New features and applications of the H.264 video coding standard

Cristina Gomila and Peng Yin
Thomson Inc., Corporate Research
2 Independence Way, Princeton, NJ 08540, USA
Email: cristina.gomila@thomson.net, peng.yin@thomson.net

***Abstract*—This paper provides a brief overview of the H.264 video coding standard, focusing on its new features and applications. Outperforming all the previous standards over a wide range of bit-rates, H.264 is expected to flood the market in a large number of applications ranging from the real-time conversational services, to TV broadcasting or Internet streaming. At the basis of its high performance there is: a layered structure, improving the network friendliness; the adoption of clean and simple solutions enabling efficient implementations; the capability of allowing flexible delays for a variety of services; in addition to a powerful set of tools maximizing the coding efficiency.**

I. HISTORICAL REMARK

In the domain of digital communications, video coding standards have emerged to ensure interoperability across countries, services and applications. Video coding standards are developed to provide the minimum set of tools that reaches the previous requirements at the lowest cost. Worldwide, two organizations dominate the video coding standardization processes, namely the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG). VCEG has traditionally focussed on low bit-rate video coding applications, where there is a need for high compression rates and error resilience tools. On its side, MPEG groups a larger community targeting higher bit-rates for entertainment-quality broadcasting applications.

For the last twenty years, both organizations have produced very successful standards in their respective domains. Fig. 1 presents them in chronological order. H.261, developed by the ITU-T, introduced the hybrid video coding design currently in use. A few years later, MPEG-1 improved the performance of H.261 at higher bit-rates. In a joint effort, the MPEG-2/H.262 [1] standard provided support for the first time to interlaced material and became the most commonly used video coding standard. Following MPEG-2, in 1995, H.263 [2] seemed to approach the best performance achievable with the classical hybrid coding scheme. For that reason, the ISO/IEC broadened its objectives from the pure compression gains. Along these lines, the MPEG-4 standard [3] focused on the development of new coding functionalities, MPEG-7 extended to the development of new description tools, while MPEG-21 aimed at defining a new multimedia framework. The ITU-T group continued the extension of the H.263 standard, standardizing two new versions, H.263+ in 1997, and H.263++ in 2000, focusing on compression efficiency.

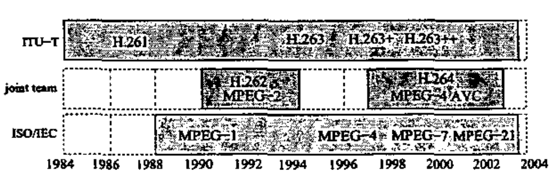

Fig. 1. Historical evolution of the video coding standards.

In parallel with these efforts, in 1997 the ITU-T started working on a new standard, known as of H.26L, in response to the demand for an efficient compression solution. From the very beginning, H.26L was intended to outperform all previous standards without forcing strong changes on the classical coding scheme. Because of its successful development, in late 2001, ITU-T and the ISO/IEC decided to join their efforts in the so called Joint Video Team (JVT). As a result of this collaboration, a single technical design has been approved in mid 2003. For the ITU-T, it constitutes a new and separate recommendation known as H.264; for the ISO/IEC, the new set of video coding tools is integrated as a new part (Part 10) of the MPEG-4 standard, referred as MPEG-4 AVC (Advanced Video Coding).

II. OVERVIEW OF THE STANDARD GOALS

As stated in its Specifications [4], the H.264 standard has been designed to be generic and serve a wide range of applications, bit-rates, resolutions, qualities and services. Within the JVT Terms of Reference (ToR) [5], the following requirements were imposed:

- Simplification *back-to-basics* approach, with the adoption of simple and clean solutions using well-known building blocks. Avoid any excessive quantity of optional features or profile configurations.
- High compression performance, having the capability of 50% or greater bit-rate savings compared to previous standards at all bit-rates, for a similar degree of encoder optimization.
- Improved network friendliness to ease packetization, information priority control and application to video streaming services.
- Enhanced error and packet loss resilience tools.
- Flexible application to delay constraints appropriate to a variety of services: low delay for real-time conversational services; higher delay to optimize compression for storage or server-based applications.

