MPEG-4

Nimrod Peleg

update: May. 2003

Introduction to MPEG-4

- Started: 1993
- Originally intended for audio & video coding at 64Kbps and under
- Mid-1994, two things became clear:
  - Video coding schemes that were likely to be mature offered only moderate increase in compression ratio (up to 2 at most) over existing methods
  - A new class of multimedia application emerged and required greater level of functionality at bitrate range of 10k-1024Kbps

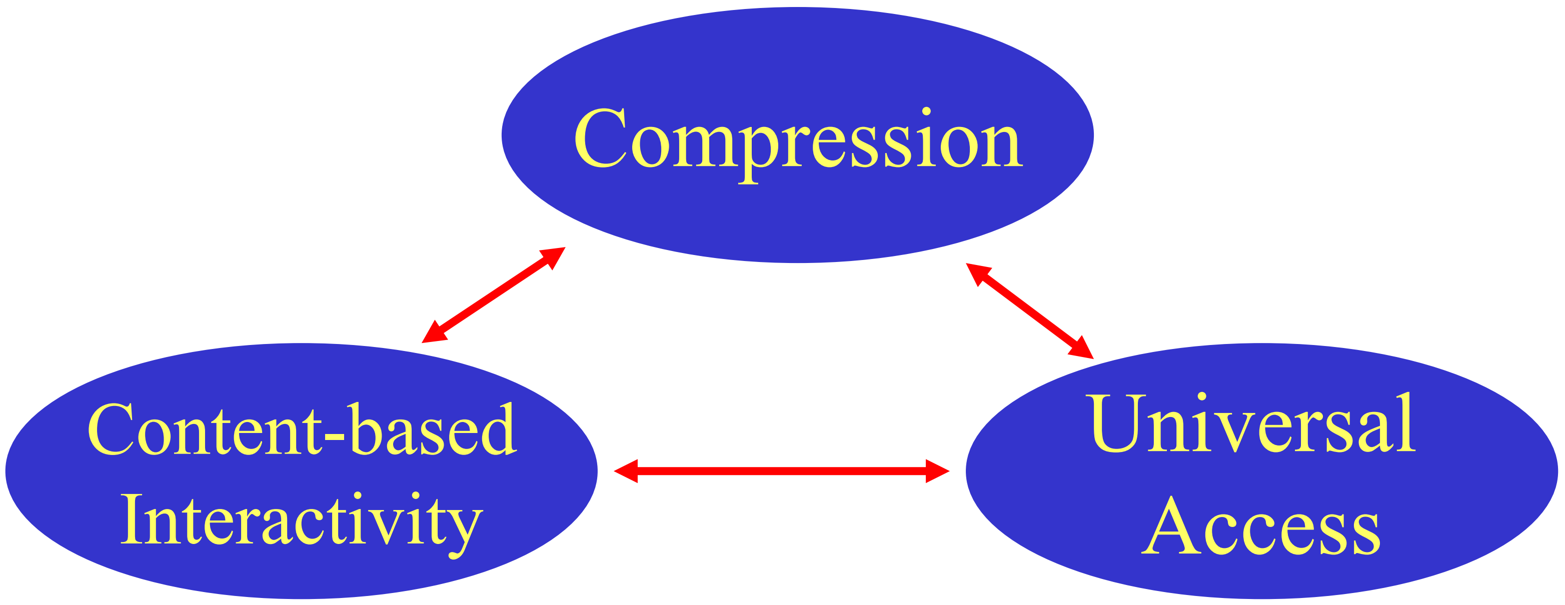
Focus of MPEG-4

1995, MPEG-4 PPD (Proposal Package Description)

- Traditional boundaries between telecommunications, computer and TV/film industries are blurring
- What seems to be convergence in reality is **NOT**
- Each industry approach audio-visual applications from different technological perspectives, providing its own (often compatible) solutions.

Final Standard

- October 1998: Final draft (ISO/IEC 14496)
- Q1 1999: Formal int'l standard.
- Ver. 2: Q1-2 / 2000 .

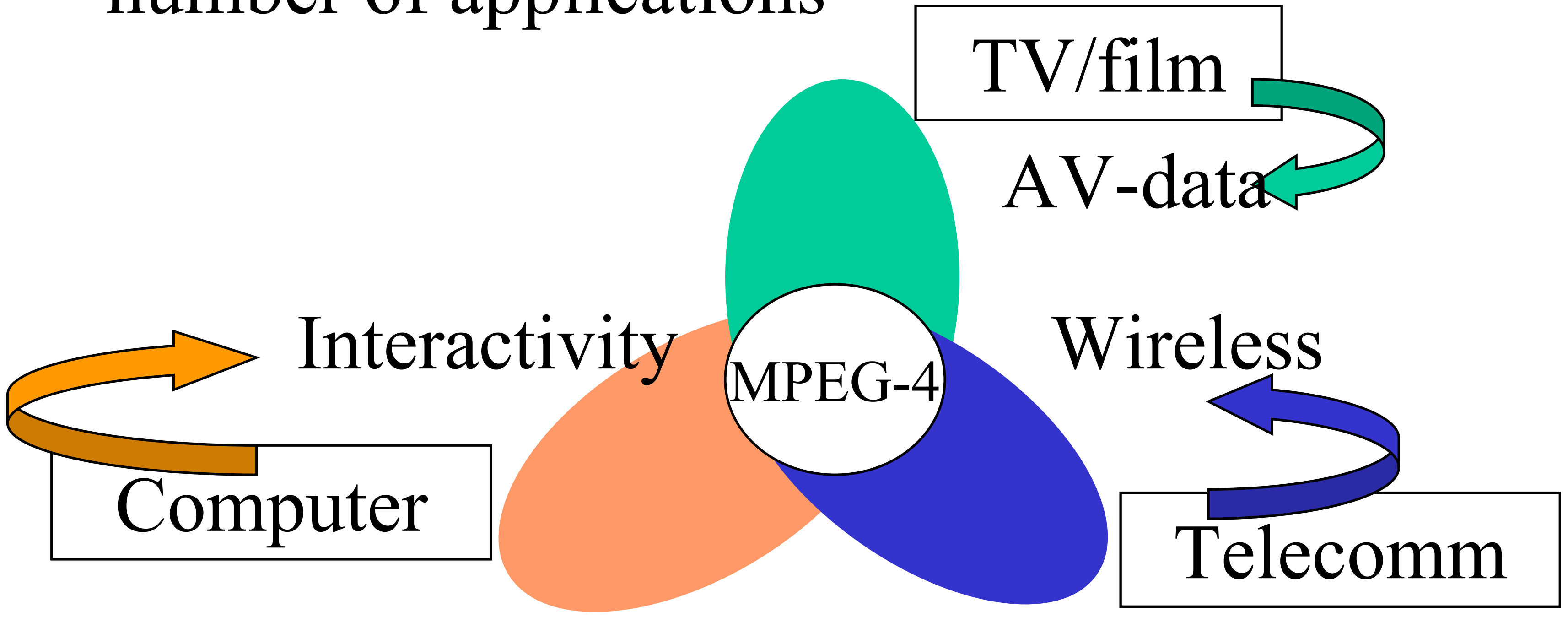


Functionality Classes

- *Content based interactivity*: Ability to interact with important objects in the scene, not only for synthetic objects
- *High compression*: to enable new applications
- *Universal accessibility*: Ability to access audio-visual data over diverse range of media (also for wireless mobile phone)

Beginning with: 3 Important Trends

- Toward wireless communication
- Toward interactive computer applications
- Toward integration of audio-visual data into number of applications



MPEG-4 Applications

- Video on LAN, Internet video
- Wireless video
- Video databases
- Interactive home shopping
- Video E-mail, home video
- Virtual reality games, flight simulations, multi-viewpoint training

“Executive Overview”

Standardized technological elements, enabling the integration of the production, distribution and content access of three fields:

- Digital TV
- Interactive graphics applications (synthetic contents)
- Interactive Multimedia (WWW, distribution and access of contents)

Scope and Features

- A set of technologies to satisfy the needs of:
  - Authors
  - Service providers
  - End users

Authors

- Production of content that have far greater reusability and flexibility, than possible with individual technologies

(Dig.TV, Animated graphics, WWW pages etc.)

- Better manage and protect content owner rights.

Network Service Providers

- Transparent information which will be interpreted and translated into the appropriate native signaling messages of each network.
- MPEG-4 will provide a generic QoS

parameter set for different MPEG-4 media.

End Users

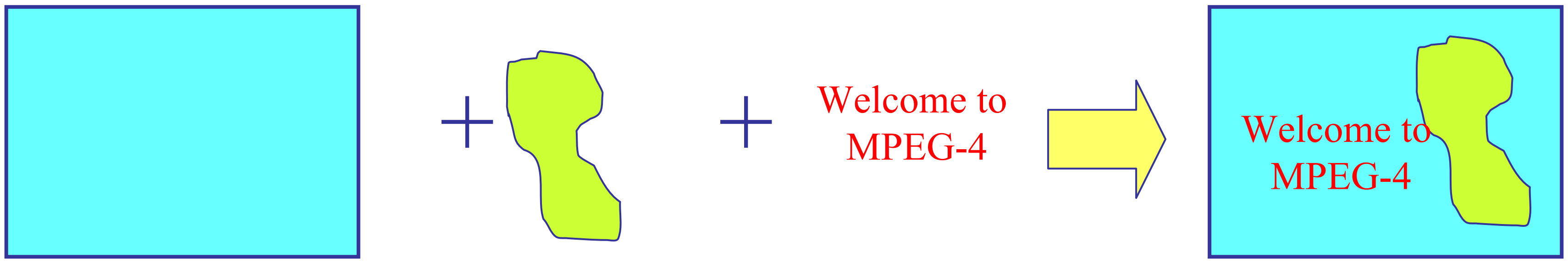
- Many functionality's which could potentially be accessed on a single compact terminal and higher levels of interaction with content, within the limits set by the author.
- An MPEG-4 applications document exists which describes many end user applications including, among others, real time communications, surveillance and mobile multimedia.

Achieving those Goals by:

- Audio/Video Objects (AVO): Basic units, of natural or synthetic origin, representing audio, video or audiovisual contents (Media Objects).
- Compose these objects together to create compound audiovisual objects that form audiovisual scenes.
- Multiplex and synchronize the data associated with AVOs.
- Interact with the audiovisual scene generated at the receiver's end.

Object Functionalities in MPEG- 4

- Image = Various objects+ text+ background
  - VOPs: Video Object Plane
- composition or segmentation
- separate object coding
- separate object manipulation



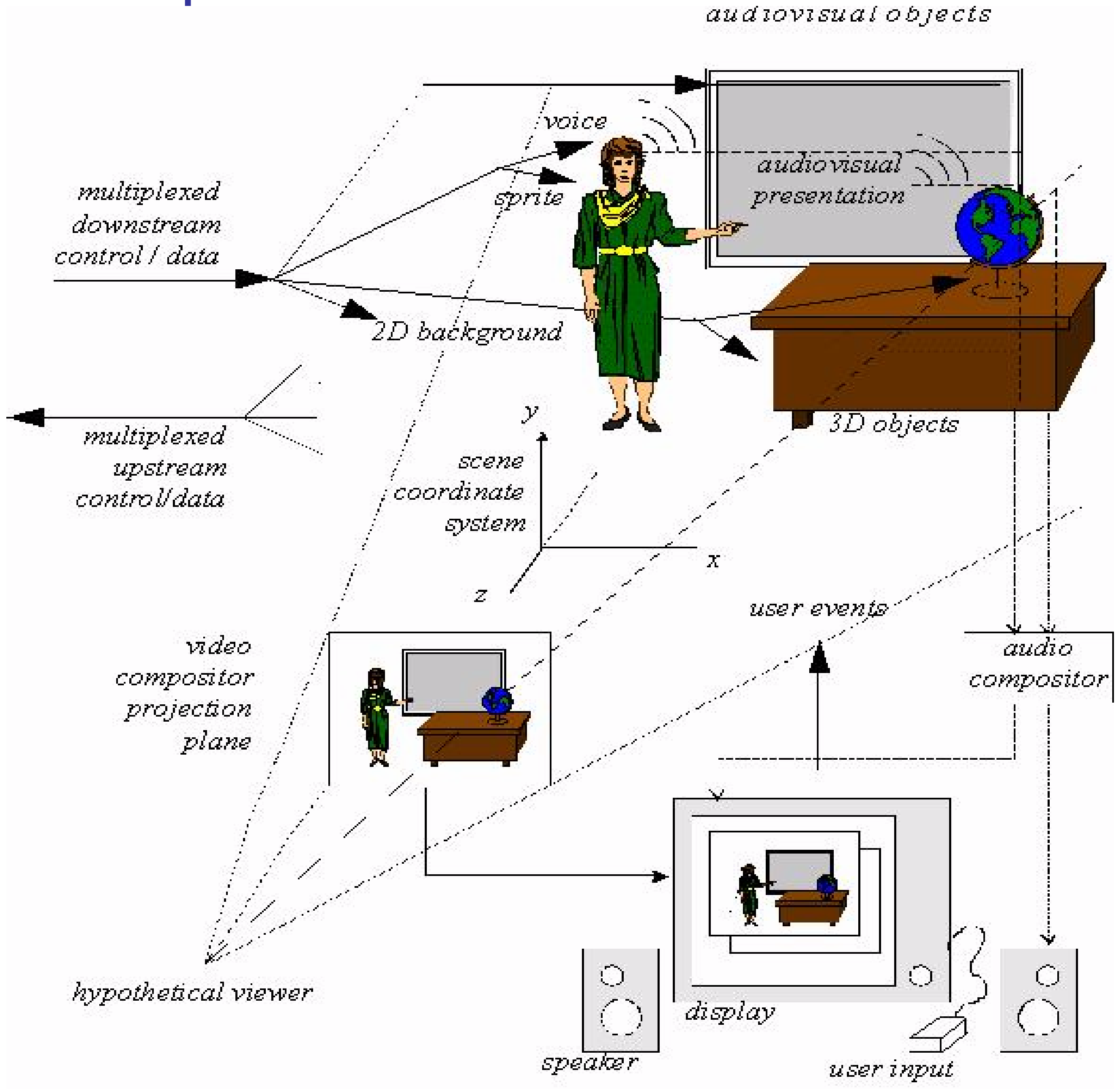
Representation of Primitive AVOs

- MPEG4 standardizes a set of primitive Media Objects (MO, also called: AVO, e.g.:
  - A 2D fixed background (still images)
  - A talking person (Video Objects without b.g.)
  - A voice, associated with a person (Audio Object)
- The MO's are organized in a hierarchical way, capable of representing both natural and synthetic content types either 2D or 3D.

Primitive AVO's (Cont'd)

- In addition, there are coded objects such as:
  - Text and graphics
  - Talking heads and associated text
  - Animated human bodies
- Even in their coded form, objects (aural or visual) can be represented independent of their surroundings or background.

