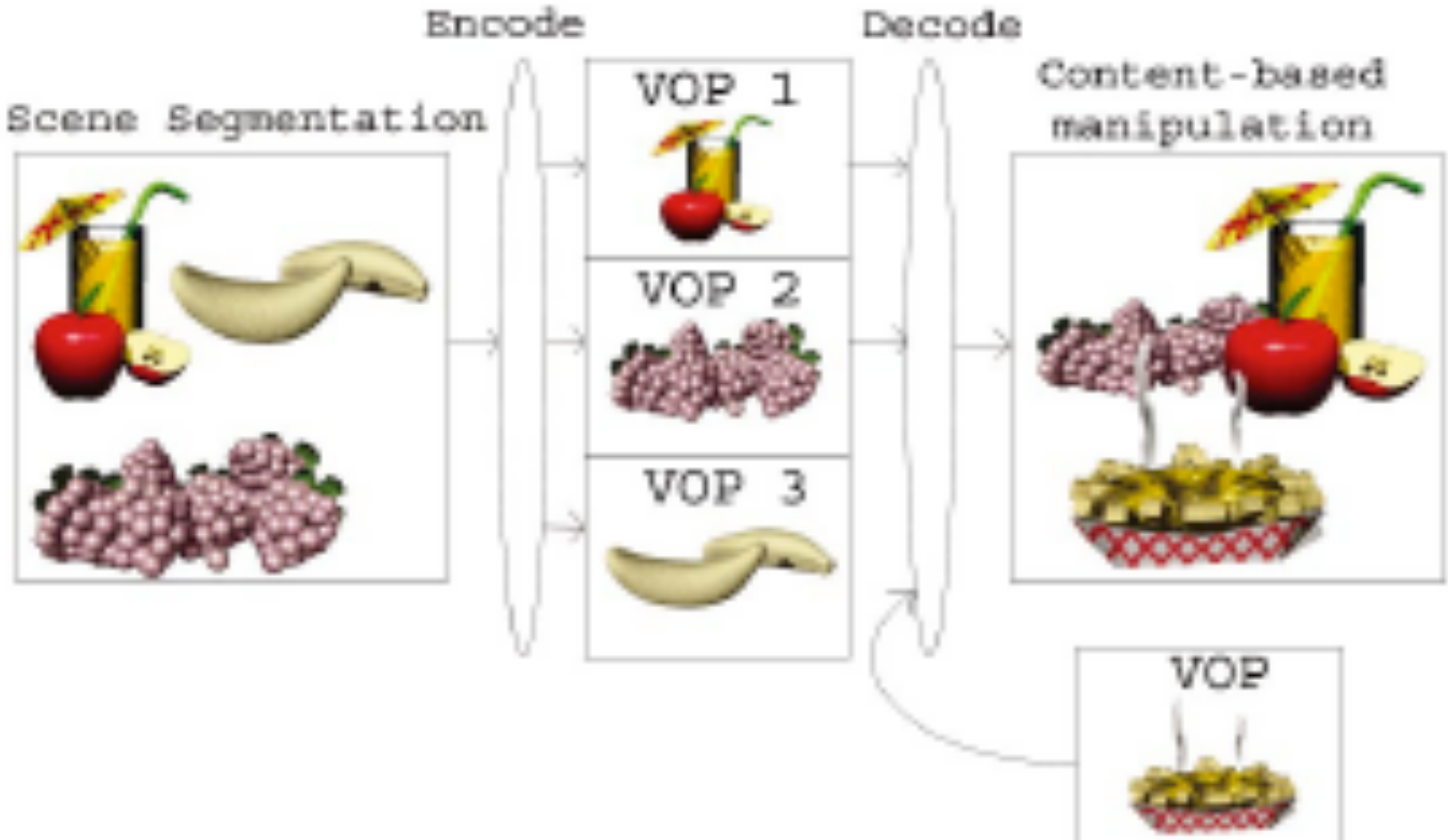
ECE160 / CMPS182
Multimedia

Lecture 12: Spring 2007
Video Compression Techniques
MPEG-4, MPEG-7 and Beyond

Overview of MPEG-4

- **MPEG-2** is designed for HDTV, i.e. moving photography
- **MPEG-4**: a newer standard designed for computer generated multimedia. Besides compression, it pays greater attention to issues about user interaction.
- **MPEG-4** departs from its predecessors in adopting a new **object-based coding.** The next slide illustrates how MPEG-4 videos can be composed and manipulated by simple operations on the visual objects.
- Offering higher compression ratios is also beneficial for digital video composition, manipulation, indexing, and retrieval.
- The bit-rate for MPEG-4 video now covers a large range between 5 kbps to 10 Mbps.

Composition and Manipulation of MPEG-4 Videos

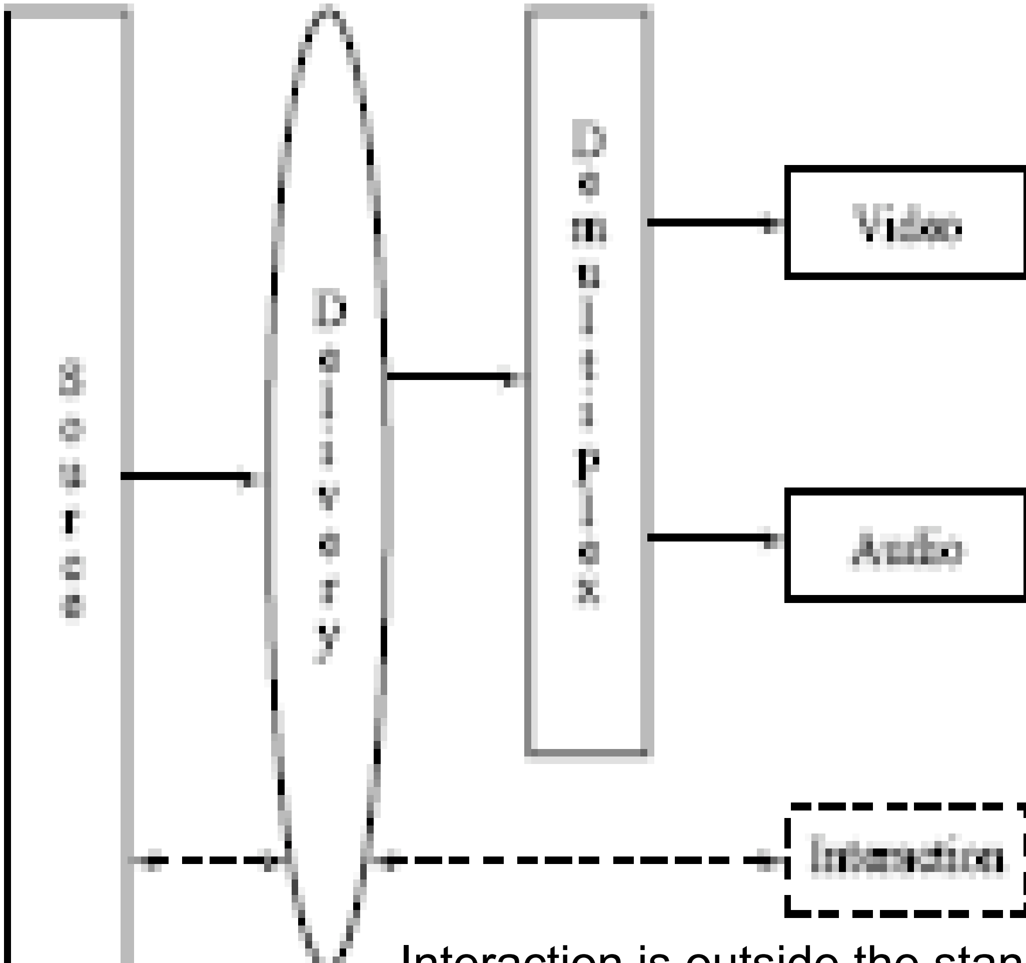


MPEG-4

- MPEG-4 is an entirely new standard for:

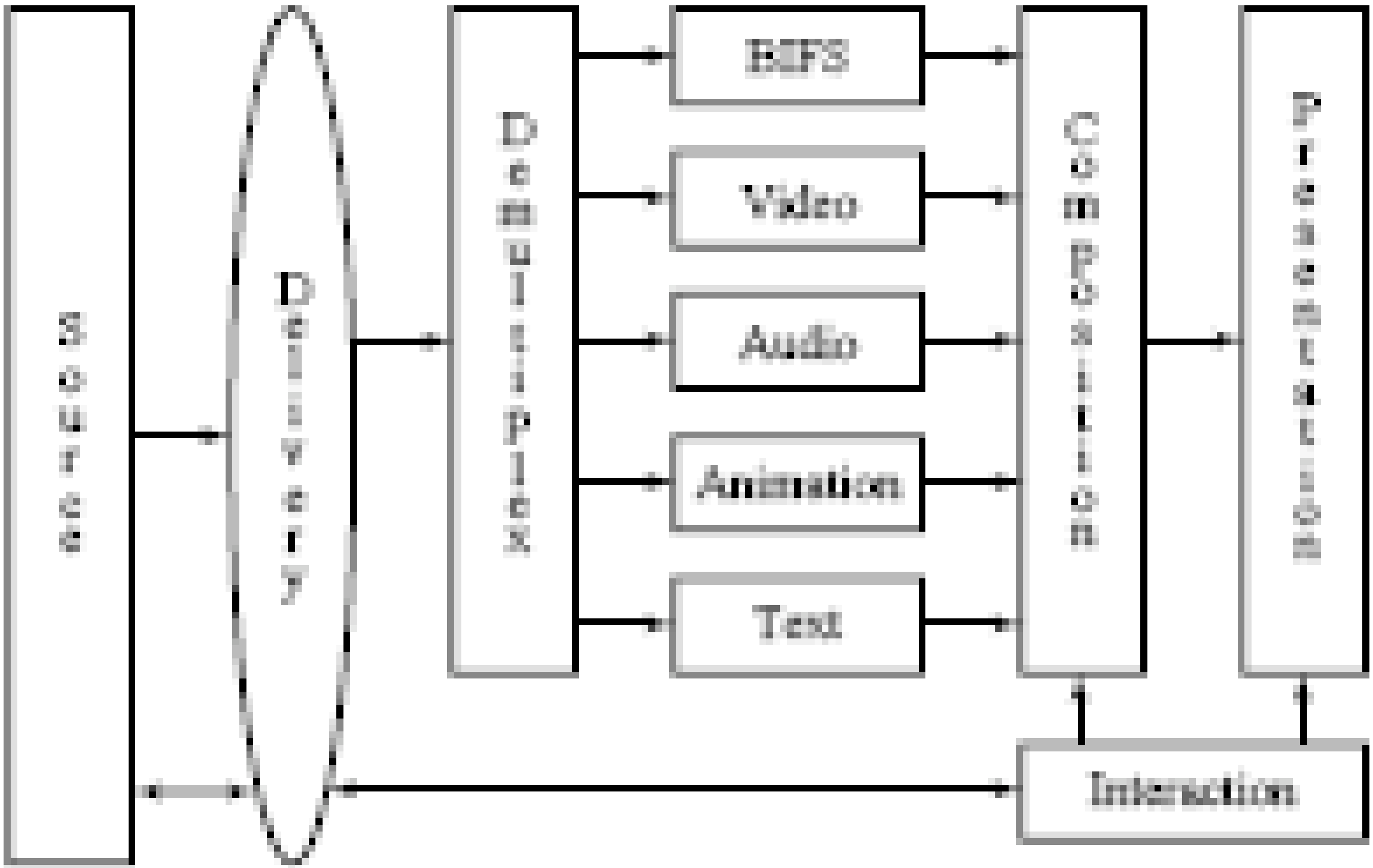
  (a) Composing media objects to create desirable audiovisual scenes.
- (b) Multiplexing and synchronizing the bitstreams for these media data entities so that they can be transmitted with guaranteed Quality of Service (QoS).
- (c) Interacting with the audiovisual scene at the receiving end - provides a toolbox of advanced coding modules and algorithms for audio and video compressions.

MPEG-1 and MPEG-2



Interaction is outside the standard

MPEG-4


Source
Delivery
Demultiplex
BIFS
Video
Audio
Animation
Text
Composition
Presentation
Interaction

Overview of MPEG-4

- Video Object Oriented Hierarchical Description of a Scene in MPEG-4 Visual Bitstreams.
  - The hierarchical structure of MPEG-4 visual bitstreams is very different from MPEG-1 and -2, It is very much video object-oriented.

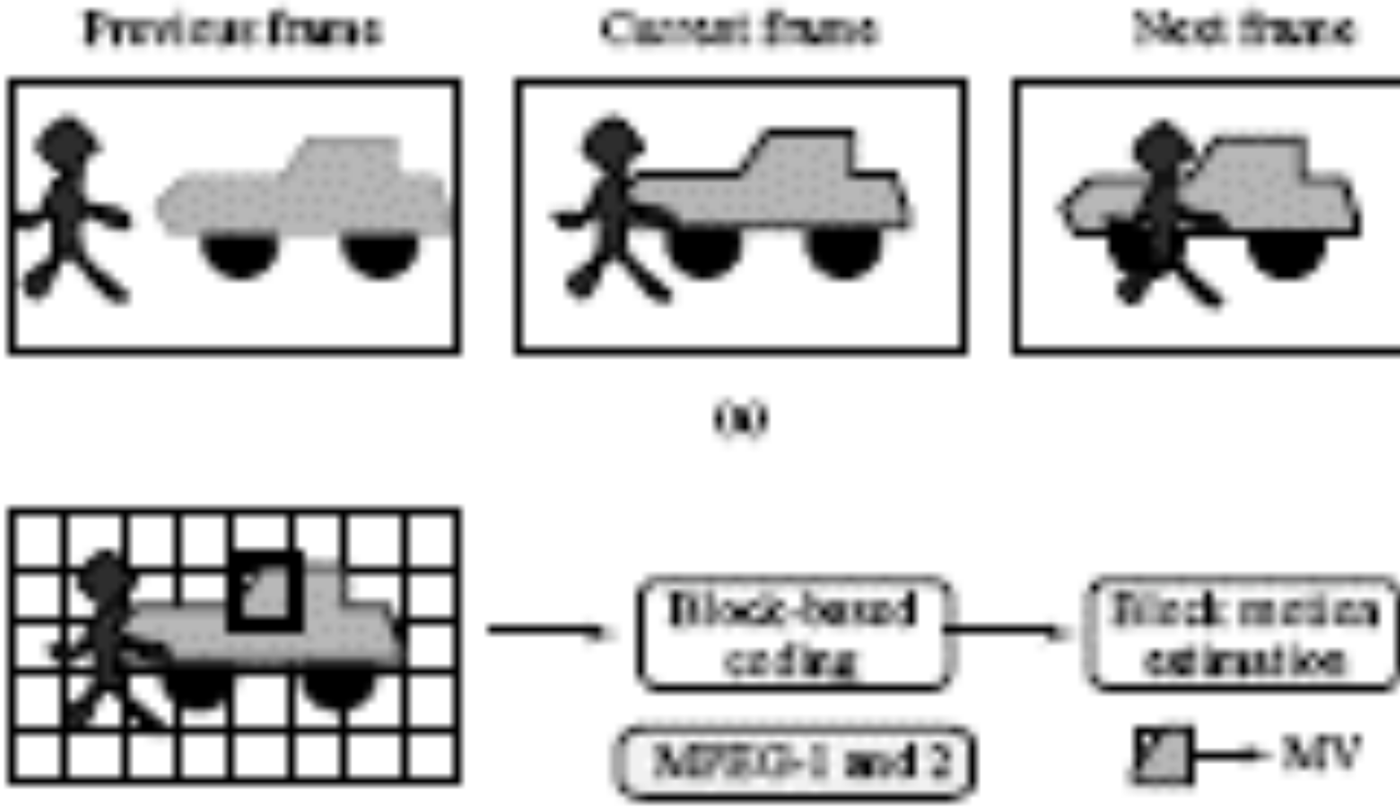
Video-object Sequence (VS)
Video Object (VO)
Video Object Layer (VOL)
Group of VOPs (GOV)
Video Object Plane (VOP)

Overview of MPEG-4

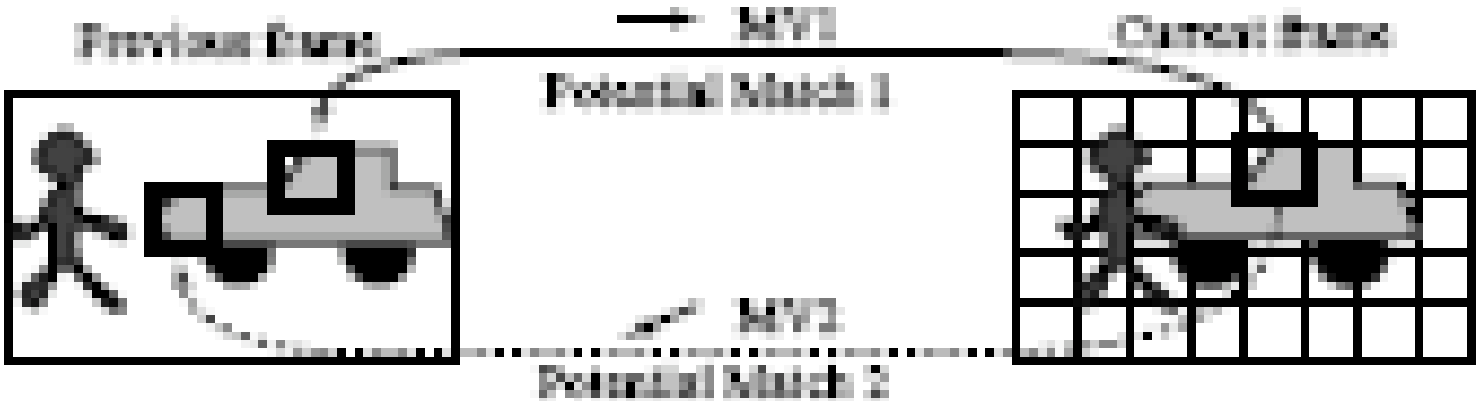
- **Video-object Sequence** (VS) - delivers the complete MPEG-4 visual scene, which may contain 2-D or 3-D natural or synthetic objects.
- **Video Object** (VO) - a particular object in the scene, which can be of arbitrary (non-rectangular) shape corresponding to an object or background of the scene.
- **Video Object Layer** (VOL) - facilitates a way to support (multi-layered) scalable coding. A VO can have multiple VOLs under scalable coding, or have a single VOL under non-scalable coding.
- **Group of Video Object Planes** (GOV) - groups Video Object Planes together (optional level).
- **Video Object Plane** (VOP) - a snapshot of a VO at a particular moment.