To achieve these requirements, H.264 has been designed in a layered structure. The Video Coding Layer (VCL), presented in Section III efficiently represents the video content. Its goal is to improve the coding efficiency and promote simple solutions. Furthermore, to enable efficient implementations, a reduced number of profiles and levels have been defined, specifying subsets of the algorithmic features and degrees of capability. We will present both of these concepts in Section IV. The Network Abstraction Layer (NAL), presented in Section V, provides the appropriate syntax to packet the VCL for its conveyance by the transport layers or storage media. Its goal is to improve the network friendliness and flexibility. Finally, we devote the last section of this paper to map the different profiles with the most common applications targeted by the standard.

III. VIDEO CODING LAYER

The Video Coding Layer (VCL) is specified to efficiently represent the content of the video data. The strength of H.264 relies on the

0-7803-7724-9/03/$17.00 © 2003 IEEE

achievement of high-performance video coding on a classical hybrid coding configuration, as illustrated in Fig. 2.

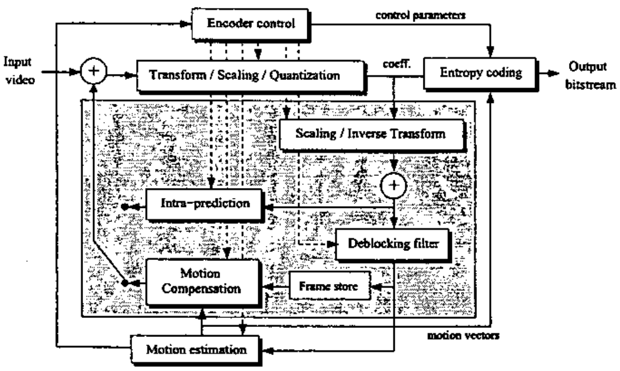

Fig. 2. H.264 encoder scheme (non normative).

Further on, we focus on presenting the new features in the VCL. We first briefly discuss the picture and slice coding types in H.264, then we talk about new coding features affecting the blocks and macroblocks within the picture.

The H.264 standard has been defined to encode both progressive and interlaced sequences in YUV 4:2:0 format. It supports four picture coding modes, as shown in Fig. 3. In frame-based coding, a picture is created by interleaving both top and bottom lines. In field-based coding, a picture is created by adding in a sequential order first the top and then the bottom field. At the picture level, adaptive frame/field coding allows the encoder to select among one of the previous two coding modes. The last coding mode is MB-adaptive frame/field coding, which makes it unique from other coding standards. In this mode, the frame is scanned as MB pairs. For each MB pair, the coding type frame/field is selected.

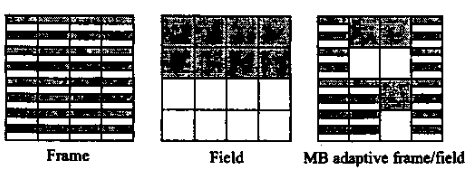

Fig. 3. Picture coding modes within H.264.

At a lower level, *slices* are the smallest self-contained video coding units. Slices are composed of an integer number of continuous MBs or MB pairs, in raster scan order. Within a slice, MBs are coded with inter-dependence. In H.264, five slice coding types are defined. The first type is Intra (I), which can predict only from decoded samples within the same slice. The second is Predictive (P), where inter prediction is performed from previously decoded pictures with at most one motion vector per block. The third is Bi-predictive (B), where inter prediction is conducted from previously decoded pictures with at most two motion vectors per block. The last ones the Intra and the Predictive Switching pictures, SI and SP, which allow identical reconstruction of slices even when different reference frames are being used.

A. Intra-prediction coding

In H.264, if a block or MB is encoded in intra mode, a prediction block is formed based on previously encoded and reconstructed surrounding available samples. The prediction block is subtracted

TABLE I
SLICE CODING TYPES.