An example of MPEG-4 Audiovisual scene



More examples for Scene composition

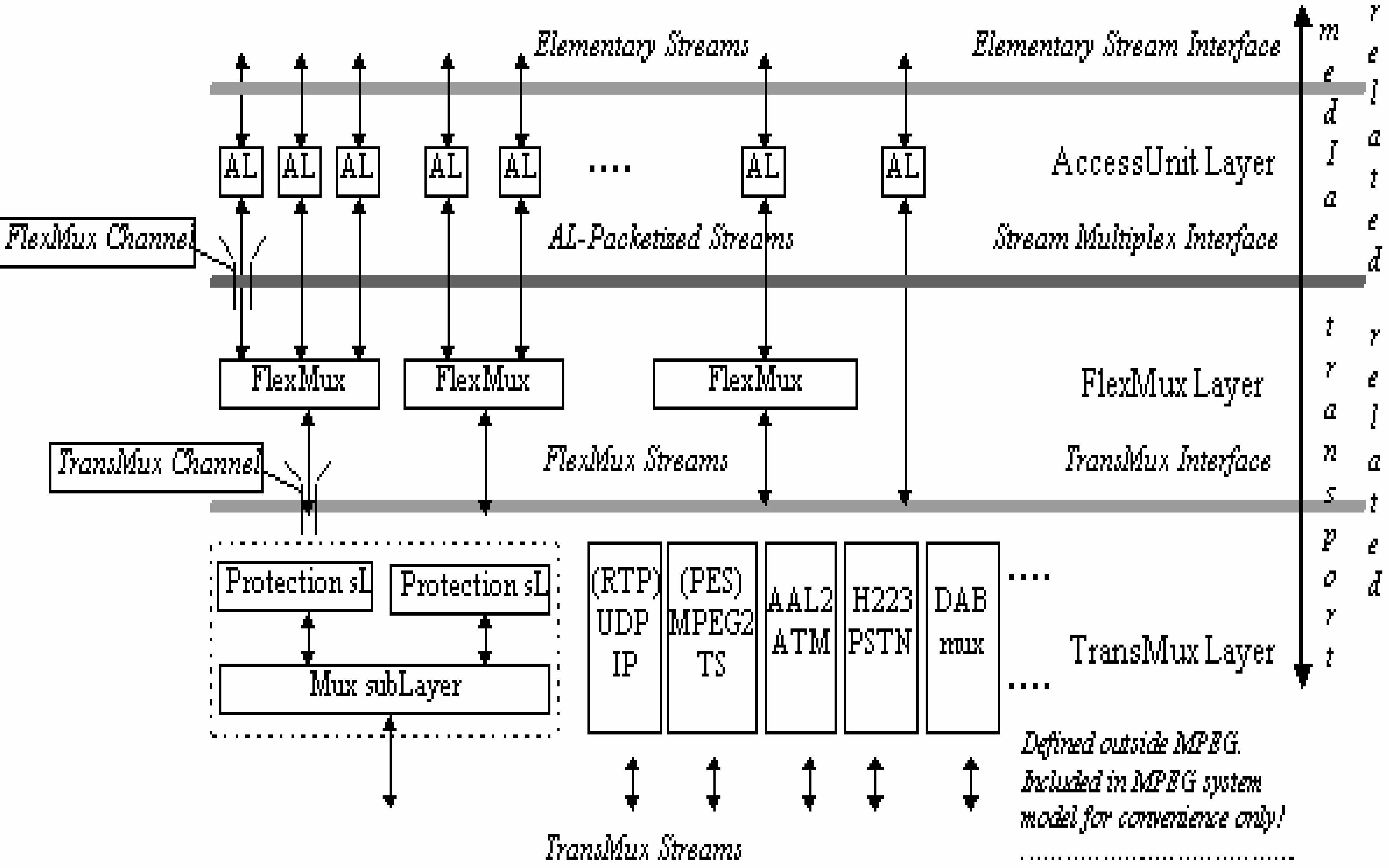
- Place AVOs anywhere in a given coordinate system
- Group primitive AVO in order to form compound AVO
- Change, interactively, the user's viewing and listening points anywhere in the scene
- Apply transforms to change geometrica/acoustical appearance.
- Apply streamed data to MO, in order to modify their attributes (e.g sound, moving texture, animation parameters)

* The scene composition borrows several concepts from VRML

Multiplexing and Synchronization of AVOs

- AVO data is delivered to one or more Elementary Streams (EM), which are characterized by:
  - the QoS they request for transmission (e.g. maximum bit-rate, BER allowed etc.)
  - Stream type information (decoder resources, timing precision needed etc.)
- This information is specified in terms of an Access Unit Layer (AUL), and a conceptual 2 layer multiplexer.

MPEG-4 system layer model



Delivery of Streaming Data

- The Access Unit Layer: Allows grouping of ES’s with low mux’ing overhead, e.g.: to group ES’s with similar QoS requirements, to reduce the number of network connection and end-to-end delay. (Delivery Multimedia Integration Framework - DMIF spec.)
- The TransMux Layer: Models the layer that offers transport services matching the requested QoS. All suitable transport protocols are supported (RTP/UDP/IP, ATM (AAL5), or MPEG-2 etc.)

Access Unit Layer

Allows :

- Identification of Access Units (video or audio frames, scene description commands) in ES.
- Recovery of the AV objects or scene description’s time base, to enable sync. Between them.

The “FlexMux” Layer

- The “Flexible Multiplexing” layer contains a tool that allows grouping of ES with a low multiplexing overhead.
- e.g.: grouping of ES with similar QoS requirements
- This layer is optional and may be bypassed if the TransMux layer provides its functionality

The “TransMux” Layer

- The “Transport Multiplexing” layer offers transport services matching the requested QoS.
- Only the I/F to this layer is specified by MPEG
- The choice is left to the end user/service provider, so MPEG-4 is not limited to one protocol

MPEG-4 system layer model allows:

- Identify AU’s, Timestamps and clock ref. Information, and recognize data loss
- Interleave data from different ES into FlexMux streams
- Convey control information to:
  - Indicate the required QoS for each ES and stream
  - Translate each QoS requirements into actual network resources
  - Convey the mapping of ES, associated with AVOs to FlexMux and TransMux channels

Interaction with AVOs

- Unlike regular movies, MPEG-4 allows the user interaction with the scene:
  - Change viewing/listening point (navigation through scene)
  - drag object into different position
  - trigger a cascade of events by clicking on a specific object
  - Select the desired language
  - and more...

Intellectual Property Rights

- It is important to have the possibility to store IPR information associated with MPEG-4 AVOs.
- The MPEG-4 standard will incorporate the functionality to store the unique identifiers issued by international numbering systems
- Protection of content will be addressed in MPEG-4 Version 2 ...

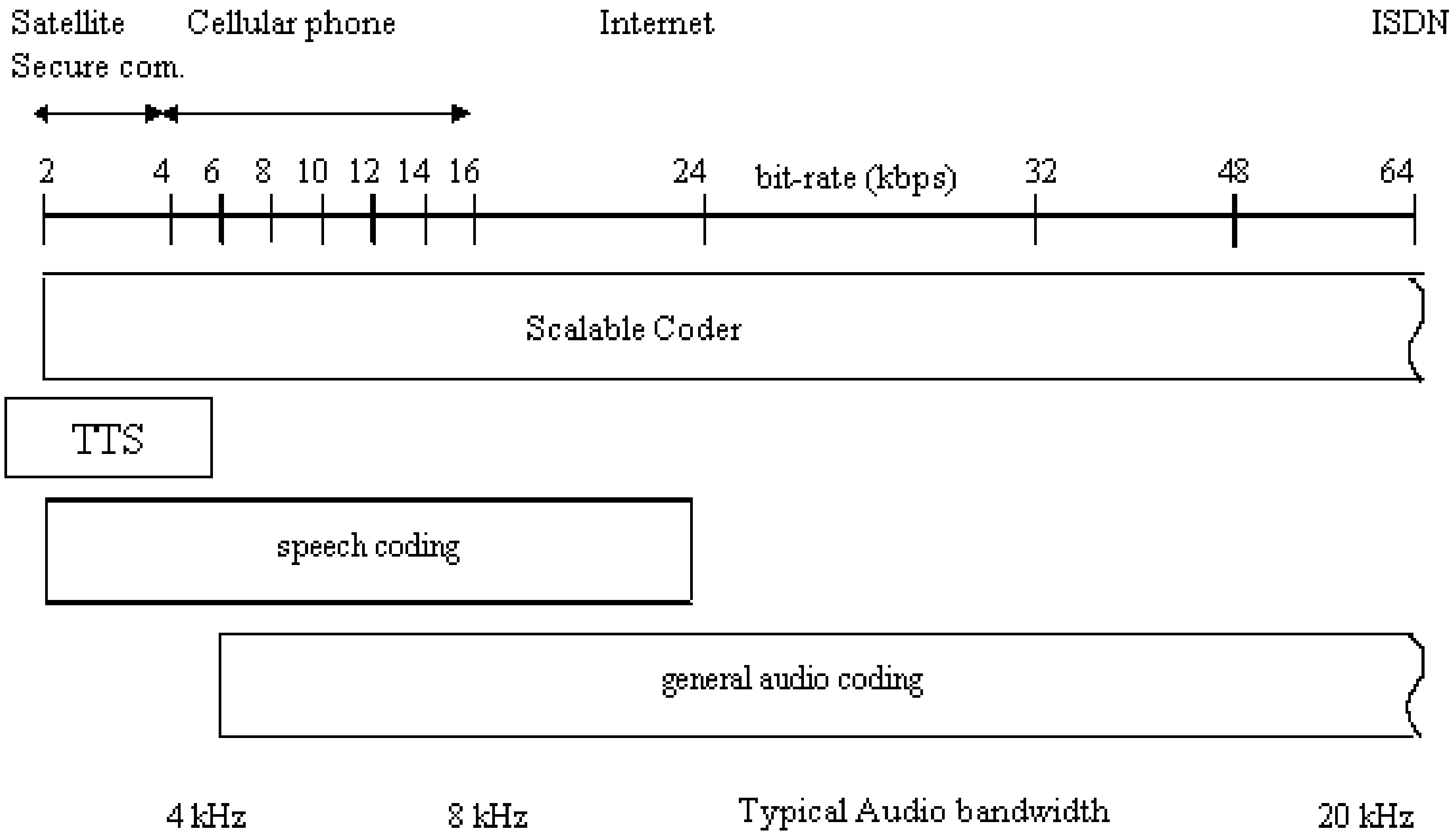
Coding of Audio Objects

- MPEG-4 coding of audio objects provides tools for representing natural sounds (such as speech and music) and for synthesizing sounds based on structured descriptions
- Compression, scalability and playing back at different speeds are supported
- synthesized sound can be formed by text or instrument descriptions and by coding parameters to provide effects such as reverberation and spatialization.

Sound Types Coding

- Natural Sound: Natural audio coding at bitrates ranging from 2Kbps upto 64Kbps, with 3 types of coders (2-4Kbps, 4-16Kbps and 16-64Kbps)
- Synthesized Sound: Text input is converted to speech in the Text-To-Speech (TTS) decoder, while more general sounds including music may be normatively synthesized.
- Effects: The effects processing includes reverberators, spatializers, mixers, limiters, dynamicrange control, filters, flanging, chorus or any hybrid of these effects.

Coverage of Optional Bitrates

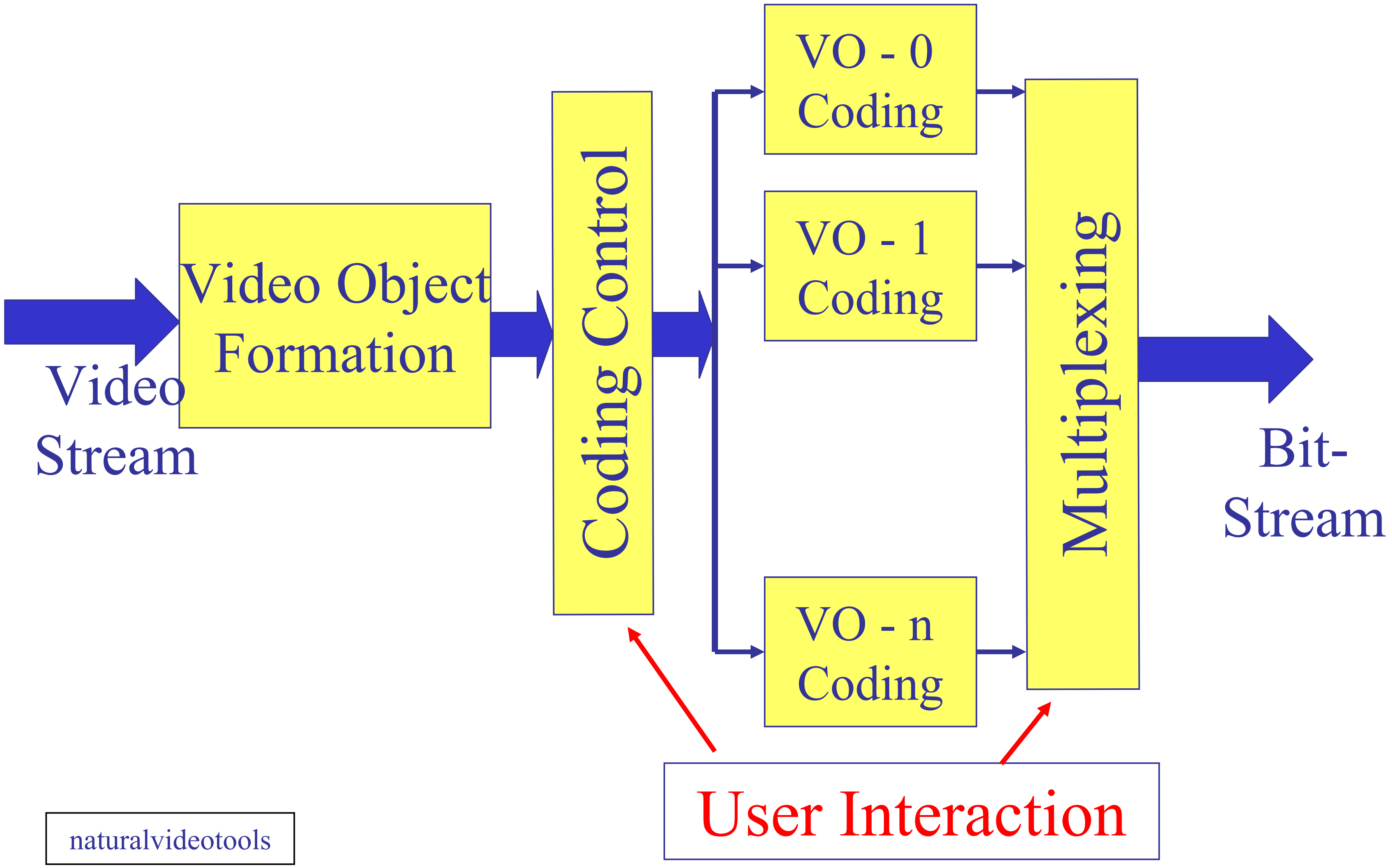


Coding of Visual Objects

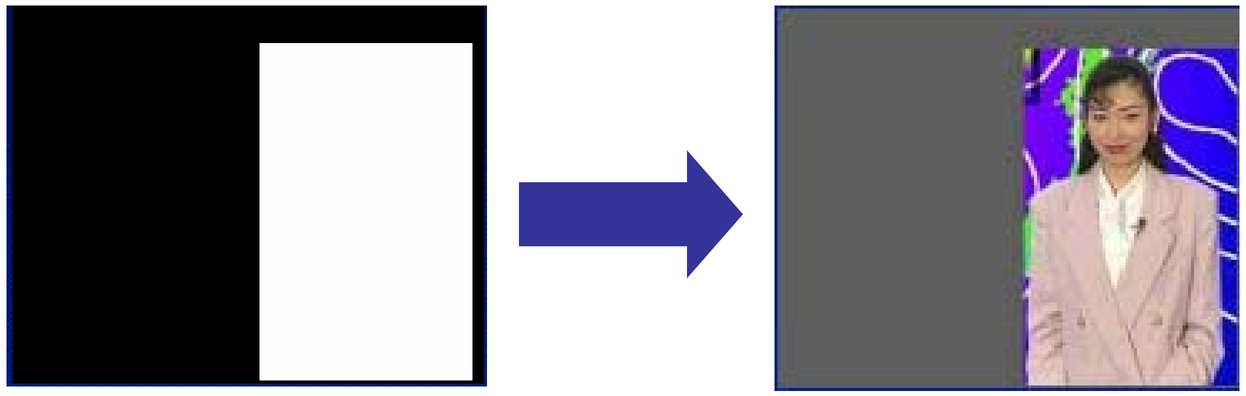
Natural Textures, Images and Video:

will 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

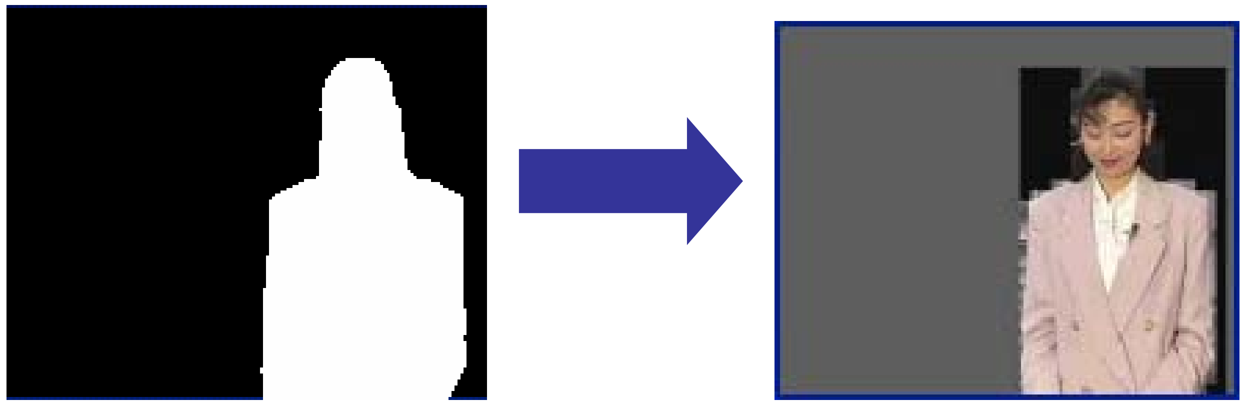
General Video Coding Block Diagram



Video Object Plane Formation

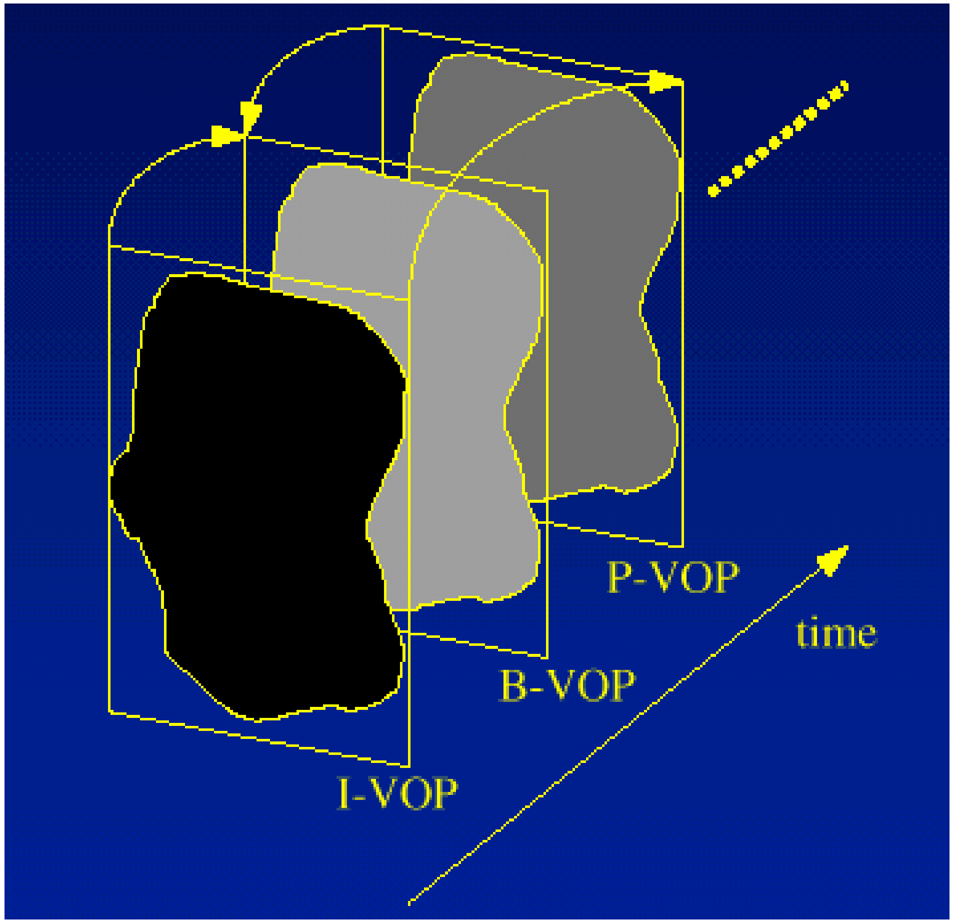


Rectangular

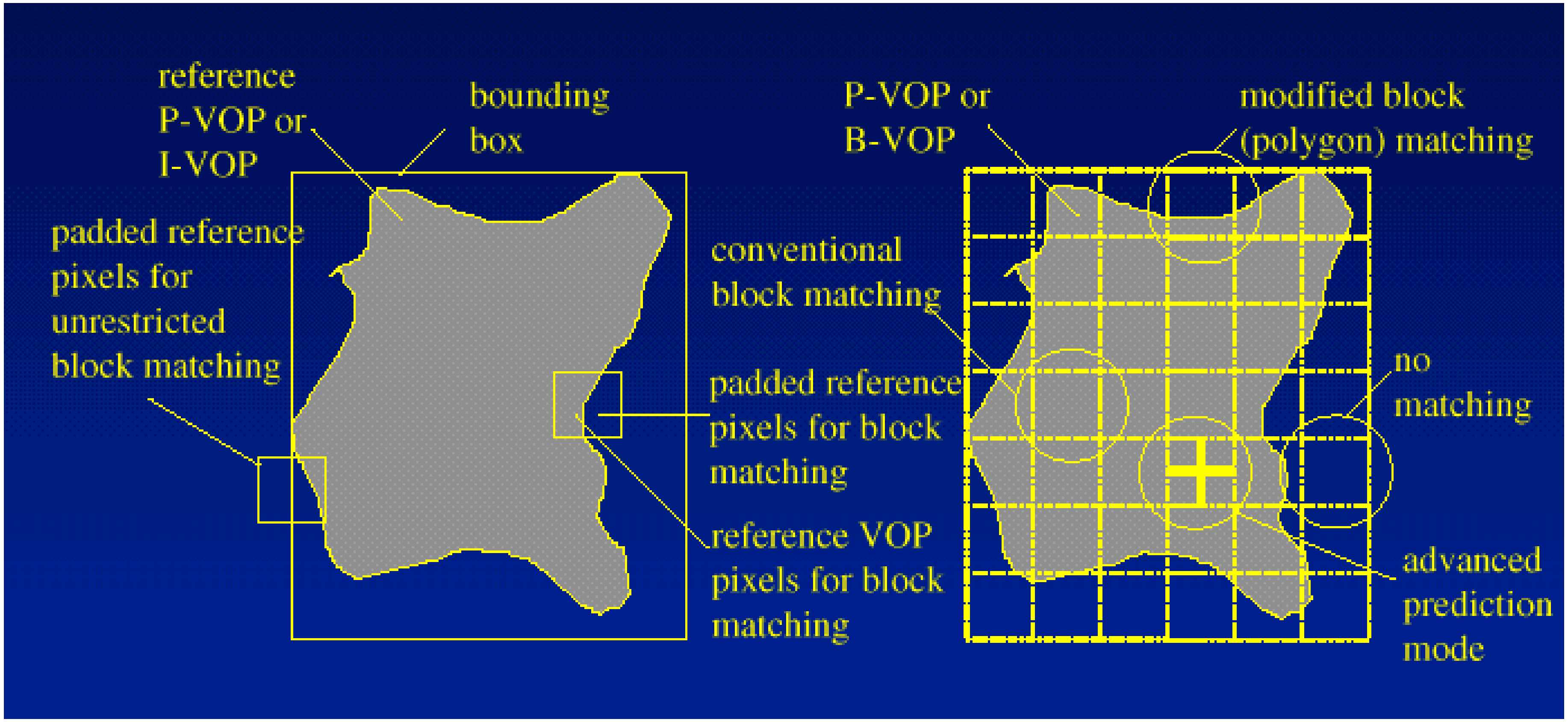


Rectangular

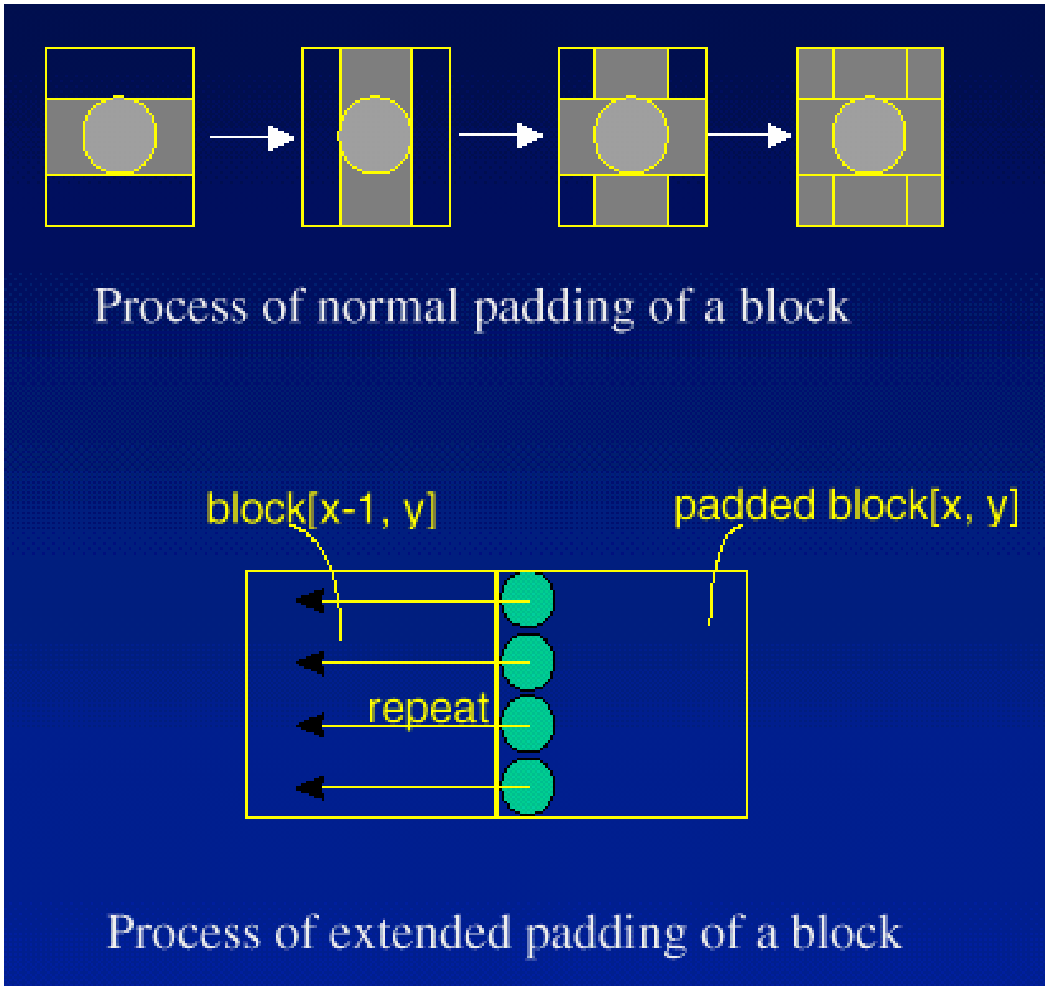
Motion Compensation Modes



Motion Vector computation


reference P-VOP or I-VOP
bounding box
padded reference pixels for unrestricted block matching
P-VOP or B-VOP
modified block (polygon) matching
conventional block matching
padded reference pixels for block matching
reference VOP pixels for block matching
no matching
advanced prediction mode

