MPEG-4 - Twice as clever?

Graham Thomas
BBC R&D
Kingswood Warren
graham.thomas@rd.bbc.co.uk
www.bbc.co.uk/rd

MPEG OVERVIEW

MPEG = “Moving Pictures Expert Group”

formally ISO/IEC JTC1/SC29 WG11

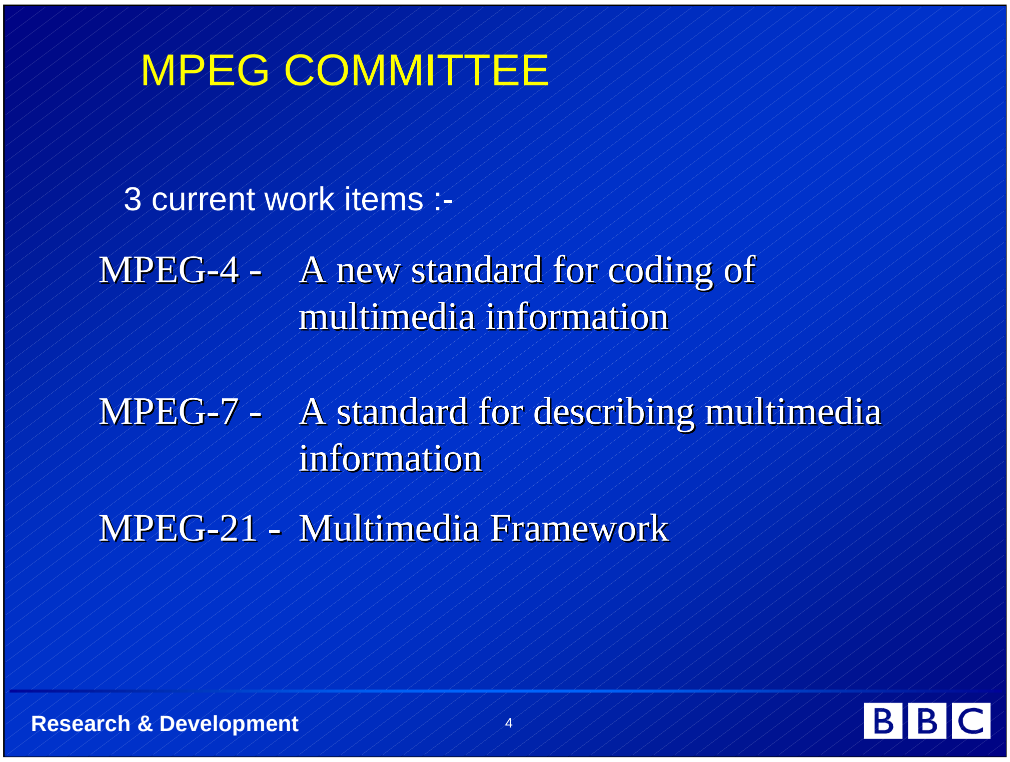
Mission - to propose new standards for multimedia representation

MPEG COMMITEE

2 earlier work items :-

MPEG-1 - coding of A/V information up to 1.5 Mbps for packetised media

MPEG-2 - coding of A/V information up to 50 Mbps for transmission & storage DVD, DVB, DTV, Betacam-SX



MPEG-4 provides standardised ways to:

• represent units of aural, visual or audiovisual content, called "media objects". These media objects can be of natural or synthetic origin; this means they could be recorded with a camera or microphone, or generated with a computer;

• describe the composition of these objects to create compound media objects that form audiovisual scenes;

• multiplex and synchronise the data associated with media objects, so that they can be transported over network channels providing a QoS appropriate for the nature of the specific media objects; and

• interact with the audiovisual scene generated at the receiver's end.

MPEG-7 will provide a stardardised way of querying a database of audio-visual material, supporting applications such as searching of audiovisual archives, or locating a particular kind of programme material in a TV broadcast.

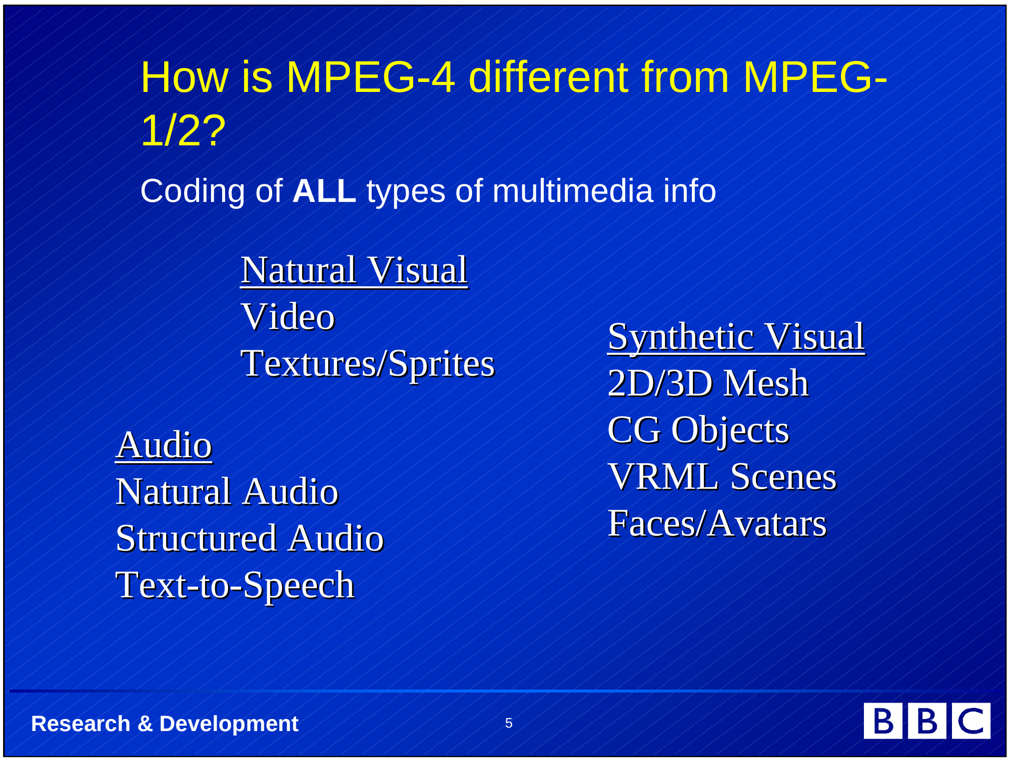
MPEG-21 aims to enable transparent use of multimedia resources across a wide range of networks and devices. Its goal is to create an interoperable multimedia framework by:

•Understanding how the components of the framework are related and identify where gaps in the framework exist

•Develop specifications which allow access, (re)use of and interaction with multimedia objects across networks and/or capable device

•Manage automated rights and payments transactions throughout the value chain

•Respect the privacy of content users.

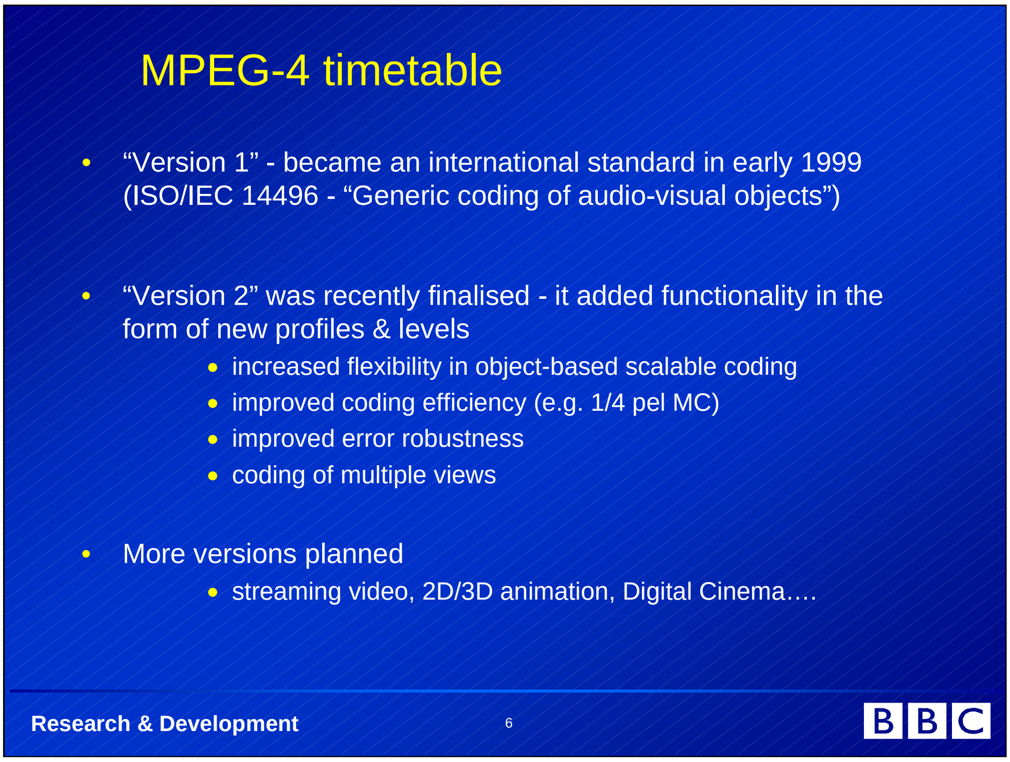


MPEG-4 audiovisual scenes are composed of several media objects, organised in a hierarchical fashion. At the leaves of the hierarchy, we find primitive media objects, such as:

- still images (e.g. as a fixed background),
- video objects (e.g. a talking person - without the background)
- audio objects (e.g. the voice associated with that person)

MPEG-4 standardises a number of such primitive media objects, capable of representing both natural and synthetic content types, which can be either 2- or 3-dimensional. In addition to the media objects mentioned above, MPEG-4 defines the coded representation of objects such as:

- text and graphics;
- talking synthetic heads and associated text used to synthesise the speech and animate the head;
- synthetic sound



MPEG-4, whose formal ISO/IEC designation is ISO/IEC 14496, was finalized in October 1998 and became an International Standard in the first months of 1999. The fully backward compatible extensions under the title of MPEG-4 Version 2 were frozen at the end of 1999, to acquire the formal International Standard Status early 2000. Some work, on extensions in specific domains, is still in progress.