VOP-based vs. Frame-based Coding

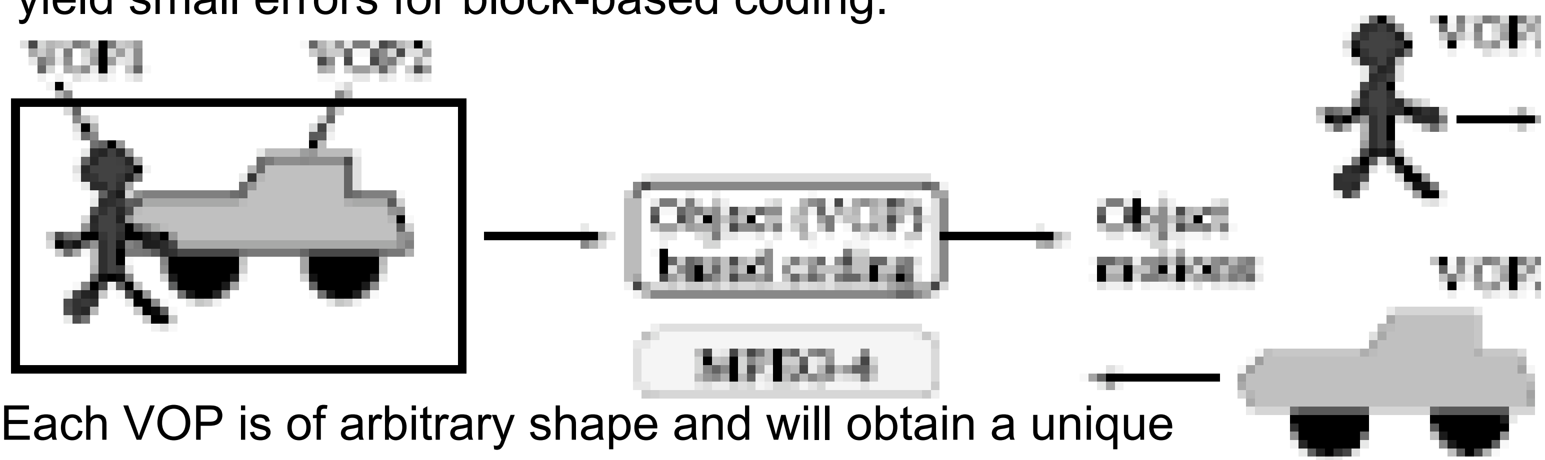
- MPEG-1 and -2 do not support the VOP concept, and hence their coding method is referred to as **frame-based** (also known as **Block-based coding**).



VOP-based vs. Frame-based Coding



A possible example in which two potential matches yield small errors for block-based coding.



Each VOP is of arbitrary shape and will obtain a unique motion vector consistent with the actual object motion.

VOP-based Coding

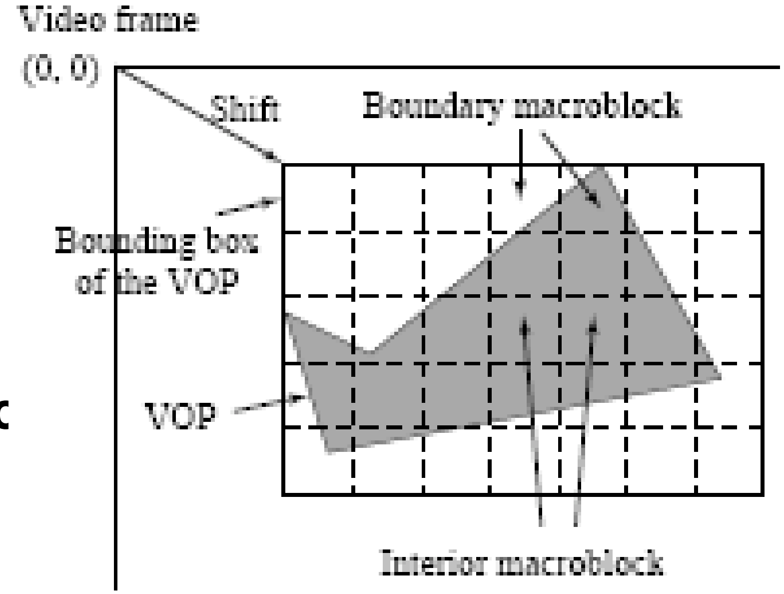
- MPEG-4 VOP-based coding also employs the Motion Compensation technique:
  - An Intra-frame coded VOP is called an **I-VOP**.
  - The Inter-frame coded VOPs are called *P-VOP*s if only forward prediction is employed, or *B-VOP*s if bi-directional predictions are employed.
- The new difficulty for VOPs: may have arbitrary shapes, shape information must be coded in addition to the texture of the VOP.

Note: *texture* here actually refers to the visual content, that is the gray-level (or chroma) values of the pixels in the VOP.

VOP-based Motion Compensation (MC)

- MC-based VOP coding in MPEG-4 again involves three steps:
  - (a) Motion Estimation.
  - (b) MC-based Prediction.
  - (c) Coding of the prediction error.
- Only pixels within the VOP of the current (Target) VOP are considered for matching in MC.
- To facilitate MC, each VOP is divided into many macroblocks (MBs).
  MBs are by default 16x16 in luminance images and 8x8 in chrominance images.

VOP-based Motion Compensation (MC)

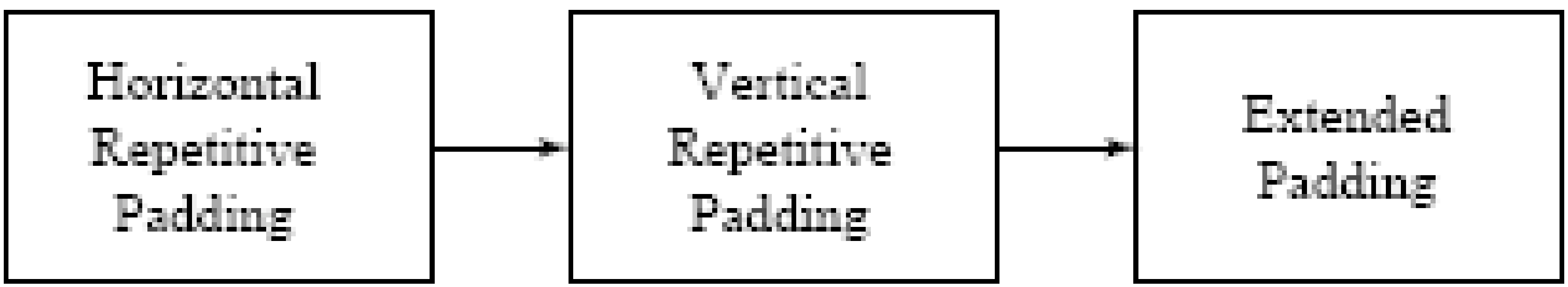


- MPEG-4 defines a rectangular *box* for each VOP.
  - The macroblocks that are entirely within the VOP are referred to as **Interior Macrobloc**
  - The macroblocks that straddle the boundary of the VOP are called **Boundary Macroblocks**.
- To help matching every pixel in the target VOP and meet the mandatory requirement of rectangular blocks in transform codine (e.g., DCT), a pre-processing step of *padding* is applied to the Reference VOPs prior to motion estimation.

  Note: Padding only takes place in the Reference VOPs.

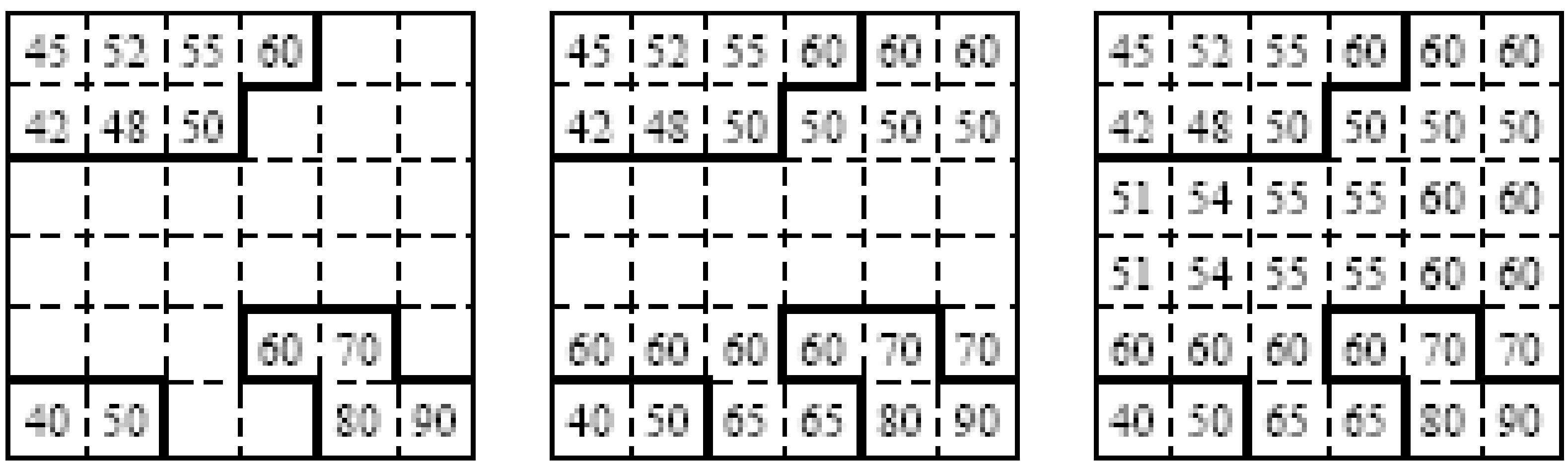
VOP-based Motion Compensation (MC)

- **Padding**
  - For all **Boundary MB**s in the Reference VOP, *Horizontal Repetitive Padding* is invoked first, followed by *Vertical Repetitive Padding*.
  - Afterwards, for all **Exterior Macroblocks** that are outside of the VOP but adjacent to one or more Boundary MBs, *Extended Padding is* applied.



