The MPEG-4 Video Standard Verification Model

Thomas Sikora, *Senior Member, IEEE*

***Abstract*— The MPEG-4 standardization phase has the mandate to develop algorithms for audio-visual coding allowing for interactivity, high compression, and/or universal accessibility and portability of audio and video content. In addition to the conventional "frame"-based functionalities of the MPEG-1 and MPEG-2 standards, the MPEG-4 video coding algorithm will also support access and manipulation of "objects" within video scenes.**

The January 1996 MPEG Video group meeting witnessed the definition of the first version of the MPEG-4 Video Verification Model—a milestone in the development of the MPEG-4 standard. The primary intent of the Video Verification Model is to provide a fully defined core video coding algorithm platform for the development of the standard. As such, the structure of the MPEG-4 Video Verification Model already gives some indication about the tools and algorithms that will be provided by the final MPEG-4 standard. The purpose of this paper is to describe the scope of the MPEG-4 Video standard and to outline the structure of the MPEG-4 Video Verification Model under development.

***Index Terms*— Coding efficiency, compression, error robustness, flexible coding, functional coding, manipulation, MPEG, MPEG-4, multimedia, natural video, object-based coding, SNHC, standardization, synthetic video, universal accessibility, verification model, video coding.**

I. Introduction

THE ISO SC29 WG11 "Moving Picture Experts Group" (MPEG), within ISO SG 29 responsible for "coding of moving pictures and audio," was established in 1988 [1]. In August 1993, the MPEG group released the so-called MPEG-1 standard for "coding of moving pictures and associated audio at up to about 1.5 Mb/s" [2], [3]. In 1990, MPEG started the so-called MPEG-2 standardization phase [3]. While the MPEG-1 standard was mainly targeted for CD-ROM applications, the MPEG-2 standard addresses substantially higher quality for audio and video with video bit rates between 2 Mb/s and 30 Mb/s, primarily focusing on requirements for digital TV and HDTV applications.

Anticipating the rapid convergence of telecommunications industries, computer, and TV/film industries, the MPEG group officially initiated a new MPEG-4 standardization phase in 1994—with the mandate to standardize algorithms for audio-visual coding in multimedia applications, allowing for interactivity, high compression, and/or universal accessibility and portability of audio and video content. Bit rates targeted for the video standard are between 5–64 kb/s for mobile applications and up to 2 Mb/s for TV/film applications. Seven new (with respect to existing or emerging standards) key video coding functionalities have been defined which support the MPEG-4 focus and which provide the main requirements for the work in the MPEG Video group. The requirements are summarized in Table I and cover the main topics related to "content-based interactivity," "compression," and "universal access." The release of the MPEG-4 International Standard is targeted for November 1998 [3], [4].

1) Content-Based Interactivity: In addition to provisions for efficient coding of conventional image sequences, MPEG-4 will enable an efficient coded representation of the audio and video data that can be "content-based"—to allow the access and manipulation of audio-visual objects in the compressed domain at the coded data level with the aim to use and present them in a highly flexible way. In particular, future multimedia applications as well as computer games and related applications are seen to benefit from the increased interactivity with the audio-visual content.

The concept of the envisioned MPEG-4 "content-based" video functionality is outlined in Fig. 1 for a simple example of an image scene containing a number of video objects. The attempt is to encode the sequence in a way that will allow the separate decoding and reconstruction of the objects and to allow the manipulation of the original scene by simple operations on the bit stream. The bit stream will be "object layered" and the shape and transparency of each object—as well as the spatial coordinates and additional parameters describing object scaling, rotation, or related parameters—are described in the bit stream of each object layer. The receiver can either reconstruct the original sequence in its entirety, by decoding all "object layers" and by displaying the objects at original size and at the original location, as shown in Fig. 1(a), or alternatively, it is possible to manipulate the video by simple operations. For example, in Fig. 1(b), some objects were not decoded and used for reconstruction, while others were decoded and displayed using subsequent scaling or rotation. In addition, new objects were included which did not belong to the original scene. Since the bit stream of the sequence is organized in an "object layered" form, the manipulation is performed on the bit stream level—without the need for further transcoding. It is targeted to provide these capabilities for both natural and synthetic audio-visual objects as well as for hybrid representations of natural and synthetic objects. Notice that MPEG-4 images as well as image sequences are, in general, considered to be arbitrarily shaped—in contrast to the standard MPEG-1 and MPEG-2 definitions.

2) Coding Efficiency and Universal Access: Provisions for improved coding efficiency, in particular at very low bit rates below 64 kb/s, continues to be an important functionality to be supported by the standard. Other important requirements for the emerging MPEG-4 standard address the heterogeneous

Manuscript received May 10, 1996; revised October 25, 1996. This paper was recommended by Guest Editors Y.-Q. Zhang, F. Pereira, T. Sikora, and C. Reader.

The author is with the Heinrich-Hertz-Institute (HHI) for Communication Technology, 10587 Berlin, Germany.

Publisher Item Identifier S 1051-8215(97)00937-3.

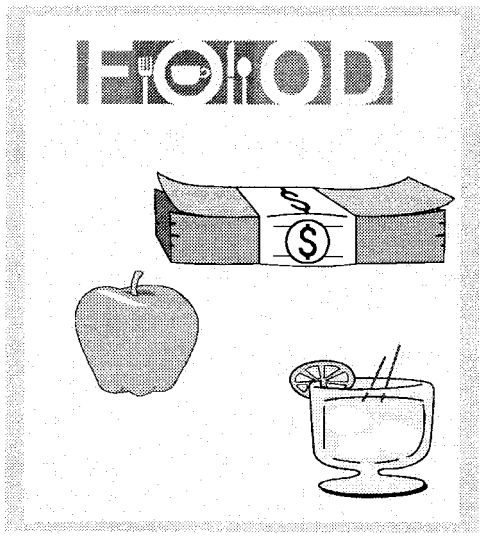
TABLE I
REQUIREMENTS FOR THE MPEG-4 VIDEO STANDARD

Functionality	MPEG-4 Video-Requirements
Content-Based Interactivity	
Content-Based Manipulation and Bitstream Editing	Support for content-based manipulation and bitstream editing without the need for transcoding.
Hybrid Natural and Synthetic Data Coding	Support for combining synthetic scenes or objects with natural scenes or objects. The ability for compositing synthetic data with ordinary video, allowing for interactivity.
Improved Temporal Random Access	Provisions for efficient methods to randomly access, within a limited time and with fine resolution, parts, e.g. video frames or arbitrarily shaped image content from a video sequence. This includes 'conventional' random access at very low bit rates.
Compression	
Improved Coding Efficiency	MPEG-4 Video shall provide subjectively better visual quality at comparable bit rates compared to existing or emerging standards.
Coding of Multiple Concurrent Data Streams	Provisions to code multiple views of a scene efficiently. For stereoscopic video applications, MPEG-4 shall allow the ability to exploit redundancy in multiple viewing points of the same scene, permitting joint coding solutions that allow compatibility with normal video as well as the ones without compatibility constraints.
Universal Access	
Robustness in Error-Prone Environments	Provisions for error robustness capabilities to allow access to applications over a variety of wireless and wired networks and storage media. Sufficient error robustness shall be provided for low bit rate applications under severe error conditions (e.g. long error bursts).
Content-Based Scalability	MPEG-4 shall provide the ability to achieve scalability with fine granularity in content, quality (e.g. spatial and temporal resolution), and complexity. In MPEG-4, these scalabilities are especially intended to result in content-based scaling of visual information.