Version 2 of the MPEG-4 systems extends Version 1 to cover issues like extended BIFS functionalities, and Java (MPEG-J) support. Version 2 also specifies a file format to store MPEG-4 content.

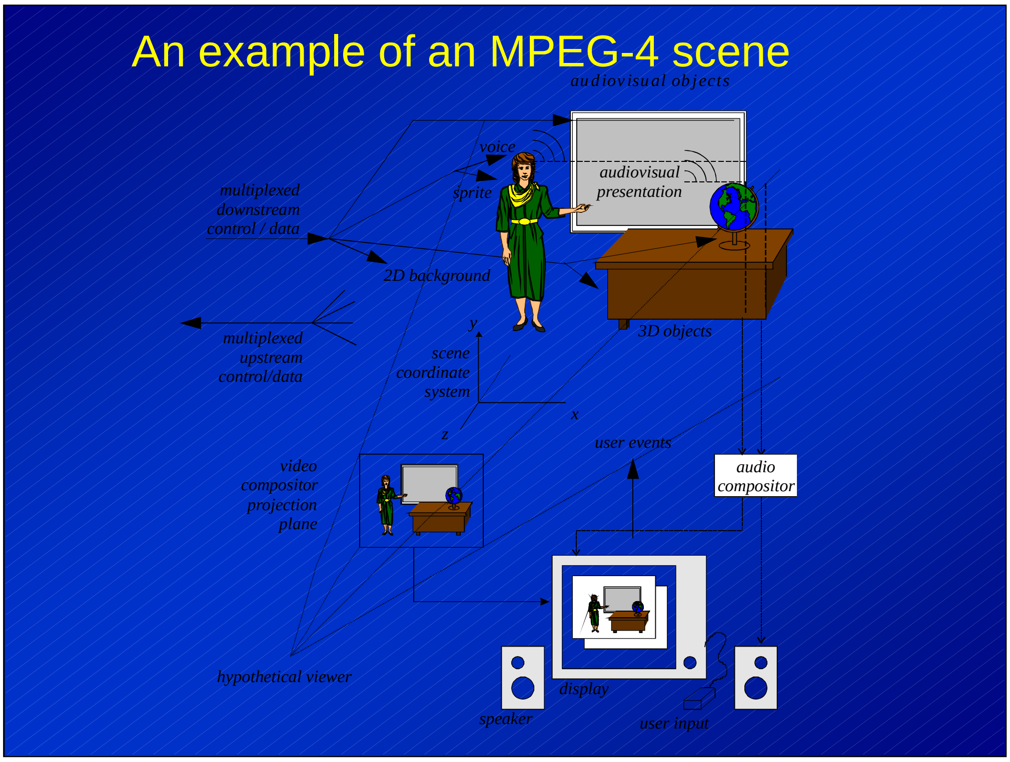
Future extensions are also being considered, including:

•Fine Grain scalability and associated 'Streaming Video Profiles'. Fine Grain Scalability is a tool that allows small quality steps by adding or deleting layers of extra information.

•Tools for usage of MPEG-4 in the Studio. For these tools, care has been taken to preserve some form of compatibility with MPEG-2 profiles. Currently, the Simple Studio Profile is in a balloting phase, this is a profile with I-frame only coding at very high bitrates (several hundred Mbits/s) which employs shape coding. Addition of a Core Studio Profile (with I and P frames) is expected.

•2D and 3D animation coding are under study; a Call for Proposals has been issued that asks for responses by Fall 2000.

•Digital Cinema is under study. This application will require truly lossless coding, and not just the visually lossless that MPEG-4 has provided so far. A Call for Proposals is expected to be issued in Fall 2000.

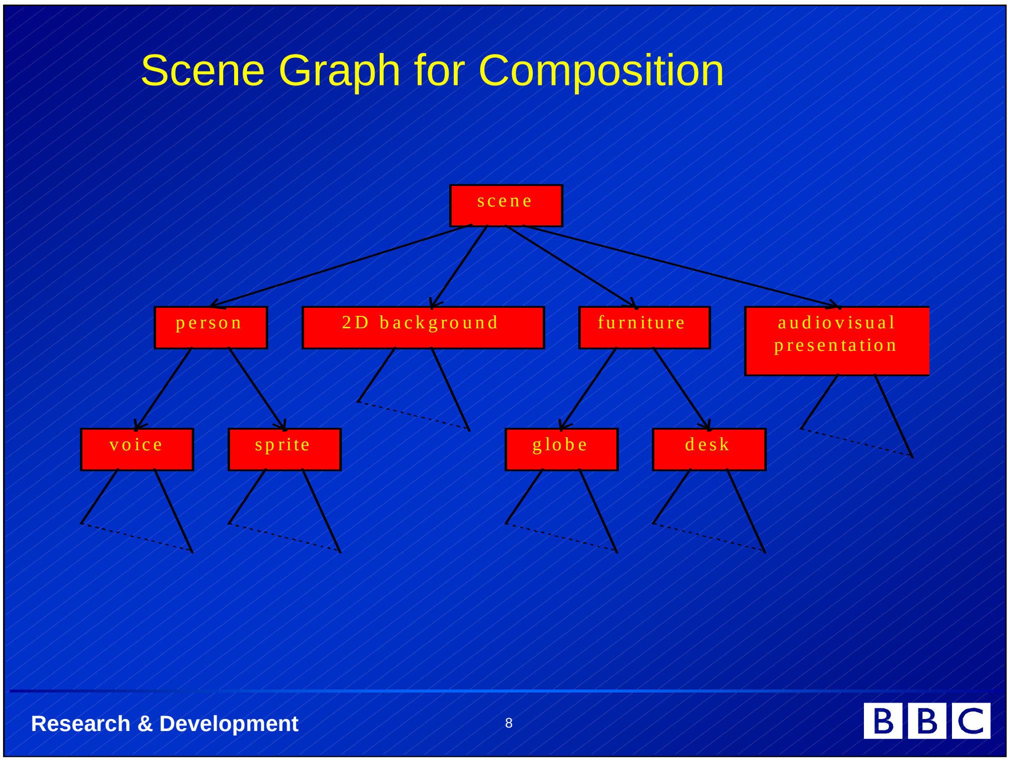


This figure explains the way in which an audiovisual scene in MPEG-4 may be described as composed of individual objects. The figure contains compound media objects that group primitive media objects together. Primitive media objects correspond to leaves in the descriptive tree while compound media objects encompass entire sub-trees. As an example: the visual object corresponding to the talking person and the corresponding voice are tied together to form a new compound media object, containing both the aural and visual components of that talking person.

Such grouping allows authors to construct complex scenes, and enables consumers to manipulate meaningful (sets of) objects.

More generally, MPEG-4 provides a standardized way to describe a scene, allowing for example to:

- place media objects anywhere in a given coordinate system;
- apply transforms to change the geometrical or acoustical appearance of a media object;
- group primitive media objects in order to form compound media objects;
- apply streamed data to media objects, in order to modify their attributes (e.g. a sound, a moving texture belonging to an object; animation parameters driving a synthetic face);
- change, interactively, the user’s viewing and listening points anywhere in the scene.

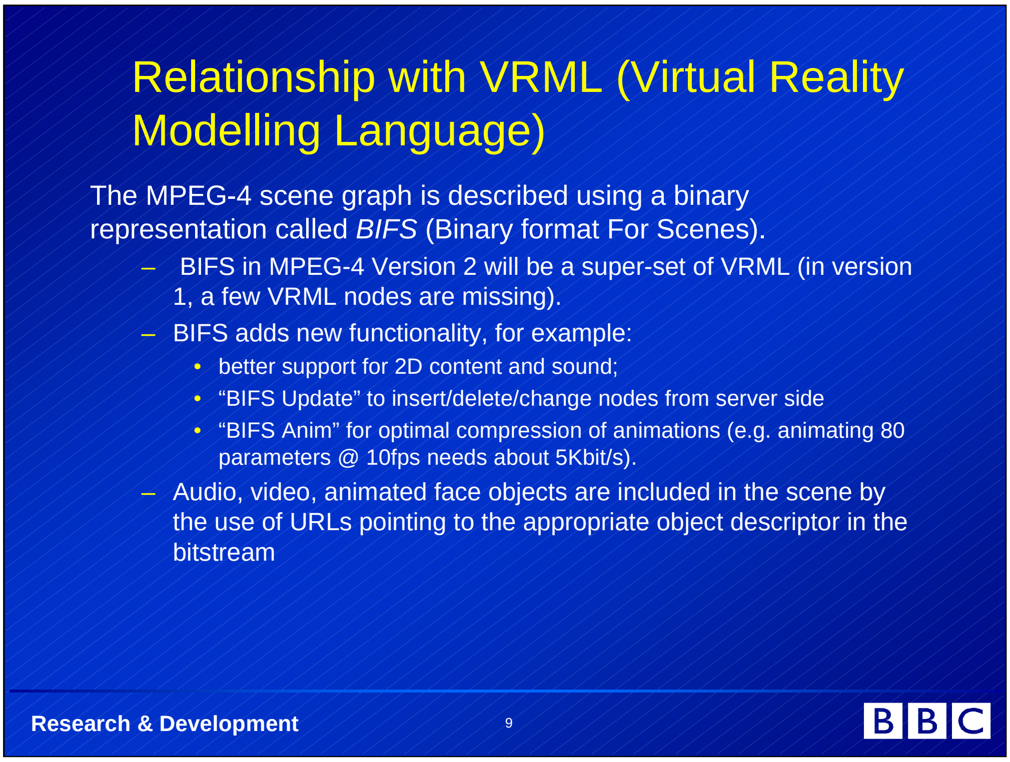


Starting from VRML (the Virtual reality Modeling Language), MPEG has developed a binary language for scene description called BIFS. BIFS stands for **BI**nary **F**ormat for **S**cenes.