VOP-based Motion Compensation (MC)

- **An Example of Repetitive Padding**
  - (a) Original pixels within the VOP,
  - (b) After Horizontal Repetitive Padding,
  - (c) Followed by Vertical Repetitive Padding.



(a) (b) (c)

Motion Vector Coding

- Let $C(x+k, y+l)$ be pixels of the MB in Target VOP, and $R(x+i+k, y+j+l)$ be pixels of the MB in Reference VOP.
- A **Sum of Absolute Difference (SAD)** for measuring the difference between the two MBs can be defined as:

$$SAD(i,j) = \sum_{k=0}^{N-1}\sum_{l=0}^{N-1} |C(x+k,y+l) - R(x+i+k,y+j+l)| \cdot Map(x+k,y+l)$$

N - the size of the MB. *Map*(*p,q*) = 1 when *C*(*p,q*) is a pixel within the target VOP, otherwise *Map*(*p; q*) = 0.

- The vector (*i,j*) that yields the minimum SAD is adopted as the motion vector **MV**(*u,v*):

$(u,v) = [\ (i;j)\ |\ SAD(i;j)$ is minimum, $i \varepsilon [-p,p], j \varepsilon [-p,p]\ ]$

p is the maximal allowable magnitude for *u* and *v*.

Texture Coding

Texture coding in MPEG-4 can be based on:

- DCT or Shape Adaptive DCT (SA-DCT).

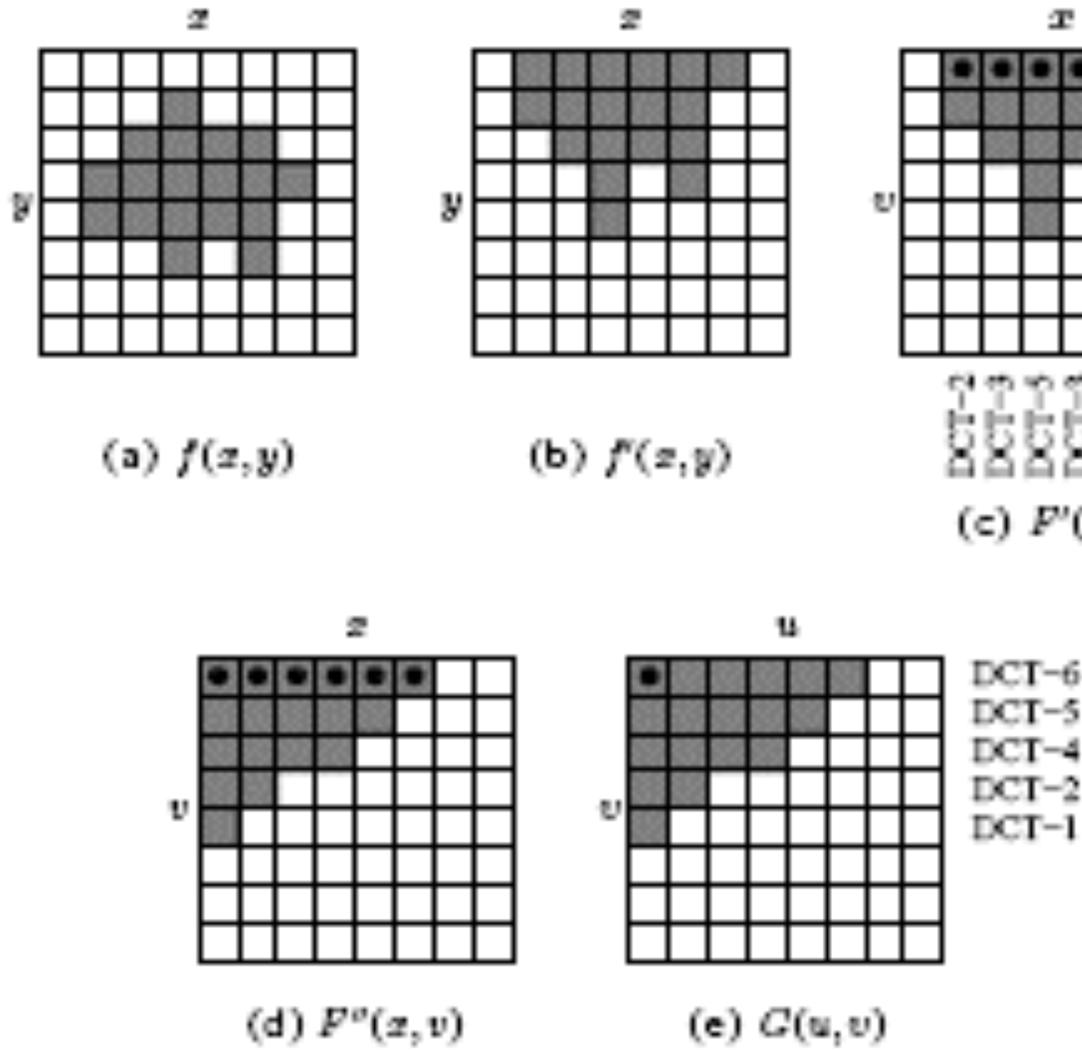
I. Texture coding based on DCT

- In I-VOP, the gray values of the pixels in each MB of the VOP are directly coded using the DCT followed by VLC, similar to what is done in JPEG.
- In P-VOP or B-VOP, MC-based coding is employed - it is the prediction error that is sent to DCT and VLC.
- Coding for the Interior MBs:
  - Each MB is 16x16 in the luminance VOP and 8x8 in the chrominance VOP.
  - Prediction errors from the six 8x8 blocks of each MB are obtained after the conventional motion estimation step.
- Coding for Boundary MBs:
  - For portions of the Boundary MBs in the Target VOP outside of the VOP, zeros are padded to the block sent to DCT since ideally prediction errors would be near zero inside the VOP.
  - After MC, texture prediction errors within the Target VOP are obtained.

SA-DCT Based Coding for Boundary MBs

- Shape Adaptive DCT (SA-DCT) is another texture coding method for boundary MBs.
- Due to its effectiveness, SA-DCT has been adopted for coding boundary MBs in MPEG-4 Version 2.
- It uses the 1D DCT-N transform and its inverse, IDCT-N.
- SA-DCT is a 2D DCT and it is computed as a separable 2D transform in two iterations of 1D DCT-N.

SA-DCT Based Coding for Boundary MBs



Shape Coding

- MPEG-4 supports two types of shape information, **binary** and **gray scale**.
- **Binary** shape information can be in the form of a binary map (also known as *binary alpha map*) that is of the size as the rectangular bounding box of the VOP.
- A value `1' (opaque) or `0' (transparent) in the bitmap indicates whether the pixel is inside or outside the VOP.
- Alternatively, the **gray-scale** shape information actually refers to the *transparency* of the shape, with gray values ranging from 0 (completely transparent) to 255 (opaque).

Binary Shape Coding

- *BABs (Binary Alpha Blocks)*: to encode the binary alpha map more efficiently, the map is divided into 16x16 blocks
- It is the boundary BABs that contain the contour and hence the shape information for the VOP - the subject of binary shape coding.
- Two bitmap-based algorithms:

  (a) **Modified Modified READ (MMR)**.

  (b) **Context-based Arithmetic Encoding (CAE)**.

Modified Modified READ (MMR)

MMR simplifies the **Relative Element Address Designate** (READ) algorithm. The READ algorithm identifies four pixel locations in the previous and current lines:

a_0: the last pixel value known to both the encoder and decoder;

a_1: the transition pixel to the right of a_0;

b_1: the first transition pixel whose color is opposite to a_0 in the previously coded line; and

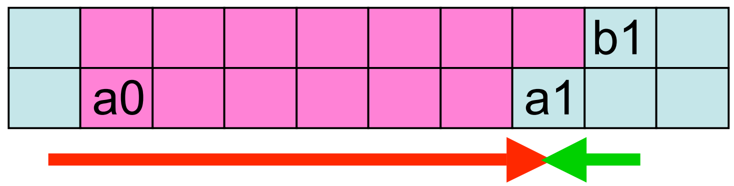
b_2: the first transition pixel to the right of b_1 on the previously coded line.

- The READ algorithm examines the relative position of the pixels: Both the encoder and decoder know the position of a_0, b_1, and b_2 while the positions a_1 is known only in the encoder.