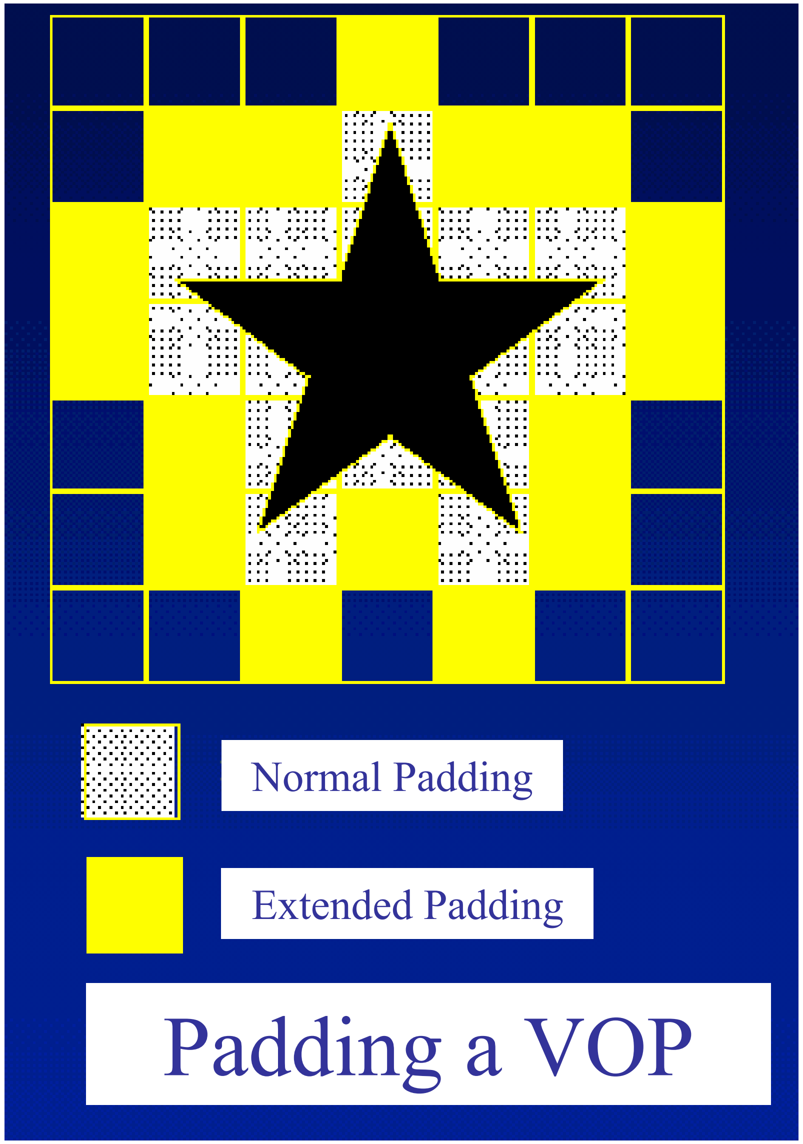
MC: Block Padding



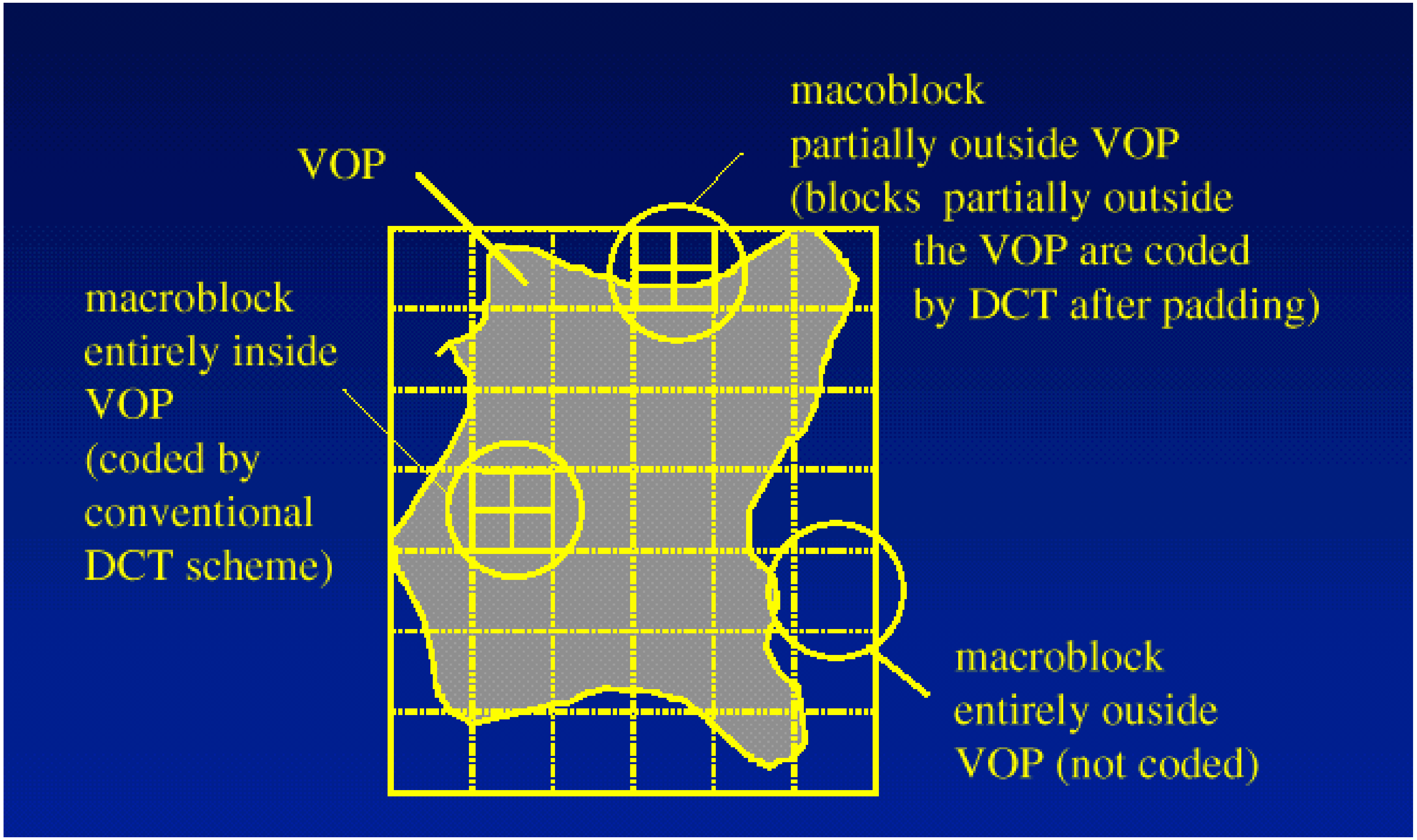
Process of normal padding of a block

Process of extended padding of a block

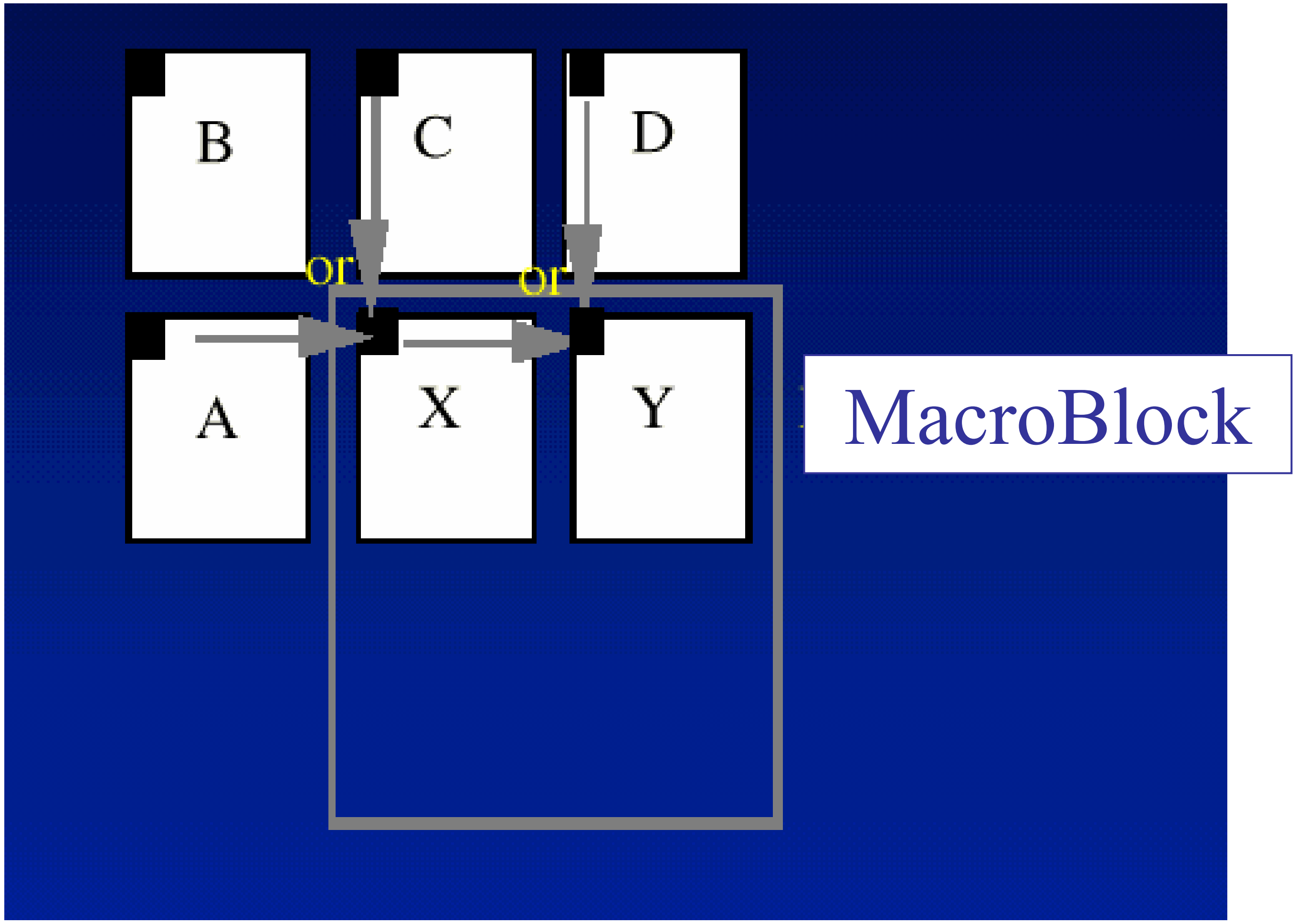
MC: VOP Padding



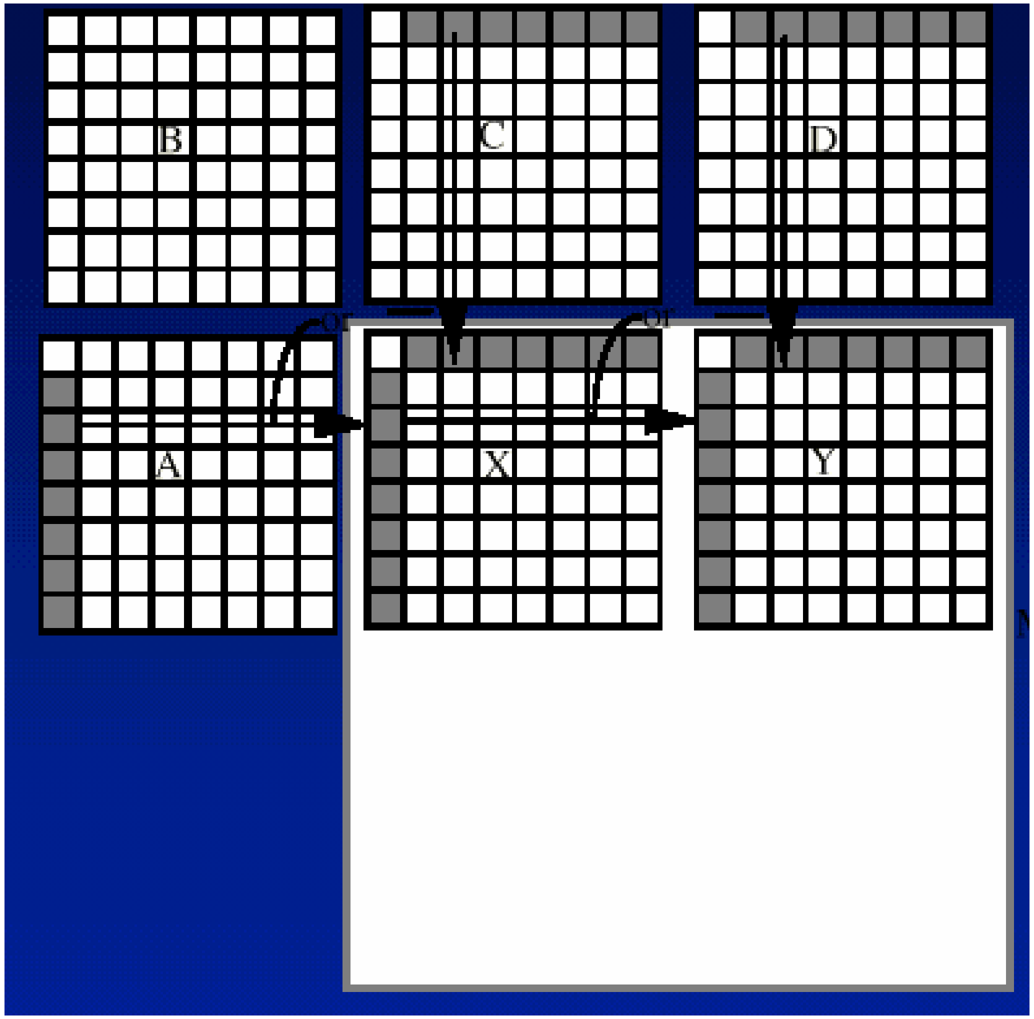
Texture Coding



Adaptive DC Prediction

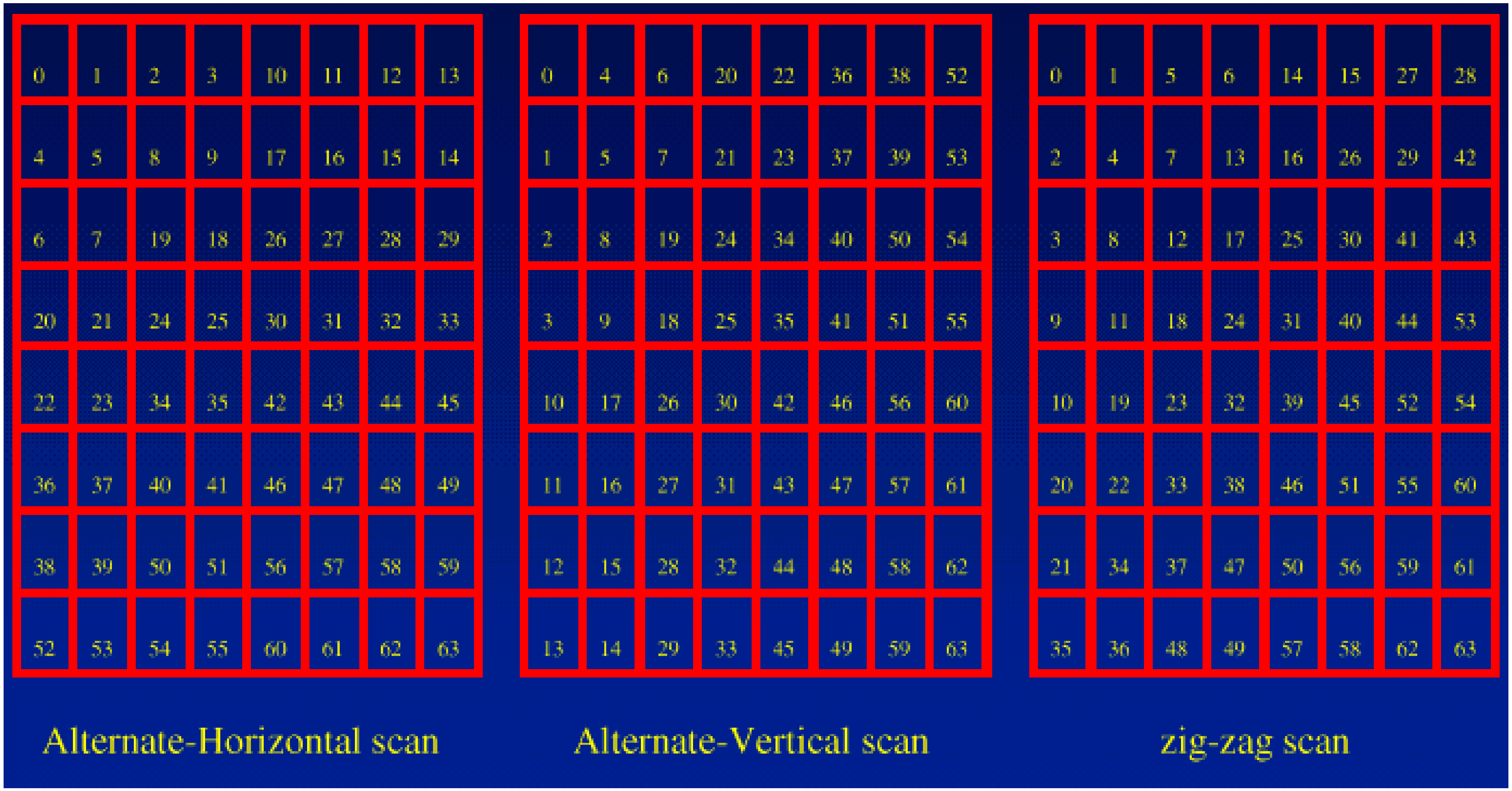


Adaptive AC Prediction



MacroBlock

DCT Scan



0	1	2	3	10	11	12	13
4	5	8	9	17	16	15	14
6	7	19	18	26	27	28	29
20	21	24	25	30	31	32	33
22	23	34	35	42	43	44	45
36	37	40	41	46	47	48	49
38	39	50	51	56	57	58	59
52	53	54	55	60	61	62	63

Alternate-Horizontal scan

0	4	6	20	22	36	38	52
1	5	7	21	23	37	39	53
2	8	19	24	34	40	50	54
3	9	18	25	35	41	51	55
10	17	26	30	42	46	56	60
11	16	27	31	43	47	57	61
12	15	28	32	44	48	58	62
13	14	29	33	45	49	59	63