An MPEG-4 scene follows a hierarchical structure, which can be represented as a directed acyclic graph. Each node of the graph is a media object, as illustrated in this slide (note that this tree refers back to the scene in the previous slide). The tree structure is not necessarily static; node attributes (e.g., positioning parameters) can be changed while nodes can be added, replaced, or removed.

In the MPEG-4 model, audiovisual objects have both a spatial and a temporal extent. Each media object has a local coordinate system. A *local coordinate system* for an object is one in which the object has a fixed spatio-temporal location and scale. The local coordinate system serves as a handle for manipulating the media object in space and time. Media objects are positioned in a scene by specifying a coordinate transformation from the object's local coordinate system into a global coordinate system defined by one more parent scene description nodes in the tree.

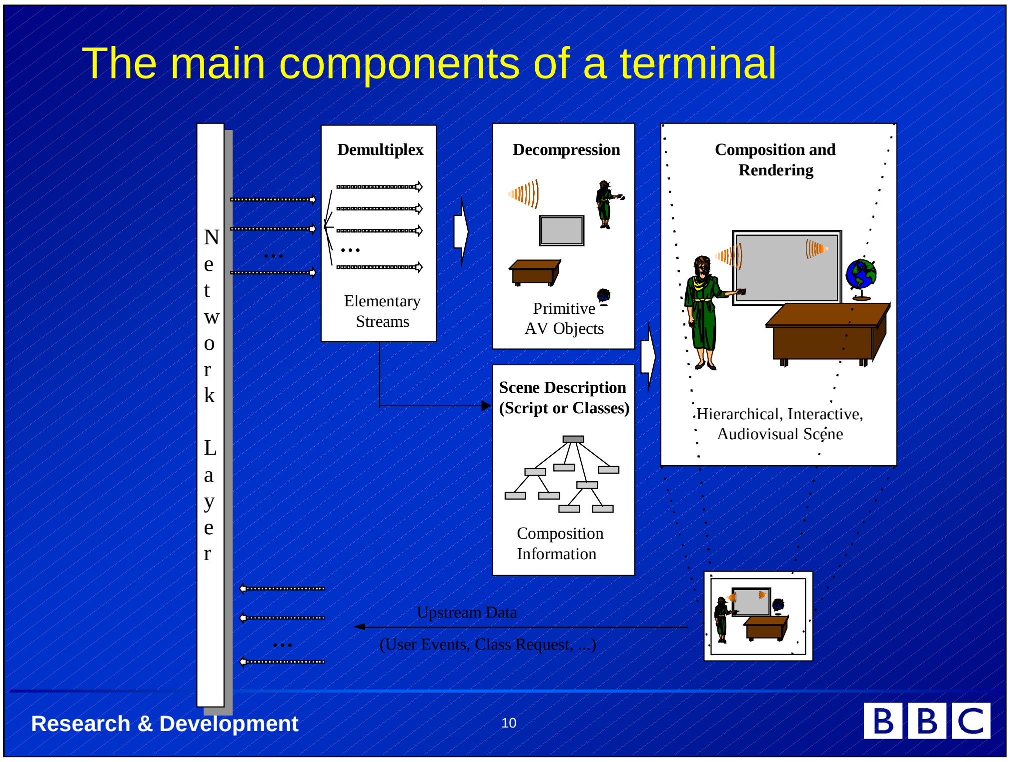
Individual media objects and scene description nodes expose a set of parameters to the composition layer through which part of their behaviour can be controlled. Examples include the pitch of a sound, the colour for a synthetic object, activation or deactivation of enhancement information for scaleable coding, etc.



The BIFS scene model is very similar to VRML. MPEG-4 Version 2 will support all VRML nodes (in version 1 the PROTO and EXTERNPROTO nodes are missing).

BIFS adds new functionality in several areas, for example:

- Improved support for 2D content (static and moving images)
- Considerable enhancements to the VRML 2.0 sound model
- A physical sound rendering model (in Version 2)
- A “Bifs update” mechanism, that allows the server to insert, delete or change any node in the scene. The initial scene is first sent, then updates may be sent to change parts of it. This may be used for many purposes, ranging from simple animation, to adding large parts of the scene. This can be used to provide a progressive-download capability, reducing the latency during initial scene loading.
- An animation mechanism that is efficient for animating the same sets of fields in successive frames. This can be used, for example, to provide a very low-bitrate animated 3D cartoon.



This slide shows how streams coming from the network (or a storage device), as TransMux Streams, are demultiplexed into FlexMux Streams and passed to appropriate FlexMux demultiplexers that retrieve Elementary Streams.

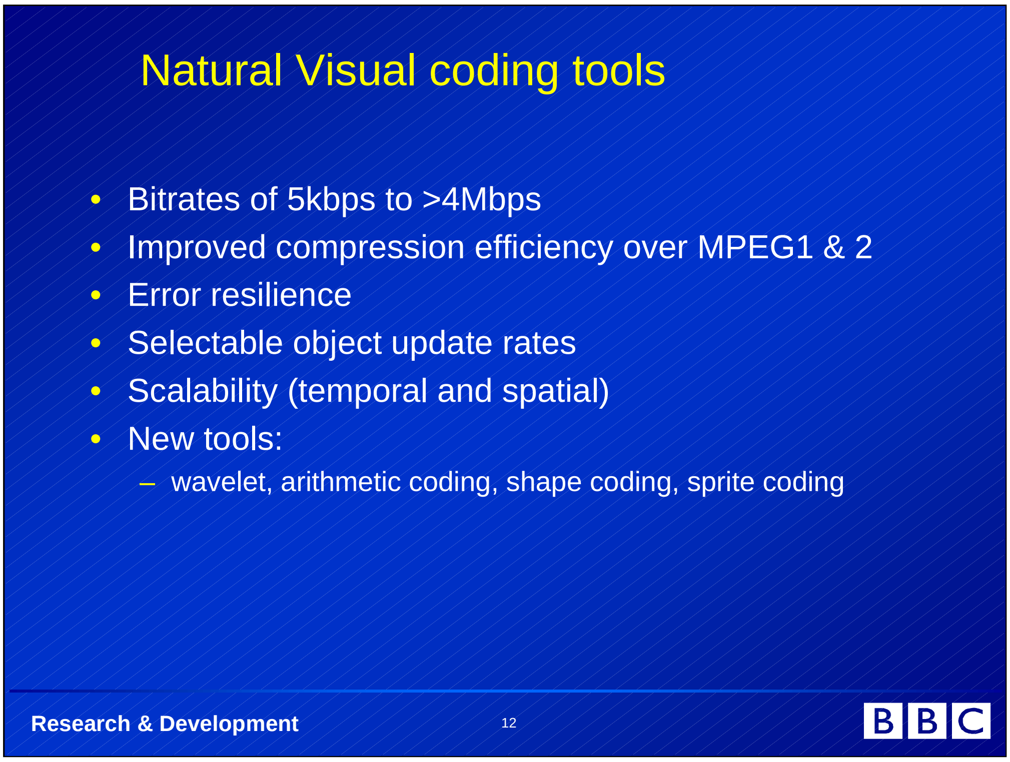
The Elementary Streams (ESs) are parsed and passed to the appropriate decoders. Decoding recovers the data in an AV object from its encoded form and performs the necessary operations to reconstruct the original AV object ready for rendering on the appropriate device. Audio and visual objects are represented in their coded form. The reconstructed AV object is made available to the composition layer for potential use during scene rendering. Decoded AV objects, along with scene description information, are used to compose the scene as described by the author.