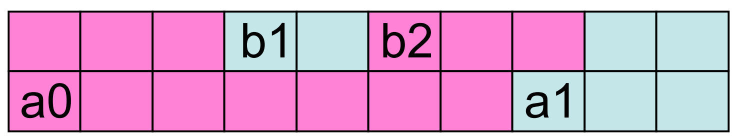
Modified Modified READ (MMR)

- Three coding modes are used:

1. If the run lengths on the previous line and the current line are similar, the distance between a_1 and b_1 should be much smaller than the distance between a_0 and a_1. The *vertical mode* encodes the current run length as $a_1 - b_1$.



2. If the previous line has no similar run length, the current run length is coded using one-dimensional run length coding – *horizontal mode*.
3. If $a_0 \leq b_1 < b_2 < a_1$, simply transmit a codeword indicating it is in *pass mode* and advance a_0 to the position under b_2 and continue the coding process.

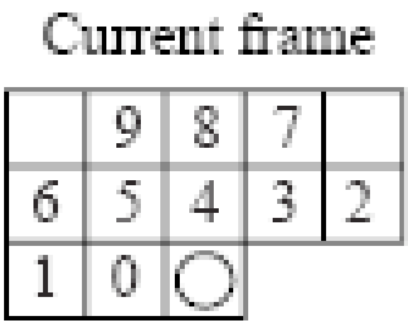


Modified Modified READ (MMR)

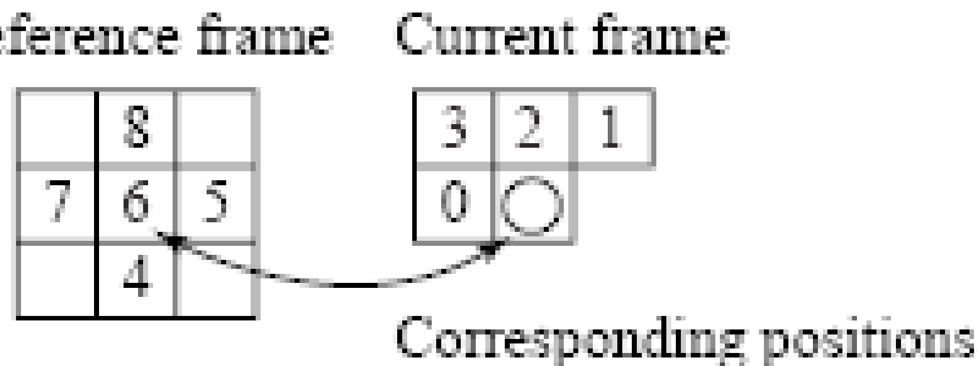
- Some simplifications can be made to the READ algorithm for practical implementation.
  - For example, if $\| a_1 - b_1 \| < 3$, then it is enough to indicate that we can apply the vertical mode.
  - Also, to prevent error propagation, a *k*-factor is defined such that every *k* lines must contain at least one line coded using conventional run length coding.
  - These modifications constitute the *Modified READ* algorithm used in the G3 standard.
    The MMR (Modified Modified READ) algorithm simply removes the restrictions imposed by the *k*-factor.

Context-based Arithmetic Encoding (CAE)

- Certain contexts (e.g., all 1s or all 0s) appear more frequently than others.
- With some prior statistics, a probability table can be built to indicate the probability of occurrence for each of the 2*k* contexts, where *k* is the number of neighboring pixels.
- Each pixel can look up the table to find a probability value for its context. CAE simply scans the 16x16 pixels in each BAB sequentially and applies Arithmetic coding to derive a single floating-point number for the BAB.
- Inter-CAE mode is a natural extension of intra-CAE: it involves both the target and reference alpha maps.



Intra-CAE



Inter-CAE

Gray-scale Shape Coding

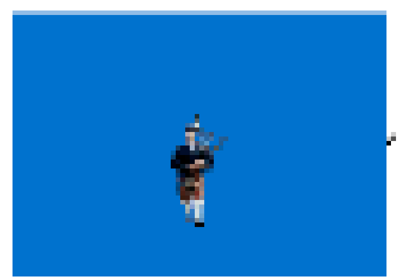
- The **gray-scale** here is used to describe the **transparency** of the shape, not the texture.
- **Gray-scale shape coding** in MPEG-4 employs the same technique as in the texture coding described above.
- Uses the alpha map and block-based motion compensation, and encodes the prediction errors by DCT.
- The boundary MBs need padding as before since not all pixels are in the VOP.

Static Texture Coding

- MPEG-4 uses wavelet coding for the texture of static objects.
- The coding of subbands in MPEG-4 static texture coding is conducted in the following manner:
  - The subbands with the lowest frequency are coded using DPCM. Prediction of each coefficient is based on three neighbors.
  - Coding of other subbands is based on a multiscale zerotree wavelet coding method.
- The multiscale zero-tree has a Parent-Child Relation tree (PCR tree) for each coefficient in the lowest frequency subband to better track locations of all coefficients.
- The degree of quantization also affects the data rate.

Sprite Coding

- A **sprite** is a graphic image that can freely move around within a larger graphic image or a set of images.
- To separate the foreground object from the background, we introduce the notion of a **sprite panorama**: a still image that describes the static background over a sequence of video frames.
  - The large sprite panoramic image can be encoded and sent to the decoder only once at the beginning of the video sequence.
  - When the decoder receives separately coded foreground objects and parameters describing the camera movements thus far, it can reconstruct the scene in an efficient manner.



Global Motion Compensation (GMC)

- “Global" { overall change due to camera motions (pan, tilt, rotation and zoom)
- Without GMC this will cause a large number of significant motion vectors
- There are four major components within the GMC algorithm:
  - Global motion estimation
  - Warping and blending
  - Motion trajectory coding
  - Choice of LMC (Local Motion Compensation) or GMC.

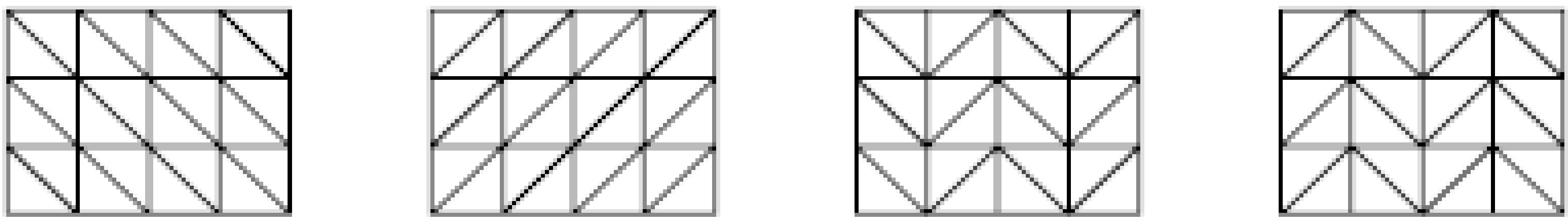
Synthetic Object Coding in MPEG-4

2D mesh: a tessellation (or partition) of a 2D planar region using polygonal patches:

- The vertices of the polygons are referred to as *nodes* of the mesh.
- The most popular meshes are *triangular meshes* where all polygons are triangles.
- The MPEG-4 standard makes use of two types of 2D mesh: **uniform mesh** and **Delaunay mesh**
- 2D mesh object coding is compact. All coordinate values of the mesh are coded in half-pixel precision.
- Each 2D mesh is treated as a *mesh object plane (MOP)*.

2D Mesh Geometry Coding

- MPEG-4 allows four types of uniform meshes with different triangulation structures.



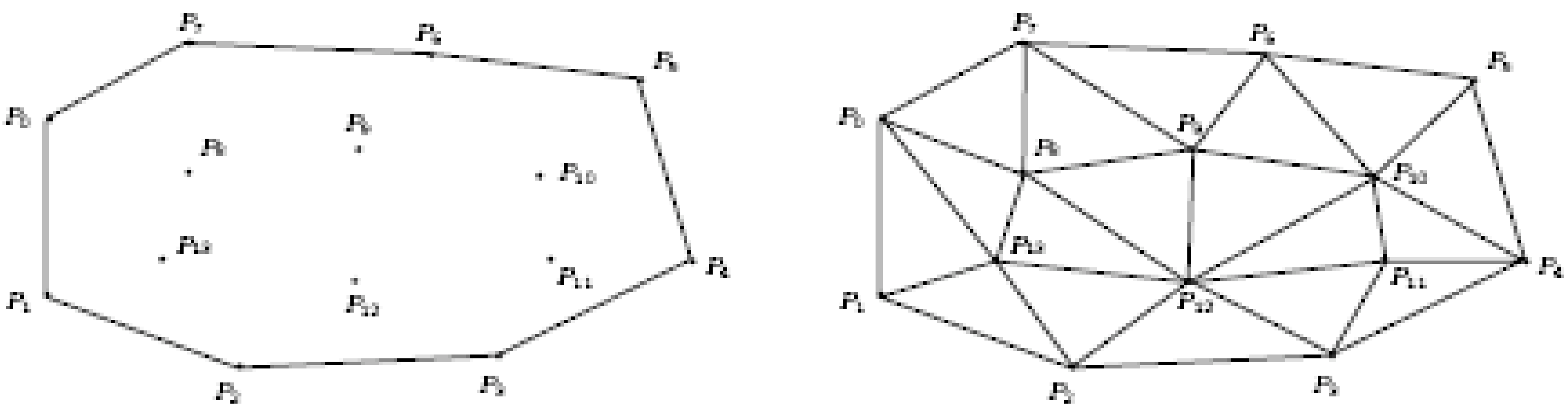
(a) Type 0 (b) Type 1 (c) Type 2 (d) Type 3

Delaunay triangulation

Definition: If *D* is a Delaunay triangulation, then any of its triangles $t_n = (P_i, P_j, P_k)\ \varepsilon\ D$ satisfies the property that the circumcircle of t_n does not contain in its interior any other node point P_l.

