An Overview of MPEG4

Thanks for slides preparation of Dr. Shawmin Lei, Sharp Labs of America
And, Mei-Yun Hsu
February 1999

Material Sources

- The MPEG-4 Tutorial, San Jose, March 1998
  - MPEG-4: Context and Objectives - Rob Koenen
  - Natural Video in MPEG-4 - Thomas Sikora
  - MPEG-4 Natural Video Tools - Touradj Ebrahimi
- Thanks to all of them
Goal of MPEG-4

- MPEG-4: ‘Coding of Audio-Visual Objects’
- One generic toolbox for many different kinds of applications (e.g. both conversational, interactive, and broadcast)
- Support for a new kind of interactivity: based on content and meaning
- Compression no longer the only reason for doing coding!
  - ‘Low Bitrate’, although still important, is not MPEG4’s only focus!

MPEG-4: Coding of Audiovisual Objects

- Audiovisual Scene is composed of ‘Objects’ (A&V)
- ‘Compositor’ puts objects in scene (A&V, 2&3D)
- Objects can be of different nature
  - natural or synthetic A&V, text & graphics, animated faces, arbitrary shape or rectangular
- Coding scheme can differ for individual objects
- Principle is independent of bitrate!
  - from low bitrates to (virtually) lossless quality
More about the goals of MPEG-4 (1)

- **Interactivity**
  - Content based
  - Random access (in time & to objects)
- **Integration** of natural and synthetic material
  - Separate activity SNHC within MPEG (Synthetic-Natural Hybrid Coding)
  - Mixing synthetic and natural objects together in the same scene
  - Virtual Environments
More about the goals of MPEG-4 (2)

- Accessing information anywhere:
  - Access on mobile networks (efficient coding and low bitrates still important)
  - Ability to cope with error-prone environments
  - Access across different networks
  - Scalability based on (audio/visual) objects
  - Different quality, priority, error protection for different objects possible
- Intellectual Property Rights (IPR)
  - Identification (V.1) and protection (V.2)

The different parts of MPEG-4 Standard

- Delivery (DMIF: Delivery MM Integration Framework)
  - set-up of connection channels (broadcast & interactive)
  - network becomes transparent to application
- Systems
  - Scene Description: composition of different objects in the scene
    - BIFS: Binary Format for Scene description
    - Build on several concepts from VRML
  - Buffering, Multiplexing, Timing
  - Interaction
  - 'Intermedia Format'
Different parts of MPEG-4 Standard (2)

- Visual
  - coding of natural, and synthetic (mostly moving) images
- Audio
  - coding of natural and synthetic sounds
- SNHC (Synthetic-Natural Hybrid Coding)
  - develops coding for synthetic data types
  - not separate part of standard, but technology is integrated in Audio and Visual parts

Overview MPEG-4 System

(Conventional system)
Overview of MPEG-4 System

- Encoder
- Multiplexer
- Demultiplexer
- Decoder
- Compositor

Objects and AV-objects are coded and then decoded. The demultiplexer extracts objects from the encoded data, and the decoder reconstructs AV-objects. The compositors combine these objects to generate the final output.
Overview MPEG-4 System

MPEG-4 Systems: scene composition

scene

person

2D background

furniture

audiovisual presentation

voice

video

globe

desk
Intellectual Property Rights Management

- MPEG-4 Version 1: Identification
  - using existing registration systems (e.g. ISBN)
- MPEG-4 Version 2: Protection
  - The persistence of content identification in modified MPEG-4 objects
  - Content Protection
- MPEG4 does not standardize IPMP (Intellectual Property Management and Protection).
- MPEG4 does standardize IPMP interface, which consists of:
  - IPMP-Descriptors (IPMP-Ds)
  - IPMP-Elementary Stream (IPMP-ES)

Profiles

- Useful subsets of the toolbox
- Allow interworking & conformance tests
- In all areas: Systems, Visual, Audio, Delivery (DMIF)
  - which objects can be combined in a scene?
- MPEG does not prescribe combinations of A, V, S, D
Five Profiles for Natural Video

- **Simple Visual Profile:**
  - Provide efficient and error resilient coding of rectangular video objects
  - Suitable for applications on mobile networks
- **Simple Scalable Visual Profile:**
  - Add support for coding of temporal and spatial scalable objects
- **Core Visual Profile:**
  - Add support for coding of arbitrary-shaped and temporally scalable objects to the Simple Visual Profile
- **Main Visual Profile:**
  - Add support for coding of interlaced, semi-transparent, and sprite objects to the Core Visual Profile
- **N-Bit Visual Profile:**
  - Add support for coding video objects having pixel-depths ranging from 4 to 12 bits to the Core Visual Profile

MPEG-4 Schedule

- November ’97: Committee Draft (CD)
- March ’98: Final CD (FCD)
- October ’98: Draft International Standard
- February ’99: International Standard

- Version 2 will follow Version 1 with all phases one year later
- Version 2 will add new ‘Profiles’, with new functionality (V2 = V1+ new Profiles)
MPEG-4 Video Standard

- MPEG-4 Video Provides Tools for a Number of Functionalities
- Integrated Approach (Core and Extensions)
- Based on DCT Technology (except for Still Texture Coding)