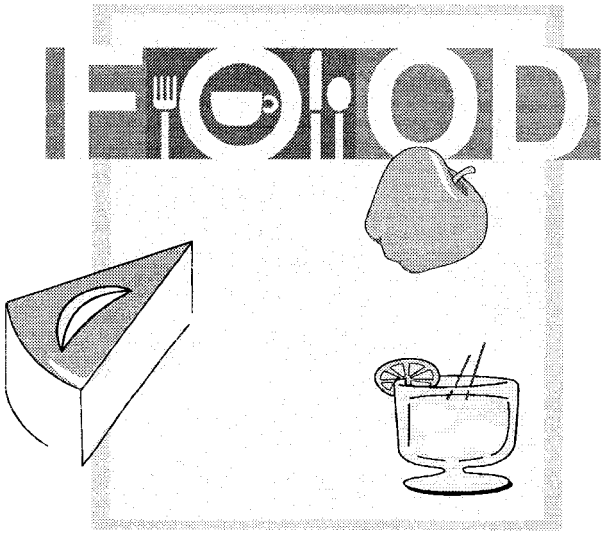
network environments that can be foreseen for many emerging MPEG-4 multimedia applications, in particular for wireless communications and database access. This introduces the requirements for tolerance of the audio and video compression algorithms with respect to noisy environments, varying bandwidths, and varying degrees of decoder resources and battery power. MPEG-4 will address this problem of error prone environments and provide content-based scalability for constrained bit rate and decoder resources.

The January 1996 MPEG meeting witnessed the release of the first version of the MPEG-4 Video Verification Model. Similar to the MPEG-2 Test Model, the MPEG-4 Video Verification Model defines a first "common core" video coding algorithm for the collaborative work within the MPEG-4 Video Group. Based on this core algorithm, a number of "Core Experiments" are defined with the aim to collaboratively improve the efficiency and functionality of the first Verification Model—and to iteratively converge through several versions of the Verification Model toward the final MPEG-4 Video coding standard by the end of 1998. To this reason, the MPEG-4 Video Verification Model provides an important platform for collaborative experimentation within the Video Group and should already give some indication about the structure of the final MPEG-4 Video coding standard.

The purpose of this paper is to provide an overview of the MPEG-4 Video Verification Model process and to outline the structure of the Verification Model algorithm. To this end, Section II discusses the role and integration of the MPEG-4 Video coding standard within the MPEG-4 framework. In Section III, the Verification Model methodology is described, and Section IV details the basic algorithm defined in the September 1996 version of the MPEG-4 Video Verification



(a)



(b)

Fig. 1. The "content-based" approach taken by the MPEG-4 Video coding standard will allow the flexible decoding, representation, and manipulation of video objects in a scene. (a) Original. (b) Manipulated.

Model. Section V discusses the role of the Core Experiment process, and Section VI finally summarizes and concludes the paper.

II. The MPEG-4 Video "Toolbox" Approach

The overall MPEG-4 applications scenario envisions the standardization of "tools" and "algorithms" for natural audio and video as well as for synthetic two-dimensional (2-D) or three-dimensional (3-D) audio and video to allow the hybrid coding of these components [4]. The MPEG-4 group has taken further steps toward an open, flexible, and extensible MPEG-4 standard by anticipating the foreseen rapid developments in the area of programmable general purpose DSP technology—and the obvious advantages with respect to software implementations of the standard. In this respect, it is foreseen to provide an open MPEG-4 standard by enabling mechanisms to

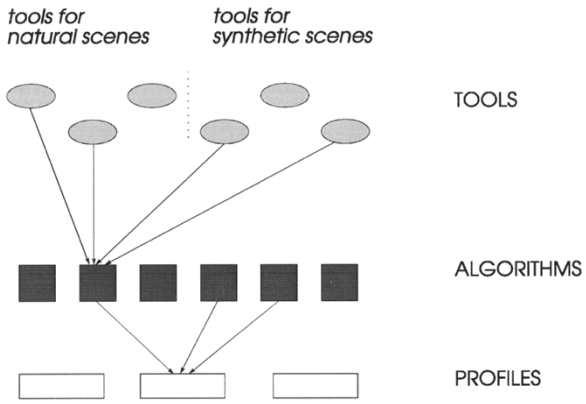


Fig. 2. Scenario of "tools," "algorithms," and profiles for the MPEG-4 Video coding standard. For the MPEG-4 standard mainly video coding tools (i.e., DCT, motion compensation, etc.) will be standardized. By means of the MSDL, a selection of tools can be flexibly combined to form an algorithm. Profiles define subsets of tools or algorithms suitable for specific applications requirements (i.e., low complexity, low delay, etc.).

download missing software decoder tools at the receiver. As a consequence, the MPEG-4 Video group will not follow the conventional approach taken by the successful MPEG-1 and MPEG-2 standards, which defined only complete algorithms for the audio, video, and systems aspects. In contrast, for MPEG-4 the attempt is to standardize video "tools"—a video "tool" being, for example, a fully defined algorithm, or only a shape coding module, a motion compensation module, a texture coding module, or related techniques. The "glue" that will bind independent coding tools together is the foreseen MPEG-4 Systems Description Language (MSDL) which will comprise several key components. First, a definition of the interfaces between the coding tools; second, a mechanism to combine coding tools and to construct algorithms and profiles; and third, a mechanism to download new tools. While some applications call for very high compression efficiency, others require high robustness in error-prone environments or a very high degree of interaction with audio or video content. No single efficient algorithm exists to cover this broad range of applications requirements. The MSDL will transmit with the bitstream the structure and rules for the decoder—thus the way the tools have to be used at the decoder in order to decode and reconstruct audio and video. At a more advanced stage, MSDL will allow the downloading of tools which are not available at the decoder. Thus, the MPEG-4 MSDL, together with the audio and video toolbox approach, will provide a very flexible framework to tackle this problem by allowing a wealth of different algorithms to be supported by the standard.

The envisioned MPEG-4 scenario in terms of standardized components for coding of visual data is summarized in Fig. 2. Note that the MPEG-4 "visual" part of the "toolbox" contains tools (including fully defined algorithms) for coding both natural (pixel based video) and synthetic visual input (i.e., 2-D or 3-D computer model data sets). The tools can be flexibly combined at the encoder and decoder to enable efficient hybrid natural and synthetic coding of visual data. The same will be the case for natural and synthetic audio data.

TABLE II
TIME SCHEDULE FOR THE MPEG-4 VIDEO STANDARD

Nov. 1995	• Subjective tests of proposals submitted to MPEG-4 Video
Jan. 1996	• Definition of 1st MPEG-4 Video Verification Model (VM)
Jan. 1996 - Nov. 1996	• Iterative improvement of the MPEG-4 Video VM • 1st Version of the MPEG-4 Video Standard Working Draft (WD)
Nov. 1996 - Nov. 1997	• Iterative improvement of the MPEG-4 Video VM and WD
Nov. 1997	• Major technical work on video algorithms finished • MPEG-4 Video Standard Committee Draft (CD)
Jan. 1998	• MPEG-4 Video Draft International Standard (DIS)
July 1998	• MPEG-4 Video Draft International Standard (DIS)

III. DEVELOPMENT OF VIDEO TOOLS AND ALGORITHMS FOR MPEG-4—THE VERIFICATION MODEL METHODOLOGY

Starting from the January 1996 Munich meeting, the work in the MPEG Video group continued in a collaborative phase with respect to the development of the MPEG-4 Video coding standard. To collaboratively develop video tools and algorithms for the final MPEG-4 standard, the MPEG-4 Video group adopted the Verification Model methodology which already proved successful in the course of the development of the MPEG-1 and MPEG-2 standards [5]. The purpose of a Verification Model (VM) within MPEG-4 is to describe completely defined encoding and decoding "common core" algorithms, such that collaborative experiments performed by multiple independent parties can produce identical results and will allow the conduction of "Core Experiments" under controlled conditions in a common environment. A VM specifies the input and output formats for the uncoded data and the format of the bitstream containing the coded data. It specifies the algorithm for encoding and decoding, including the support for one or more functionalities.