- A Delaunay mesh for a video object can be obtained in the following steps:

1. **Select boundary nodes of the mesh:** A polygon is used to approximate the boundary of the object.
2. **Choose interior nodes:** Feature points, e.g., edge points or corners, within the object boundary can be chosen as interior nodes for the mesh.
3. **Perform Delaunay triangulation:** A *constrained Delaunay triangulation* is performed on the boundary and interior nodes with the polygonal boundary used as a constraint.

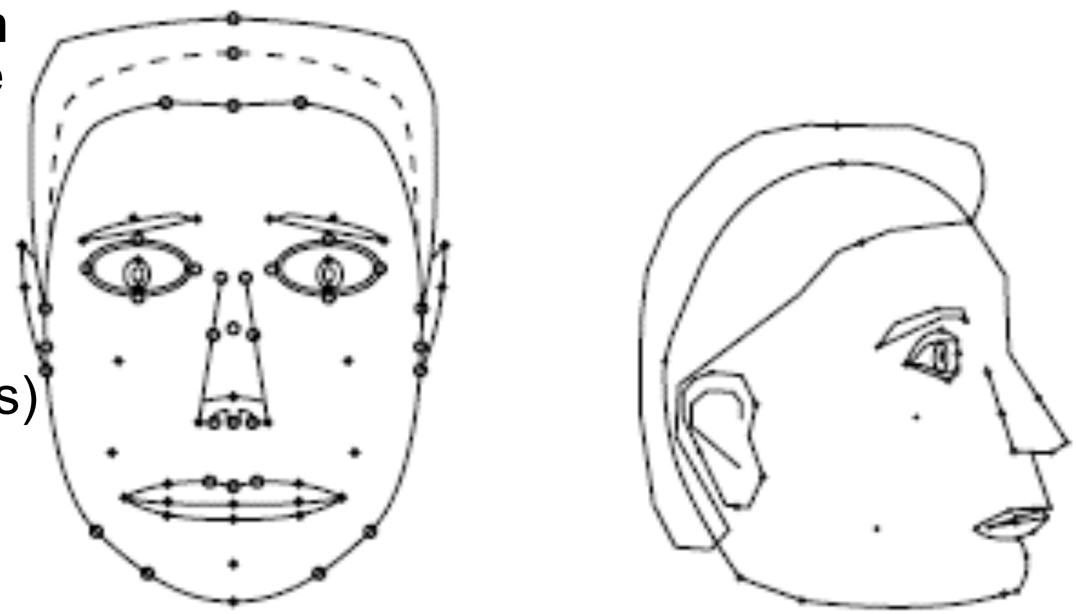


3D Model-Based Coding

- MPEG-4 has defined special 3D models for **face objects** and **body objects** because of the frequent appearances of human faces and bodies in videos.
- Some of the potential applications for these new video objects include teleconferencing, human-computer interfaces, games, and e-commerce.
- MPEG-4 goes beyond wireframes so that the surfaces of the face or body objects can be shaded or texture-mapped.

Face Object Coding and Animation

- MPEG-4 has adopted a generic default face model, developed by VRML Consortium.
- **Face Animation Parameters (FAPs)** can be specified to achieve desirable animations - deviations from the original “neutral" face.
- In addition, **Face Definition Parameters (FDPs)** can be specified to better describe individual faces.
- The figure shows the feature points for FDPs. Feature points that can be affected by animation (FAPs) are shown as solid circles, and those that are not affected are shown as empty circles.



Body Object Coding and Animation

- MPEG-4 Version 2 introduced **body objects**, which are a natural extension to face objects.
- Working with the Humanoid Animation (H-Anim) Group in the VRML Consortium, a generic virtual human body with default posture is adopted.
  - The default posture is a standing posture with feet pointing to the front, arms on the side and palms facing inward.
  - There are 296 **Body Animation Parameters (BAPs)**. When applied to any MPEG-4 compliant generic body, they will produce the same animation.

Body Object Coding and Animation

  - A large number of BAPs are used to describe joint angles connecting different body parts: spine, shoulder, clavicle, elbow, wrist, finger, hip, knee, ankle, and toe – yields 186 degrees of freedom to the body, and 25 degrees of freedom to each hand alone.
  - Some body movements can be specified in multiple levels of detail.
- For specific bodies, **Body Definition Parameters (BDPs)** can be specified for body dimensions, body surface geometry, and optionally, texture.
- The coding of BAPs is similar to that of FAPs: quantization and predictive coding are used, and prediction errors are further compressed by arithmetic coding.

MPEG-4
Object Types, Profiles and Levels

- The standardization of Profiles and Levels in MPEG-4 serve two main purposes:

  (a) ensuring interoperability between implementations

  (b) allowing testing of conformance to the standard
- MPEG-4 not only specified Visual profiles and Audio profiles, but it also specified Graphics profiles, Scene description profiles, and one Object descriptor profile in its Systems part.
- **Object type** is introduced to define the tools needed to create video objects and how they can be combined in a scene.

Tools for MPEG-4
Natural Visual Object Types

	Object Types					
Tools	Simple	Core	Main	Simple scalable	N-bit	Scalable Still Texture
Basic MC-based tools	*	*	*	*	*	
B-VOP		*	*	*	*	
Binary Shape Coding		*	*		*	
Gray-level Shape Coding			*			
Sprite			*			
Interlace			*			
Temporal scalability (P-VOP)		*	*		*	
Spat. & Temp Scal. (r. VOP)				*		
N-bit					*	
Scalable Still Texture						*
Error Resilience	*	*	*	*	*	

MPEG-4 Natural Visual Object Types and Profiles

Object Types	Profiles					
	Simple	Core	Main	Simple scalable	N-bit	Scalable Texture
Simple	*	*	*	*	*	
Core		*	*		*	
Main			*			
Simple Scalable				*		
N-bit					*	
Scalable Still Texture			*			*

MPEG-4 Levels in Simple, Core, and Main Visual Profiles

Profile	Level	Typical picture size	Bit-rate (bits/sec)	Max number of objects
Simple	1	176×144 (QCIF)	64 k	4
	2	352×288 (CIF)	128 k	4
	3	352×288 (CIF)	384 k	4
Core	1	176×144 (QCIF)	384 k	4
	2	352×288 (CIF)	2 M	16
Main	1	352×288 (CIF)	2 M	16
	2	720×576 (CCIR601)	15 M	32
	3	1920×1080 (HDTV)	38.4 M	32

MPEG-4 Part10 / H.264

- The H.264 video compression standard, formerly known as “H.26L", is being developed by the Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG.
- Preliminary studies using software based on this new standard suggests that H.264 offers up to 30-50% better compression than MPEG-2, and up to 30% over H.263+ and MPEG-4 advanced simple profile.
- The outcome of this work is actually two identical standards:
  - **ISO MPEG-4 Part10** and **ITU-T H.264**.
- H.264 is currently one of the leading candidates to carry High Denifition TV (HDTV) video content on many potential applications.

MPEG-4 Part10 / H.264

Core Features

- VLC-Based Entropy Decoding:

  Two entropy methods are used in the variable-length entropy decoder: Unified-VLC (UVLC) and Context Adaptive VLC (CAVLC).
- Motion Compensation (P-Prediction):

  Uses a tree-structured motion segmentation down to 4x4 block size (16x16, 16x8, 8x16, 8x8, 8x4, 4x8, 4x4).

  This allows much more accurate motion compensation of moving objects. Furthermore, motion vectors can be up to half-pixel or quarter-pixel accuracy.
- Intra-Prediction (I-Prediction):

  H.264 exploits much more spatial prediction than in previous video standards such as H.263+.

  Uses a simple integer-precision 4x4 DCT, and a quantization scheme with nonlinear step-sizes.

  In-Loop Deblocking Filters.

MPEG-4 Part10 / H.264

Baseline Profile Features

- The Baseline profile of H.264 is intended for real-time conversational applications, such as videoconferencing.
- It contains the core coding tools of H.264 and additional error-resilience tools, to allow for error-prone carriers such as IP and wireless networks:
  - Arbitrary slice order (ASO).
  - Flexible macroblock order (FMO).
  - Redundant slices.

Main Profile Features

- Represents non-low-delay applications such as broadcasting and stored-medium.
- The Main profile contains the Baseline profile features (except ASO, FMO, and redundant slices) plus:
  - B slices.
  - Context Adaptive Binary Arithmetic Coding (CABAC).
  - Weighted Prediction.