Slice coding types	Allowed MB prediction types
Intra (I)	Intra
Predictive (P)	Intra, Predictive
Bi-Predictive (B)	Intra, Predictive, Bi-Predictive
S-Intra (SI)	S-Intra, Intra
S-Predictive (SP)	Intra, S-Predictive

from the current block prior encoding. For luma intra prediction, the prediction block may be formed either for a 4x4 block or for a 16x16 MB: (1) by means of a single prediction for an entire 16x16 MB 4 modes (vertical, horizontal, DC, plane) are available; (2) by means of 16 individual predictions on 4x4 blocks 9 modes (DC + 8 directional) are available. Chroma intra prediction supports a single prediction type for each 8x8 region. In this case, only 4 modes (vertical, horizontal, DC, plane) are available.

B. Transform modes

Unlike previous standards such as MPEG-1, MPEG-2, MPEG-4 and H.263 which use the 8x8 DCT as the basic transform, H.264 supports three transforms modes depending on the type of residual data that is to be encoded. (1) 4x4 residual transform for all residual data: the transform is based on the DCT but it is modified to be an integer transform, so no mismatch occurs between encoders and decoders. (2)4x4 luma DC coefficient transform for 16x16 intra-mode only: after transforming each 4x4 residual block, the resulting set of DC coefficients is then transformed by a 4x4 Hadamard transform. (3) 2x2 chroma DC coefficient transform: a 2x2 Hadamard transform applies to the DC coefficients after 4x4 residual transform.

C. Motion estimation/compensation

H.264 uses block-based motion compensation, the same principle adopted by other major coding standards. The major difference from earlier standards includes the support of multiple references, variable block sizes, fine resolution with $1/4$ pixel precision and extension of the B pictures.

The MBs in the current picture can be predicted from multiple reference pictures [7]. Multiple reference pictures are most helpful for sequences with chaotic motion. This new features mainly increases the computational burden at the encoder, while requires an additional usage of memory. Much of the gain from multiple reference pictures is achieved by using 2-5 reference frames. Using 5 reference frames for prediction can yield $5-10\%$ in bit-rate savings as compared to using only 1 reference frame.

H.264 supports tree-structured hierarchical MB partitions, as illustrated in Fig. 4. Inter-coded 16x16 pixel MBs can be broken into MB partitions, of sizes 16x8, 8x16, or 8x8. The 8x8 partitions, also known as sub-MBs, can be further broken into sub-MB partitions, of sizes 8x4, 4x8, and 4x4. The choice of the partition size has a significant impact on compression efficiency. In general, a large partition size is appropriate for homogeneous areas, while small partition sizes are beneficial for detailed areas.

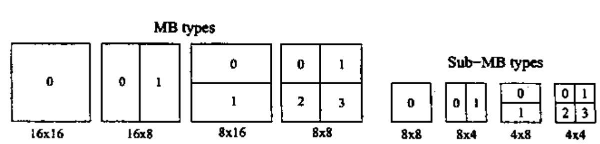

Fig. 4. MB and sub-MB partitions within the H.264 video coding standard.

Each partition in an inter-coded MB is predicted from an area of the same size in a reference picture, indicated by a motion vector. H.264 supports $1/4$ pixel precision for motion vectors, which outperforms both the integer and the half pixel compensation, at the expense of increased complexity.

Similar to other major standards, H.264 supports both P and B pictures. The largest difference lies in B pictures, which have more flexibility and better coding gain. First, H.264 adopts a new prediction mode, "direct mode" for B pictures, which does not require the coding of motion vectors. Second, B pictures can be used as reference pictures. Third, Bi-prediction can be formed from both the same direction or different direction. Accordingly, H.264 changes the name B as Bi-predictive instead of Bi-directional as in previous standards.

D. *In-loop deblocking filter*

The deblocking filter is computed within the decoding loop, as illustrated in Fig. 2. It is fully defined by the standard, and must be computed at both the encoder and the decoder. As an in-loop filter, it improves the quality of the decoded picture prior to motion estimation, showing a performance significantly superior to post-filters. The deblocking filter operates on the horizontal and vertical edges between blocks of 4x4 pixels inside the prediction loop to remove artifacts caused by quantization.