Based on the Proposal Package Description [4] for the MPEG-4 standardization phase, which identifies the preliminary requirements for the envisioned MPEG-4 Video standard (see also Table I), a variety of algorithms were developed by companies worldwide in a competitive manner. In November 1994, a "Call for Proposals" was issued by the MPEG-4 group where laboratories were asked to submit results for their video coding algorithm, tools, and proposals to be compared in formal subjective viewing tests [6]. The "Call for Proposals" specified detailed functionalities that needed to be addressed by the proposers and defined test sequences and coding conditions to be used. The functionalities addressed were: coding efficiency, content-based scalability, content-based spatial and temporal scalability, and error robustness and resilience. The subjective viewing tests were carried out in November 1995 and resulted in a ranking of the proposals with respect to the subjective image quality achieved for the diverse functionalities [7], [8]. In addition, laboratories were asked to submit proposals for video tools and algorithms for MPEG-4 which were not subject to formal subjective viewing tests—but were evaluated purely for their technical merit by MPEG Video group experts [9], [10].

As a result, a wealth of promising video coding techniques addressing diverse functionalities were identified. The reader is referred to the references in [11] for an excellent summary of a selection of the techniques submitted to the subjective viewing test and to the informal video group evaluation. Based on the proposals submitted, the first version of the MPEG-4 Video Verification Model was defined in January 1996. This event marked the end of the MPEG-4 competitive phase and the beginning of the collaborative effort in the MPEG Video group.

Anticipating that the final MPEG-4 Video coding standard is intended to be generic by supporting a broad range of applications with varying applications requirements, the MPEG Video group adapted an approach for defining the VM which is functionality driven. The aim was to cover a maximum set of the functionalities in Table I by one VM algorithm to support a maximum of applications requirements. Based on the ranking in the subjective viewing test, and based on the technical merit of the algorithms, it was possible to identify a small number of techniques which performed most promising in the tests and which used similar technology to cover a range of functionalities. These algorithms formed the substance for the first version of the MPEG-4 Video Verification Model algorithm.

Based on the remaining proposals submitted, a list of "Core Experiments" was defined to foster the improvement of the VM between the meetings in the collaboration phase. In subsequent meetings, new tools can be brought to MPEG-4 and these will be evaluated inside the VM process following a Core Experiment procedure if a minimum of two independent companies agree to perform the same experiments. In the final standard, if two tools accomplish the same functionality under the same conditions, only the best will be chosen.

The Core Experiment process will continue until November 1997 when the Committee Draft of the MPEG-4 Video standard will be released. Table II summarizes the foreseen time schedule for the development of the standard.

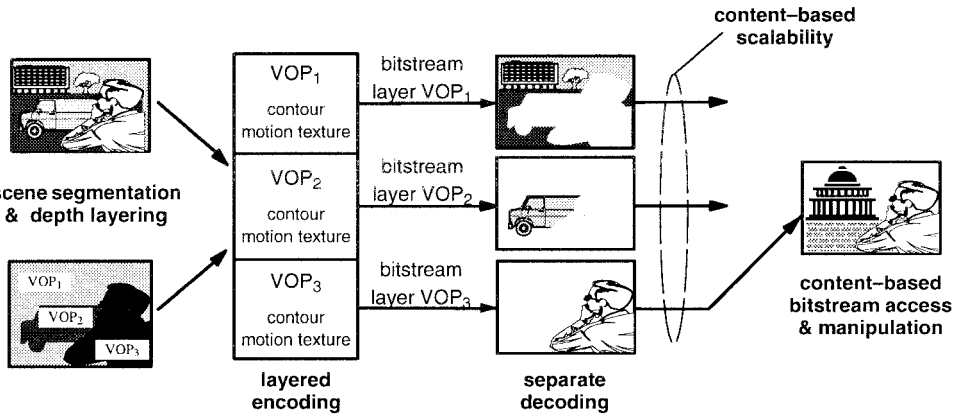


Fig. 3. The coding of image sequences using MPEG-4 VOP's enables basic content-based functionalities at the decoder. Each VOP specifies particular image sequence content and is coded into a separate VOL-layer (by coding contour, motion, and texture information). Decoding of all VOP-layers reconstructs the original image sequence. Content can be reconstructed by separately decoding a single or a set of VOL-layers (content-based scalability/access in the compressed domain). This allows content-based manipulation at the decoder without the need for transcoding.

IV. The MPEG-4 Video Verification Model

In the January 1996 MPEG Video group meeting in Munich, Germany, the first version of the official MPEG-4 Video Verification Model was defined. The VM has since then, by means of the Core Experiment process, iteratively progressed in each subsequent meeting and has been optimized with respect to coding efficiency and the provisions for new content-based functionalities and error robustness. At the current stage, the MPEG-4 Video Verification Model supports the features summarized below [12].

- Standard $Y:U:V$ luminance and chrominance intensity representation of regularly sampled pixels in $4:2:0$ format. The intensity of each Y, U, or V pixel is quantized into 8 b. The image size and shape depends on the application.
- Coding of multiple "video object planes" (VOP's) as images of arbitrary shape to support many of the content-based functionalities. Thus, the image sequence input for the MPEG-4 Video VM is, in general, considered to be of arbitrary shape—and the shape and location of a VOP within a reference window may vary over time. The coding of standard rectangular image input sequences is supported as a special case of the more general VOP approach.
- Coding of shape and transparency information of each VOP by coding binary or gray scale alpha plane image sequences.
- Support of intra (I) coded VOP's as well as temporally predicted (P) and bidirectionally (B) predicted VOP's. Standard MPEG and H.263 I, P, and B frames are supported as special case.
- Support of fixed and variable frame rates of the input VOP sequences of arbitrary or rectangular shape. The frame rate depends on the application.
- 8×8 pel block-based and 16×16 pel macroblock-based motion estimation and compensation of the pixel values within VOP's, including provisions for block-overlapping motion compensation.
- Texture coding in I, P, and B-VOP's using a discrete cosine transform (DCT) adopted to regions of arbitrary shape, followed by MPEG-1/2 or H.261/3 like quantization and run-length coding.
- Efficient prediction of dc- and ac-coefficients of the DCT in intra coded VOP's.
- Temporal and spatial scalability for arbitrarily shaped VOP's.
- Adaptive macroblock slices for resynchronization in error prone environments.
- Backward compatibility with standard H.261/3 or MPEG-1/2 coding algorithms if the input image sequences are coded in a single layer using a single rectangular VOP structure.

The reader is referred to the references in [2], [3], and [15] for details related to the H.261/3 and MPEG-1/2 standards video compression algorithms.