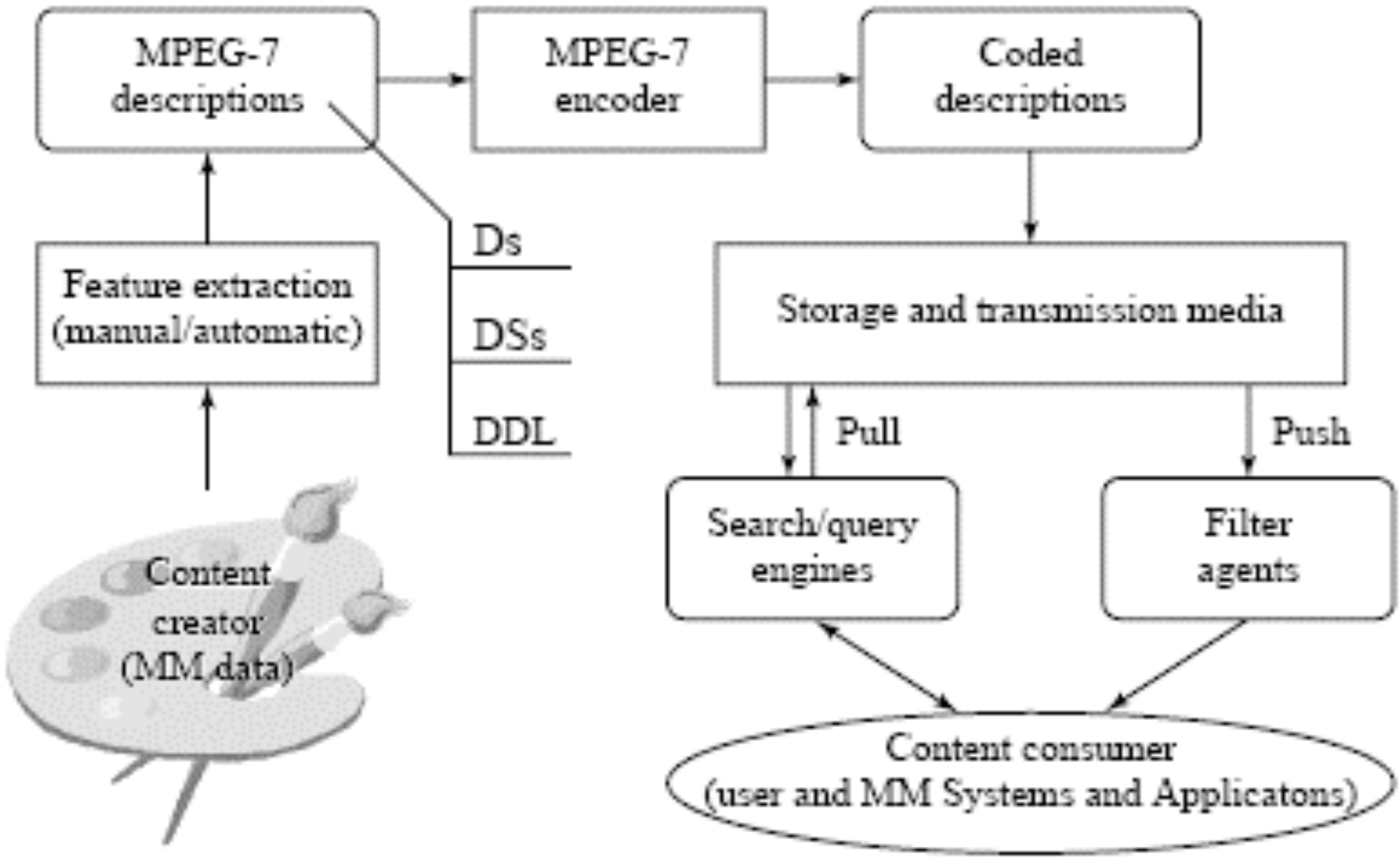
Extended Profile Features

- The eXtended profile (or profile X) is designed for video streaming applications. This profile allows bitstream switching features, and more error-resilience tools.

MPEG-7

- The objective of MPEG-7 is to serve audiovisual content-based retrieval (or audiovisual object retrieval) in digital libraries and search.
- It is also applicable to multimedia applications involving generation (*content creation*) and usage (*content consumption*) of multimedia.
- MPEG-7 became an International Standard in Sept. 2001 - as the **Multimedia Content Description Interface**.
- MPEG-7 supports many multimedia applications. Its data may include still pictures, graphics, 3D models, audio, speech, video, and composition information (how to combine these elements).
- These MPEG-7 data elements can be represented in textual format, or binary format, or both.

Applications using MPEG-7



MPEG-7 and Multimedia Content Description

MPEG-7 has developed Descriptors **(D)**, Description Schemes **(DS)** and Description Denifition Language **(DDL)**. The following are some of the important terms:

- **Feature -** characteristic of the data.
- **Description -** a set of instantiated Ds and DSs that describes the structural and conceptual information of the content, the storage and usage of the content, etc.
- **D -** definition (syntax and semantics) of the feature.
- **DS -** specification of the structure and relationship between Ds and between DSs.
- **DDL -** syntactic rules to express and combine DSs and Ds.

- The scope of MPEG-7 is to standardize the Ds, DSs and DDL for descriptions. The mechanism and process of producing and consuming the descriptions are beyond the scope of MPEG-7.

MPEG-7 Descriptor (D)

The descriptors are chosen based on a comparison of their performance, efficiency, and size. Low-level visual descriptors for basic visual features include:

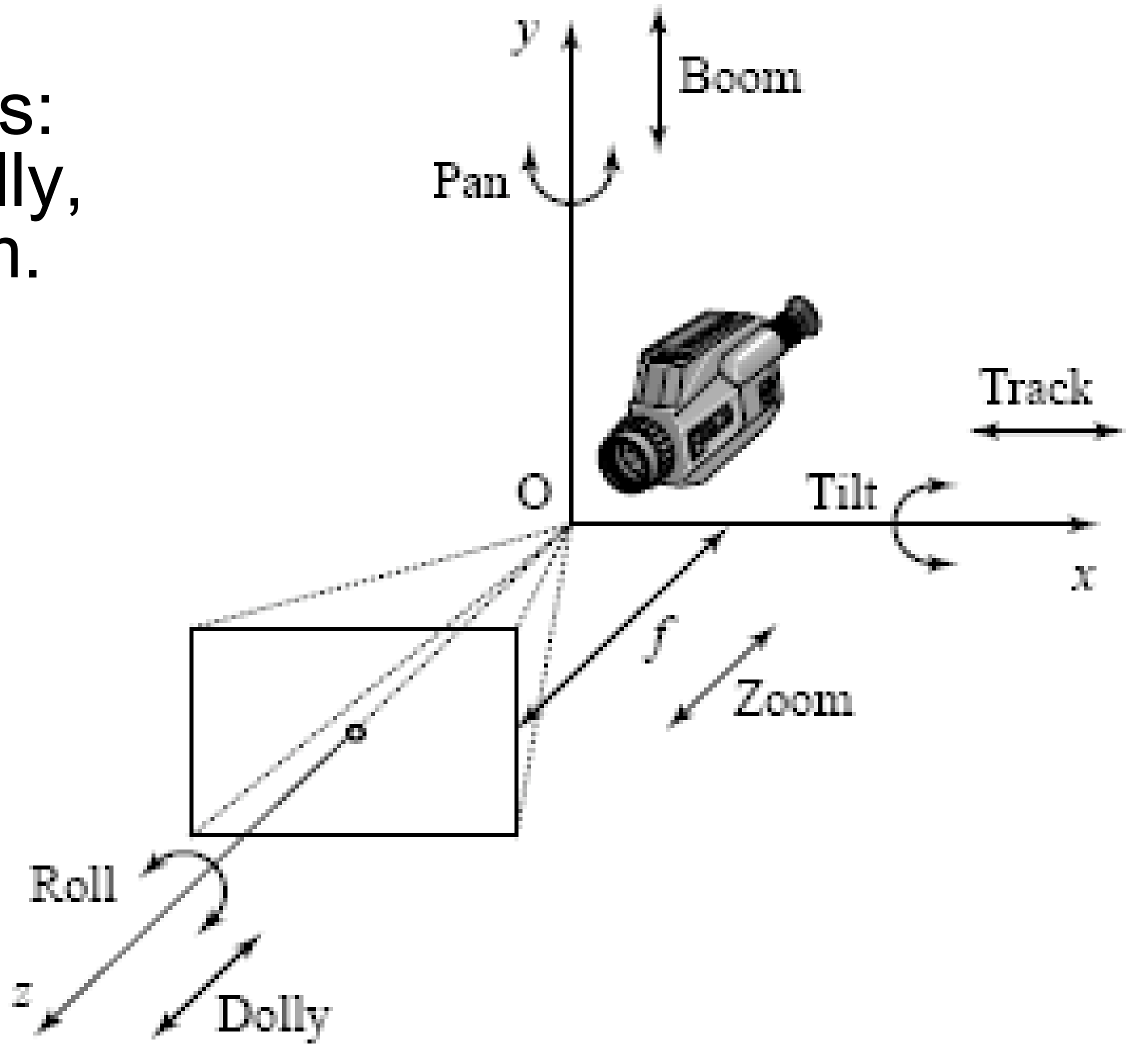
- **Color**
  - Color space. (a) RGB, (b) YCbCr, (c) HSV (hue, saturation, value), (d) HMMD (HueMaxMinDiff), (e) 3D color space derivable by a 3x3 matrix from RGB, (f) monochrome.
  - Color quantization. (a) Linear, (b) nonlinear, (c) lookup tables.
  - Dominant colors.
  - Scalable color.
  - Color layout.
  - Color structure.
  - Group of Frames/Group of Pictures (GoF/GoP) color.