Alternate-Vertical scan

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

zig-zag scan

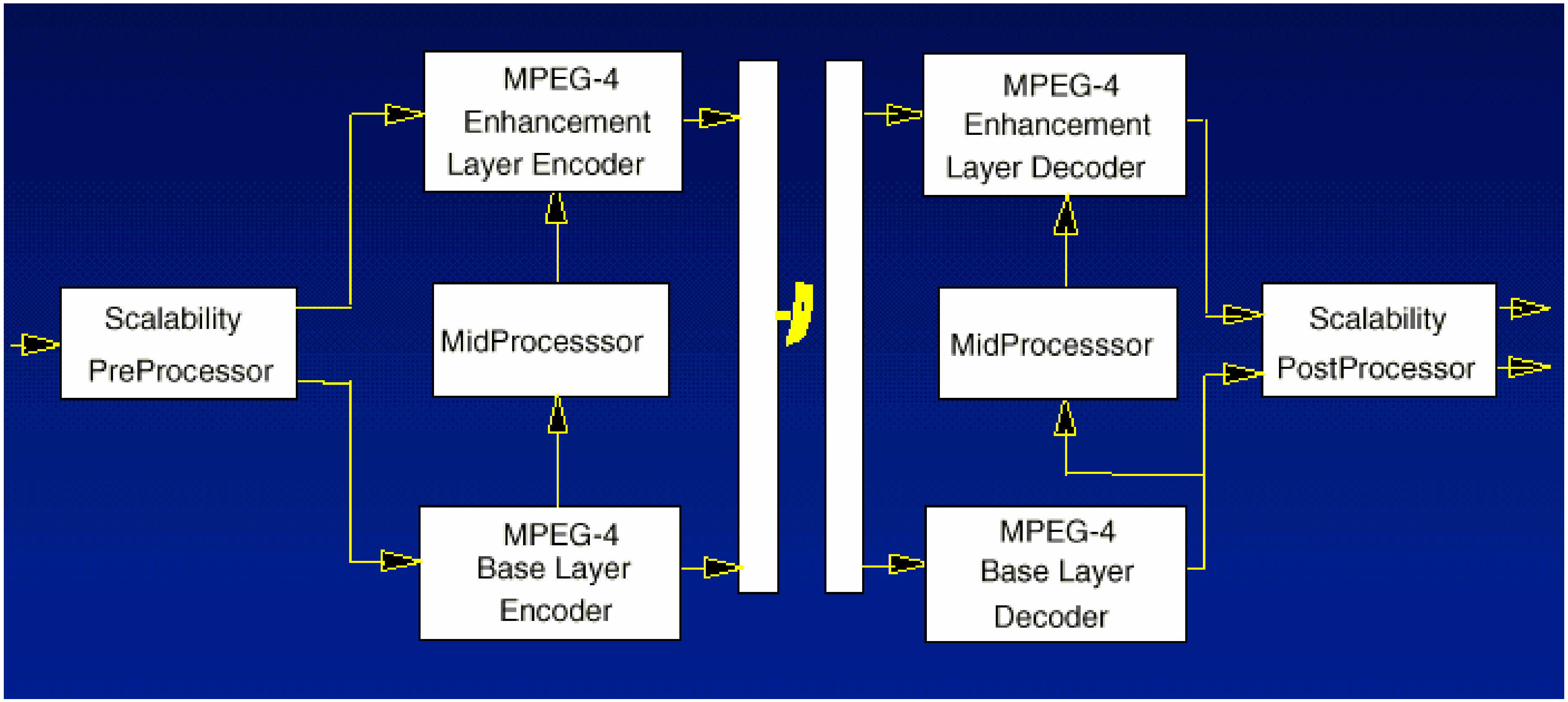
Quantization

- 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 mechanismt

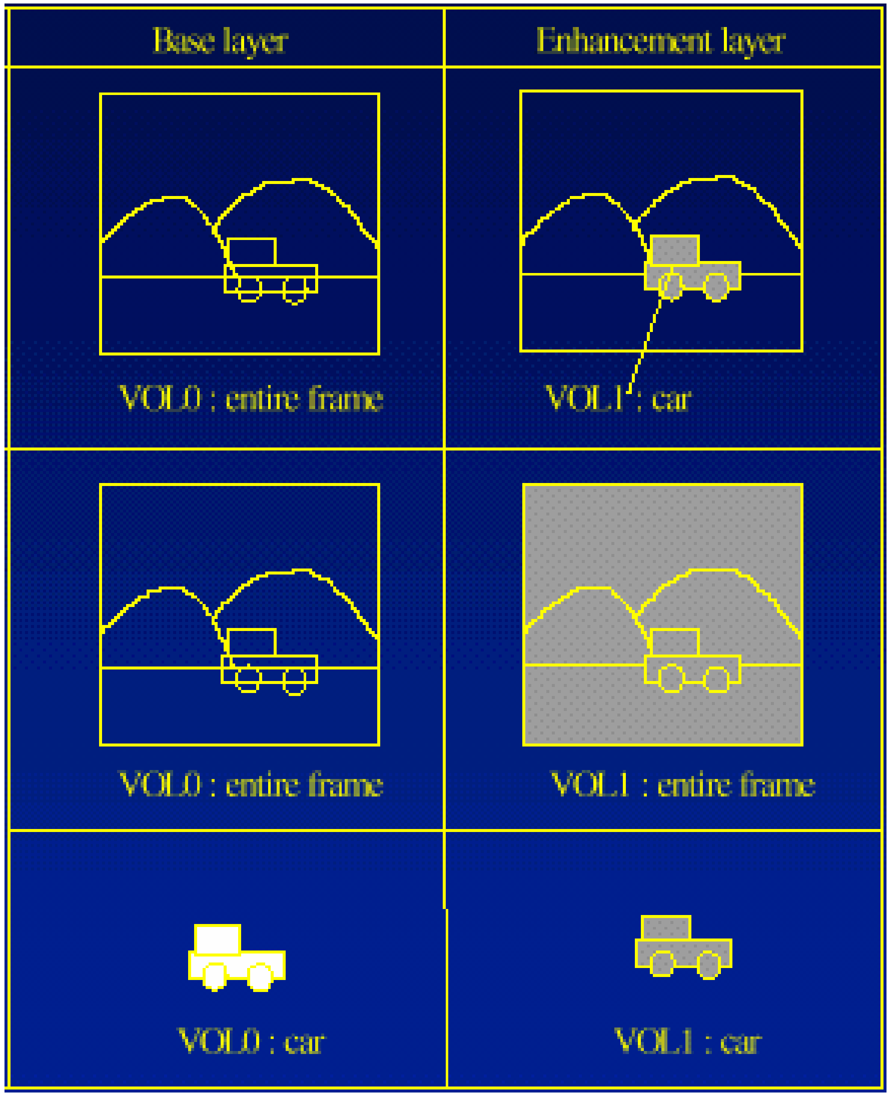
Scalability: Block Diagram


Scalability PreProcessor
MPEG-4 Enhancement Layer Encoder
MidProcesssor
MPEG-4 Base Layer Encoder
MPEG-4 Enhancement Layer Decoder
MidProcesssor
MPEG-4 Base Layer Decoder
Scalability PostProcessor

Temporal scalability

- Temporal scalability is optional 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



Type 1: Only a portion of the VOP in the base layer is enhanced

Type 2: The entire VOP in the base layer is enhanced

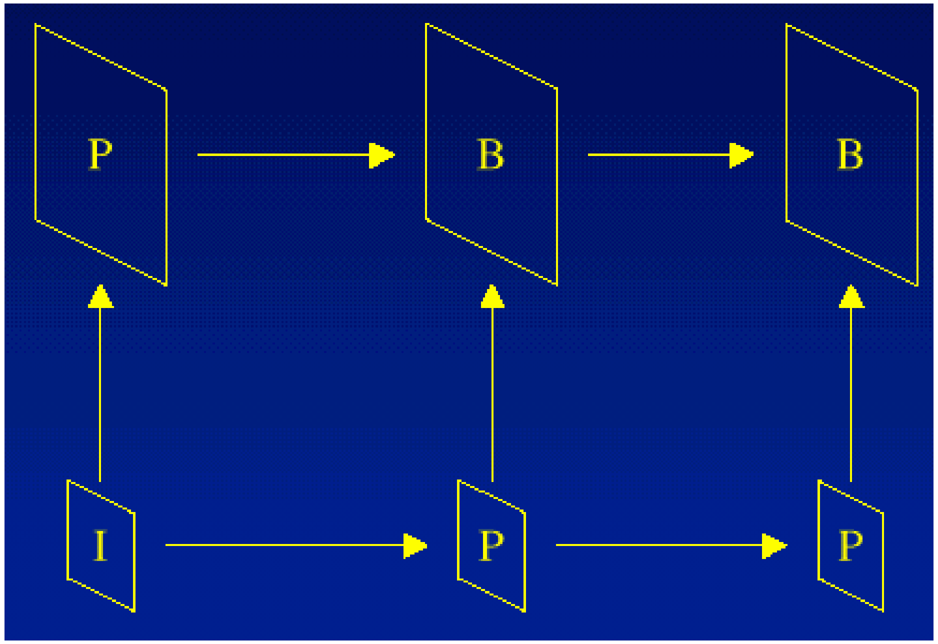
▨ region to be enhanced by an enhancement layer

Spatial Scalability

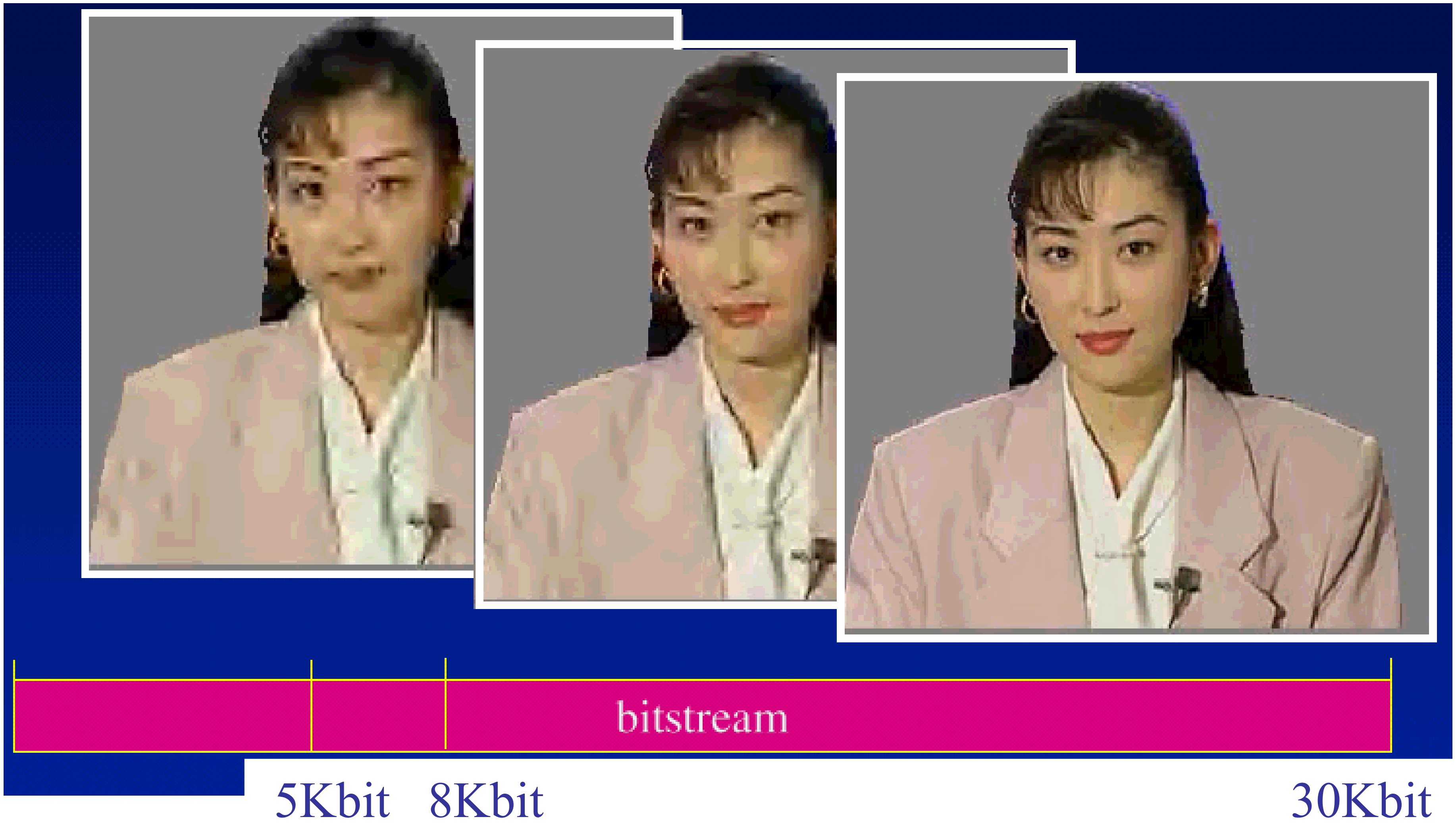
- The base layer is coded as conventional MPEG- 4 video
- The enhancement layer is encoded using prediction mechanism from the base layer

Enhancement Layer

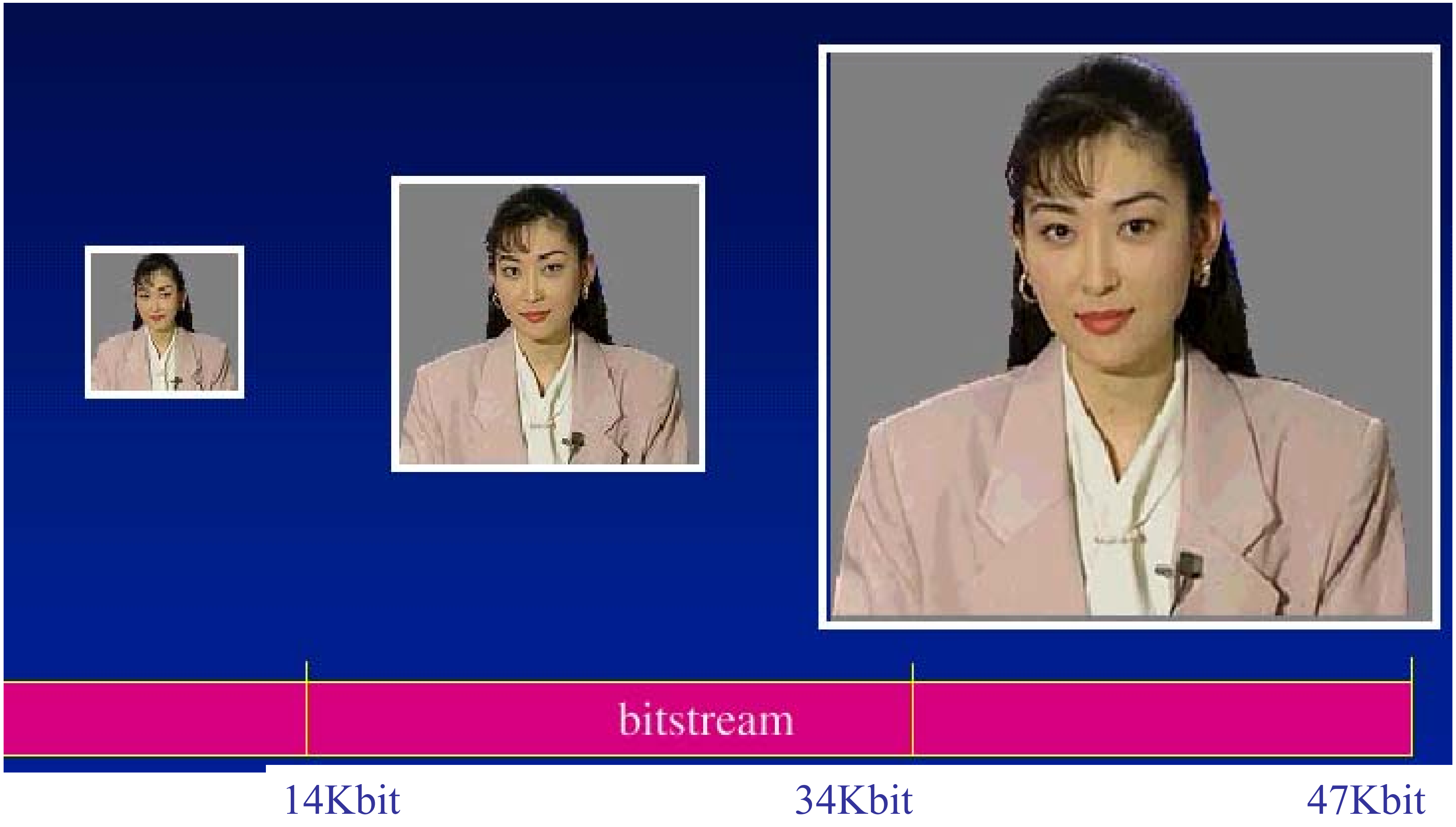
Base Layer



SNR Scalability:
Shape adaptive wavelet coding



Spatial Scalability: Shape Adaptive wavelet Coding



Coding of Visual Objects (cont'd)

Solutions and algorithms for:

- efficient compression of images and video
- efficient compression of textures for texture mapping on 2D and 3D meshes
- efficient compression of implicit 2D meshes
- efficient compression of time-varying geometry streams that animate meshes
- efficient random access to all types of visual objects

Coding of Visual Objects (Cont'd)

- extended manipulation functionality for images and video sequences
- content-based coding of images and video
- content-based scalability of textures, images and video
- spatial, temporal and quality scalability
- error robustness and resilience in error prone environments