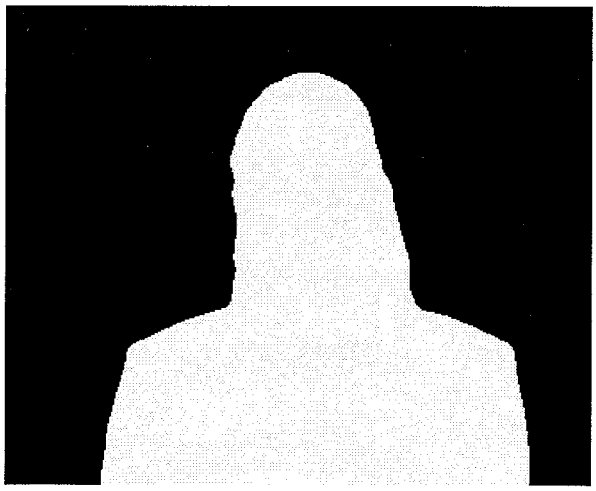
A. Provisions for Content Based Functionalities—Decomposition into "Video Object Planes"

The MPEG-4 Video coding algorithm will eventually support all functionalities already provided by MPEG-1 and MPEG-2, including the provision to efficiently compress standard rectangular sized image sequences at varying levels of input formats, frame rates, and bit rates.

Furthermore, at the heart of the so-called "content"-based MPEG-4 Video functionalities is the support for the separate encoding and decoding of content (i.e., physical objects in a scene). Within the context of MPEG-4, this functionality—the ability to identify and selectively decode and reconstruct video content of interest—is referred to as "content-based scalability." This MPEG-4 feature provides the most elementary mechanism for interactivity and manipulation with/of content of images or video in the compressed domain without the need for further segmentation or transcoding at the receiver.

To enable the content-based interactive functionalities envisioned, the MPEG-4 Video Verification Model introduces the concept of VOP's. It is assumed that each frame of

(a)



(b)

Fig. 4. Description of the shape of a VOP by means of an alpha plane mask (binary segmentation mask in this case). (a) Image of the original sequence Akiyo. (b) Binary segmentation mask specifying the location of the foreground content VOP (person Akiyo).

an input video sequence is segmented into a number of arbitrarily shaped image regions (video object planes)—each of the regions may possibly cover particular image or video content of interest, i.e., describing physical objects or content within scenes. In contrast to the video source format used for the MPEG-1 and MPEG-2 standards, the video input to be coded by the MPEG-4 Verification Model is thus no longer considered a rectangular region. This concept is illustrated in Fig. 3. The input to be coded can be a VOP image region of arbitrary shape and the shape and location of the region can vary from frame to frame. Successive VOP's belonging to the same physical object in a scene are referred to as video objects (VO's)—a sequence of VOP's of possibly arbitrary shape and position. The shape, motion, and texture information of the VOP's belonging to the same VO is encoded and transmitted or coded into a separate video object layer (VOL). In addition, relevant information needed to identify each of the VOL's—and how the various VOL's are composed at the receiver to reconstruct the entire original sequence is also included in the bitstream. This allows the separate decoding of each VOP and the required flexible manipulation of the video sequence as indicated in Fig. 3—similar to the example already discussed in Fig. 1. Notice that the video source input assumed for the VOL structure either already exists in terms of separate entities (i.e., is generated with chroma-key technology) or is generated by means of on-line or off-line segmentation algorithms.

To illustrate the concept, the MPEG-4 Video source input test sequence AKIYO in Fig. 4(a), which as an example consists of a foreground person and of textured stationary background content, is here decomposed into a background VOP_1 and a foreground VOP_2. A binary alpha plane image sequence as depicted in Fig. 4(b) is coded in this example to indicate to the decoder the shape and location of the foreground object VOP_2 with respect to the background VOP_1. In general, the MPEG-4 Video Verification Model also supports the coding of gray scale alpha planes to allow at the receiver the composition of VOP's with various levels of transparency.

Fig. 5(a) and (b) depict an example of the content of the two VOP's to be coded in two separate VOL-layers in Fig. 3. Note that the image regions covered by the two VOP's are nonoverlapping, and that the sum of the pels covered by the two VOP's is identical to the pels contained in the original source sequence in Fig. 4(a). Both VOP's are of arbitrary shape and the shape and the location of the VOP's change over time. The receiver can either decode and display each VOP separately (i.e., the foreground person in VOP_2 only) or reconstruct the original sequence by decoding and appropriate compositing of both VOP's based on the decoded alpha channel information.

The MPEG-4 Video Verification Model also supports the coding of overlapping VOP's as indicated in Fig. 6(a) and (b). Here, the foreground VOP_2 in Fig. 6(b) is identical to the one in Fig. 5(b). However, the background VOP_1 is of rectangular shape with the size of the original input images, and the shape of the background VOP remains the same for the entire sequence. Again, both VOP's are encoded separately and the original is reconstructed at the receiver by decoding each VOP and pasting the foreground VOP content at the appropriate location on top of the background layer content based on the decoded alpha channel information. If the background VOP content is stationary (as is the case in the AKIYO test sequence—meaning that the background content does not change over time), only one frame needs to be coded for the background VOP. Thus the foreground and background VOP's may have different display repetition rates at the receiver.

Notice that, if the original input image sequences are not decomposed into several VOL's of arbitrary shape, the coding structure simply degenerates into a single layer representation which supports conventional image sequences of rectangular shape. The MPEG-4 content-based approach can thus be seen as a logical extension of the conventional MPEG-1 and MPEG-2 coding approach toward image input sequences of arbitrary shape.

(a)

(b)

Fig. 5. Image content of VOP_1 [(a) background VOP] and VOP_2 [(b) foreground VOP] according to the alpha plane mask in Fig. 4(b). Contour, motion, and texture information for each VOP is coded in a separate VOP-layer. Notice that the two VOP's are nonoverlapping and the image sequence input for each VOP-layer is of arbitrary shape, with the location and shape varying between VOP images depending on the movement of the person Akiyo.

(a)

(b)

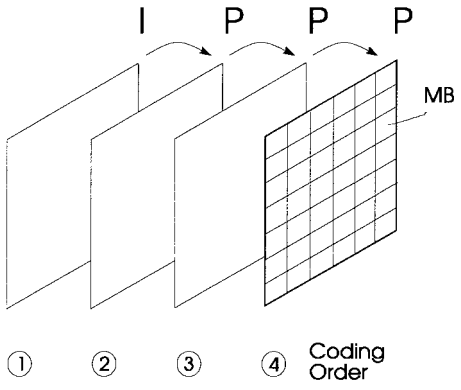
Fig. 6. An example of the decomposition of the original image sequence AKIYO in Fig. 4 into overlapping VOP's (i.e., if the entire background is known prior to coding). (a) The background VOP_1 in this case is a possibly stationary rectangular image. (b) The foreground VOP_2 remains the same than the one depicted in Fig. 5(b).

B. Coding of Shape, Motion, and Texture Information for each VOP

As indicated in Fig. 3, the information related to the shape, motion, and texture information for each VO is coded into a separate VOL-layer in order to support separate decoding of VO's. The MPEG-4 Video VM uses an identical algorithm to code the shape, motion, and texture information in each of the layers. The shape information is, however, not transmitted if the input image sequence to be coded contains only standard images of rectangular size. In this case, the MPEG-4 Video coding algorithm has a structure similar to the successful MPEG-1/2 or H.261 coding algorithms—suitable for applications which require high coding efficiency without the need for extended content based functionalities.

