An Introduction to MPEG - 4

R. Venkatesh Babu
Brief History

- STARTED : JULY 1993
- COMMITTEE DRAFT LEVEL : NOV 1997
- INTERNATIONAL STD LEVEL : APRIL 1999
Comparison of MPEG -1,2 & 4

- **MPEG-1**: STANDARD FOR STORAGE AND RETRIEVAL OF AV MATERIAL ON STORAGE MEDIA
- **MPEG-2**: STANDARD FOR DIGITAL TV
- **MPEG-4**: ADDRESSES THE CODED REPRESENTATION OF BOTH NATURAL AND SYNTHETIC AV OBJECTS
Important Objectives

• COMMON TECHNOLOGY
• ALLOWING MORE AND DIFFERENT INTERACTIVITY
• INTEGRATING NATURAL AND SYNTHETIC CONTENT
• COVERING WIDE RANGE OF ACCESS CONDITIONS
• INTELLECTUAL PROPERTY MANAGEMENT AND PROTECTION (IPMP)
Basics of MPEG-4

- CODED REPRESENTATION OF AV OBJECTS
- SYNTACTIC DESCRIPTION OF AV OBJECTS
  - AUDIO AND VISUAL
  - NATURAL AND SYNTHETIC
- SCENE DESCRIPTION BY A COMPOSITOR
  - USING BIFS
- ALLOWING INTERACTIVITY
The corresponding scene graph

```
root
  /     /     /
background voice-over baby bream graphics
   /     /     /        /
sprite music video bubbles sound soil fish seaweed
```

- root
- background
  - sprite
  - music
- voice-over
- baby
  - video
  - bubbles
  - sound
- bream
- graphics
  - soil
  - fish
  - seaweed
     /
    blue
    red
An MPEG-4 Scene
Scene Graph
Scene Description

- ELEMENTARY STREAM CONVEYS COMPOSITION INFORMATION
- ALLOWS TO MODIFY SPATIO-TEMPORAL LAYOUT
- ENCODE THE SCENE DESCRIPTION IN BINARY FORMAT (BIFS)
MPEG-4 Functionalities

- OBJECT BASED INTERACTIVITY
- COMPRESSION
- UNIVERSAL ACCESS
- ROBUSTNESS IN ERROR PRONE ENVIRONMENT
- CONTENT BASED SCALABILITY
Interactivity with Media Objects

• AUTHOR MAY ALLOW THE USER TO
  – CHANGE THE VIEWING & LISTENING POSITION OF THE SCENE
  – PLACE MEDIA OBJECTS ANYWHERE IN A GIVEN COORDINATE SYSTEM
  – ADD OR REMOVE OBJECTS
  – CHANGE THE POSITION AND SIZE OF THE OBJECT
  – APPLY STREAMED DATA TO MEDIA OBJECTS
  – SELECT LANGUAGE WHEN MULTIPLE LANGUAGES ARE AVAILABLE
Parts of MPEG-4 Standard

1. Systems
2. Visual
3. Audio
6. DMIF

- presentation
- decoding
- demux & buffer
- transport interface
- not in standard

4. Conformance
5. Reference SW
Various Layers of MPEG-4

- Compression Layer
  *Elementary Stream Interface (ESI)*

- Sync Layer
  *DMIF Application Interface (DAI)*

- Delivery Layer
DMIF
Delivery Multimedia Integration Framework

The Broadcast Technology
- Cable, Satellite, etc

The Interactive Network Technology
- Internet, ATM, etc

The Disk Technology
- CD, DVD, etc
Return Channel

Network

TransMux

FlexMux

Elementary Streams

Return Channel Encoding

DAI

User to User Interaction
Streaming Data for Media Objects
Objects

- Objects have temporal and spatial extensions

<table>
<thead>
<tr>
<th>object</th>
<th>Object plane 1</th>
<th>Object plane 2</th>
<th>.......</th>
<th>Object plane n</th>
</tr>
</thead>
</table>

- Visual objects and Audio objects
- Natural objects and Synthesized objects
- Primitive objects and Compound objects
Object Descriptors
Elementary Stream Descriptor