MPEG-7 Descriptor (D)

- **Texture**
  - Homogeneous texture.
  - Texture browsing.
  - Edge histogram.
- **Shape**
  - Region-based shape.
  - Contour-based shape.
  - 3D shape.
- **Motion**
  - Camera motion.
  - Object motion trajectory.
  - Parametric object motion.
  - Motion activity.
- **Localization**
  - Region locator.
  - Spatiotemporal locator.
- **Face recognition.**

MPEG-7 Descriptor (D)

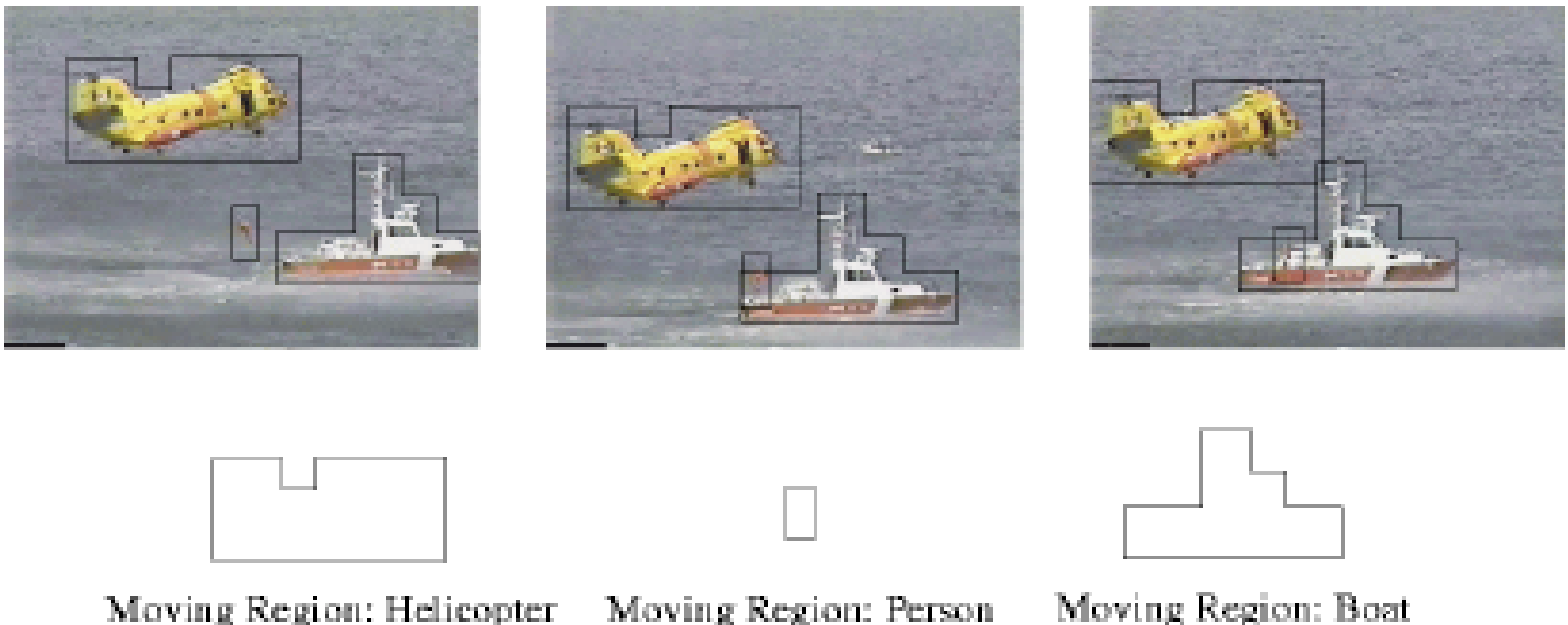
- Camera motions: pan, tilt, roll, dolly, track, and boom.



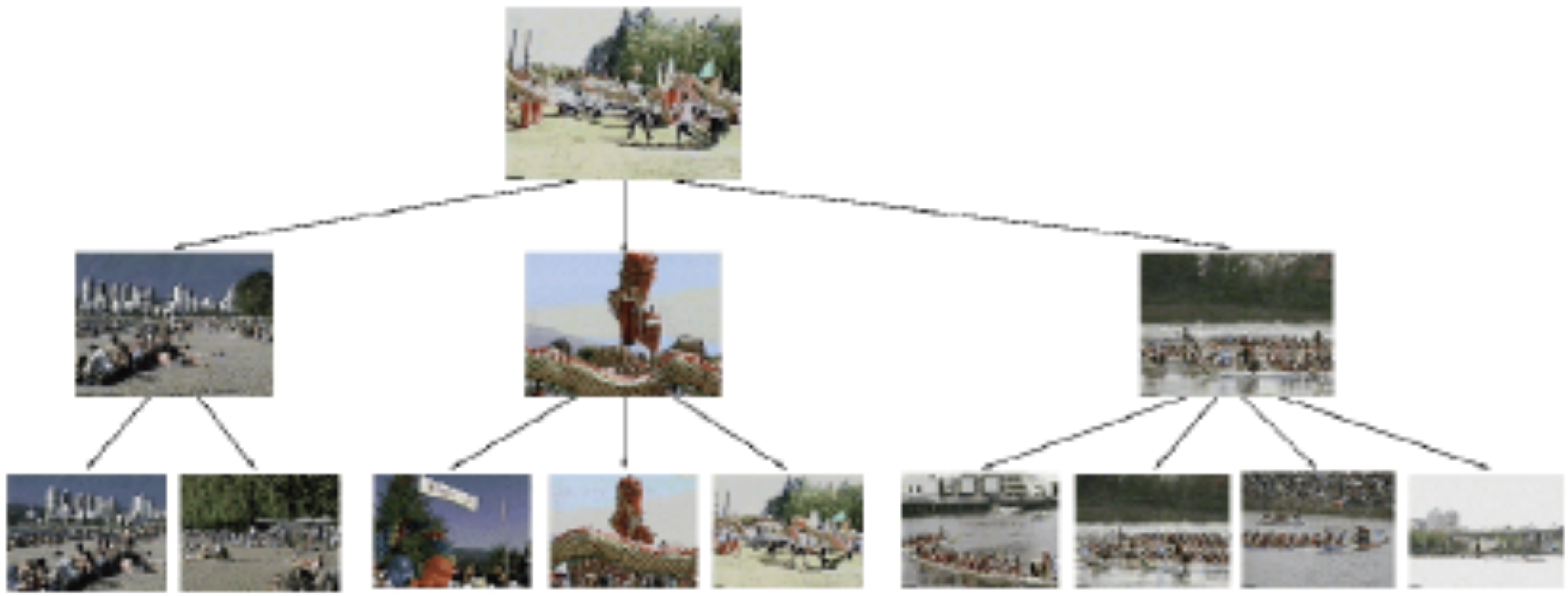
MPEG-7 Description Scheme (DS)

- **Basic elements**
  Datatypes and mathematical structures, Constructs, Schema tools.
- **Content Management**
  Media Description, Creation and Production Description, Content Usage Description.
- **Content Description**
  Conceptual Description.
  Structural Description.
  A *Segment* DS, for example, can be implemented as a class object.
  It can have five subclasses:
  Audiovisual segment DS, *Audio segment DS*, *Still region DS*, *Moving region DS*, and *Video segment DS*.
  The subclass DSs can recursively have their own subclasses.
- **Navigation and access**
  Summaries, Partitions and Decompositions, Variations of the Content.
- **Content Organization**
  Collections, Models.
- **User Interaction**
  UserPreference.

MPEG-7 Video Segment



MPEG-7 Video Summary



MPEG-7 Description Denition Language (DDL)

- MPEG-7 adopted the XML Schema Language developed by the WWW Consortium (W3C) as its Description Definition Language (DDL). Since XML Schema Language was not designed specifically for audiovisual contents, some extensions are made to it:
  - Array and matrix data types.
  - Multiple media types, including audio, video, and audiovisual presentations.
  - Enumerated data types for **MimeType, CountryCode, RegionCode, CurrencyCode,** and **CharacterSetCode**.
  - Intellectual Property Management and Protection **(IPMP)** for Ds and DSs.

MPEG-21 Multimedia Framework

The *vision* for MPEG-21 is to define a multimedia framework to enable transparent and augmented use of multimedia resources across a wide range of networks and devices used by different communities. The seven key elements in MPEG-21 are:

- **Digital item declaration -** a uniform and flexible abstraction and interoperable schema for declaring Digital items.
- **Digital item identification and description-** a framework for standardized identification and description of digital items, regardless of their origin, type or granularity.
- **Content management and usage -** an interface and protocol to facilitate management and usage (searching, caching, archiving, distributing) of content.
- **Intellectual property management and protection (IPMP)**
- **Terminals and networks -** interoperable and transparent access to content with Quality of Service (QoS) over a range of networks and terminals.
- **Content representation -** to represent content in an adequate way for pursuing the objective of MPEG-21, namely “content anytime anywhere".
- **Event reporting -** metrics and interfaces for reporting *events* (user interactions) so as to understand performance and alternatives.