The MPEG-4 VM compression algorithm employed for coding each VOP image sequence (rectangular size or not) is based on the successful block-based hybrid DPCM/transform coding technique already employed in the MPEG coding standards [3]. As outlined in Fig. 7(a) for the example of a VOP of rectangular shape, the MPEG-4 coding algorithm encodes the first VOP in intraframe VOP coding mode (I-VOP). Each subsequent frame is coded using interframe VOP prediction (P-VOP's)—only data from the nearest previously coded VOP frame is used for prediction. In addition, the coding of bidirectionally predicted VOP's (B-VOP's) is also supported.

Similar to the MPEG baseline coders, the MPEG-4 VM algorithm processes the successive images of a VOP sequence block-based. Taking the example of arbitrarily shaped VOP's, after coding the VOP shape information, each color input VOP image in a VOP sequence is partitioned into nonoverlapping "macroblocks" as depicted in Figs. 7–9. Each macroblock contains blocks of data from both luminance and cosited chrominance bands—four luminance blocks (Y_1, Y_2, Y_3, Y_4) and two



(a)

(b)

Fig. 7. (a) Illustration of an I-picture VOP (I-VOP) and P-picture VOP's (P-VOP's) in a video sequence. P-VOP's are coded using motion-compensated prediction based on the nearest previous VOP frame. Each frame is divided into disjoint "macroblocks" (MB). (b) With each MB, information related to four luminance blocks (Y_1, Y_2, Y_3, Y_4) and two chrominance blocks (U, V) is coded. Each block contains 8 $\times$ 8 pels.

chrominance blocks (U, V), each with size 8 $\times$ 8 pels. The basic diagram of the MPEG-4 VM hybrid DPCM/Transform encoder and decoder structure for processing single Y, U, or V blocks and macroblocks is depicted in Fig. 8. The previously coded VOP frame $N-1$ is stored in a VOP frame store in both encoder and decoder. Motion compensation is performed on a block or macroblock basis—only one motion vector is estimated between VOP frame N and VOP frame $N-1$ for a particular block or macroblock to be encoded. The motion-compensated prediction error is calculated by subtracting each pel in a block or macroblock belonging to the VOP frame N with its motion shifted counterpart in the previous VOP frame $N-1$. An 8 $\times$ 8 DCT is then applied to each of the 8 $\times$ 8 blocks contained in the block or macroblock followed by quantization (Q) of the DCT coefficients with subsequent run-length coding and entropy coding (VLC). A video buffer is needed to ensure that a constant target bit rate output is produced by the encoder. The quantization stepsize for the DCT-coefficients can be adjusted for each macroblock in a VOP frame to achieve a given target bit rate and to avoid buffer overflow and underflow.

The decoder uses the reverse process to reproduce a macroblock of VOP frame N at the receiver. After decoding the variable length words contained in the video decoder buffer, the pixel values of the prediction error are reconstructed. The motion-compensated pixels from the previous VOP frame $N-1$ contained in the VOP frame store are added to the prediction error to recover the particular macroblock of frame N.

In general, the input images to be coded in each VOP layer are of arbitrary shape and the shape and location of the images vary over time with respect to a reference window. For coding shape, motion, and texture information in arbitrarily shaped VOP's, the MPEG-4 Video VM introduces the concept of a "VOP image window" together with a "shape-adaptive" macroblock grid. All VOL layers to be coded for a given input video sequence are defined with reference to the reference window of constant size. An example of a VOP image window within a reference window and an example of a macroblock grid for a particular VOP image are depicted in Fig. 9. The shape information of a VOP is coded prior to coding motion vectors based on the VOP image window macroblock grid and is available to both encoder and decoder. In subsequent processing steps, only the motion and texture information for the macroblocks belonging to the VOP image are coded (which includes the standard macroblocks as well as the contour macroblocks in Fig. 9).

1) Shape Coding: Essentially, two coding methods are supported by the MPEG-4 Video VM for binary and gray scale shape information. The shape information is referred to as "alpha planes" in the context of the MPEG-4 VM. The techniques to be adopted for the standard will provide lossless coding of alpha-planes as well as the provision for lossy coding of shapes and transparency information, allowing the tradeoff between bit rate and the accuracy of shape representation. Furthermore, it is foreseen to support both intra shape coding as well as inter shape coding functionalities employing motion-compensated shape prediction—to allow both efficient random access operations as well as an efficient compression of shape and transparency information for diverse applications.

2) Motion Estimation and Compensation: The MPEG-4 VM employs block-based motion estimation and compensation techniques to efficiently explore temporal redundancies of the video content in the separate VOP layers. In general, the motion estimation and compensation techniques used can be seen as an extension of the standard MPEG-1/2 or H.261/3 block matching techniques toward image sequences of arbitrary shape [2], [3]. However, a wealth of different motion prediction methods is also being investigated in the Core Experiment process (see Section V).

To perform block-based motion estimation and compensation between VOP's of varying location, size, and shape, the shape-adaptive macroblock (MB) grid approach for each VOP image as discussed in Fig. 9 is employed. A block-matching procedure shown in Fig. 10 is used for standard macroblocks. The prediction error is coded together with the macroblock motion vectors used for prediction. An advanced motion compensation mode is defined which supports block-overlapping motion compensation as with the ITU H.263 standard as well as the coding of motion vectors for 8 $\times$ 8 blocks [13].

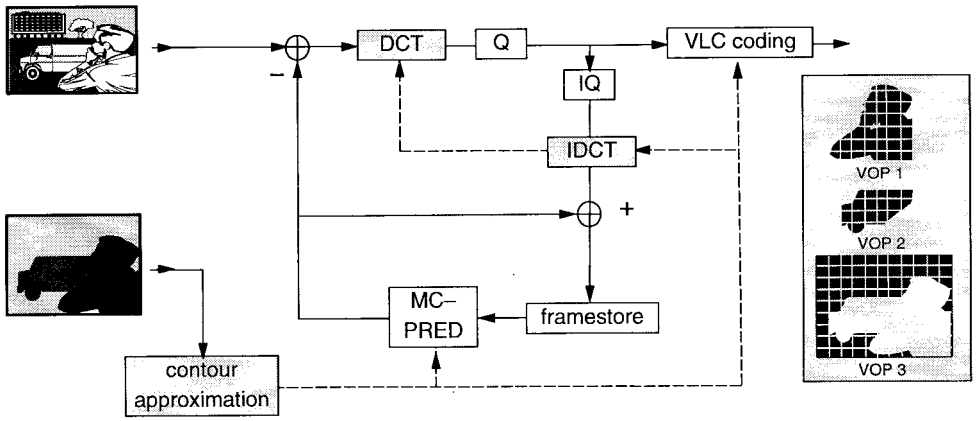


Fig. 8. Block diagram of the basic MPEG-4 VM hybrid DPCM/transform encoder and decoder structure.

