Results

It is obvious that the re-encoding of video segments will be more efficient for video sequences with a high number of similar shots scattered through the overall sequence (as the indexing metadata is able to group them).

Figure 8 shows the results of using a BPT to re-organize the coding order of the input sequence *Geri*. Three different initial segmentations where studied:

- $S = 1$ Segment size of 1 frame (frame shuffling)
- $S = 8$ Segment size of 8 frames
- $S = N$ Segment corresponding to complete shots (shot shuffling)

The H.264/AVC coder is used to code the original *Geri* video sequence and the shuffled version using the BPT. The best results are obtained using shots as initial segments. The resulting bitrate savings reach 7% for the same visual quality.

Figure 9 shows results using different B frames between P frames. In all cases, initial video segments correspond to shots only ($S = N$). As shown in the figure, the best results are obtained when $B = 1$ (one B frame between P frames).

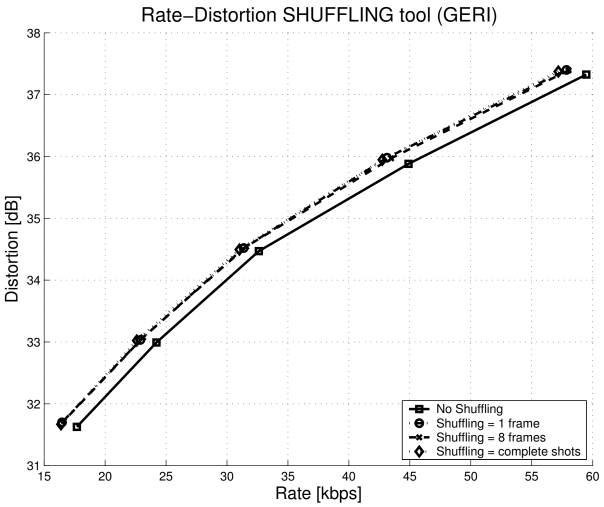

Fig. 8. Shuffling results for different cases of initial segment sizes for the test sequence Geri.

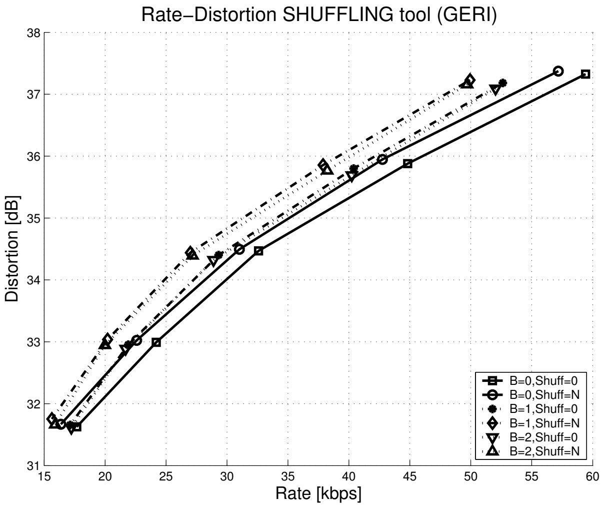

Fig. 9. Shuffling results using different number of B frames ($B = 0$), $B = 1$ (one B frame between P frames) and $B = 2$ (2 B frames between P frames) for the sequence *Geri*. Both test using video segments of complete shots.

Figure 10 shows the results for the sequence *Jornal da Noite* in the MPEG-7 database. The figure is used to study two different configurations of the H.264/AVC codec. $B = 0$ and $B = 1$ configurations are studied, but only whole shots are re-organized ($S = N$). The use of shot shuffling results on bitrate savings for both configurations. Results show lower bitrate savings (up to 3.2% at the same visual quality) than the results obtained for the *Geri* sequence. This is because *Jornal da Noite* has longer shots, so the improvement achieved by shuffling has less impact on global averages. However, the coding efficiency around the starting frames of shots improves considerably.

Figure 11 shows the same test for the *News11* sequence. In this case, the coding efficiency using indexing metadata

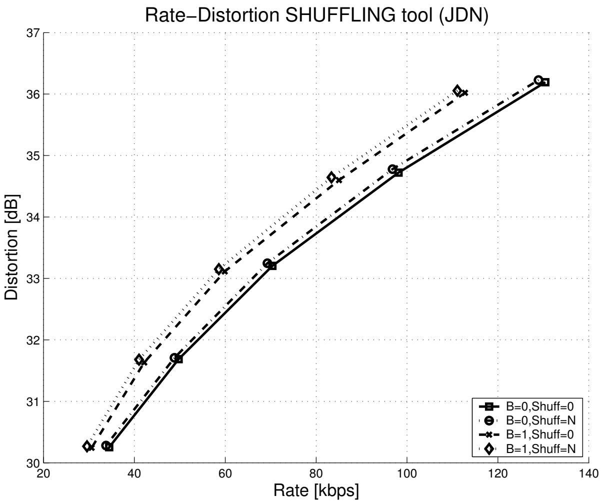

Fig. 10. Shuffling results for the sequence *Jornal da Noite* ($B = 0$ and $B = 1$) and re-ordering complete video shots ($S = N$)