- **CONTAIN POINTERS TO**
  - SCALABILITY CODED CONTENT STREAMS
  - ALTERNATE QUALITY CONTENT STREAMS
  - OBJECT CONTENT INFORMATION
- **AND SUB DESCRIPTORS FOR**
  - DECODER CONFIGURATION
Elementary Streams Associated With MPEG-4 Objects
Audio

• EFFICIENT REPRESENTATION OF
  – SPEECH SIGNALS
  – SYNTHESIZED SPEECH SIGNALS
  – GENERAL AUDIO
  – SYNTHESIZED AUDIO

• ADDITIONAL FUNCTIONALITY
  – SPEED CHANGE
  – PITCH CHANGE
  – BIT RATE SCALABILITY
  – BANDWIDTH SCALABILITY
  – AUDIO EFFECTS
    (MIXING, REVERBERATION, SPATIALIZATION)
Audio Coding Techniques

To Achieve the Highest Quality

• Parametric coding
• Linear Predictive coding (\textit{LPC})
• Time/frequency coding

To Provide Extra Functionality

• Synthetic/natural hybrid coding (\textit{SNHC})
• Text-to-speech integration
Coding of Visual Objects

- Efficient Compression of Images and Videos
- Efficient compression of textures
- Efficient compression of implicit 2D meshes
- Efficient compression of time-varying geometry streams that animate meshes
- Efficient random access to all types of VO
- Content-based coding of images and video
- Content-based scalability of textures, images and video
- Spatial, temporal and quality scalability
- Error robustness and resilience
Compression Efficiency

• VIDEO
  – EFFICIENT COMPRESSION FOR ALL BIT RATES
  – COMPACT CODING FOR WIDE RANGE OF QUALITY
  – EFFICIENCY IMPROVED BY SADCT FOR ARBITRARY SHAPED OBJECTS
  – EFFICIENT COMPRESSION OF TEXTURES FOR TEXTURE MAPPING ON 2D & 3D
Synthetic Objects

- Object types
  - Face
  - Body
  - Text
  - 2D, 3D

- 2D/3D mesh representation and compression
- Texture mapping
- Text Overlay
- Image and Graphics overlay
- Video Object Tracking
Neutral face:
- Iris diameter
- Eye separation
- Eye-nose separation
- Mouth-nose separation
- Mouth width
- Angle unit

Facial definition parameter set (FDP)
Facial animation parameter set (FAP)
Face object
Facial Animation Demo

With 26 FAPs
At 1.1 Kbits/sec
Text

- Capability for text overlay
- Capability to animate text
- Controllable by user or author
- Capability for spatial and temporal location and manipulation
- Synchronization with spatial and temporal events in associated audio and video

Usage:
- Program guide
- Hyperlinked text in video
- Real-time insertion of advertisements
Object-based Mesh representation

- **Terminology**
  - Node points
  - Edges
  - Adjacent

- **Triangular mesh modeling**
  - Node points -- salient feature points
  - The choice of feature points is not subjective to standardization
Dynamic Mesh

- **Mesh tracking**
  - Tracking of node points
  - Method: not subjective to standardization

- **Texture mapping**
  - Affine transform
    - 6 parameters, calculated from three node points
    - Model translation, rotation, scaling, reflection
    - Preserve straight line

- **Dynamic mesh**
  - Geometry of initial 2D mesh
  - Motion vectors of node points
Advantage

- Model the shape well (Polygonal approximation)
- Compactly represented. Piecewise motion field can be reconstructed from node points
- Triangular mesh constrains movements of adjacent image patches
- Without the need for further segmentation in compressed domain
- Extensible to 3D objects when data is available
Examples of Overlay

- Animated Text + Video + Still Images

- Video Overlay of Animated Text + Video + Still Images
Image and Graphics Overlay

- Similar to Text
- Usage:
  - Special effects for advertisements
  - Hyperlinked images and graphics in video
3D Scenes

- Scene Description Features:
  - Hierarchical 3D scene graph with depth
  - Basic 3D graphic primitive including 3D mesh and texture
  - Face and body animation

- Media Objects:
  - Still and Moving Images as textures
  - Basic 3D graphic primitives including 3D Mesh and texture mapping streaming text
  - Face and Body animations
3D Mesh Example
2D/3D examples (1)