Coding of Visual Objects (Cont'd)

Synthetic objects: form a subset of the larger class of computer graphics, as an initial focus the following visual synthetic objects will be described:

- Parametric descriptions of:

a) a synthetic description of human face and body

b) animation streams of the face and body

- Static and Dynamic Mesh Coding with texture mapping
- Texture Coding for View Dependent applications

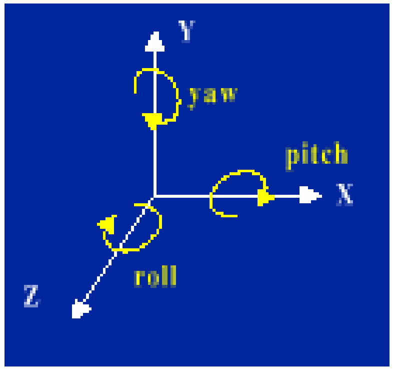
Synthetic Natual Hybrid Coding

- Version 1 (Dec. 1998):
  - Face animation
  - 2D dynamic mesh
  - Scalable coding of synthetic texture
  - View dependent scalable coding of texture
- Version 2 (Dec. 1999):
  - Body animation
  - 3D model compression

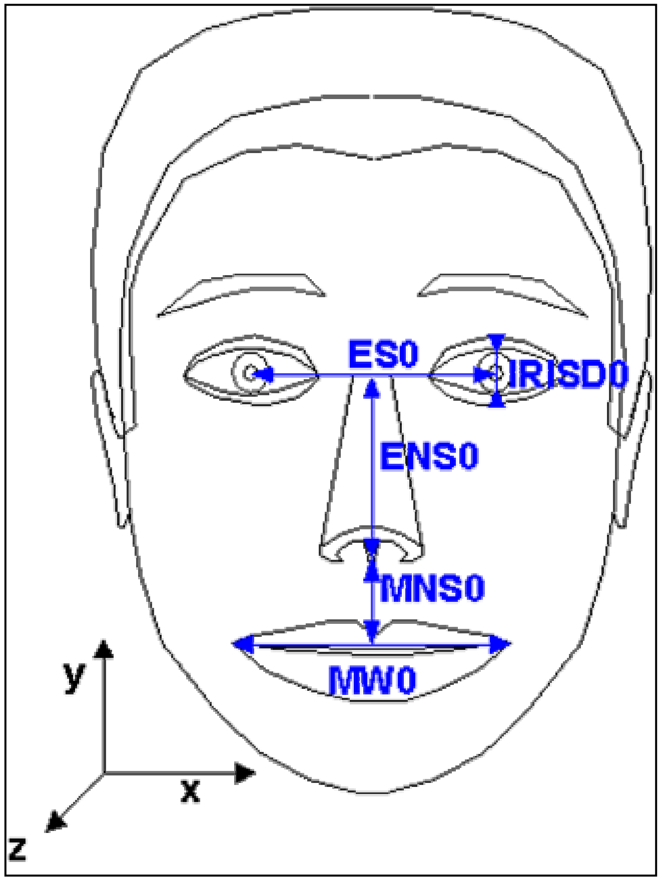
Synthetic objects: Facial animation

- The face is an object capable of facial geometry ready for rendering and animation. The shape, texture and expressions of the face are generally controlled by the bitstream containing instances of Facial Definition Parameter (FDP) sets and/or Facial Animation Parameter (FAP) sets
- Upon construction, the Face object contains a generic face with a neutral expression.

Face Model: Shape, texture and expressions



- Gaze along the Z axis
- All face muscles relaxed
- Eyelids tangential to the iris
- Pupil one- third of full eye
- Lips: in contact; horizontal
- Mouth: closed; upper and lower teeth touching
- Tongue: flat; tip touching front teeth

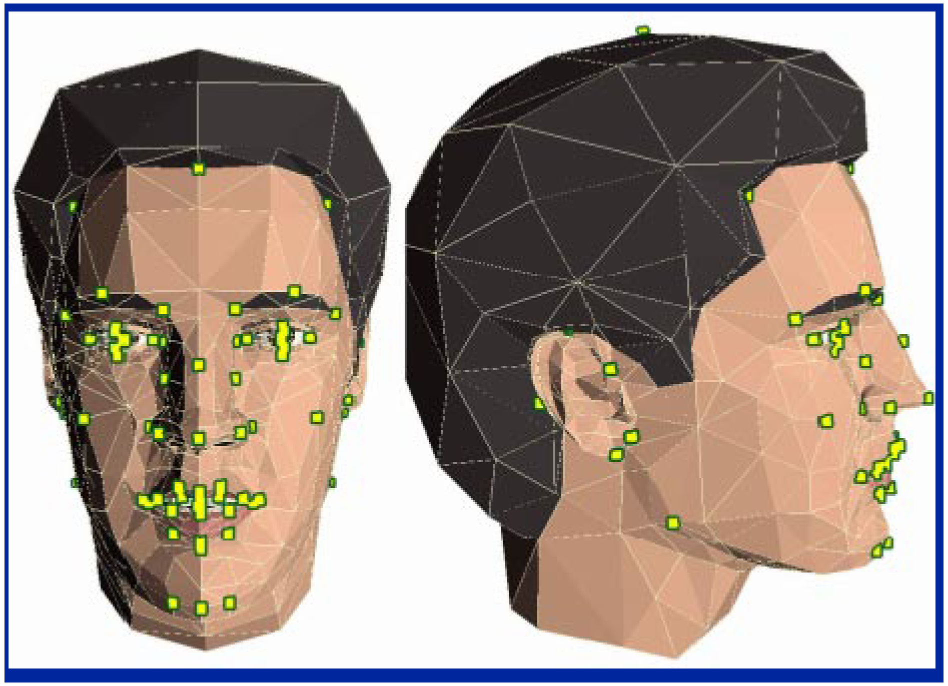


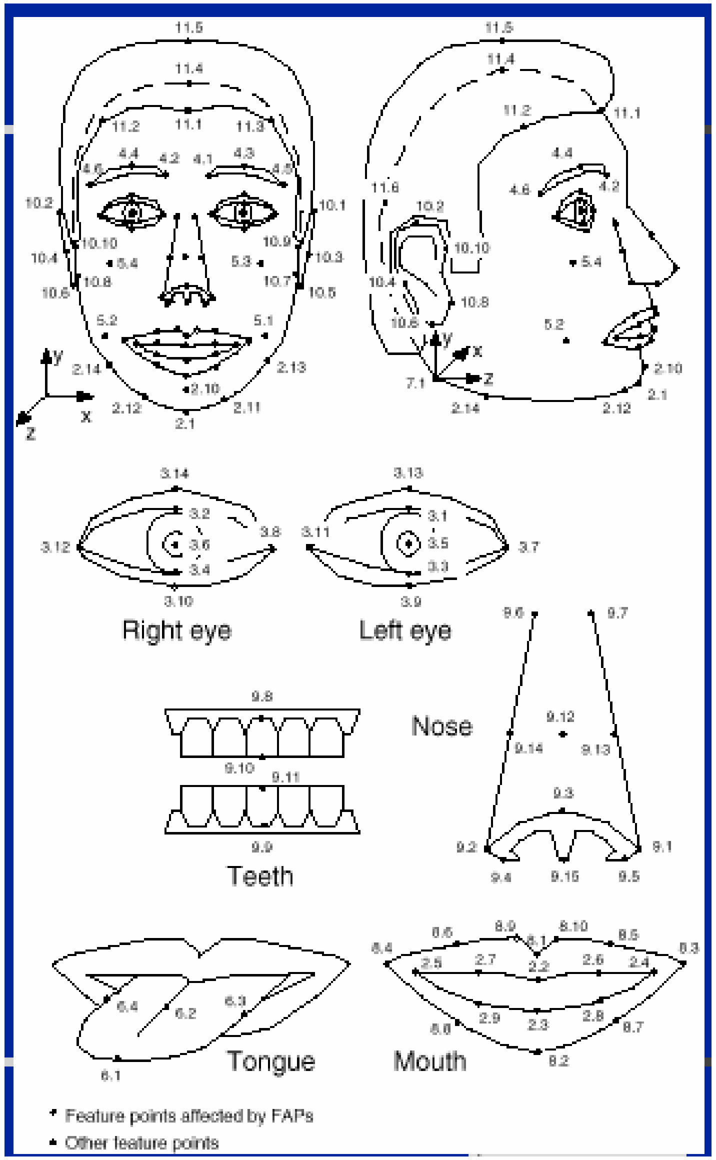
Face Animation Parameters

- Three sets of parameters used to describe a face and its animation characteristics:
  - Facial Definition Parameters (FDPs)
  - Facial Animation Parameters (FAPs)
  - Facial Interpolation Transform (FIT)
- Defines a <u>specific face</u> via:
  - 3D feature points
  - 3D mesh/ scene graph
  - Face Texture
  - Face Animation Table (FAT)

Face Feature Points

3D mesh and feature points





FDPs

- Two modes:
  - To customize the face model at the receiver to a particular face
  - To download a face model along with its animation information
- Generally, sent once per session
  - for calibration and/ or download
  - could be sent more often for “special effects”
- Specified using BIFS

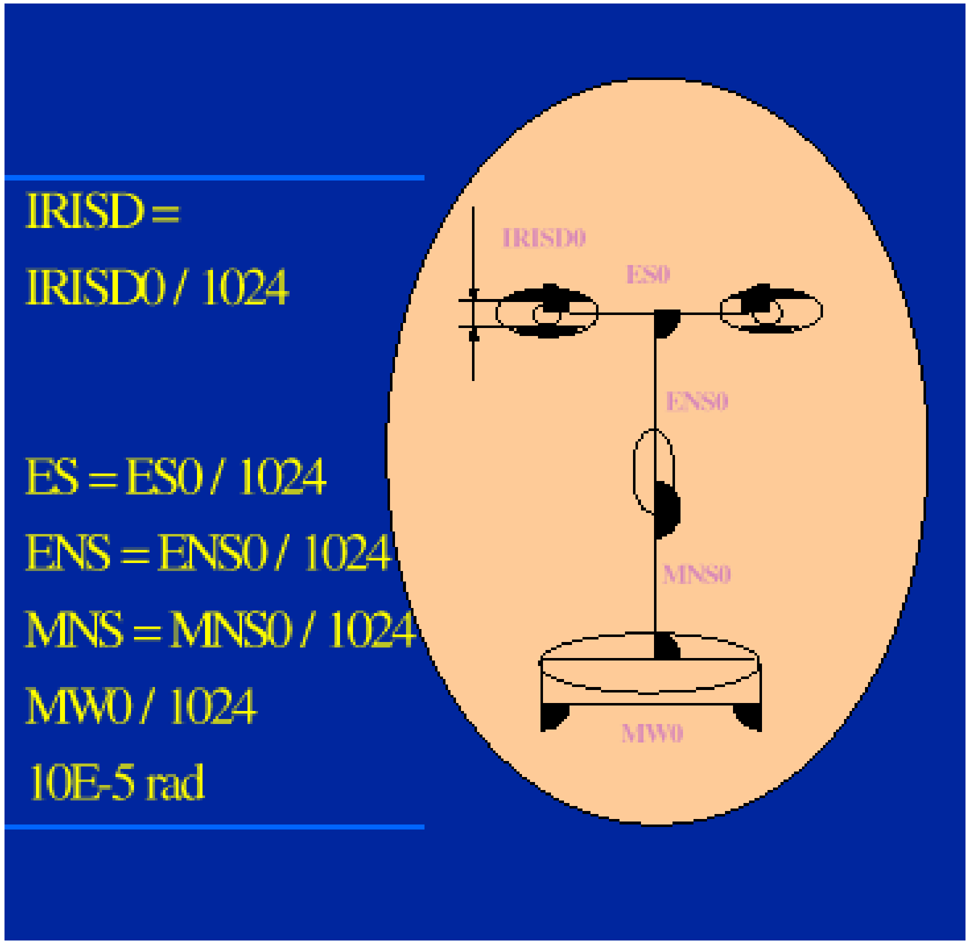
FAPs

- Represent a complete set of facial actions: allow representation of most of the natural facial expressions
- All FAPs involving translation movement: in terms of Facial Animation Parameter Units (FAPUs)
- Allows consistent interpretation of FAPs on any facial model.

FAPUs

IRISD0 Iris diameter (by definition it is equal to the distance between upper and lower eyelid) in neutral face

ES0	Eye separation
ENS0	Eye - nose separation
MNS0	Mouth - nose separation
MW0	Mouth width
AU	Angle Unit


IRISD =

IRISD0 / 1024

ES = ES0 / 1024

ENS = ENS0 / 1024

MNS = MNS0 / 1024

MW0 / 1024

10E-5 rad

High-level and Low-level FAPs

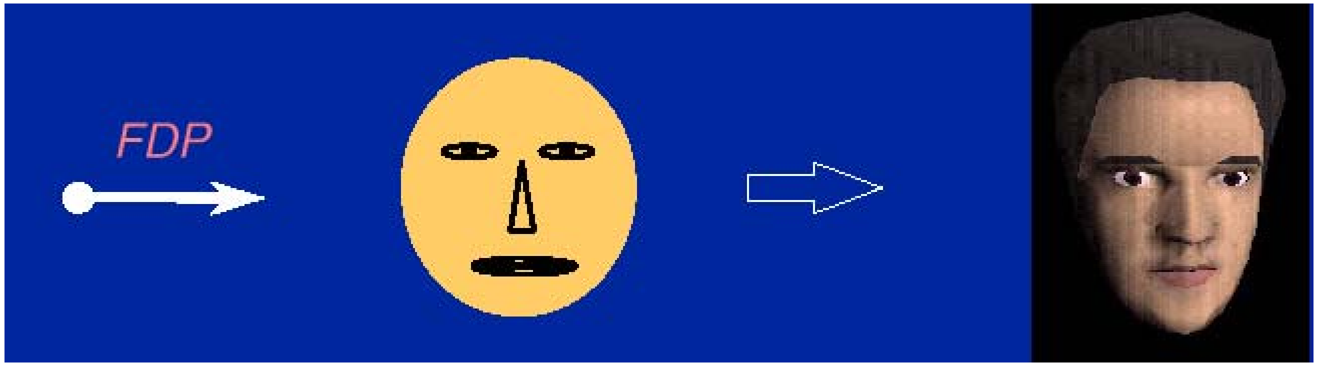
- 2 high- level FAPS:
  - *Viseme* (visual correlate of phoneme)
  - *Expression* (joy, anger, fear, disgust, sadness, surprise)
    - Textual description of expression parameters
    - Points to groups of FAPs used together to achieve an expression
- 66 Low-level FAPs
  - associated with the displacement or rotation of the facial feature points

FITs

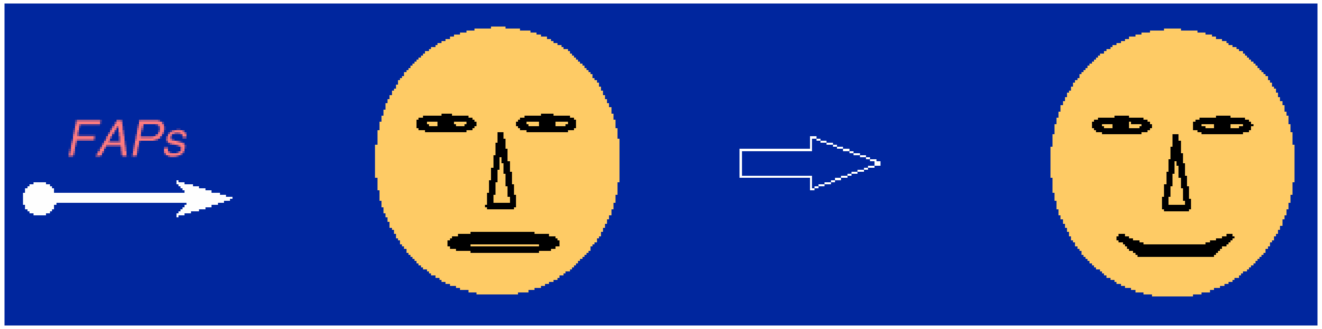
- Specification of interpolation rules for some/all FAPs by the sender.
- Sender specifies FAP Interpolation Graph (FIG) and set of interpolation functions.
- Allows higher degree of control over the animation results.