Being highly content adaptive, the filtering procedure mainly removes blocking artifacts and does not unnecessarily soften the visual content. At the slice level, the global filtering strength is adjusted to the individual characteristics of the video sequence. At the edge level, the filtering strength is made dependent on the inter/intra mode decision, the motion vectors, and the coded residuals. Finally, at the sample level, the quantizer dependent thresholds can turn off the filtering process for every individual sample.

E. *Entropy coding*

In contrast to other coding standards, two entropy coding schemes, namely the Content Adaptive Binary Arithmetic Coding (CABAC) and Content Adaptive Variable Length Coding (CAVLC), are supported in H.264, which is the outcome of a tradeoff between complexity and coding efficiency. CABAC provides higher coding efficiency than CAVLC, but at the expense of higher complexity.

In CABAC [8], context modeling provides conditional probabilities by utilizing context models and inter-symbol redundancy. Arithmetic codes combines code-words to permit non-integer number of bits to be assigned to each symbol. This is especially helpful for symbol probabilities much greater than 0.5, which occur often when context modeling is used. A fraction of a bit can be used in this case versus minimum 1 bit for VLC. Adaptivity is achieved by taking into account the cumulative probabilities of already coded symbols, which leads to a better fit of the arithmetic codes to the current symbol statistics. For example, statistics may vary for different sequences and for different bit-rates.

CAVLC was proposed to improve the performance of the VLC without incurring the complexity of CABAC. CAVLC adds context models only to the transform coefficient coding. Other elements, such as motion vectors, MB types, etc., are coded with Exp-Golomb codes (VLC). CAVLC coding separates runs and levels, which allows better adaptivity and thereby better coding efficiency. It also results in low complexity with low demand on memory. Zig-zag scanning is still used, but in the transmission of the coefficient data (levels and runs), the scanning is done in reverse order. To improve efficiency, CAVLC switches between a reduced number of VLC tables. The selection takes into account the statistics in the previously coded blocks.

IV. Profiles and Levels

The H.264 standard has defined three different profiles: the Baseline, the Main profile and the Extended profile. However, unlike previous standards, profiles in H.264 are not fully hierarchical. The Baseline is a subset of the Extended profile. However, some tools in Baseline are not present in Main profile. We have summarized the set of tools available for each profile in Table II. In this section, we present only the new features unique to different profiles.

TABLE II

SELECTION OF ALGORITHMIC FEATURES ACCORDING TO THE PROFILE.

Visual tools	Baseline	Main	Extended
1/4 pixel precision	•	•	•
VLC entropy coding	•	•	•
CABAC entropy coding		•	
I pictures	•	•	•
P pictures	•	•	•
B pictures		•	•
SI/SP pictures			•
Flexible Macroblock ordering	•		•
Arbitrary slice ordering	•		•
Data Partitioning			•
Weighted Prediction		•	•
Interlace picture-level adaptation		•	•
Interlace MB-level adaptation		•	
Redundant pictures	•		•

A. *Baseline*

Baseline supports some low-delay error resiliency tools combined with a subset of relatively low complexity coding efficiency tools.

To reduce the error effect and facilitate error recovery, Flexible Macroblock Ordering (FMO) is supported, whereby the sender can transmit MBs in non-raster scan order. A MB allocation map is generated to group the MBs. In some sense, FMO is very similar to block interleaving, but with more flexibility. Another new error resilient feature is arbitrary slice order (ASO). If used, slices may not be coded following the raster-scan order.

B. *Main profile*

Main profile supports the tools which maximize coding efficiency, such as B frames and CABAC. Furthermore, for the first time, a Weighted Prediction (WP) tool is incorporated into a video coding standard. WP improves coding efficiency by applying a multiplicative weighting factor and an additive offset to the motion compensated prediction to form a weighted prediction, which is particular useful for fading sequences. Two methods are provided for the WP tool. In the *explicit* mode (P and B slices), a weighting factor and offset may be coded in the slice header for each allowable reference picture index. In the *implicit* mode (B slices), the weighting factors are not coded but are derived based on the relative Picture Order Count (POC) distances of the two reference pictures.

C. *Extended profile*

The extended profile includes tools to support coding efficiency, error resiliency and re-synchronization.