(a) 3D on top of 2D

(b) 2D on top of 3D

(c) Multi View Point 3D
2D + 3D Scenes

Scene Description Features

- Ability to layer rendered 2D or 3D scene in a 2D space
- 2D interfaces for a 3D scene
- 3D Objects on top of a 2D scene
- Inclusion of a 2D scene in a 3D scene Graph
- 2D or 3D scene as texture maps
2D/3D Examples (2)

2D or 3D scene as a texture map on 3D

2D inside a 3D plane
Structure of the Tools for Representing Natural Video

- VLBV (Very Low Bitrate Video)
- High Bitrate
- Content-Based functionalities
MPEG-4 Video Image Coding Scheme

- Support for Conventional and Content-based Functionalities
Structure for Shape-Motion-Texture coding
Shape Coding

- Macroblock based
- Three block types
  - Transparent
  - Opaque
  - Shape block
Arithmetic Encoding

- Treat each macroblock as a string composed of symbol “1” and “0”
  - “1110001110011111000.........”
- Probability of distribution: $P(0) = p$, $P(1) = 1-p$
- Shortcoming: Probability distribution is not very suitable

Observation:
- High degree of local correlation exists in the image
- For a given pixel, the probability distribution is conditioned upon the values of pixels in a local neighborhood.
Context-based Arithmetic Encoding (CAE) (1)

- Basic idea: Making use of template to define a context value associated with a pixel
- Intra mode
  - Size of template: 10 pel
  - Context number: 1024

![Diagram showing probability distribution]

Probability distribution is created before coding by training
CAE (2)

- Inter mode

Alignment after motion compensation

Previous Frame

Current Frame

9 Pixels, 512 context values

Efficiency depends on the choice of template
Use down sampling to raise compression efficiency (lossy)

Transmitted information:
- Coded data
- Coding mode information
  - Transparent
  - Opaque
  - Intra
  - Inter
  - No update
- Motion compensation
CAE

- Calculate the context of BAB pixel Value (X)
- Look-up the relevant entry in the Probability Table
- Encode X with an arithmetic encoder:
  - Sub-range (0 – P(0)) \( \Rightarrow \) X=0
  - else, X=1
MC coding of Arbitrary Shape

- Note: The transparent pixel values are not defined
- Hzl and vert padding needed in boundary blocks

Fig: Hzl & vertical filling
Padding of Transparent MB

- Transparent MBs are padded as extensions of the left/top/right or bottom boundary MBs.
- If no boundary MBs is a neighbor, then fill with pixel value $2^{n-1}$ (usually 128).

TEXTURE CODING OF BOUNDARY MB:

  Classical way after padding the MB
  Eff way: SADCT $\rightarrow$ in Adv. Coding
Sprite Coding
Still Image Coding

- Zero tree based wavelet algorithm
- Scalability
  - SNR
  - Spatial
Error Resilient Video Coding

- Video Packet Resynchronization
  *Resync Markers, Video Packets*

- Data Partitioning (*DP*)
  *Motion Boundary Marker (MBM)*

- Header Extension Code (*HEC*)
  *if 1- Header info repeated*

- Reversible Variable Length Codes (*RVLCs*)
  *Backward decoding*
Version 2 of MPEG-2

- Ready: One year later
- Intellectual Property Management & Protection (IPMP)
- Advanced BIFS
- MPEG-4 File Format
- MPEG-J
- Coding of 3D Meshes
- Body Animation
Applications

- Real time communication
- Mobile Multimedia
- Content based Storage and retrieval
- Streaming video on internet
- Broadcast
- Digital Television
Profiles and Levels

- Most applications only need a part of the MPEG-4 tool set.
- Profiles define subsets useful for a large class of applications/services.
- Types of Profiles:
  - Scene description (e.g. behaviour),
  - Object Descriptor (mainly timing models),
  - Audio (natural and synthetic): types of objects,
  - Visual (natural and synthetic): types of objects,
  - Graphics.
- Levels limit the number of objects and complexity
- Profiles/Levels can be easily added when need arises
Some Visual Profiles for Natural Video