Interaction with media objects

The user may be given the possibility to interact with the scene, for example, including:

- change the viewing/listening point of the scene, e.g. by navigation through a scene;
- drag objects in the scene to a different position;
- trigger a cascade of events by clicking on a specific object, e.g. starting or stopping a video stream;
- select the desired language when multiple language tracks are available;

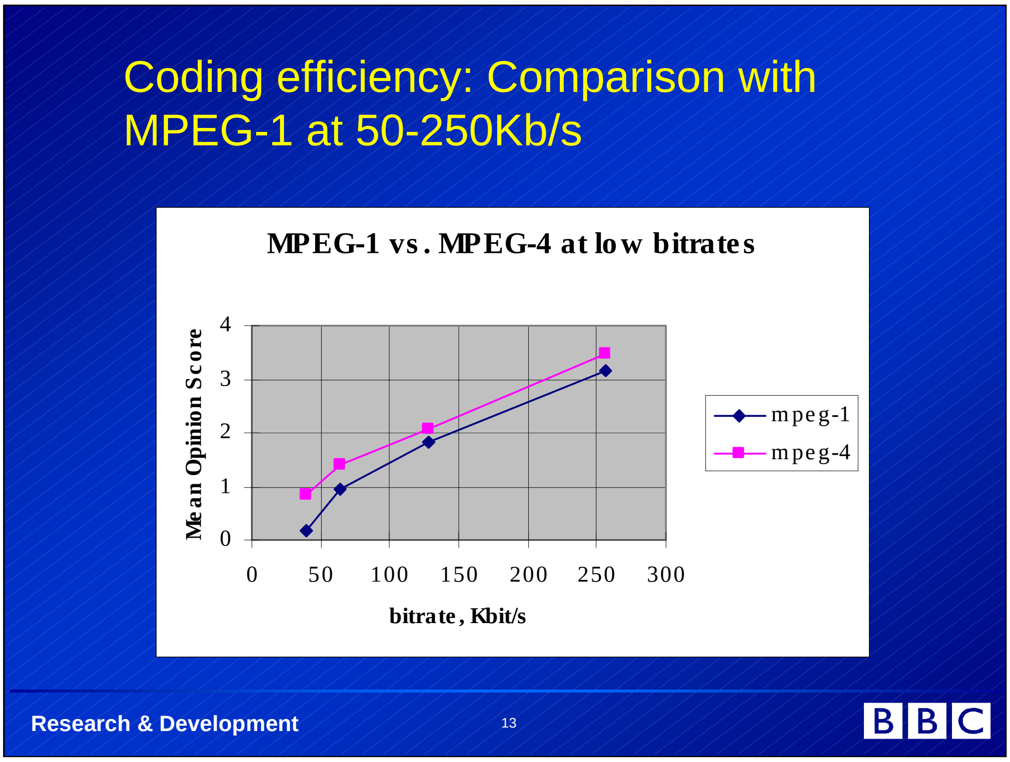
More complex kinds of behaviour can also be triggered, e.g. a virtual phone rings, the user answers and a communication link is established.



The tools for representing natural video in the MPEG-4 visual standard provide standardised core technologies allowing efficient storage, transmission and manipulation of textures, images and video data for multimedia environments. These tools allow the decoding and representation of atomic units of image and video content, called "video objects" (VOs). An example of a VO could be a talking person (without background), which can then be composed with other AVOs (audio-visual objects) to create a scene. Conventional rectangular imagery is handled as a special case of such objects.

The MPEG-4 standard provides tools and algorithms for:

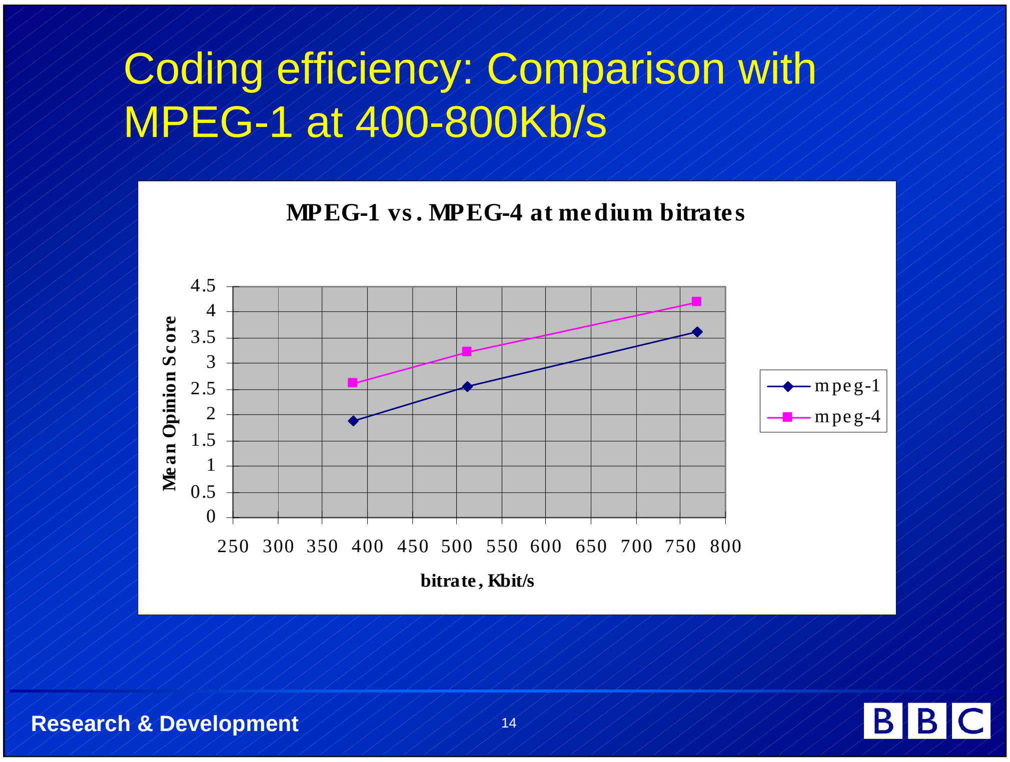
- efficient compression of images and video
- efficient compression of textures for texture mapping on 2-D and 3-D meshes
- efficient compression of implicit 2-D meshes
- efficient compression of time-varying geometry streams that animate meshes
- efficient random access to all types of visual objects
- extended manipulation functionality for images and video sequences
- spatial, temporal and quality scalability
- error robustness and resilience in error prone environments



The tools for representing natural video in the MPEG-4 visual standard provide standardised core technologies allowing efficient storage, transmission and manipulation of textures, images and video data for multimedia environments. These tools allow the decoding and representation of atomic units of image and video content, called “video objects” (VOs). An example of a VO could be a talking person (without background), which can then be composed with other AVOs (audio-visual objects) to create a scene. Conventional rectangular imagery is handled as a special case of such objects.

The MPEG-4 standard provides tools and algorithms for:

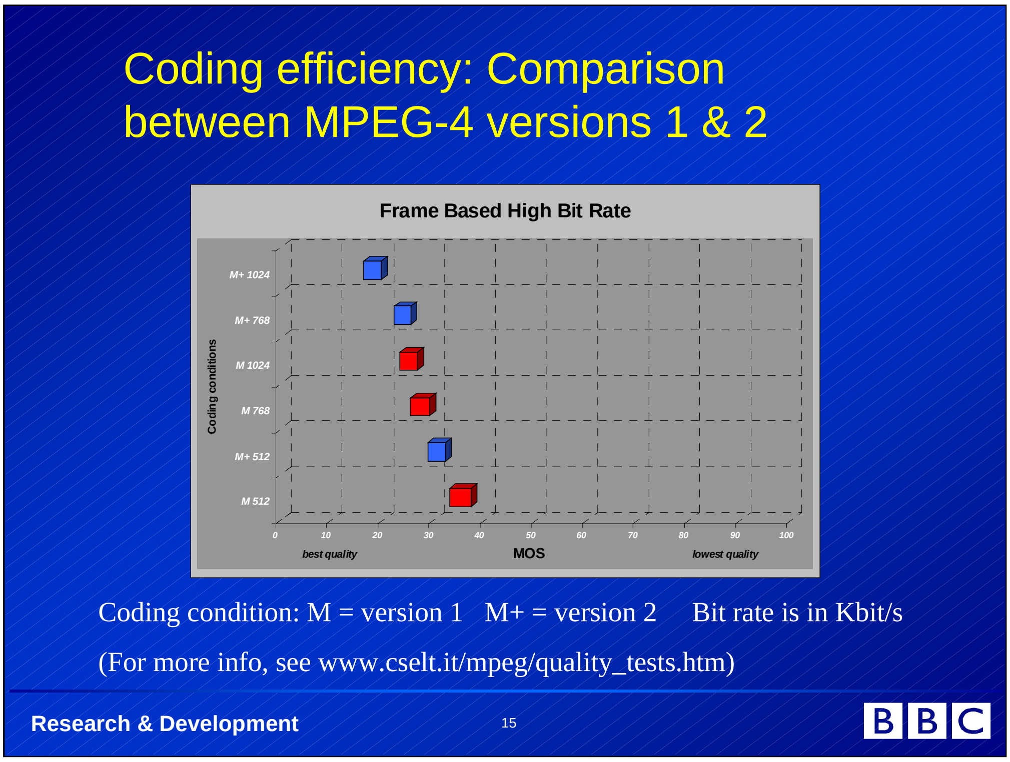
- efficient compression of images and video
- efficient compression of textures for texture mapping on 2-D and 3-D meshes
- efficient compression of implicit 2-D meshes
- efficient compression of time-varying geometry streams that animate meshes
- efficient random access to all types of visual objects
- extended manipulation functionality for images and video sequences
- spatial, temporal and quality scalability
- error robustness and resilience in error prone environments



The tools for representing natural video in the MPEG-4 visual standard provide standardised core technologies allowing efficient storage, transmission and manipulation of textures, images and video data for multimedia environments. These tools allow the decoding and representation of atomic units of image and video content, called "video objects" (VOs). An example of a VO could be a talking person (without background), which can then be composed with other AVOs (audio-visual objects) to create a scene. Conventional rectangular imagery is handled as a special case of such objects.