is the same or sometimes worse as the one obtained without using the video segment re-ordering tool. These results can be explained by the type of test sequence. The News11 sequence has long video shots with almost no repetition. Even though, some video shots are visually similar (various football shots, for instance), the resulting shuffled sequence does not increase the coding efficiency of the hybrid codec.

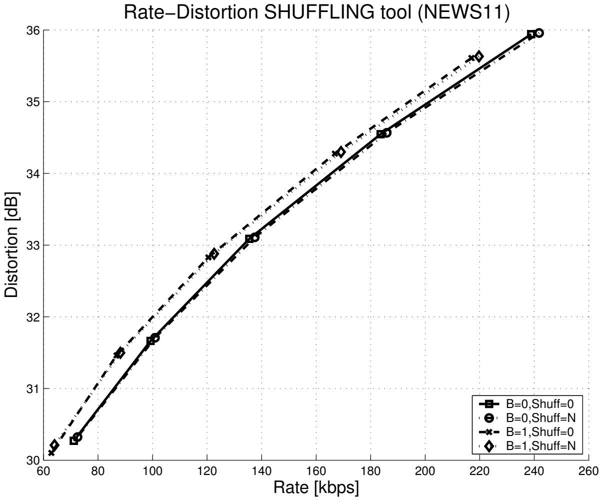

Fig. 11. Shuffling results for the sequence *News11* (with $B = 0$ and $B = 1$) and re-ordering complete video shots ($S = N$)

Tests for sequences *Formula1*, *Agora* and *Fr3* also show similar results. Fig. 12 show the encoding of the three sequences at QCIF resolution with the standard H.264/AVC codec (with $B = 0$) and the modified shuffled sequences using the BPT with $S = N$ (re-ordering of complete shots). In the case of the *Formula1* and *Agora*, the internal of the shot is captured by the BPT and the re-ordering of shots successfully an increase of the efficiency in the video codec. Again, as for the sequence *News11*, the re-ordering of the *Fr3* sequence does not improve the coding efficiency as the sequence is composed of long and non-repeating shots.

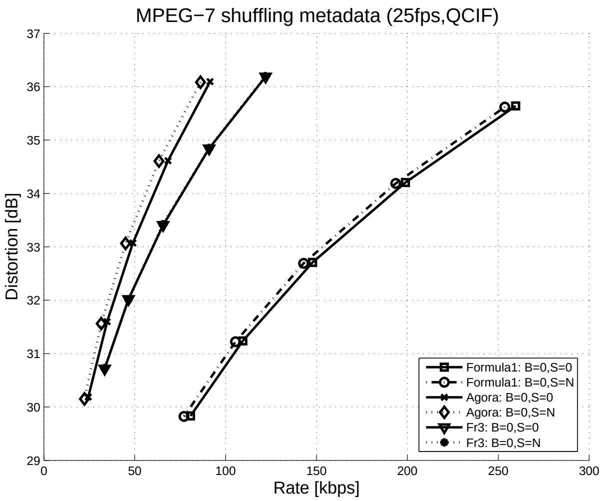

Fig. 12. Shuffling results for the sequences *Formula1*, *Agora* and *Fr3* at QCIF resolution

Fig. 13 shows the same results for the *Formula1* sequence but this time at CIF resolution. Similar gains than Fig. 12 are obtained.

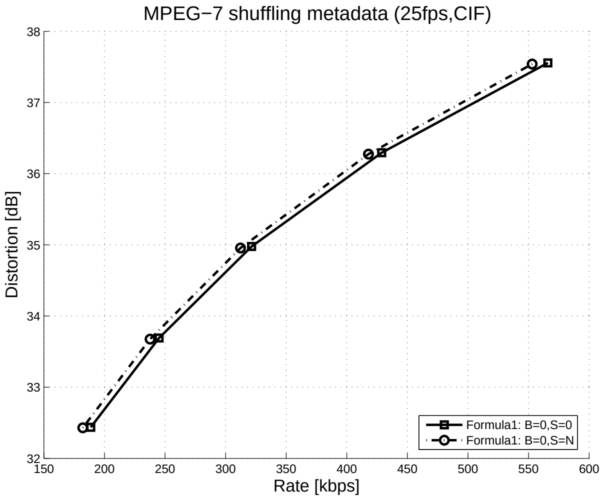

Fig. 13. Shuffling results for the sequence *Formula1* at CIF resolution

In conclusion, results for this tool show interesting gains (bitrate savings up to 8.5% for the same quality factor) in sequences that present scattered, short and similar shots (common in TV programs such as interviews, video clips, etc.). In other kind of sequences, the efficiency gain is relatively small or nothing; but segment shuffling could be a convenient way of storing a sequence, i.e. to ease the access through a BPT index. Re-ordering complete shots, $S = N$, performs better in any configuration.