A new picture type referred as *switching* pictures (SI or SP) is incorporated into H.264. An SP picture makes use of motion compensated predictive coding to exploit temporal redundancy in the sequence similar to P pictures. The difference between SP and P pictures is that SP pictures allow identical picture reconstruction even when they are predicted from different reference frames. SI is similar to I and has the same features as SP.

Applying to all the previous profiles, H.264 also defines a set of levels and sub-levels, which establish different degrees of capability. According to the maximal frame size a level can support, levels are sorted as follows: level 1–QCIF; level 2–CIF; level 3–SDTV; level 4–HDTV; level 5–Digital Cinema. Note that decoders supporting a given level shall also be capable of decoding bit-streams using all lower levels.

V. Network Abstraction Layer

The Network Abstraction Layer (NAL) is specified to efficiently represent the encoded video sequence, allowing an easy integration into a variety of protocol and multiplex architectures [10]. Fig. 5 illustrates the relationship between the NAL and the transport layer. The NAL provides both header information and the appropriate format for the conveyance of encoded video data by transport layers or storage media. The new concepts introduced by the NAL are related to the the definition of NAL units and parameter sets, to which we devote the next two sections.

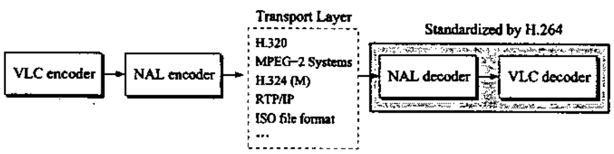

Fig. 5. Relationship between the NAL and the transport layer.

A. NAL units

According to the NAL specification, all the encoded data must be contained in NAL units. A NAL unit is a syntax structure consisting of a one-byte header and a raw byte sequence payload (RBSP). NAL units can be directly used as payload in packet-based networks, or can be mapped into a bit-stream oriented transport layer. NAL units are self-contained in the sense that they are independently decodable. Only their size must be conveyed externally. Depending on the encoded payload, the standard has defined up to thirteen different NAL unit types, among those we would like to introduce the following:

Coded Slice This NAL unit conveys all the information required for decoding a slice. This concerns both slice header information and encoded data.

Coded Data Partition When data partitioning is used, the MB data of a slice is organized in three partitions: partition A contains header information as well as motion vectors; partition B contains intra texture information; and partition C contains inter-coded texture. According to the standard, each partition is conveyed in its own NAL unit, namely: DPA, DPB, and DPC. This should allow, for example, the DPA transport to be done with higher QoS than DPB and DPC.

Instantaneous Decoder Refresh An IDR picture is a special I or SI picture delivered as a decoder refresh point. It re-initializes the Picture Order Count (POC), resets the decoded pictures buffer, and guaranties that all later coded pictures can be decoded without inter prediction from any picture decoded prior to the IDR picture.

Supplemental Enhancement Information SEI messages are delivered into a NAL unit synchronous with the video data content to assist mainly in the processes of decoding and display. However, SEI is not limited to this purposes and can transmit any arbitrary data. Note that SEI is not required to decode VCL data correctly, and decoders are not required to process this information for conformance with the standard. Examples of SEI messages are: display information, control information, error resilience issues, etc.

Sequence and Picture Parameter Set SPS and PPS NAL units have been defined to convey all information relevant to more than one slice. Because of the novelty of this concept, we devote the next section to its presentation.

B. The Parameter Sets

With the layered nature of recent video coding standards, the loss of a packet containing header information renders useless all the following packages with related data. To address this problem, several strategies have been designed. One strategy consists in checking for packet integrity at the decoder end and invoking error concealment when missing or corrupted packages are detected. The main drawback of such approach, however, concerns the difficulty of predicting header changes. A more robust strategy proposes the use of error resiliency tools when the transmitter is aware of possible header losses. However, this is done at a high cost in both bit-rate and design complexity.