Functionality (Core)

- Coding Efficiency
  - 5 kbits/s - 5 Mb/s
  - Resolution: Small - TV
  - Progress/Interlace
- Error Resilience/Robustness
  - Mobile Environments
- Scalability (Spatial/Temporal)
Content-Based Coding of Video

- Content-Based Coding Allows the User to Access Arbitrarily-Shaped Objects in a Coded Scene
- Content-Based Coding Enables High Interaction With Scene Content
- Manipulation of Scene Content on Bitstream Level

Object Manipulation

- Original Decoded
- Decoded and Manipulated
Content-Based Layering of Video

- Each Video Object in a Scene is Coded and Transmitted Separately

MPEG-4 Core and Extension

MPEG-4 Core Coder

Extended MPEG-4 Core Coder
MPEG-4
Content-Based Functionalities

- Shape Coding
- Sprites
- Scalability (Content Based)
- Error Resilience/Robustness
- Scalable Texture Coding (Wavelets)

Classification of MPEG4 Video Tools

<table>
<thead>
<tr>
<th>Bitrate</th>
<th>Functionalities</th>
</tr>
</thead>
<tbody>
<tr>
<td>High bitrate tools (interlace)</td>
<td>Content-based functionalities (shape, scalability)</td>
</tr>
<tr>
<td>VLBV core</td>
<td></td>
</tr>
</tbody>
</table>


MPEG-4 Tools - Summary -

- I-VOP
- P and B Prediction
- Interlace Prediction

- Scalability
  - Temporal (rectang.)
  - Spatial (rectang.)
  - Temporal (object)

- Binary Shape Coding

- Error Resilience
  - rectang./object

MPEG-4 Tools (Cont.)

- 12 bit Video

- Texture Coding
  (Wavelets)
  - rectang./object
  - Coding Efficiency
  - SNR Scalability
  - Spatial Scalability

- Static Sprites
  - Basic Sprites
  - Low Latency Sprites

- Computational
  Graceful Degradation
Video Object Coding Outline

- Data structure used in visual part of MPEG-4
- Block diagram of natural video decoding
- List of major natural video tools
- Shape coding tools
- Motion compensation tools
- Texture coding tools
- Scalable coding tools
- Error resilience tools
- Sprite coding tools
- 12-bit and Interlaced coding tools

Data structure in visual part of MPEG-4

- Visual object Sequence (VS)
  - VS0
  - VS1
- Visual Object (VO)
  - VO0
  - VO1
- Visual Object Layer (VOL)
  - VOL0
  - VOL1
- Group of Visual Object Plane (GOV)
  - GOV0
  - GOV1
- Visual Object Plane (VOP)
  - VOP0
  - VOPn-1
Basic Block Diagram of MPEG4 Video Coder

Simplified block diagram of natural video decoding
## List of major natural video tools

| Binary shape | Overlapped motion compensation |
| Padding | Advanced motion compensation |
| Motion compensation | Method 1 |
| Quantization | Method 2 |
| AC/DC prediction | Non-linear |
| Scanning | Type 1 |
| I, P, B modes | Type 2 |
| Temporal scalability | Slice synchronization |
| Spatial Scalability | Extended header code |
| Error resilience | Data partitioning |
| Static sprites | Reversible VLC |
| Interlaced coding | Basic |
| 12-bit video | Low delay |
| Static texture | Scalable |

## Shape coding tool (1/3)

Every VOP is coded by dividing it into smaller macroblocks.

![Shape coding tool](image)

**Video Object Plane**

**Bounding box**

**Shape block (BAB)**

(Binary Alpha Block)
Shape coding tool (2/3)

- Coding modes
  - Opaque
  - Transparent
  - No-update
  - Intra Context based Arithmetic Encoding
  - Inter Context based Arithmetic Encoding
- Lossless
- Lossy
  - Motion compensation without update
  - sub-sampling by factor 2 or 4

Shape coding tools - CAE

- Context based Arithmetic Encoding
  - Intra
    - \[
      \begin{array}{ccccccccc}
        c_9 & c_8 & c_7 \\
        c_6 & c_5 & c_4 & c_3 & c_2 \\
        c_1 & c_0 & ? \\
      \end{array}
    \]
  - Inter
    - \[
      \begin{array}{ccc}
        c_3 & c_2 & c_1 \\
        c_0 & ? \\
      \end{array}
    \]
    - Motion compensated
      - \[
        \begin{array}{ccc}
          c_8 \\
          c_7 & c_6 & c_5 \\
        \end{array}
      \]
      - \[
        \begin{array}{c}
          c_4 \\
        \end{array}
      \]
Motion compensation tools (1/3)

Motion compensated coding modes (I, B, P)

Motion compensation tools - Motion vector computation (2/3)
Motion compensation tools - padding (3/3)

Texture coding tools (1/2)
Texture coding tools (2/2)

Adaptive DC prediction
Adaptive AC prediction

Coefficients scanning

Alternate-Horizontal scan

Alternate-Vertical scan

zig-zag scan
Quantization

- Method 1: Similar to that of H.263
- Method 2: Similar to that of MPEG-2
- Optimized non-linear quantization of DC coefficients
- Quantization matrices and loading mechanism