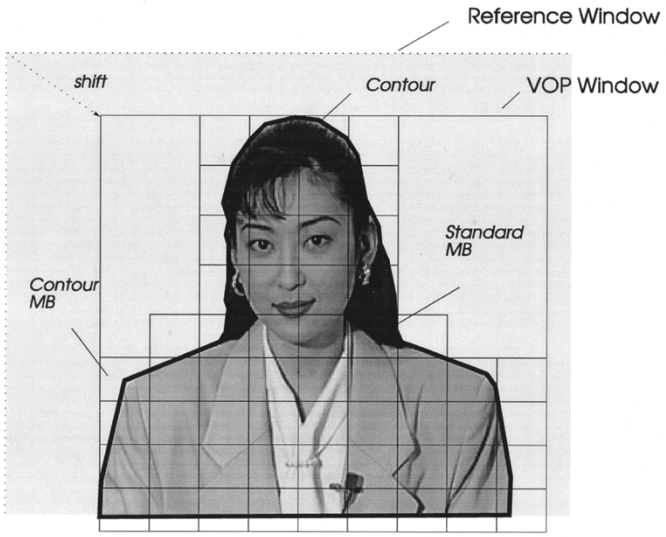


Fig. 9. Example of an MPEG-4 VM macroblock grid for the AKIYO foreground VOP_2 image. This macroblock grid is used for alpha plane coding, motion estimation, and compensation as well as for block-based texture coding. A VOP window with a size of multiples of 16 pels in each image direction surrounds the foreground VOP_2 of arbitrary shape and specifies the location of the macroblocks, each of size 16×16 pels. This window is adjusted to collocate with the top-most and left-most border of the VOP. A shift parameter is coded to indicate the location of the VOP window with respect to the borders of a reference window (original image borders).

The definition of the motion estimation and compensation techniques are, however, modified at the borders of a VOP. An image padding technique is used for the reference VOP frame $N-1$, which is available to both encoder and decoder, to perform motion estimation and compensation. The VOP padding method can be seen as an extrapolation of pels outside of the VOP based on pels inside of the VOP. After padding the reference VOP in frame $N-1$ (as shown in Fig. 11 for our example in Fig. 9), a "polygon" matching technique is employed for motion estimation and compensation. A polygon defines the part of the contour macroblock (or the 8×8 block for advanced motion compensation, respectively) which belongs to the active area inside of the VOP frame N to be coded and excludes the pels outside of this area. Thus, the pels not belonging to the active area in the VOP to be coded are essentially excluded from the motion estimation process.

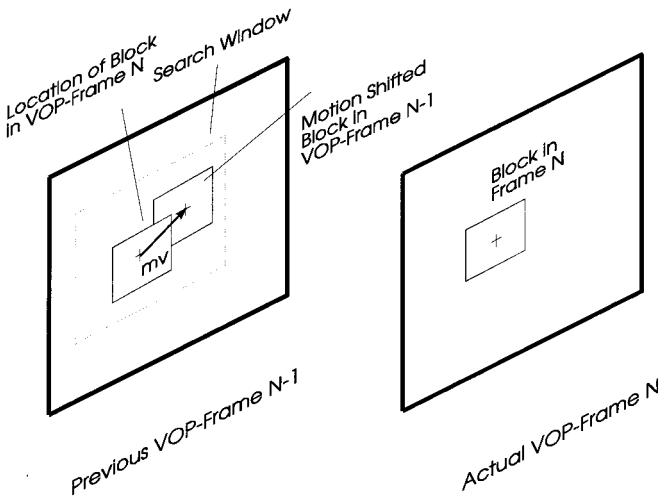
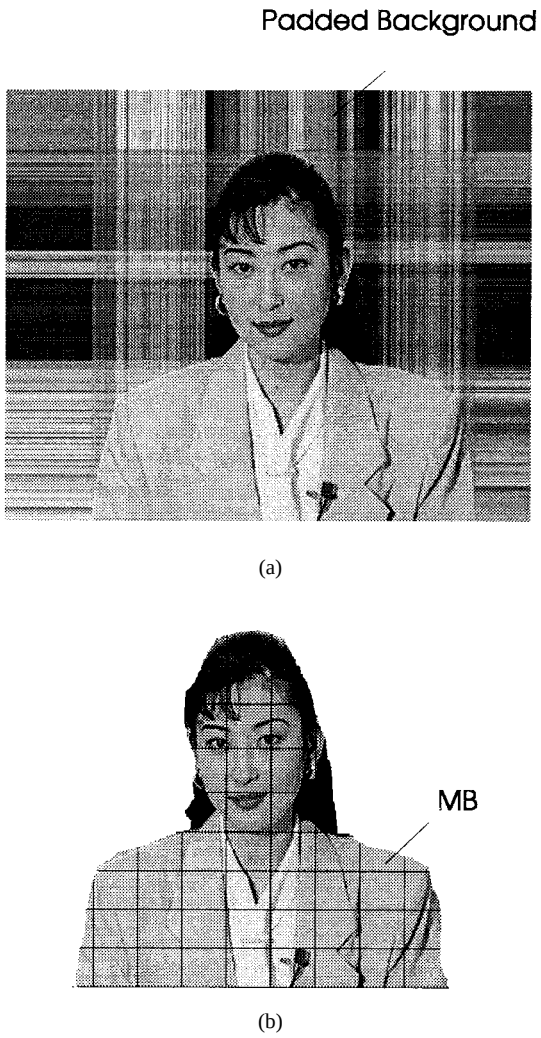


Fig. 10. Block matching approach for motion compensation: One motion vector (MV) is estimated for each block in the actual VOP frame N to be coded. The motion vector points to a reference block of same size in the previously coded VOP frame $N-1$. The motion-compensated prediction error is calculated by subtracting each pel in a block with its motion-shifted counterpart in the reference block of the previous VOP frame.

The MPEG-4 Video VM supports the coding of both forward-predicted (P) as well as bidirectionally (B) predicted VOP's (P-VOP and B-VOP). Motion vectors are predictively coded using standard MPEG-1/2 and H.263 VLC code tables including the provision for extended vector ranges. Notice that the coding of standard MPEG I-frames, P-frames, and B-frames is still supported by the VM—for the special case of image input sequences (VOP's) of rectangular shape (standard MPEG or H.261/3 definition of frames).

3) Texture Coding: The intra VOP's as well as the residual errors after motion-compensated prediction are coded using a DCT on 8×8 blocks similar to the standard MPEG and H.263 standards. Again, the adaptive VOP window macroblock grid specified in Fig. 9 is employed for this purpose. For each macroblock, a maximum of four 8×8 luminance blocks and two 8×8 chrominance blocks are coded. Particular adaptation is required for the 8×8 blocks straddling the VOP borders. The image padding technique in Fig. 11 is used to fill the



(a)

(b)

Fig. 11. An image padding technique is employed for the purpose of contour block motion estimation and compensation as well as for the contour block texture coding. The aim of the padding procedure is to allow separate decoding and reconstruction of VOP's by extrapolating texture inside the VOP to regions outside the VOP (here shown for the foreground VOP_2 of AKIYO). This allows block-based DCT coding of texture across a VOP border as well. Furthermore, the block-based motion vector range for search and motion compensation in a VOP in frame N can be specified covering regions outside the VOP in frame $N-1$. (a) Previous frame. (b) Actual frame.

macroblock content outside of a VOP prior to applying the DCT in intra-VOP's. For the coding of motion-compensated prediction error P-VOP's, the content of the pels outside of the active VOP area are set to 128. Scanning of the DCT coefficients followed by quantization and run-length coding of the coefficients is performed using techniques and VLC tables defined with the MPEG-1/2 and H.263 standards, including the provision for quantization matrices. An efficient prediction of the dc- and ac-coefficients of the DCT is performed for intra coded VOP's.

In the Core Experiment process, a considerable effort is dedicated to explore alternative techniques for texture coding, such as shape adaptive DCT's and wavelet transforms.

4) Multiplexing of Shape, Motion, and Texture Information: Basically all "tools" (DCT, motion estimation, and compensation, etc.) defined in the H.263 and MPEG-1 standards (and most of the ones defined for MPEG-2 Main Profile) are currently supported by the MPEG-4 Video VM. The compressed alpha plane, motion vector, and DCT bit words are multiplexed into a VOL layer bitstream by coding the shape information first, followed by motion and texture coding based on the H.263 and MPEG definitions.