- **Simple**: Low complexity coding of rectangular frames
- **Adv. Simple**: + support for interlaced video
- **Adv. Real-time simple**: for real-time streaming
- **Core**: basic coding of arbitrary Shaped VO
- **Main**: Rich coding of VO
- **Simple scalable**: Scalable rectangular objs
- **FGS**: Advanced scalable
<table>
<thead>
<tr>
<th>Visual profile</th>
<th>Level</th>
<th>Typical visual session size</th>
<th>Max. number of objects</th>
<th>Maximum number objects per type</th>
<th>Max. unique quant. tables</th>
<th>Max. VMV buffer size (MB units)</th>
<th>Max. VCV buffer size (MB/s)</th>
<th>VCV decoder rate (MB/s)</th>
<th>Max. VCV boundary MB decoder rate (MB/s)</th>
<th>Max. total VBV buffer size (units of 16384 bits)</th>
<th>Max. VLO VBV buffer size (units of 16384 bits)</th>
<th>Max. video packet length (bits)</th>
<th>Max. sprite size (MB units)</th>
<th>Max. bitrate (kbit/s)</th>
<th>Max. enhancement layers per object</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple</td>
<td>L0</td>
<td>QCIF</td>
<td>1</td>
<td>1 x Simple</td>
<td>1</td>
<td>198</td>
<td>99</td>
<td>1485</td>
<td>N.A</td>
<td>10</td>
<td>10</td>
<td>2048</td>
<td>N.A</td>
<td>N.A</td>
<td>64</td>
</tr>
<tr>
<td>Simple</td>
<td>L1</td>
<td>QCIF</td>
<td>4</td>
<td>4 x Simple</td>
<td>1</td>
<td>198</td>
<td>99</td>
<td>1485</td>
<td>N.A</td>
<td>10</td>
<td>10</td>
<td>2048</td>
<td>N.A</td>
<td>N.A</td>
<td>64</td>
</tr>
<tr>
<td>Simple</td>
<td>L2</td>
<td>CIF</td>
<td>4</td>
<td>4 x Simple</td>
<td>1</td>
<td>792</td>
<td>396</td>
<td>5940</td>
<td>N.A</td>
<td>40</td>
<td>40</td>
<td>4096</td>
<td>N.A</td>
<td>N.A</td>
<td>128</td>
</tr>
<tr>
<td>Simple</td>
<td>L3</td>
<td>CIF</td>
<td>4</td>
<td>4 x Simple</td>
<td>1</td>
<td>792</td>
<td>396</td>
<td>11880</td>
<td>N.A</td>
<td>40</td>
<td>40</td>
<td>8192</td>
<td>N.A</td>
<td>N.A</td>
<td>384</td>
</tr>
<tr>
<td>Core</td>
<td>L1</td>
<td>QCIF or Simple</td>
<td>4</td>
<td>4 x Core or Simple</td>
<td>1</td>
<td>594</td>
<td>198</td>
<td>5940</td>
<td>2970</td>
<td>16</td>
<td>16</td>
<td>4096</td>
<td>N.A</td>
<td>N.A</td>
<td>384</td>
</tr>
<tr>
<td>Core</td>
<td>L2</td>
<td>CIF</td>
<td>16</td>
<td>16 x Core or Simple</td>
<td>4</td>
<td>2376</td>
<td>792</td>
<td>23760</td>
<td>11880</td>
<td>80</td>
<td>80</td>
<td>8192</td>
<td>N.A</td>
<td>N.A</td>
<td>2000</td>
</tr>
</tbody>
</table>
Simple Profile

- I – VOP
- P– VOP
- Short Header (compatibility with h.263)
- Compression efficiency tools: (4MVs/MB, UMV, Intra Prediction)
- Transmission Efficiency tools: (Video pkts, DP, RVLC)
- Core used : VLBV (h.263 based)
Decoding

Bitstream → VLD → IQUANT → IDCT → Format Conversion → Frame Store

DCT: Discrete Cosine Transform
QUANT: Quantization
IQUANT: Inverse quantization
IDCT: Inverse Discrete Cosine Transform
VLC: Variable length coding
VLD: Variable length decoding
Coding Efficiency tools
(when short header mode is disabled)

- 4 MVs /MB: MV for each 8x8 block. Here the encoder decides to send 1 or 4 MVs
- UMV: Allows MVs to point outside the boundary of reference VOP. (Improves MC when objs moving in and out of picture, MVs can be longer than 16 pixels)
- Intra Prediction: The DC+AC (1st row/col) are predicted from neighboring intra block coefficients.
• Direction of smallest gradient is chosen as prediction direction.