E. Data-Metadata Coding

In the applications where the video segment re-ordering tool might be used, principally storing, it makes sense to have a hierarchical index of the sequence segments to allow quick access to any part of the sequence. For this reason, previous results do not include the bitrate needed to encode the hierarchical tree. However, in the case where the indexing metadata is not available, some information must be streamed with the content. As sending the entire hierarchical decomposition to only re-organize the shuffled sequence may not be very efficient. A different coding method is proposed.

In order to de-shuffle and visualize the decoded sequence in the correct order, we only need to send a list specifying the segments positions and the place where they should be moved to. For example, we have a sequence segmented as figure 14. The proposed re-ordering (following the indexing metadata information) is shown on figure 15.

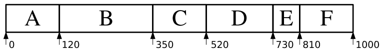

Fig. 14. Initial video segmentation of a example sequence.

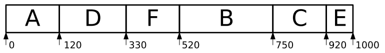

Fig. 15. Proposed re-ordering using the Video Segment DS metadata.

On the decoder side, we have to arrange the segments in the presentation order again. In order to do so, we send a list containing the location of the first segment frame before and after applying the shuffling tool (table III). Finally, after each segment is decoded, it is moved to its marked place. Segment by segment, the sequence is recovered as figure 16 shows.

Segment	Re-Ordered seq. (1^{st} frame)	Original seq. (1^{st} frame)
A	0	0
D	120	520
F	330	810
B	520	120
C	750	350
E	920	730

TABLE III
RE-ORDERING INFORMATION FOR THE SEQUENCE DESCRIBED IN 14.

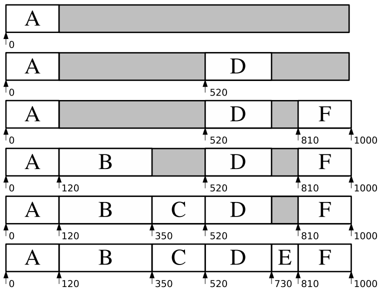

Fig. 16. Sequence for recovering the re-ordered video segments.

The amount of bits necessary to address any frame of the whole sequence is $\lceil \log_2 L \rceil$, where L is the length in frames of the sequence. Consequently, the total number of bits necessary to encode the markers list is $2N\lceil \log_2 L \rceil$(where N is the number of segments). Dividing by L, we obtain the average bits/frame that are required for this indexing metadata information:

$$\frac{2N\lceil \log_2 L \rceil}{L} = \frac{2\lceil \log_2 L \rceil}{L/N} \qquad (2)$$

The smaller L/N is, the larger would be the amount of bits required for indexing metadata. But a small L/N also means a large number of scattered small shots in the original sequence, so the bitrate savings achieved by the re-ordering tool will be higher. Additionally, if two or more contiguous segments are not separated by the shuffling tool, they can be considered as a single segment and only one position on the list is required.

If the indexing metadata to re-arrange the sequence has to be sent, the worst case to study is the lowest bitrate one, because the absolute number of bits saved is lower, but the number of bits required for encoding the indexing metadata is the same at all qualities. Table IV shows the rate needed to send data and extra information (for low bitrates) in the sequences *Geri* and *Jornal da Noite*. In the table, each line presents results at low bitrates. First column shows the rate to encode the sequence ($B = 1$). The second column shows the total savings when shuffling the sequence ($S = N$). Third column shows the total rate of sending the extra information. Last column shows the distortion difference between normal and shuffling encoding.

In the case of the sequence *Geri*, the bitrate savings are 8.5% with respect to H.264/AVC while the penalty of sending the extra information is 0.07% of the data rate. In the case of the sequence *Jornal da Noite* the penalty of sending the extra information is 0.11% with a bitrate savings of 3.3% when using the indexing metadata to re-order the input sequence.