The VM defines two separate modes for multiplexing texture and motion information: A joint motion vector and DCT-coefficient coding procedure based on standard H.263-like macroblock type definitions is supported to achieve a high compression efficiency at very low bit rates. This guarantees that the performance of the VM at very low bit rates is at least identical to the H.263 standard. Alternatively, the separate coding of motion vectors and DCT-coefficients is also possible—to eventually incorporate new and more efficient motion or texture coding techniques separately into the VM.

C. Coding Efficiency

Besides the provision for new content-based functionalities and error resilience and robustness, the coding of video with very high coding efficiency over a range of bit rates continues to be supported for the MPEG-4 standard. As indicated above, the MPEG-4 Video VM allows the single object-layer (single VOP) coding approach as a special case. In this coding mode, the single VOP input image sequence format may be rectangular as depicted in Fig. 7 (thus not segmented into several VOP's), and the MPEG-4 Video VM coding algorithm can be made almost compatible to the ITU-H.263 or ISO-MPEG-1 standards. Most of the coding techniques used by the MPEG-2 standard at Main Profile are also supported. A number of motion compensation and texture coding techniques are being investigated in the Core Experiment process to further improve coding efficiency for a range of bit rates, including bit rates below 64 kb/s.

D. Spatial and Temporal Scalability

An important goal of scaleable coding of video is to flexibly support receivers with different bandwidth or display capabilities or display requests to allow video database browsing and multiresolution playback of video content in multimedia environments. Another important purpose of scaleable coding is to provide a layered video bit stream which is amenable for prioritized transmission. The techniques adopted for the MPEG-4 Video VM allow the "content-based" access or transmission of arbitrarily-shaped VOP's at various temporal or spatial resolutions—in contrast to the frame-based scalability approaches introduced for MPEG-2. Receivers either not capable or willing to reconstruct the full resolution arbitrarily shaped VOP's can decode subsets of the layered bit stream to display the arbitrarily shaped VOP's content/objects at lower spatial or temporal resolution or with lower quality.

1) Spatial Scalability: Fig. 12 depicts the MPEG-4 general philosophy of a content-based VOP multiscale video coding scheme. Here, three layers are provided, each layer supporting a VOP at different spatial resolution scales, i.e., a multiresolution representation can be achieved by downscaling the input video signal into a lower resolution video (downsampling

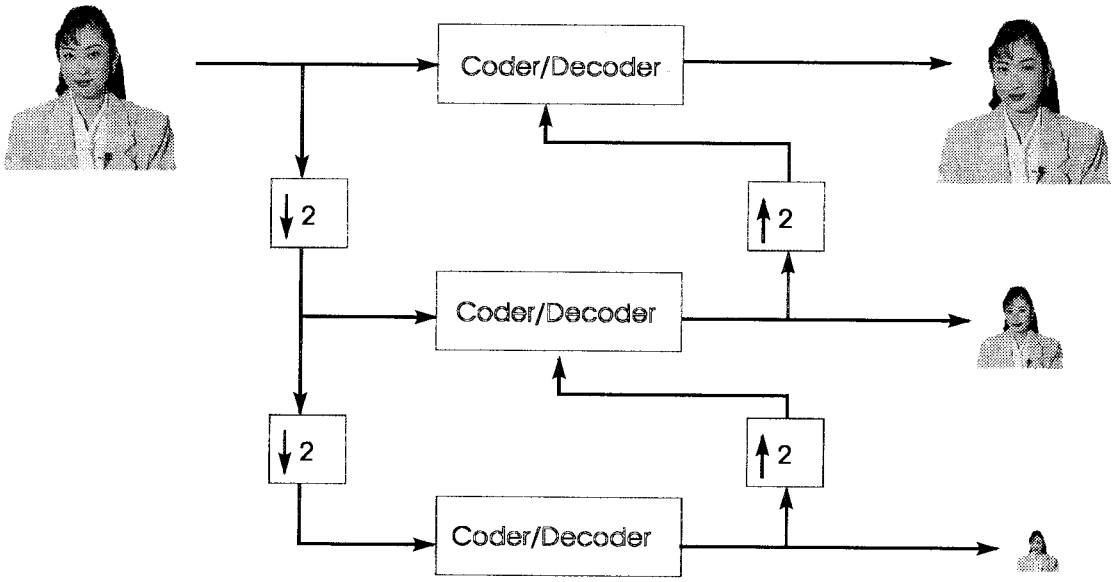


Fig. 12. Spatial scalability approach for arbitrarily shaped VOP's.

spatially in our example). The downscaled version is encoded into a base layer bit stream with reduced bit rate. The upscaled reconstructed base layer video (upsampled spatially in our example) is used as a prediction for the coding of the original input video signal. The prediction error is encoded into an enhancement layer bit stream. If a receiver is either not capable or willing to display the full quality VOP's, downscaled VOP signals can be reconstructed by only decoding the lower layer bit streams. It is important to notice, however, that the display of the VOP at highest resolution with reduced quality is also possible by only decoding the lower bit rate base layer(s). Thus, scaleable coding can be used to encode content-based video with a suitable bit rate allocated to each layer in order to meet specific bandwidth requirements of transmission channels or storage media. Browsing through video data bases and transmission of video over heterogeneous networks are applications expected to benefit from this functionality.

2) Temporal Scalability: This technique was developed with an aim similar to spatial scalability. Different frame rates can be supported with a layered bit stream. Layering is achieved by providing a temporal prediction for the enhancement layer based on coded video from the lower layers. Using the MPEG-4 "content-based" VOP temporal scalability approach, it is possible to provide different display rates for different VOL's within the same video sequence (i.e., a foreground person of interest may be displayed with a higher frame rate compared to the remaining background or other objects).

E. Error Resilience—Error Robustness

A considerable effort has been made to investigate the robust storage and transmission of MPEG-4 Video in error prone environments. To this end, an adaptive macroblock slice technique similar to the one already provided with the MPEG-1 and MPEG-2 standards has been introduced into the MPEG-4 Video VM. The technique provides resynchronization bit words for groups of macroblocks and has been optimized in particular to achieve efficient robustness for low bit rate video under a variety of severe error conditions, i.e., for the transmission over mobile channels.

V. The Core Experiment Process

Based on the "Core Experiment" process, the MPEG-4 Video VM algorithm is being refined with the aim to *collaboratively* improve the efficiency and functionality of the VM—and to iteratively converge through several versions of the VM toward the final MPEG-4 Video coding standard by the end of 1998.

At the current stage, the MPEG-4 Video VM supports functionalities such as high coding efficiency, random access, error robustness, content-based scalability, and content-based random access features. The MPEG Video group has established a number of Core Experiments to improve the efficiency of the MPEG-4 VM between meetings with respect to the functionalities already supported—and to identify new coding techniques that allow provisions for functionalities not yet supported by the VM. Table III details a selection of the diverse Core Experiment techniques.