The MPEG-4 standard provides tools and algorithms for:

- efficient compression of images and video
- efficient compression of textures for texture mapping on 2-D and 3-D meshes
- efficient compression of implicit 2-D meshes
- efficient compression of time-varying geometry streams that animate meshes
- efficient random access to all types of visual objects
- extended manipulation functionality for images and video sequences
- spatial, temporal and quality scalability
- error robustness and resilience in error prone environments



MPEG-4 V.2 improves the motion estimation and compensation of rectangular and arbitrary shaped objects and the texture coding of arbitrary shaped objects significantly. In the area of motion estimation and compensation two new techniques are introduced:

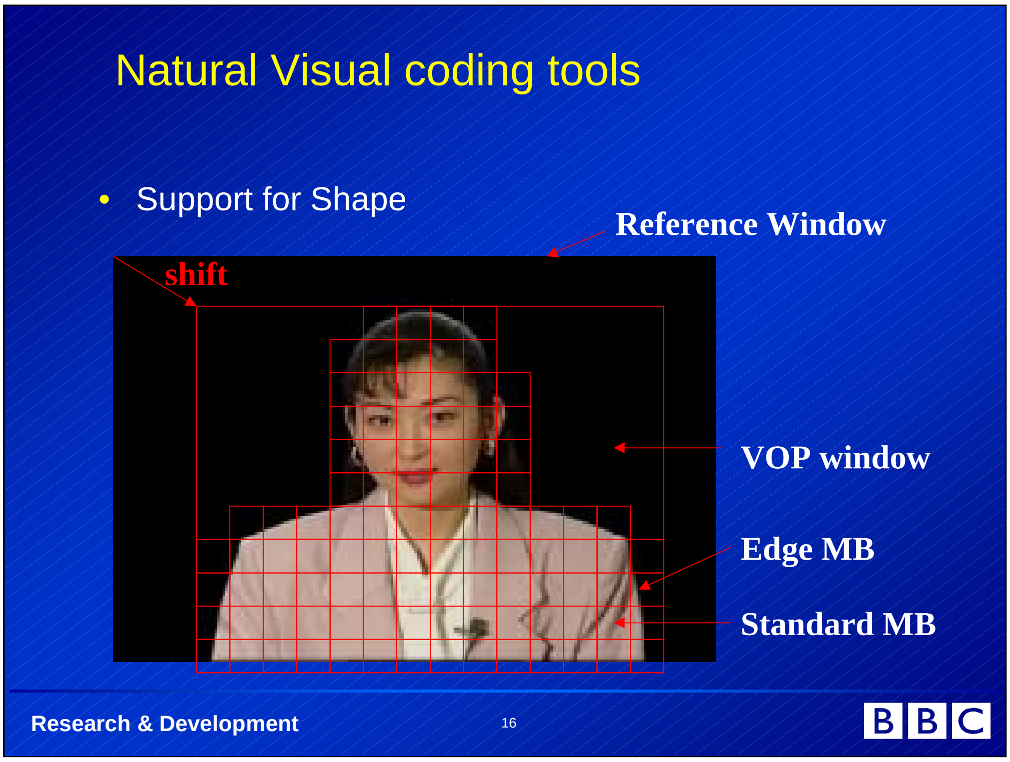
• Global Motion Compensation (GMC): Encoding of the global motion for a object using a small number of parameters.

•Quarter Pel Motion Compensation enhances the precision of the motion compensation scheme.

The graph shows some of the results obtained when testing the "Advanced Coding Efficiency" profile from MPEG-4 Version 2 against the "Main" profile in Version 1. The "double-stimulus continuous quality scale" subjective test method was used. The graph should be read as follows:

- The visual quality (MOS) decreases from the left to the right
- Bars of the same colour correspond to the same profile; the Advanced Coding Efficiency Profile is labelled as M+ (blue) and the Main Profile is labelled as M (red).

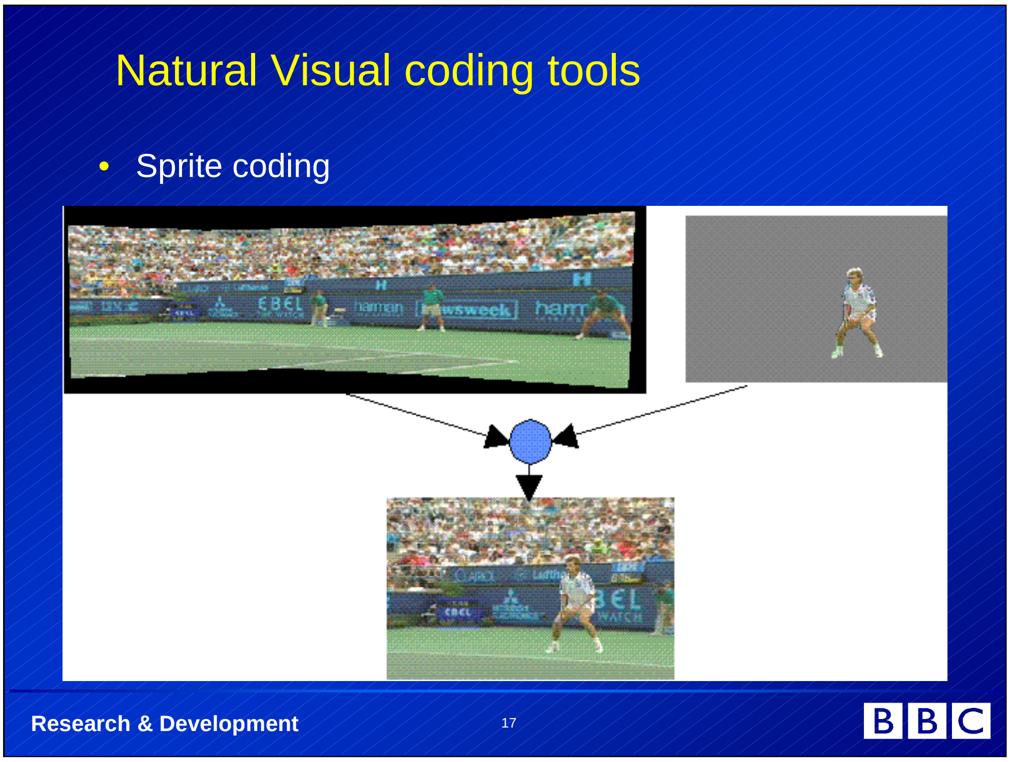
The graph shows that, for the Frame Based at High Bit Rate case, the quality provided by the Advanced Coding Efficiency Profile at 768 Kbps equals the quality provided by Main Profile at 1024 Kbps. Tests were also conducted at lower bitrates, and showed improvements of up to 50% in bitrate.



One of the major differences from MPEG-1 and MPEG-2 is that MPEG-4 supports arbitrary-shaped video objects. This is achieved by coding a mask, or “shape” signal, in addition to the video signal, indicating which pixels of the image contain the object. Two kinds of shape signal are available - binary and greyscale shape. With a binary shape signal, each pixel is either inside the object or outside it; a greyscale shape signal supports varying degrees of transparency, exactly like a key or mask signal in a vision mixer.

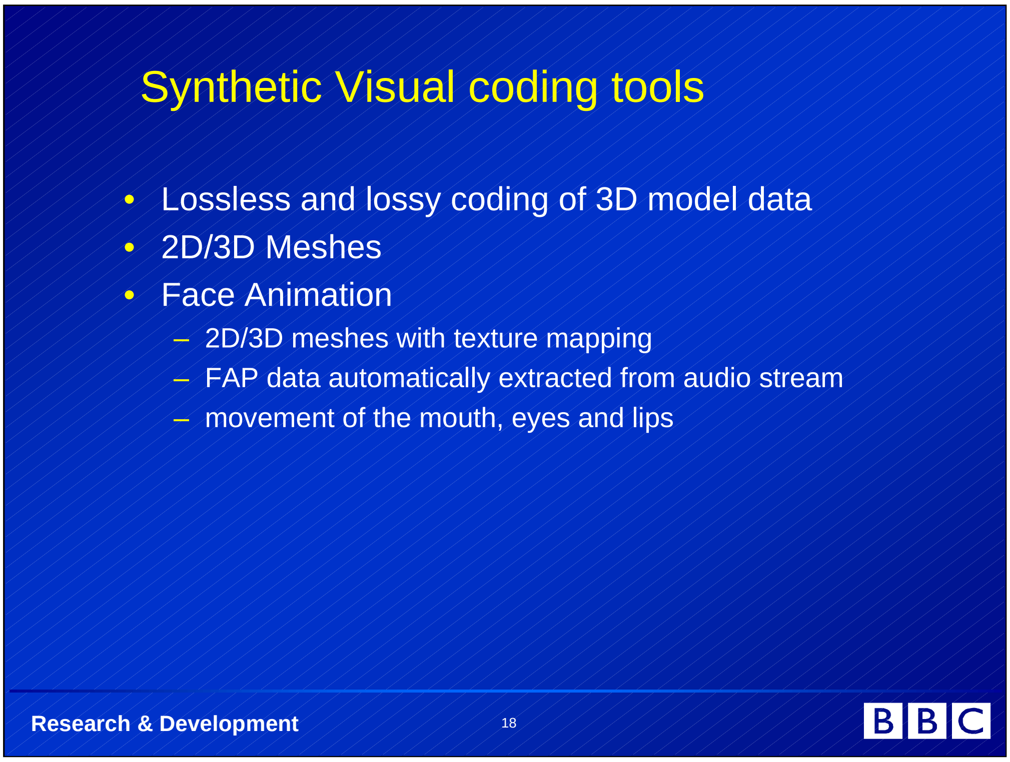
The video and shape signals presented to the coder are fundamentally rectangular. The coder analyses the shape signal to find the smallest rectangular area of the image that encompasses the active image area. Each block in this rectangular region is then coded as either a standard macroblock (wholly within the object), an edge macroblock (where the shape signal is varying), or a transparent macroblock.