To overcome the previous drawbacks, the H.264 standard takes a new approach proposing to convey asynchronously all information relevant to more than one slice in nested Parameter Set NAL units. At this stage, the slice layer was identified as the appropriate smallest self-contained unit (unless data partitioning is used) because the size of the slice can be adjusted to the MTU size of the most demanding system. The slice header contains a reference to a Picture Parameter Set (PPS) to be used for decoding of its VLC data, and some parameters that change dynamically. At its turn, each PPS references a Sequence Parameter Set (SPS), which contains information that varies slowly along a video coding session. Fig. 6 illustrates an example.

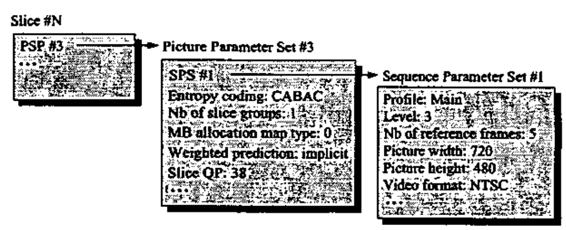

Fig. 6. Asynchronous transmission of nested parameter sets.

The advantages of such approach are twofold: on the one hand, it decouples the transmission of MB data from relevant header information; and on the other hand, it allows using different logic channels or even a different out-of-band protocols to convey reliably the most valuable information. Typically, establishment could be done during the capability exchange or in a session announcement. However, for those applications where no control protocol is available, special NAL unit types are specified to setup and update the SPS and the PPS in-band. Note that the use of in-band and out-of-band PS transmission is mutually exclusive.

C. Network Adaptation

Although the mapping of the encoded data into transport networks is not specified by the standard, it is within its terms of reference to provide appropriate mechanisms, syntax and interfaces to facilitate the gateway design.

Hence, unlike previous video coding standards, H.264 has taken into account the transmission over packet networks in the video codec design from the very beginning. Systems using packet networks as the RTP/IP can transmit NAL units directly, by using them as payloads.

Similarly, for all transport protocols that are not packet-based, such as H.320 for video-conferences, or MPEG-2 systems for broadcasting, the H.264 standard defines a byte-stream format to transmit a sequence of NAL units as an ordered stream of bytes. A byte-oriented framing is defined to identify the NAL unit boundaries and allow the correct decoding: NAL units are encapsulated by start codes of 1 or 2 bytes. Since H.264 contains two different entropy coding modes, an start-code-emulation-free environment would be very difficult to achieve. Instead, an emulation prevention code is specified by using byte stuffing.

VI. Applications

The H.264 standard is intended to cover a large range of applications including, but not limited to, the following: CATV, Cable TV on optical networks, copper, etc.; DBS, Direct broadcast satellite video services; DSL, Digital subscriber line video services; DTTB, Digital terrestrial television broadcasting; ISM, Interactive storage media (optical discs, etc.); MMS, Multimedia messaging service; MSPN, Multimedia services over packet networks; RTC, Real-time conversational services (video-conference, videophone, etc.); RVS, Remote video-surveillance; SSM, Serial storage media (digital VTR, etc.).

Furthermore, requirements from the most typical applications have been considered in the creation of the three different profiles.

A. Real-time conversational services

These set of applications are expected to adopt the Baseline profile of the H.264 standard, in replacement of the H.263 standard currently in use. The Baseline has been conceived to satisfy all the needs of such kind of applications: it includes a powerful set of error resiliency tools; allows low complexity implementations by avoiding the handling of interlaced coding modes; and ensures low-delay processing. Furthermore, the Baseline profile should benefit from a royalty-free policy, as demanded by the ITU-T community in the ToR [5].

B. Television broadcasting

The Main profile of the H.264 standard targets entertainment-quality broadcasting applications. This profile achieves a maximal compression performance by using B frames and CABAC, at the cost of higher delays and increasing complexity. No error resilience tools are included in this profile. Its key features are the adaptive weighted prediction, which is specially suited for fading sequences, and the MB frame/field adaptive coding mode, which gives bit-rate savings up to 15% depending on the content. It is expected the Main profile replaces progressively the MPEG-2 standard, since it performs up to 3x better than MPEG-2 at the same quality (PSNR). Similarly, there is an important quality improvement when sequences are encoded at the same bit-rate. Figure 7 illustrates an example on the *Football* sequence, encoded at 2.3Mbps in both MPEG-2 and H.264. For a detailed analysis about the performance of H.264, when compared to previous standards, we refer the reader to [11].