A Core Experiment is defined with respect to the VM, which is considered as the common core algorithm. A Core Experiment proposal describes a potential algorithmic improvement to the VM, i.e., a motion compensation technique different from the one defined by the VM. Furthermore, the full description of encoder and decoder implementation of the algorithm and the specification of experimental conditions (bit rates, test sequences, etc.) to compare the proposed Core Experiment technique against the performance of the VM are provided. A Core Experiment is being established by the

TABLE III
CORE EXPERIMENTS

Subject	Techniques compared in Core Experiments
Motion Prediction	Global motion compensation, Block partitioning, Short-term/long-term frame memory, Variable block size motion compensation, 2D Triangular mesh prediction, Sub-pel prediction.
Frame Texture Coding	Wavelet transforms, Matching pursuits, 3D-DCT, Lapped transforms, Improved Intra coding, Variable block-size DCT.
Shape and Alpha Channel Coding	Gray scale shape coding, Geometrical transforms, Shape-adaptive region partitioning, Variable block-size segmentation.
Arbitrary-Shaped Region Texture Coding	Padding DCT, Mean-replacement DCT, Shape-adaptive DCT, Extension/interpolation DCT, Wavelet/subband coding.
Error Resilience/Robustness	Resynchronization techniques, Hierarchical structures, Back channel signaling, Error concealment.
Bandwidth and Complexity Scaling	Generalized temporal-spatial coding, content-based temporal scalability.
Misc.	Rate control, Mismatch corrected stereo/multiview coding, 2D triangular mesh for object and content manipulation, Noise removal, Automatic segmentation, Generation of sprites.

MPEG Video group if two independent parties are committed to perform the experiment. If a Core Experiment is successful in improving on techniques described in the VM—i.e., in terms of coding efficiency, provisions for functionalities not supported by the VM, and implementation complexity—the successful technique will be incorporated into the newest version of the VM. The technique will either replace an existing technique or supplement the algorithms supported by the VM. Core Experiments are being performed between two MPEG Video group meetings. At each MPEG Video group meeting, the results of the Core Experiments are being reviewed and the VM is updated depending on the outcome of the experiment and a new version of the VM is being released.

VI. SUMMARY AND CONCLUSION

In this paper, the aim and methodologies of the MPEG-4 Video standardization process has been outlined. Starting from algorithms and tools submitted to the MPEG-4 Video group, and which have been tested by formal subjective viewing tests by the MPEG Test group, a VM methodology is used to develop the envisioned MPEG-4 Video coding standard.

The MPEG-4 Video VM defines a video coding algorithm including a firm definition of the video coder and decoder structure. The primary intent of the MPEG-4 VM methodology is to provide a fully defined core video coding algorithm platform for core experimental purposes. This core algorithm is used to verify the performance of proposed algorithms and tools submitted to the MPEG Video group—and to iteratively converge in a collaborative effort toward the final MPEG-4 Video coding standard by July 1998.

The MPEG-4 Video VM introduces the concept of VOP's to support content-based functionalities at the decoder. The primary intent is to support the coding of image sequences which are presegmented based on image content—and to allow the flexible and separate reconstruction and manipulation of content at the decoder in the compressed domain. To this end, the image input sequences in each VOP to be coded are, in general, considered to be entries of arbitrary shape. The VM encodes shape, motion, and texture information for each VOP to allow a large degree of flexibility for the Core Experiment process. The coding of image sequences using a single layer VOP—thus the coding of standard rectangular size image sequences—is supported as a special case, i.e., if coding efficiency is of primary interest. A number of Core Experiments intended to improve the VM with respect to coding efficiency, error robustness, and content-based functionalities are being investigated. It is targeted to release the Committee Draft of the MPEG-4 Video standard in November 1997 and to promote this draft to the final International Standard by July 1998.

It is envisioned that the final MPEG-4 Video standard will define "tools" and "algorithms" resulting in a toolbox of relevant video tools and algorithms available to both encoder and decoder. These tools and algorithms will be defined based on the MPEG-4 Video VM algorithm. It is likely that, similar to the approach taken by the MPEG-2 standard [3], [13], [14], profiles will be defined for tools and algorithms which include subsets of the MPEG-4 Video tools and algorithms.

The MPEG-4 MSDL will provide sufficient means to flexibly glue video tools and algorithms at the encoder and decoder to suit the particular needs of diverse and specific applications. While some applications may require a high degree of flexibility with respect to random access and interaction with image content (i.e., the provision to separately access, decode, and display VOP's and to further flexibly access shape, motion,

and texture information associated with each VOP separately), others call for very high coding efficiency and/or very high error robustness. The MPEG-4 MSDL will allow the flexible definition of a bitstream syntax which multiplexes shape, motion, and texture information in each VOP. It is foreseen to provide a bitstream which is in part or entirely compatible to the H.261, H.263, or MPEG-1 and MPEG-2 standards (i.e., by degenerating the VOP structure into one rectangular VOP and coding and multiplexing the motion and texture information accordingly). Furthermore it may become feasible to achieve an error robustness much more sophisticated than currently provided by these standards, by flexibly redefining standard MPEG or ITU syntax definitions and synchronization bit words tailored for error patterns encountered in specific transmission or storage media.

REFERENCES

[1] L. Chiariglione, "The development of an integrated audiovisual coding standard: MPEG," *Proc. IEEE*, vol. 83, pp. 151–157, Feb. 1995.
[2] D. J. Le Gall, "The MPEG video compression algorithm," *Signal Processing: Image Commun.*, 1992, vol. 4, no. 4, pp. 129–140.
[3] R. Schäfer and T. Sikora, "Digital video coding standards and their role in video communications," *Proc. IEEE*, vol. 83, pp. 907–924, June 1995.
[4] MPEG AOE Group, "Proposal package description (PPD)—Revision 3," Tokyo meeting, document ISO/IEC/JTC1/SC29/WG11 N998, July 1995.
[5] S. Okubo, "Reference model methodology—A tool for collaborative creation of video coding standards," *Proc. IEEE*, vol. 83, pp. 139–150, Feb. 1995.
[6] MPEG AOE Group, "Call for proposals," Tokyo meeting, July 1995.
[7] F. Pereira and T. Alpert, "MPEG-4 video subjective test procedures," *IEEE Trans. Circuits Syst. Video Technol.*, this issue, pp. 32–51.
[8] H. Peterson, "Report of ad-hoc group on MPEG-4 video testing logistics," ISO/IEC/JTC1/SC29/WG11 N1056, Nov. 1995.
[9] MPEG Video Group, "Report of ad-hoc group on the evaluation of tools for nontested functionalities of video submissions to MPEG-4," Dallas meeting, document ISO/IEC/JTC1/SC29/WG11 N1064, Nov. 1995.
[10] MPEG Video Group, "Report of ad-hoc group on the evaluation of tools for nontested functionalities of video submissions to MPEG-4," Munich meeting, document ISO/IEC/JTC1/SC29/WG11 N0679, Jan. 1996.
[11] *IEEE Trans. Circuits Syst. Video Technol.*, special issue on MPEG-4, this issue.
[12] MPEG Video Group, "MPEG-4 video verification model—Version 2.1," ISO/IEC JTC1/SC29/WG11, May 1996, draft in progress.
[13] ISO/IEC 13818-2 MPEG-2 Video Coding Standard, "Information technology—Generic coding of moving pictures and associated audio information: Video," Mar. 1995.
[14] S. Okubo, K. McCann, and A. Lippmann, "MPEG-2 requirements, profiles and performance verification—Framework for developing a generic video coding standard," *Signal Processing: Image Commun.*, vol. 7, pp. 201–209, 1995.
[15] ITU-T Group for Line Transmission of Non-Telephone Signals, "Draft recommendation H.263—Video coding for low bitrate communication," Dec. 1995.

Thomas Sikora (M'93–SM'96), for photograph and biography, see this issue, p. 4.