Scalability

- Object scalability
  - Achieved by the data structure used and the shape coding
- Temporal scalability
  - Achieved by generalized scalability mechanism
- Spatial scalability
  - Achieved by generalized scalable mechanism
Temporal scalability

- The temporal scalability is achievable for both rectangular frames and arbitrarily shaped VOPs
- The base layer is encoded conventional MPEG-4 video
- The enhancement layer is encoded using one of the following two mechanisms:
  - Type 1
  - Type 2
Temporal enhancement types

<table>
<thead>
<tr>
<th>Enhancement type 1</th>
<th>Base layer</th>
<th>Enhancement layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOL0: entire frame</td>
<td>VOL0: car</td>
<td>VOL1: car</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Enhancement type 2</th>
<th>Base layer</th>
<th>Enhancement layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOL0: entire frame</td>
<td>VOL0: car</td>
<td>VOL1: entire frame</td>
</tr>
</tbody>
</table>

: region to be enhanced by an enhancement layer

Temporal scalability Type 1

Only a portion of the VOP in the base layer is enhanced

VOL0

0 2 4 6 8 10 12 frame number

VOL1

0 6 12 frame number

Base Layer

Enhancement Layer
Temporal Scalability Type 2

The entire VOP in the base layer is enhanced

Spatial scalability

- The base layer is coded as conventional MPEG-4 video
- The enhancement layer is encoded using prediction mechanisms from the base layer
Spatial scalability

Enhancement layer

Base layer

Error resilience tools (1/3)

• Resynchronization markers
  – Spatial Resynchronization: GOB Start Code
  – Periodic Resynchronization Markers
• Extended header code
  – A single bit, when enable, indicates additional resynchronization information for VOP header
• Data partitioning
  – Separate motion and texture information
  – Enable better error concealment
• Reversible VLCs
Error resilience tools (2/3)

Resync Marker MB Address Quant Param Header Extension Temporal Reference

Shape DATA Motion DATA Motion Marker Texture DATA Error Burst Texture Data Reversible VLC

Forward decoding Backward decoding

Error resilience tools (3/3)

- Picture Start Code
- MPEG4 Resync Marker
- H.263 Resync Marker

H.263 Bitstream

MPEG4 Bitstream
Static sprite coding tools (1/3)

- Sprite and sprite points
- VOP and reference points

Static sprite coding tools (2/3)

- Basic sprite coding
- Low latency sprite coding
- Scalable sprite coding
Static sprite coding tools (3/3)

Shape adaptive wavelet coding (1/4)
Shape adaptive wavelet coding (2/4)

- Generalization of the wavelet transform to arbitrarily shaped VOP
  - number of transformed coefficients in the VOP = number of pixels in the VOP
- Generalization of zero-tree coding
  - no extra bit necessary for pixels outside the VOP

Shape adaptive wavelet coding - SNR scalability (3/4)
Shape adaptive wavelet coding - spatial scalability (4/4)

12-bit video coding tool

- Allows compression of video data with precision of up to 12-bits/pixel
- The syntax, semantics, and coding tools are extended:
  - bit-precision
  - extended DC VLC tables
  - extended quantization mechanism
  - Insertion of marker bits to avoid start code emulations
Interlaced coding mode

- Allows all option in progressive also for interlaced.
- Motion compensation for field or frames similar to that of MPEG-2
- Modified AC/DC prediction
- Field DCT
- Interlaced I, P, and B VOP coding
- Modified prediction for motion coding
- Modified scan rules
- 10% more efficient in compression efficiency compared to MPEG-2

SNHC Visual: Areas of Work

- Facial Animation
  - Facial Definition Parameters (FDP)
  - Facial Animation Parameters (FAP)
  - Face Interpolation Technique (FIT)
- 2D Animated Meshes
  - Triangular meshes
- Scalable Textures
  - Wavelet-based, ZTE (variation of EZW) + DC prediction
- View Dependent Scalability
  - Use a back-channel
- Body Animation (in MPEG4 Version 2)
### Visual Tools in Version 1

- Intra Coding Mode (I-VOP)
- Inter Prediction Mode (P-VOP)
- AC/DC Prediction
- Slice Resynchronization
- Data Partitioning
- Reversible VLC
- 4MV, Unrestricted MV
- Binary Shape Coding
- H.263/MPEG-2 Quantization Tables
- P-VOP based temporal scalability
- Rectangular Shape
- P-VOP based temporal scalability
- Arbitrary Shape

### Visual Tools in Version 2

- Quarter Pel Prediction
- Global Motion Compensation
- Boundary Block Merging
- Shape Adaptive DCT
- Newpred
- Object based Spatial Scalability
- Multiple Auxiliary Components
- Wavelet Tiling
- Scalable Shape Coding for Still Texture
- Dynamic Resolution Conversion
Conclusions

• MPEG4 is a very rich standard
• It provides many tools
• Most saliently, content-based (or object-based) functionalities
• Challenges: How to use it?
  – Killer applications?
  – How to implement it? Too complex?