If |DC(A) - DC(B)| < |DC(C) - DC(B)| then predict from block C
Else
Predict from block A

(AC coefficients are predicted in the same direction of DC prediction)
Predictions are normalized for quantization scale
Transmission Efficiency tools

- Objective: minimise the spatial/temporal error propagation
- Resynchronisation Marker (RM): A uniquely decodable binary code. (In case of error, decoder skips the remaining bits till it encounters another marker.)

Video Packet structure

- Data Partitioning: Split the packet into 2 partitions.
  - Coding mode info+DC (intra)/MV(inter)
  - Other AC coefficients
Reversible VLCs:

- Code words which can be uniquely decoded in both forward and backward direction.
- In the event of an error and skipping to the next resynchronization marker, it is still possible to decode the corrupted data in reverse direction.
RVLC - Example

<table>
<thead>
<tr>
<th>Code alphabet</th>
<th>Constant Hamming-Weight VLC</th>
<th>RVLC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>111</td>
</tr>
<tr>
<td>2</td>
<td>01</td>
<td>1011</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>1101</td>
</tr>
<tr>
<td>4</td>
<td>001</td>
<td>10011</td>
</tr>
<tr>
<td>5</td>
<td>010</td>
<td>10101</td>
</tr>
<tr>
<td>6</td>
<td>100</td>
<td>11001</td>
</tr>
</tbody>
</table>

The bit stream sequence 1011 1101 10011 10101 gives 2345 in both forward and backward directions.
RVLC - result

Results after using Reversible VLC
MPEG-4 Adv. Simple Profile

Simple

B-VOP

Quarter Pel

GMC

Alternate Quant

Interlace

Ad. Simple
Bidirectional MC

- Forward Mode
- Backward Mode
- Interpolative Mode
- Direct Mode: The MVs are derived from MVs of colocated MB in backward-ref VOP and only correction term is transmitted

Similar to MPEG-1 & 2
Direct Mode

MVf = [(T1 - T0) / (T3 - T0)] * MV + MVd

MVb = [(T1 - T3) / (T3 - T0)] * MV + MVd

MVf = 1/3 MV + MVd

MVb = -2/3 * MV + MVd

If bkwd ref (P3 here) has 4 MVs/MB then Curr MB is coded with 4 MVs
Quarter Pel MC

• Improved Half-pel sampling:

\[ X = \frac{-8a_4 + 24a_3 - 48a_2 + 160a_1 + 160b_1 - 48b_2 + 24b_3 - 8b_4}{256} \]

• Quarter pel sample value from half-pel

\[ +A \approx b +B \quad b = \frac{(A+B+1)}{2}; \]
\[ c = \frac{(A+C+1)}{2}; \]
\[ d = \frac{(2A+B+C+2)}{4} \]

\[ +C \quad (+ \text{ Half pel location}) \]
Global Motion Compensation

*MV*s are interpolated for each pixel of a MB from the given Global MVs (4 Mvs at arbitrary locations)

*For each pel of the MB, a MC prediction is carried out using corresponding interpolated MVs

Bi-linear interpolation:

\[ M_x = \frac{(M-m)(N-n) \cdot M_1 + (M-m)n \cdot M_2 + m \cdot n \cdot M_3 + m(N-n) \cdot M_4}{MN} \]
• Alternate quantizer: Supports alternate rescaling method.
  a) MPEG-2 quantization
  b) H.263 quantization (default)

• Support for interlaced video (2 MVs /MB, each for one field)
ARTS (Adv Real-time Simple)

**NEWPRED:**

Switch back reference frame

**Encoder:**
- a
- b
- c
- d
- e
- f
- g
- h
- i
- j

**Decoder:**
- a
- b
- c
- d
- e
- f
- g
- h
- i
- j

FB indication

Error

Degraded

Fast recovery
**Dynamic Resolution Conversion (ARTS)**

- **DRS:** To prevent sudden increase in bit-rate due to increased detail or rapid motion in a scene.
- **VOP encoded at half the normal resolution.**