Animation Control

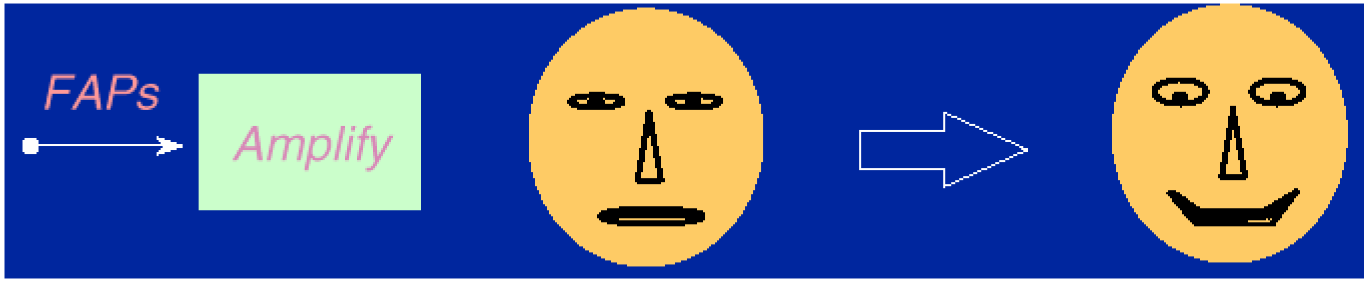
Using FDP:



Using FAP:



Amplified FAP:



The standard supports:

- FDPs
  - BIFS Syntax and Semantics
  - Rules for decoding and adaptation
- FAPs
  - Bitstream syntax and semantics
  - Rules for decoding and animation
- FITs
  - Syntax and semantics
  - Decoding rules

The standard do not support:

- The way to extract the parameters
  - Markers
  - Speech driven
  - Image analysis

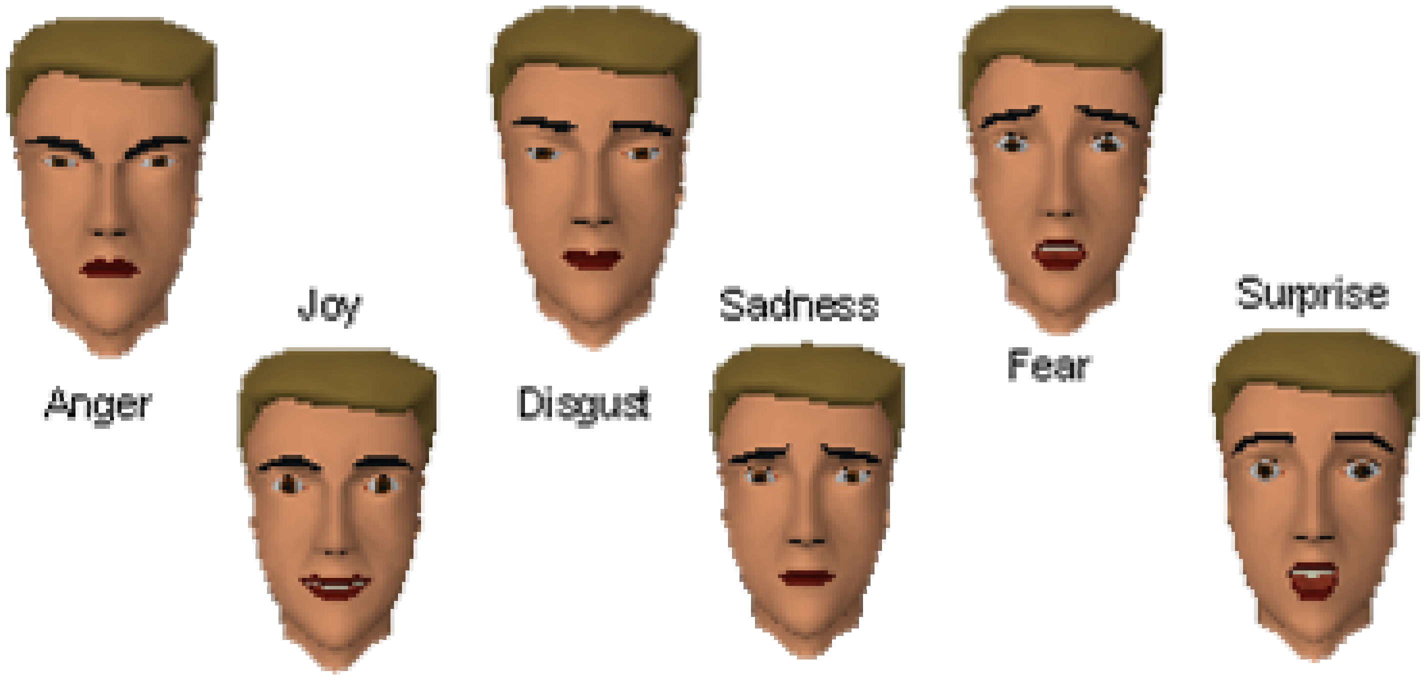
  and feature extraction

- The choice on which parameters to code and with which precision
  - Quantization
  - Rate control

Facial animation: Local controls

- local controls that can be used to modify the look or behavior of the face locally by a program or by the user
- three possibilities of local control:
  - sending locally a set of FDPs to the Face the shape and/or texture can be changed
  - a set of Amplification Factors can be defined, each factor corresponding to an animation parameter in the FAP set
  - Filter Function: The Face object passes the original FAP set to the Filter Function, which applies any modification to it

Facial Expressions



Body Animation Objectives

- For streaming human- like characters
- Supports various applications
  - from realistic simulation of human motions to network games using simple human-like models.
- Applications:
  - Virtual meeting, tele- presence, …
  - Virtual story teller, virtual actor user interface
  - Games, avatars (chat icons), education, medicine ….

Body Animation Rendering

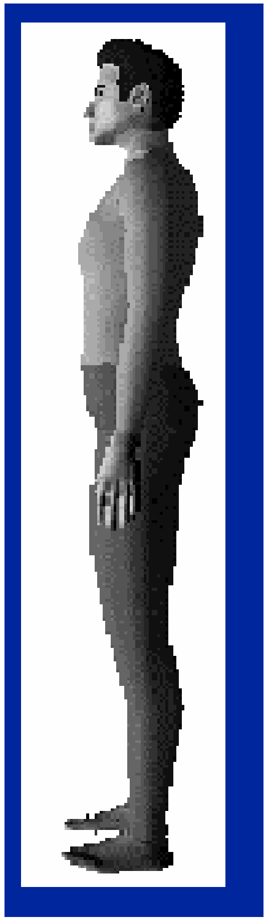
Body : an object ready for rendering and animation

- A realistic representation of a “human” body
- Capable of animation by a reasonable set of parameters
- Extension of the notion of face object to FBA object

Body Shape, texture and gesture:

- specified parameters in the incoming bitstream

- Both remote and local control of these parameters



Body Animation

- An object capable of producing virtual body models and animations in form of a set of 3D polygon meshes ready for rendering
- <u>2 sets of parameters</u> are defined for the body:
  - <u>Body Definition Parameter (BDP):</u> parameters to transform the default body to a customized body with its surface, dimensions, and texture
  - <u>Body Animation Parameter (BAP):</u> produce reasonably similar high level results in terms of body posture and animation on different body models

Body Animation Table (BAT)

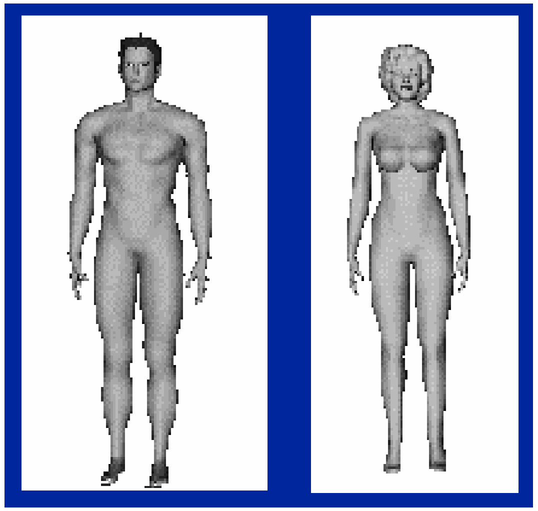
A downloadable function mapping
from incoming BAPs
to body surface geometry
that provides a combination of BAPs
for controlling
body surface geometry deformation

Face & Body Animation (FBA) Object

- a collection of nodes in a scene graph which are animated by the FBA object bitstream
- FBA is controlled by two separate bitstreams:
  - BIFS: contains instances of Body Definition Parameters (BDPs) in addition to Facial Definition Parameters (FDPs)
  - FBA: contains Body Animation Parameters (BAPs) together with Facial Animation Parameters (FAPs)

The Parameters

- The Body Animation Parameters (BAPs):
  - Produce similar body posture and animation on different body models
  - No need to initialize or calibrate the model
- Body Definition Parameters (BDPs):
  - Transform the default body to a customized body
  - Body dimensions
  - Body surface geometry
  - Joint center locations
  - Textures

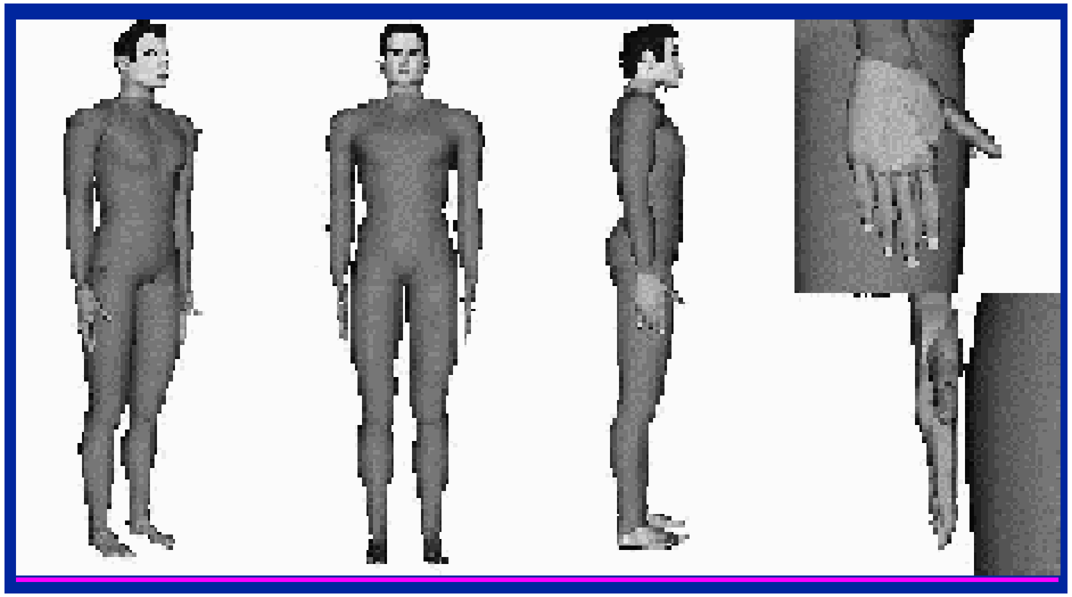


Default Body

- A generic virtual human or human- like body with the default posture
- Can be rendered immediately
- Can receive BAPs immediately for animation
- If BDPs are received, they are used to transform the decoder's generic body <u>into a particular body determined</u> by the parameter contents
- <u>No assumption is made and no limitation</u> is imposed on the range of defined mobilities for humanoid animation

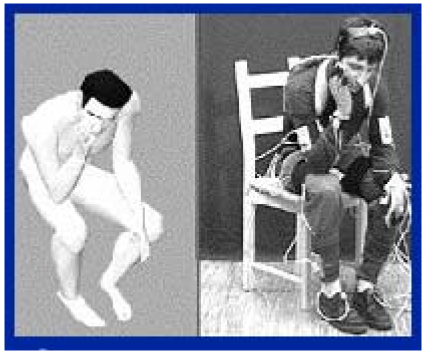
Default Posture

- Default posture is defined by standing posture:
  - The feet should point to the front direction
  - The two arms should be placed on the side of the body with the palm of the hands facing inward
  - This posture also implies that all BAPs have default values as 0