Note that the method for generating the shape signal is not specified by MPEG-4. Typically, shape signals may be obtained from a chromakey process, from a (semi)automated image segmentation process, or from a drawing or animation process used to generate the video.



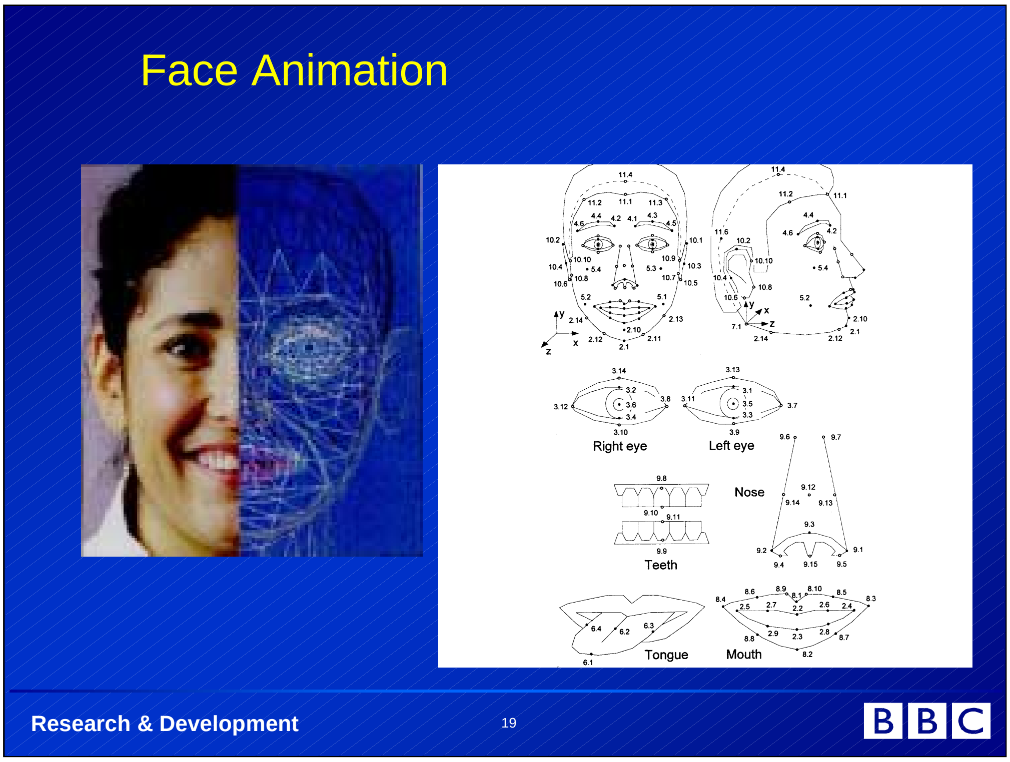
This slide depicts the basic concept for coding an MPEG-4 video sequence using a sprite panorama image. This coding method can give very high compression compared to MPEG-1 and -2 for scenes containing a large background image which remains unchanged apart from camera panning, tilting and zooming.

In this example, it is assumed that the foreground object (tennis player, image top right) can be segmented from the background and that the sprite panorama image can be extracted from the sequence prior to coding. (A sprite panorama is a still image that describes as a static image the content of the background over all frames in the sequence). The large panorama sprite image is transmitted to the receiver only once as first frame of the sequence to describe the background – the sprite remains is stored in a sprite buffer. In each consecutive frame only the camera parameters relevant for the background are transmitted to the receiver. This allows the receiver to reconstruct the background image for each frame in the sequence based on the sprite. The moving foreground object is transmitted separately as an arbitrary-shape video object. The receiver composes both the foreground and background images to reconstruct each frame (bottom picture in figure below). For low delay applications it is possible to transmit the sprite in multiple smaller pieces over consecutive frames or to build up the sprite at the decoder progressively.



2D and 3D computer graphic objects (e.g. cones, rectangles, cubes) are represented in the scene graph by nodes which specify the parameters of each object (size, colour, orientation, etc.) in the same way as these objects are represented in VRML.

More general kinds of objects may be represented by 2D meshes (and 3D meshes in version 2). In addition, there are special tools for representing animated faces. Body animation will be added in version 2.



The ‘facial animation object’ can be used to render an animated face. The shape, texture and expressions of the face are controlled by Facial Definition Parameters (FDPs) and/or Facial Animation Parameters (FAPs). Upon construction, the face object contains a generic face with a neutral expression. This face can already be rendered. It can also immediately receive the animation parameters from the bitstream, which will produce animation of the face: expressions, speech etc. Meanwhile, definition parameters can be sent to change the appearance of the face from something generic to a particular face with its own shape and (optionally) texture. If so desired, a complete face model can be downloaded via the FDP set.

Face Animation in MPEG-4 Version 1 provides for highly efficient coding of animation parameters that can drive an unlimited range of face models. The models themselves are not normative, although (see above) there are normative tools to describe the appearance of the model. Frame-based and temporal-DCT coding of a large collection of FAPs can be used for accurate speech articulation. Viseme and expression parameters are used to code specific speech configurations of the lips and the mood of the speaker.

Face Animation

- Demonstrations courtesy of AT&T
  - http://www.research.att.com/~osterman/AnimatedHead/
  - Policeman
  - Pinnochio
  - Real face

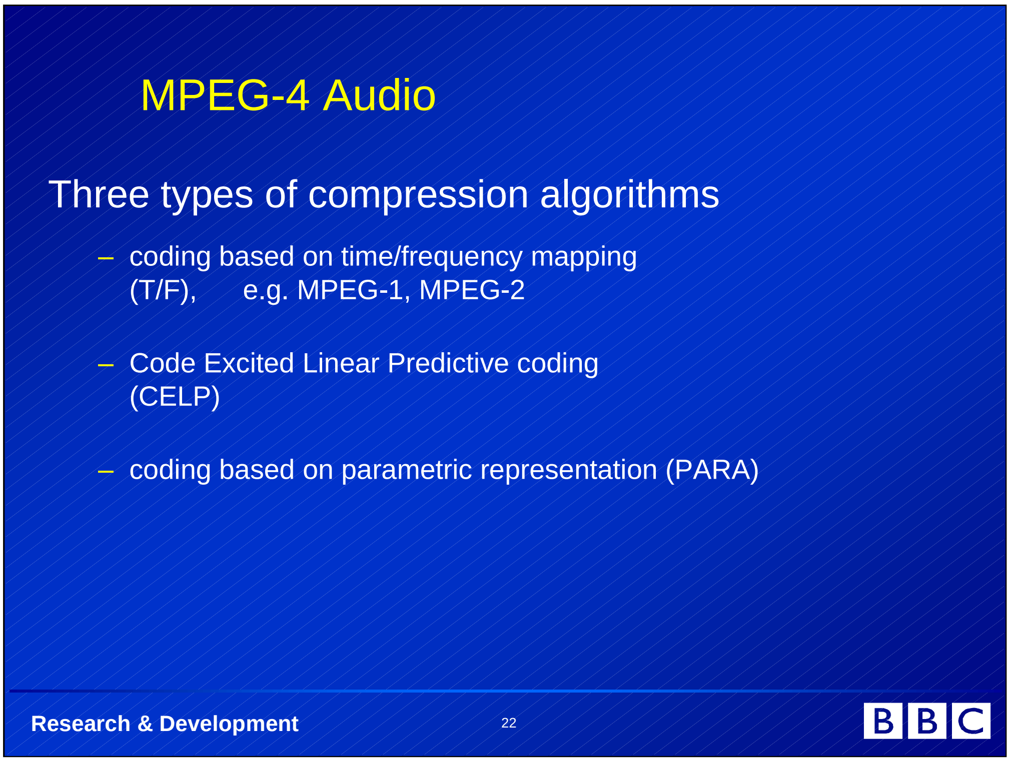
2D animated meshes

By deforming the mesh, the fish can be animated very efficiently, and be made to ‘swim’.

Research & Development 21 BBC

A dynamic mesh is a forward tracking mesh, where the node points of the initial mesh track image features forward in time by their respective motion vectors. The initial mesh may be regular, or can be adapted to the image content, which is called a *content-based mesh*. 2-D content-based mesh modelling then corresponds to non-uniform sampling of the motion field at a number of salient feature points (node points) along the contour and interior of a video object. Methods for selection and tracking of these node points are not subject to standardisation.

In 2-D mesh-based texture mapping, triangular patches in the current frame are deformed by the movements of the node points into triangular patches in the reference frame. The texture inside each patch in the reference frame is *warped* onto the current frame using a parametric mapping, defined as a function of the node point motion vectors. For triangular meshes, the affine mapping is a common choice. Its linear form implies texture mapping with low computational complexity. Affine mappings can model translation, rotation, scaling, reflection and shear, and preserve straight lines. The degrees of freedom given by the three motion vectors of the vertices of a triangle match with the six parameters of the affine mapping. This implies that the original 2-D motion field can be compactly represented by the motion of the node points, from which a continuous, piece-wise affine motion field can be reconstructed. At the same time, the mesh structure constrains movements of adjacent image patches. Therefore, meshes are well-suited to represent mildly deformable but spatially continuous motion fields.