C. Internet streaming and communications

Despite of previous standards, H.264 has been conceived to be transmitted over packet-based networks, providing a seamless and easy integration of the coded video into all current protocols and multiplex architectures. Typical applications include fixed and wireless video transmission over the Internet Protocol (IP), as for the third generation of mobile systems [12]. In this field, the Extended profile of the H.264 standard combines successfully coding efficiency with an enhanced set of error resilience tools. However, and as for all the previous applications, extensive research needs to be conducted in order to exploit all the new capabilities of the standard.

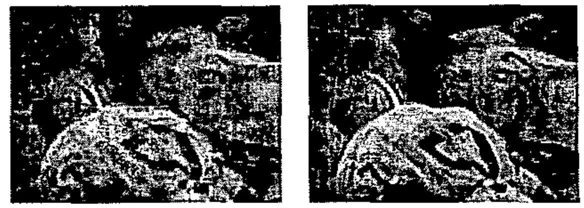

Fig. 7. Visual quality comparison between MPEG-2 (left) and H.264 (right) when encoding at the same bit-rate. Crop on a player of the *Football* sequence.

VII. Summary

In this paper, we have presented a brief overview of the H.264 video coding standard. We started with a historical remark following the evolution of previous standards until the emergence of H.264. Then, we presented the key features of the standard, differentiating those within the Video Coding Layer from the new concepts introduced by the Network Abstraction Layer. Finally, we described the mapping between the set of profiles and the most typical applications targeted by the standard.

References

[1] MPEG-2, *Information technology – Generic coding of moving pictures and associated audio information: Video*, ISO/IEC 13818-2, 2d Edition, 2000.

[2] H.263, *Video Coding for Low Bit Rate Communication*, ITU-T Recommendation, Version 1: Nov. 1995. Version 2: Jan. 1998. Version 3: Nov. 2000.

[3] MPEG-4, *Coding of Audio Visual Objects – Part 2: Visual*, ISO/IEC 14496-2:2001, Version 1: Jan. 1999. Version 2: Jan. 2000. Version 3: Jan. 2001.

[4] H.264, *Draft ITU-T Recommendation and Final Draft International Standard*, Geneva, Switzerland, 2003.

[5] H.264, *Terms of Reference for the Joint Video Team (JVT) Activities*, downladable from http://www.itu.int/ITU-T/studygroups/com16/jvt/JVTTOR.pdf.

[6] G. Sullivan and T. Wiegand (ed.), *Special issue on H.26L/JVT coding*, IEEE Transactions on Circuits and Systems for Video Technology, to appear in September 2003.

[7] T. Wiegand et al. *Lont-term memory motion-compensated prediction*, IEEE Transactions on Circuits and Systems for Video Technology, vol 9, no. 1, pp. 70-84, Feb. 1999.

[8] D. Marpe et al., *Context-Based Adaptive Binary Arithmetic Coding in JVT/H.26L*, invited paper in Proc. IEEE International Conference on Image Processing, ICIP'02, Vol. II, Rochester, USA, 2002.

[9] S. Wenger and M. Horowitz, *Flexible MB Ordering – A new error resilience tool for IP-based video*, Proc. International Workshop on Digital Communications, IWDC 2002, Capri, Italy, 2002.

[10] T. Stockhammer et al., *H.26L/JVT coding Network Abstraction Layer and IP-based transport*, Proc. IEEE International Conference on Image Processing, ICIP'02, Rochester, USA, 2002.

[11] A. Joch et al., *A performance analysis of the ITU-T Draft H.26L video coding standard*, 12th International Packet Video Workshop, PV2002, Pittsburgh, PA, USA, April, 2002.

[12] T. Stockhammer et al., *JVT/H.26L in 3G Wireless Environments*, Proc. IEEE International Conference on Image Processing, ICIP'02, Rochester, USA, 2002.