Sequence	Data (kbps)	Savings (kbps)	Indexing Metadata (kbps)	Distortion (dB)
Geri	17.65	1.50	0.012	0.10
Jornal Noite	30.19	0.99	0.033	-0.02

TABLE IV
DATA AND INDEXING METADATA RATE COMPARISONS

IV. CONCLUSIONS

This article focuses on the synergy between compression and indexing. More precisely, we have discussed two different indexing metadata based coding techniques that exploit indexing metadata information in order to improve the efficiency of current hybrid codecs. Note here that indexing metadata refers to indexing information that has been generated to describe the content with the objective to search, query, filter and browse. In our case, we have used two MPEG-7 indexing metadata. In the first case, the MPEG-7 Color Layout descriptor has been used to improve the long term prediction step of the H.264/AVC codec. In the second case, indexes based on the MPEG-7 Segment DS have been used to reshuffle video shots and to improve the compression efficiency of motion compensated hybrid codecs. The results reported here show promising gains when enabling indexing metadata based coding into current standards video codecs.

Our future research will focus on the incorporation of more MPEG-7 descriptors (such as texture descriptors) in the metadata based coding techniques presented in this article. New coding techniques using metadata are also being studied. Moreover, different type of codecs (such as wavelet or 3D codecs) are investigated in order to benefit from the indexing metadata-based coding strategy.

References

[1] F. Pereira and T. Ebrahimi, Eds., *The MPEG-4 Book*. Prentice Hall, 2002.

[2] B. Manjunath, P. Salembier, and T. Sikora, Eds., *Introduction to MPEG-7*. Wiley, 2002.

[3] 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.

[4] 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.

[5] I. I. S. 14496-10:2003, "Information technology - coding of audio-visual objects - part 10: Advanced video coding," 2003.

[6] 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.

[7] T. Wiegand, X. Zhang, and B. Girod, "Long-term memory motion-compensated prediction," *IEEE Transactions on Circuits and Systems for Video Technology*, 1997.

[8] T. Wiegand and B. Girod, *Multi-Frame Motion-Compensated Prediction for Video Transmission*. Kluwer Academic Plublisher, 2001.

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

[10] I. T. on Circuits and S. for Video Technology, "Special issue on H.264/AVC, vol. 13, no. 7," 2003.

[11] J. Lee and B. Dickinson, "Rate-distortion optimized frame type selection for mpeg encoding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 3, pp. 501–510, June 1997.

[12] P. Salembier, J. Llach, and L. Garrido, "Visual segment tree creation for mpeg-7 description schemes," *Pattern Recognition*, vol. 35, no. 3, pp. 563–579, March 2002.

[13] J. Llach and P. Salembier, "Analysis of video sequences: Table of contents and index creation," in *Proceedings of International Workshop on Very Low Bit-rate Video*, Kyoto, Japan, October 1999, pp. 29–30.

Javier Ruiz Hidalgo received a degree in Telecommunications Engineering at the Universitat Politcnica de Catalunya (UPC), Barcelona, Spain in 1997. From 1998 to 1999, he developed an MSc by Research on the field of Computer Vision by the University of East Anglia (UEA) in Norwich, UK. During 1999 he joined the Image Processing Group at UPC working on image and video indexing in the context of the MPEG-7 standard. Since 2000, he is also pursuing a Ph.D. in the field of image processing.

Since 1999 he has been involved in various European Projects as a researcher from the Image Processing Group at UPC. During 1999 and 2000 he worked in the ACTS(AC308) DICEMAN project developing new descriptors and representations for image and video sequences. From 2001 to 2003 he is also involved in the IST/FET(2000-26467) project MASCOT developing an efficient compression scheme exploiting metadata information.

Since 2001 he is an Associated Professor at the Universitat Politcnica de Catalunya. He is currently lecturing on the area of digital signal and systems and image processing. His current research interests include image segmentation, still image and sequence coding, compression and indexing.

Philippe Salembier received a degree from the Ecole Polytechnique, Paris, France, in 1983 and a degree from the Ecole Nationale Suprieure des Tlcommunications, Paris, France, in 1985. He received the Ph.D. from the Swiss Federal Institute of Technology (EPFL) in 1991. He was a Postdoctoral Fellow at the HarvardRobotics Laboratory, Cambridge, MA, in 1991.

From 1985 to 1989, he worked at Laboratoires d'Electronique Philips, Limeil-Brevannes, France, in the fields of digital communications and signal processing for HDTV. In 1989, he joined the Signal Processing Laboratory of the Swiss Federal Institute of Technology in Lausanne, Switzerland, to work on image processing. At the end of 1991, after a stay at the Harvard Robotics Laboratory, he joined the Technical University of Catalonia, Barcelona, Spain, where he is currently professor lecturing on the area of digital signal and image processing.

His current research interests include image and sequence coding, compression and indexing, image modeling, segmentation, video sequence analysis, mathematical morphology, level sets and nonlinear filtering. In terms of standardization activities, he has been particularly involved in the definition of the MPEG-7 standard ("Multimedia Content Description Interface") as chair of the "Multimedia Description Scheme" group between 1999 and 2001.