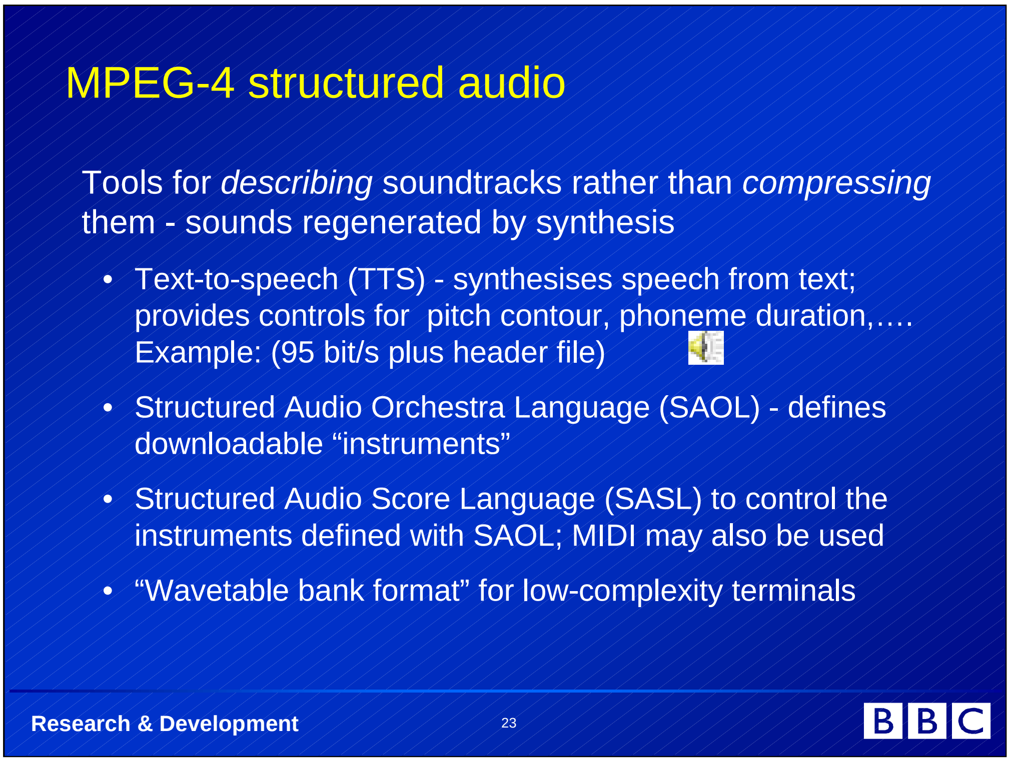


Three types of coding method are available for coding natural audio:

For general audio coding at bitrates at and above 6 kbit/s, time/frequency transform coding techniques, namely TwinVQ and AAC (Advanced Audio Coding), are applied. The audio signals in this region typically have sampling frequencies starting at 8 kHz. AAC is capable of providing quality indistinguishable from that of the source, from mono to multichannel surround sound,at considerably lower bit-rates than the mp3 audio format widely used on the Internet today.

Code Excited Linear Predictive (CELP) coding is provided for speech coding at an operating bitrate of 4 - 24 kbit/s. In CELP coding, two sampling rates, 8 and 16 kHz, are used to support narrowband and wideband speech, respectively.

A form of parametric coding called Harmonic Vector eXcitation Coding (HVXC) is provided for speech coding at a recommended operating bitrate of 2 - 4 kbit/s. In addition, HVXC can operate down to an average of around 1.2 kbit/s in its variable bitrate mode.



MPEG-4 defines decoders for generating sound based on several kinds of 'structured' inputs. Text input is converted to speech in the Text-To-Speech (TTS) decoder, while more general sounds including music may be normatively synthesized. Synthetic music may be delivered at extremely low bitrates while still describing an exact sound signal.

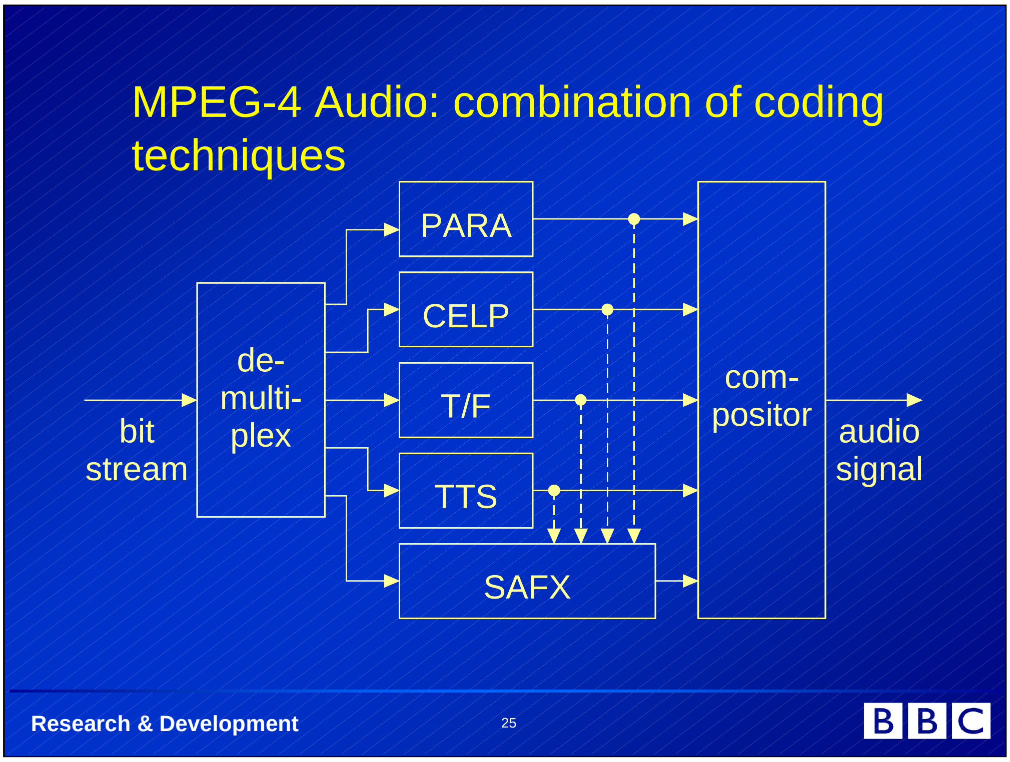
TTS coders bitrates range from 200 bit/s to 1.2 Kbit/s, which allows a text or a text with prosodic parameters (pitch contour, phoneme duration, and so on) as its inputs to generate intelligible synthetic speech. It supports the generation of parameters that can be used to allow synchronization to associated face animation, international languages for text and international symbols for phonemes.

The Structured Audio tools decode input data and produce output sounds. This decoding is driven by a special synthesis language called SAOL (Structured Audio Orchestra Language) standardized as a part of MPEG-4. This language is used to define an "orchestra" made up of "instruments" (downloaded in the bitstream, not fixed in the terminal) which create and process control data.

Control of the synthesis is accomplished by downloading "scores" or "scripts" in the bitstream, downloaded in a language called SASL (Structured Audio Score Language). This can be used to create new sounds, and can also include additional control information for modifying an existing sound.

Structured audio example

- MPEG-4 soundtrack providing
  - natural and synthetic speech
  - synthetic music and sound effects
  - mixing, reverb, etc in BIFS with SAOL
- Use SAOL to describe effects (AudioFX)
- Mix and synchronise multi-channel sound
- Position sounds in virtual 3-D space
- **Bitrate = 16 kbps**



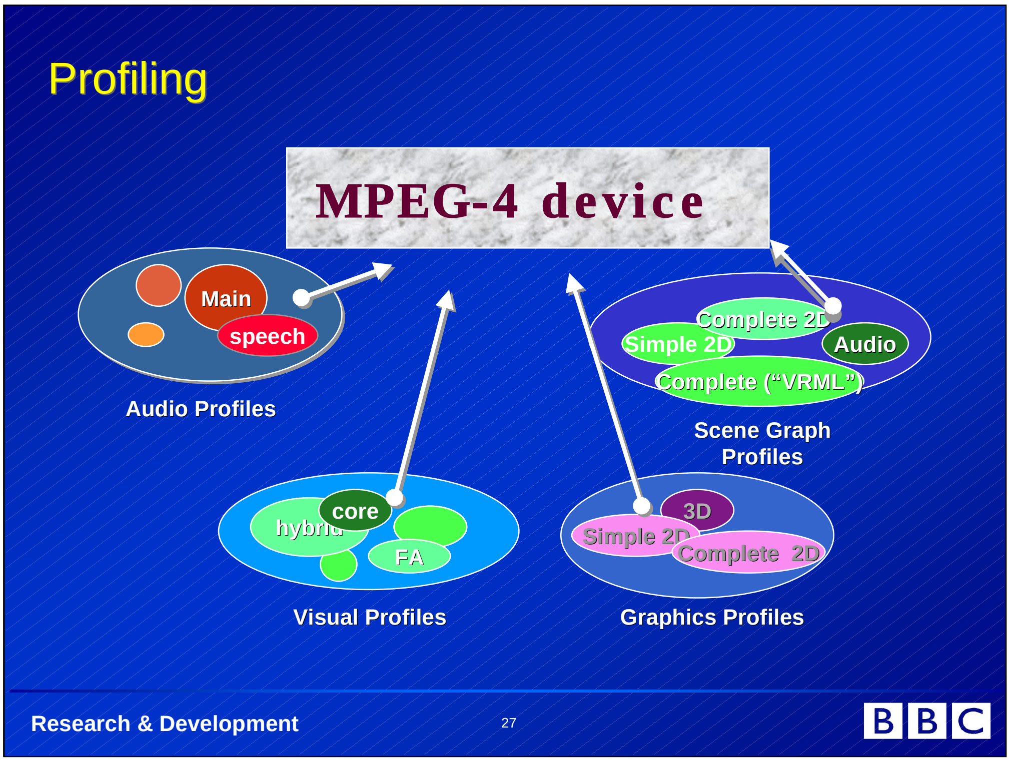
An MPEG-4 bitstream may contain many separate coded audio signals. Each signal is decoded by the corresponding decoder, and the resulting signals are combined after decoding. Each coding tool should be used for the kind of material for which it was designed - for example, very poor results will be obtained if the CELP coding tool designed for speech coding is used for music.