**Decoding:**

- Decoded MB (16x16) → Upsample → 32x32 Ref pic (32x32) → Motion Compensation (with 2xMV) → Ref pic (32x32)
MPEG-4 Core Profile

- Simple
- B-VOP
- Binary Shape
- PVOP Temporal Scalability
- Alternate Quant
- Core
Coding Arbitrary Shaped Objects

- VOP: An instance of VO
- VO: Arbitrary shaped object sequence with arbitrary time duration (an entity that an user is allowed to access and manipulate.)
- VOL: Support for scalable coding
- GOV: Group of VOPs (provides random access points)
- VS: A complete MPEG-4 scene.
Video Objects - Example

Original Frame

Extracted VO
MPEG-4 Main Profile

- Core
- Gray Shape
- Interlace
- Sprite Coding

MAIN
Gray Shape Coding

- Allows different level of transparency
- Gray alpha plane for each MB
- Transparency ranges from 0 (fully transparent) $\leftrightarrow$ 255 (fully opaque)
- Helps to hide the artifacts of object segmentation
- Coding gray shape $\rightarrow$ Binary support mask +gray alpha plane. (classical 8x8 coding)
Sprite Coding - idea
Static Sprite Coding

• Usage: set 'sprite enable' to 'static' in VOL
• 1st VOP is an I-frame followed by series of S-VOPs
• S-VOP is coded differentially from a GMC VOP.
Ways of transmitting Sprite

- **Types:** Basic Sprite & Low latency Sprite
- **Basic:** 1st I-VOP contains the entire sprite
- **Low Latency:** Transmitted partly over time (to reduce the latency caused by larger sprite)
  - Stat. Spr Obj Piece: Intra MB containing new part of sprite
  - Stat. Spr update: Improves quality of prev. Decoded Spr. Area (Inter coded without MVs & Shape info)
Sprite Parameters

• For warping, moving the content of sprite
• Up to 4 parameters /S-VOP
• 1 parameter → Linear Translation
• Upto 2,3 parameter → Affine warping (rotation & shear)
• 4 parameters → Perspective transformation
Sprite Demo 1

Video Clip

Sprite
Sprite Demo 2

Video Clip

Sprite
MPEG-4 Adv. Coding Efficiency Profile

- Gray Shape
- SADCT
- Core
- Quarter Pel
- GMC
- Interlace

ACE

Adv. Simple
SADCT

- Applied for the 8x8 boundary blocks that contain 1 or more transparent pixels.

- Algo:
  - Shift the residual values Up
  - Apply 1D-DCT to each col (n-pt DCT for column with n-opaque values)
  - Shift the resulting coeffs to left
  - Apply 1D DCT to each row
N-bit Profile

- Supports Coding of data between 4 to 12 bits/ sample (usual 8)
- Useful for low and high depth display
MPEG-4 Scalable Video

- Spatial Scalability
- Temporal Scalability
- Quality Scalability (SNR Scalability)
- Fine Granular Scalability (FGS)
Temporal Scalability
FGS

• Enhancement layer can be truncated at any level
• Application: Streaming Video (by providing flexible control over bit-rate)
• Encoding: Residual DCT coeffs. are coded as series of bit-planes (in zig-zag scanning order, MSB → LSB)
Texture Coding

- Still Image Coding using DWT tool.

Applications:
- For coding rectangular frame,
- Coding arbitrary shaped object
- Coding of texture for mapping animated 2D/3D meshes.

DWT used: Daubechies (9,3)-tap bi-orth filter
Texture Coding

- Reordering: 1) Tree-order
  2) Band-by-band order (less coding effect but supports scalability)
- Entropy Coding: CAE
For More Info...

- ISO official site  http://www.iso.ch/
- MPEG official site  http://www.cselt.it/mpeg
- MPEG-4 Systems site  http://garuda.imag.fr/MPEG4/
- MPEG-4 Visual site  http://wwwam.hhi.de/mpeg-video/
- MPEG-4 Audio site  
  http://www.tnt.unihannover.de/project/mpeg/audio/
- MPEG-4 SNHC site  http://www.es.com/mpeg4-snhc/
- Web3D official site  http://www.web3d.org/
- IPA site  http://www.arts.gla.ac.uk/IPA/ipa.html