The Structured audio effects tool (SAFX) may be used to apply some effects, such as reverberation, to audio signals from other decoders.

Summary of MPEG-4 audio

MPEG-4 Audio supports coding of

- music signals with high complexity content
  - at bit rates down to 6 kbps
- music signals with low complexity content
  - at bit rates down to 4 kbps
- speech signals
  - at bit rates down to 2 kbps
- audio scenes
  - object based structured representation



MPEG-4 provides a large and rich set of tools for the coding of audio-visual objects. In order to allow effective implementations of the standard, subsets of the MPEG-4 Systems, Visual, and Audio tool sets have been identified, that can be used for specific applications. These subsets, called 'Profiles', limit the tool set a decoder has to implement. For each of these Profiles, one or more Levels have been set, restricting the computational complexity. The approach is similar to MPEG-2, where the most well known Profile/Level combination is 'Main Profile @ Main Level'. A Profile@Level combination allows:

- a codec builder to implement only the subset of the standard he needs, while maintaining interworking with other MPEG-4 devices built to the same combination, and
- checking whether MPEG-4 devices comply with the standard ('conformance testing').

Profiles exist for various types of media content (audio, visual, and graphics) and for scene descriptions. MPEG does not prescribe or advise combinations of these Profiles, but care has been taken that good matches exist between the different areas.

Scene Graph profiles

These specify which BIFS tools are available:

	Scene Graph Profiles			
Scene Graph Tools	Audio	Simple 2D	Complete 2D	Complete
Anchor			X	X
AudioBuffer	X		X	X
AudioDelay	X		X	X
AudioFX	X		X	X
AudioMix	X		X	X
AudioSwitch	X		X	X
Billboard				X
Collision				X
Composite2DTexture			X	X
Composite3DTexture				X
Form			X	X
Group	X	X	X	X
Inline			X	X
Layer2D			X	X
Layer3D				X
Layout			X	X
…another 39 entries…				

Research & Development 28 BBC

The scene graph profiles specify the scene graph elements of the BIFS tool that are allowed. These elements provide the means to describe the spatio-temporal locations, the hierarchical dependencies as well as the behaviors of audio-visual objects in a scene.

The Audio scene graph profile provides for a set of BIFS scene graph elements for usage in audio-only applications. The Audio scene graph profile supports applications like broadcast radio.

The Simple 2D scene graph profile provides for only those BIFS scene graph elements necessary to place one or more audio-visual objects in a scene. The Simple 2D scene graph profile allows presentation of audio-visual content with potential update of the complete scene but no interaction capabilities. The Simple 2D scene graph profile supports applications like broadcast television.

The Complete 2D scene graph profile provides for all the 2D scene description elements of the BIFS tool. It supports features such as 2D transformations and alpha blending. The Complete 2D scene graph profile enables 2D applications that require extensive and customized interactivity.

The Complete scene graph profile provides the complete set of scene graph elements of the BIFS tool. The Complete scene graph profile will enable applications like dynamic virtual 3D world and games.

Graphics profiles

These specify which BIFS nodes are available:

	Graphics Profiles		
Graphics Tools	Simple 2D	Complete 2D	Complete
Appearance	X	X	X
Box			X
Bitmap	X	X	X
Background			X
Background2D		X	X
Circle		X	X
Color		X	X
Cone			X
Coordinate			X
Coordinate2D		X	X
Curve2D		X	X
Cylinder			X
DirectionalLight			X
… another 32 entries…			

Research & Development 29 BBC

The graphics profiles specify the graphics elements of the BIFS tool that are allowed. These elements provide means to represent graphics visual objects in a scene. Profiling of graphics elements of the BIFS tool serves to restrict the memory requirements for the storage of the graphical elements as well as to restrict the computational complexities of composition and rendering processes.

The Simple 2D graphics profile provides for only those graphics elements of the BIFStools that are necessary to place one or more visual objects in a scene. This profile would be suitable, for example, for a video phone that only needs to display a number of decoded video images, with no graphics.

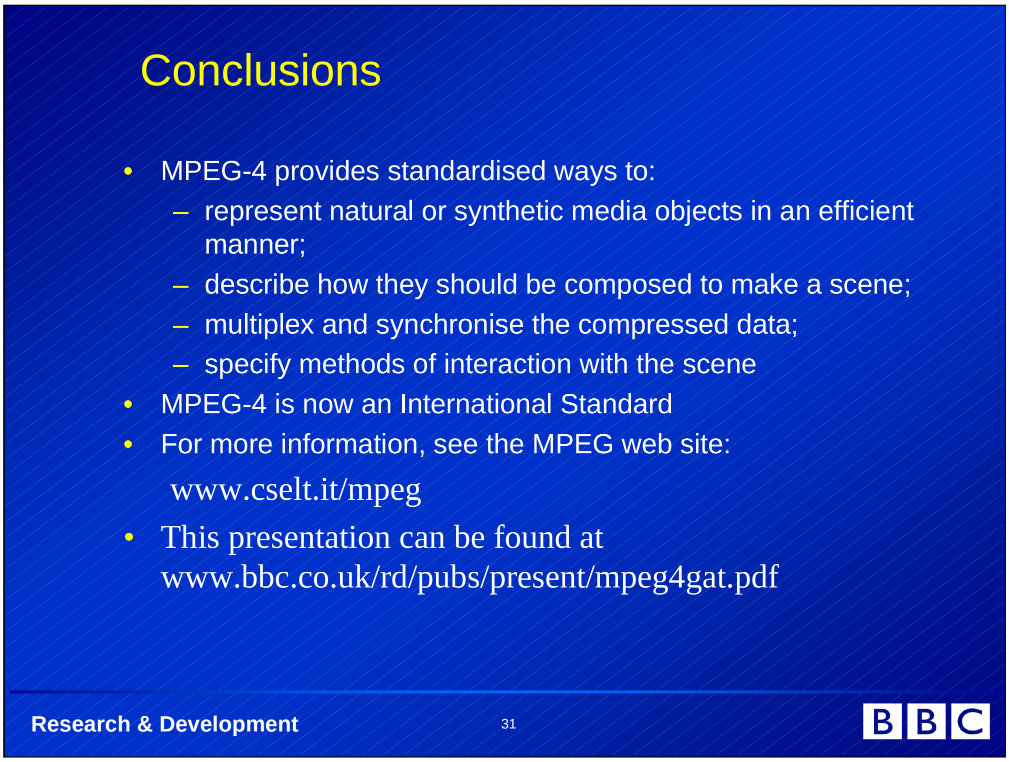
The Complete 2D graphics profile provides two-dimensional graphics functionalities and supports features such as arbitrary two-dimensional graphics and text, possibly in conjunction with visual objects.

The Complete graphics profile provides advanced graphical elements such as elevation grids and extrusions and allows creating content with sophisticated lighting. The Complete Graphics profile enables applications such as complex virtual worlds that exhibit a high degree of realism.

No levels are yet defined for any of these profiles. It is anticipated that levels will be defined in the future, by means of an amendment to the standard.

MPEG-4 Applications

- Unlike MPEG-1 & -2 no “killer” application
- Possible applications include:
  – Enhanced TV-type broadcasts, supporting e.g. interaction, additional services (e.g. signing), 3D viewing (MPEG-4 data can be added to MPEG-2 transport streams)
  – Low bitrate communications - mobile multimedia, teleconferencing, remote surveillance, digital AM radio
  – Collaborative virtual environments - like teleconferencing but with shared 3D models



Detailed information about all of the MPEG standards, including MPEG-4 and MPEG-7, can be obtained from the MPEG home page on the World Wide Web, www.cselt.it/mpeg. Some other MPEG-related sites and their special interests are:

garuda.imag.fr/MPEG4/ (systems).

www.hhi.de/mpeg-video (natural video coding).

www.es.com/mpeg4-snhc/ (synthetic visual coding).

www.tnt.uni-hannover.de/project/mpeg/audio/ (audio coding).

sound.media.mit.edu/mpeg4/ (structured audio).

The May and June 1998 issues of the IEEE Proceedings include a number of articles that supply background information on the MPEG-4.

For more information on VRML (Virtual Reality Modeling Language), see the website of the Web3d Consortium (formerly VRML Consortium), www.web3d.org.