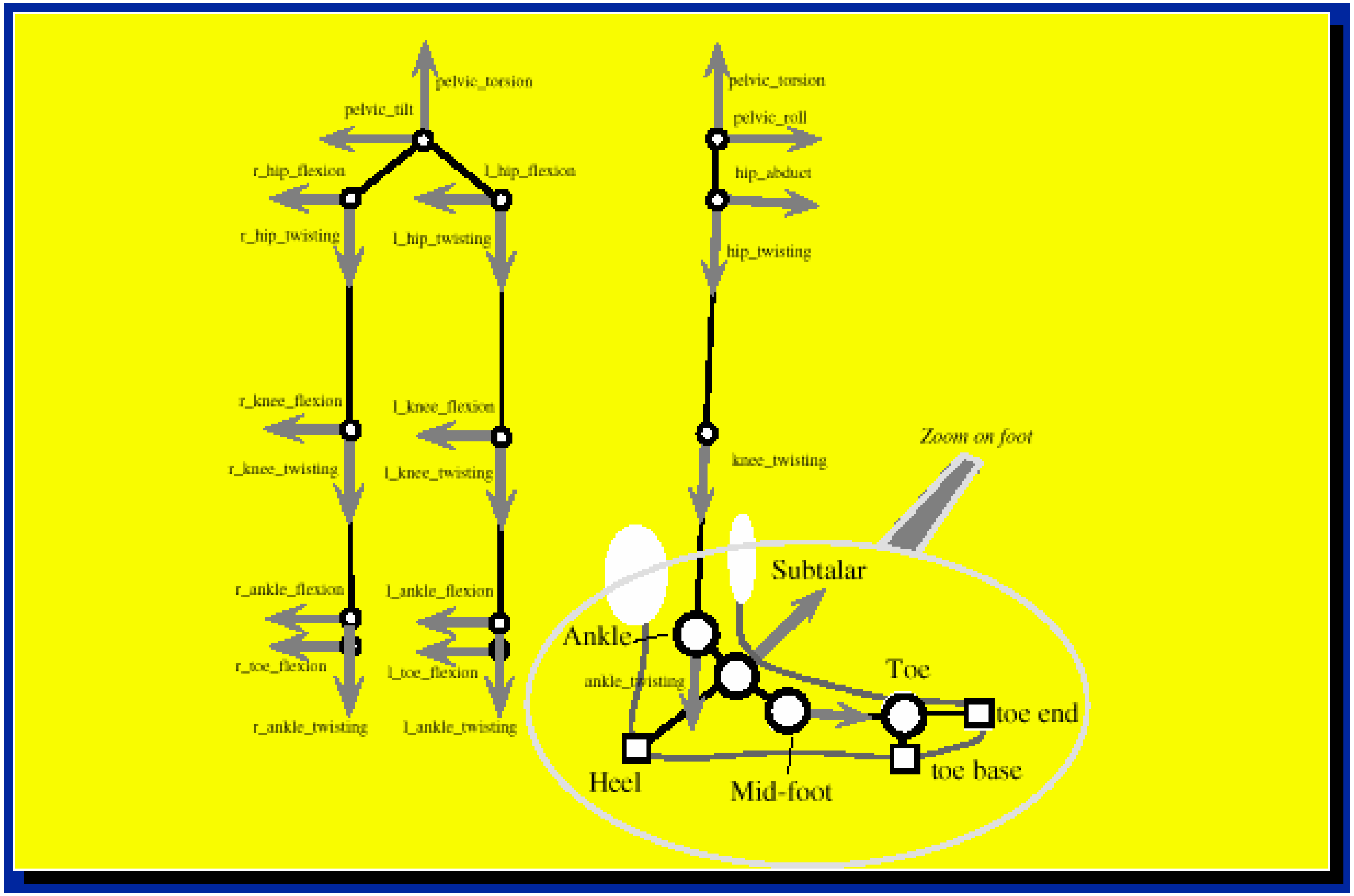


BAPs

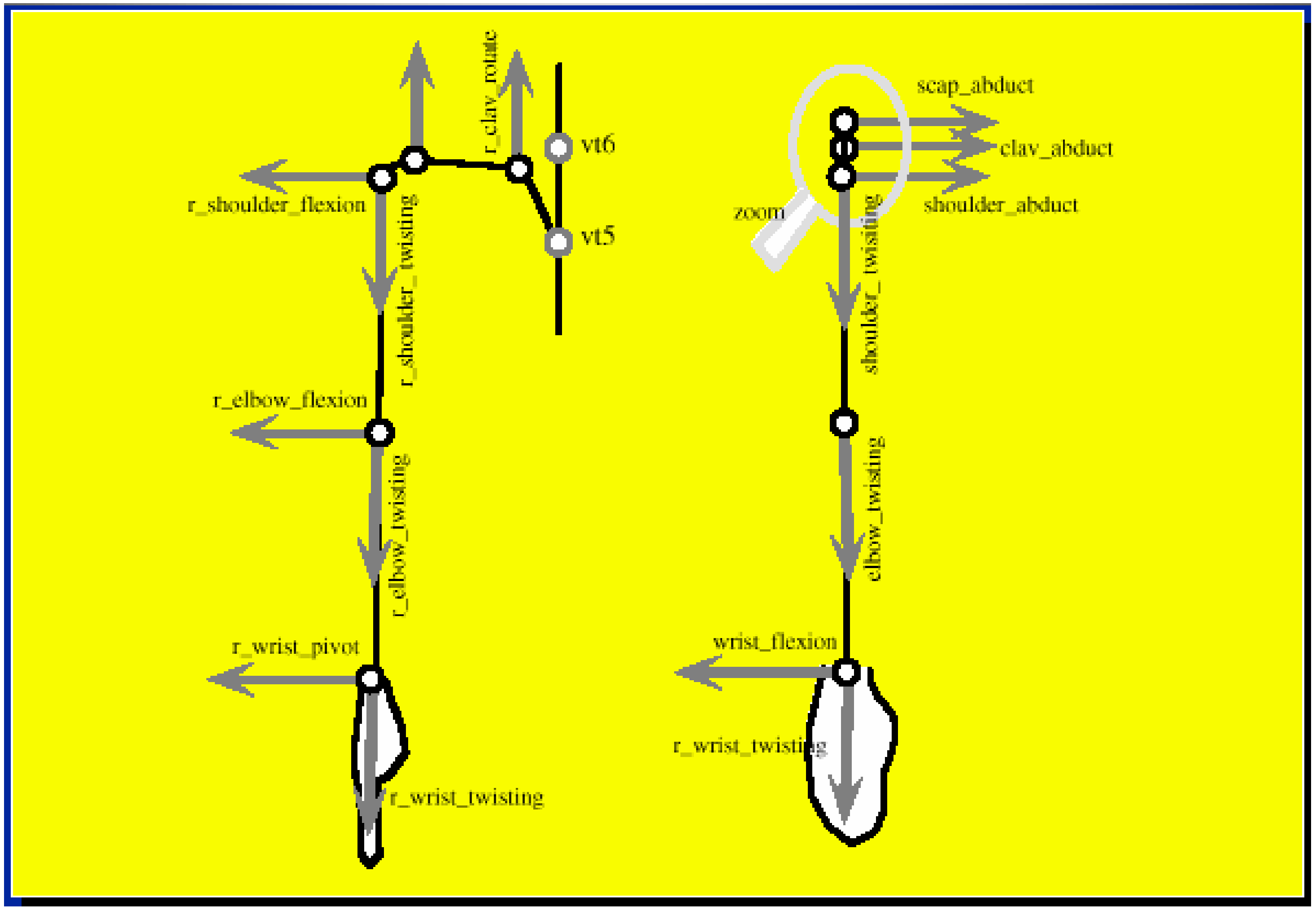
- BAPs comprise joint angles connecting different body parts, including: toe, ankle, knee, hip, spine (C1- C7, T1- T12, L1- L5), shoulder, clavicle, elbow, wrist, and the hand fingers
- The unit of rotations (BAPU) is defined as Pi/10^{-5} rad.
- The unit of translation BAPs is defined in millimeters
- 175 Body Animation Parameters
- 125 BAPs for body, 25 BAPs for each hand
- 19 groups, e. g. left_ leg1,
- left_ leg2, left_ arm1, etc.
- classified: visual effect on posture



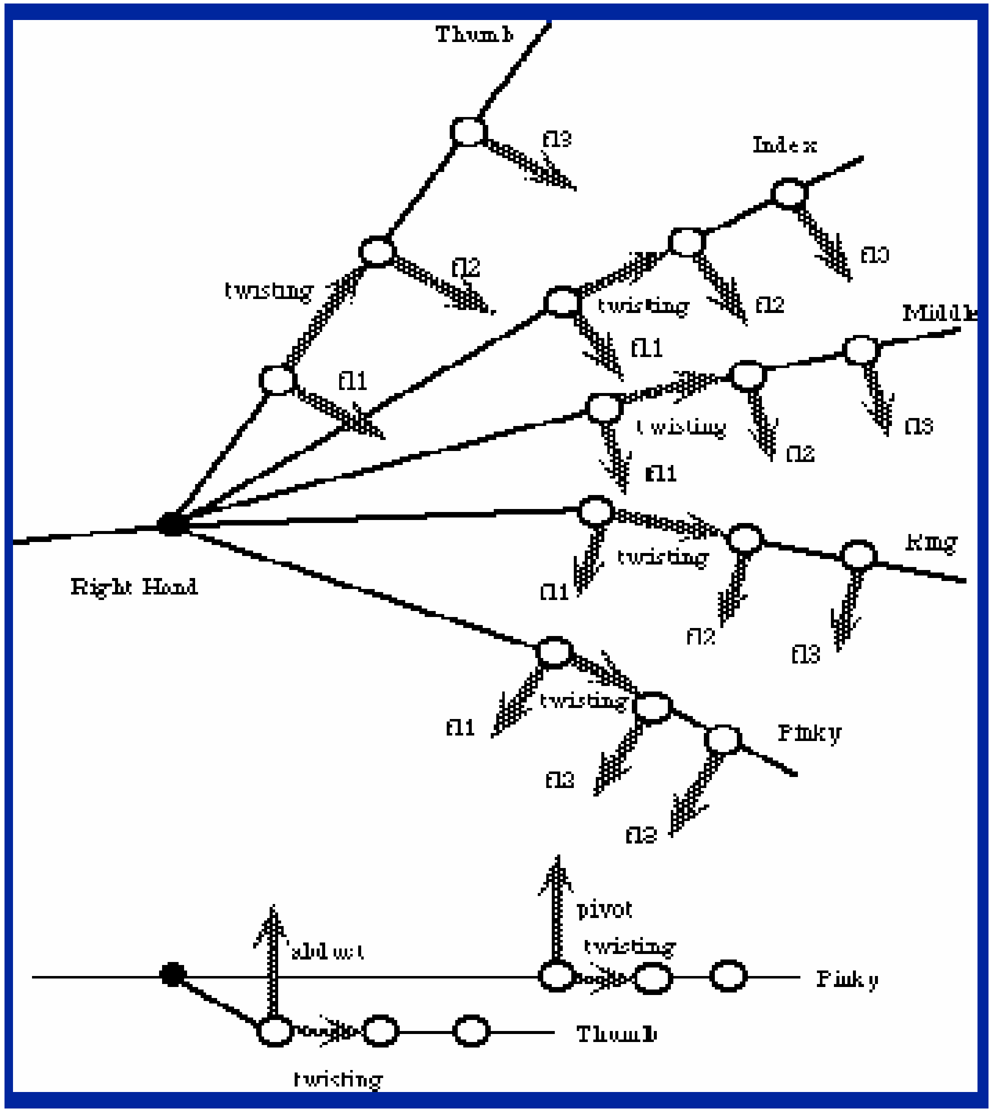
Body animation - lower body



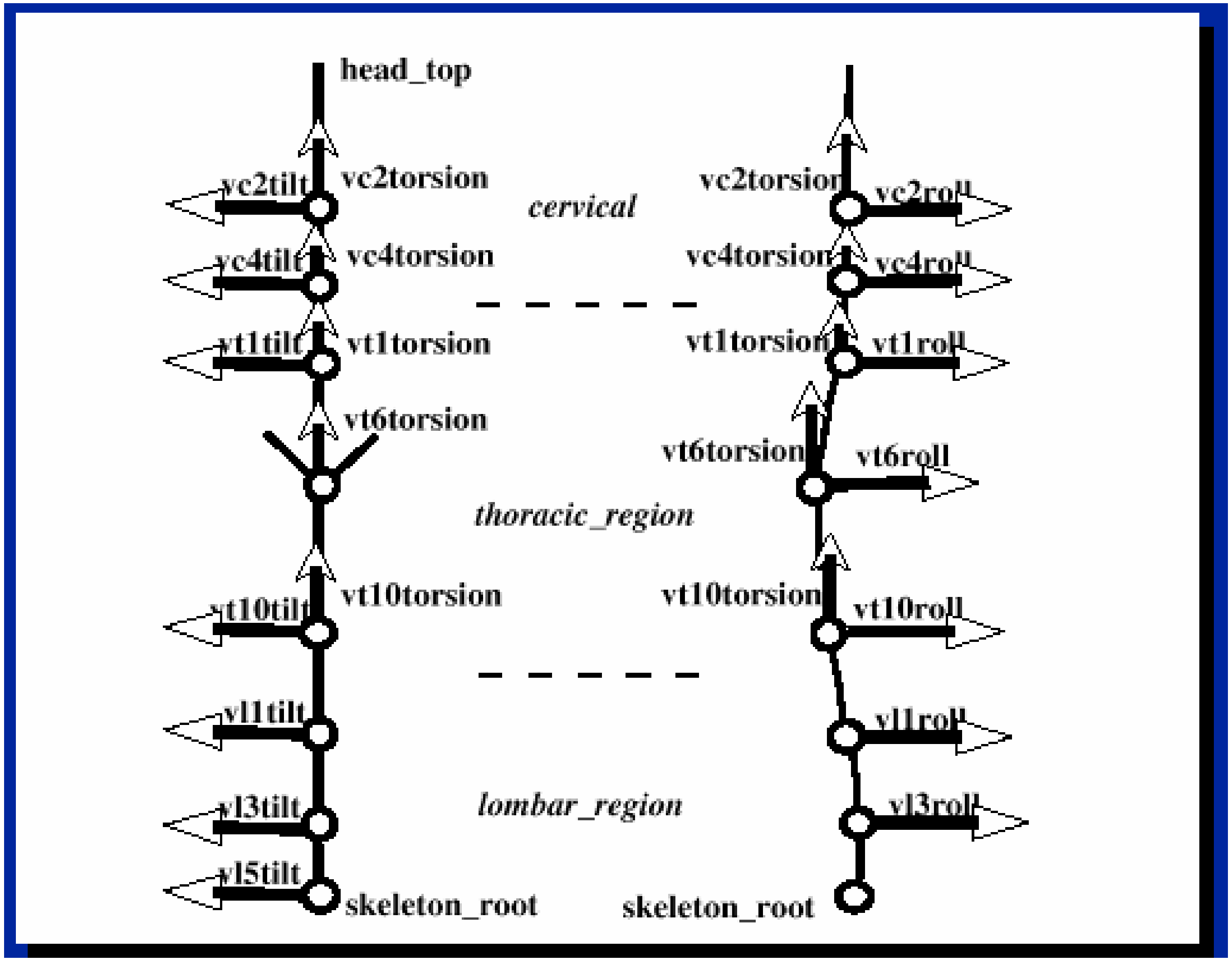
Body animation - arms



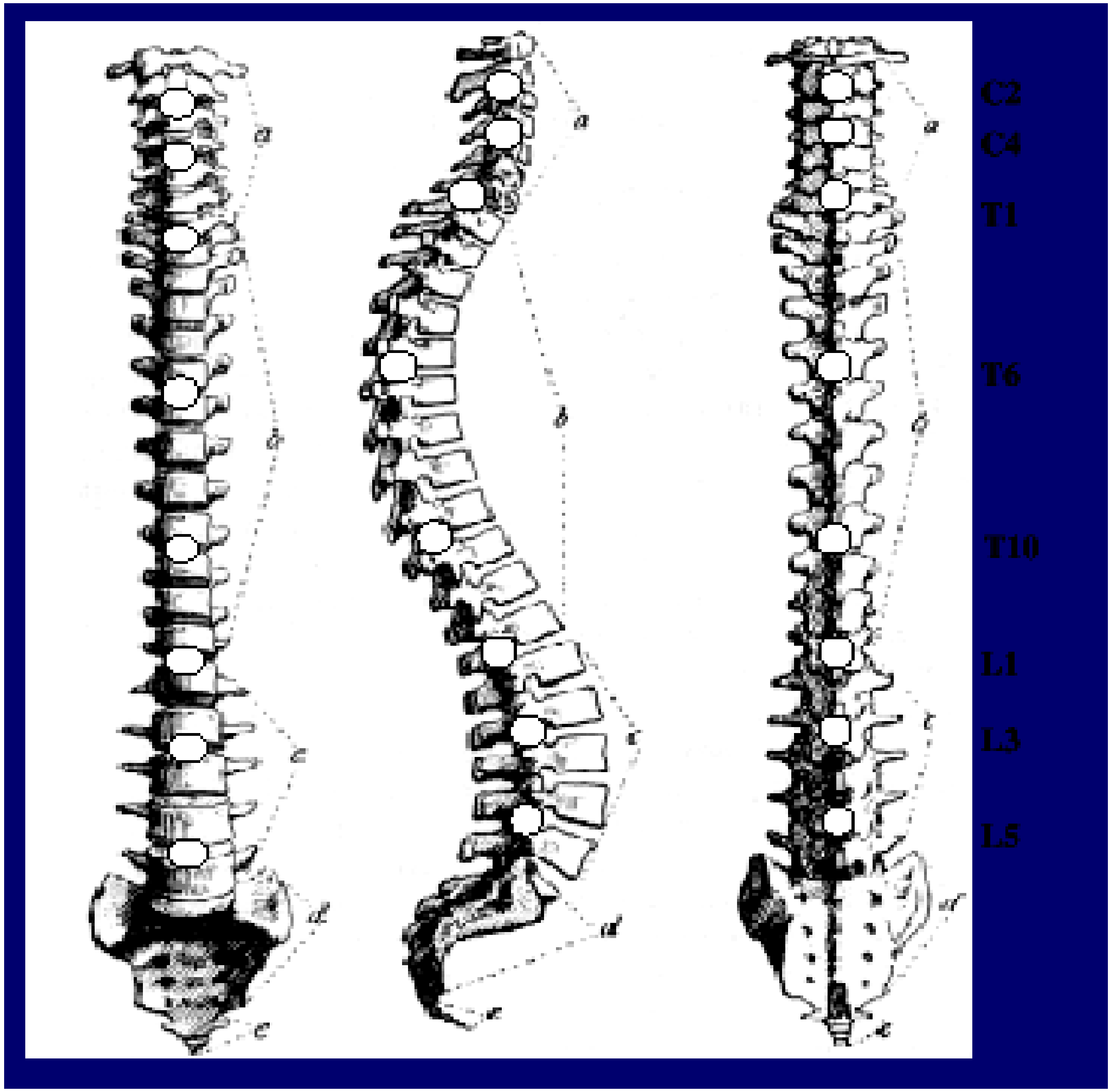
Body Animation: hands



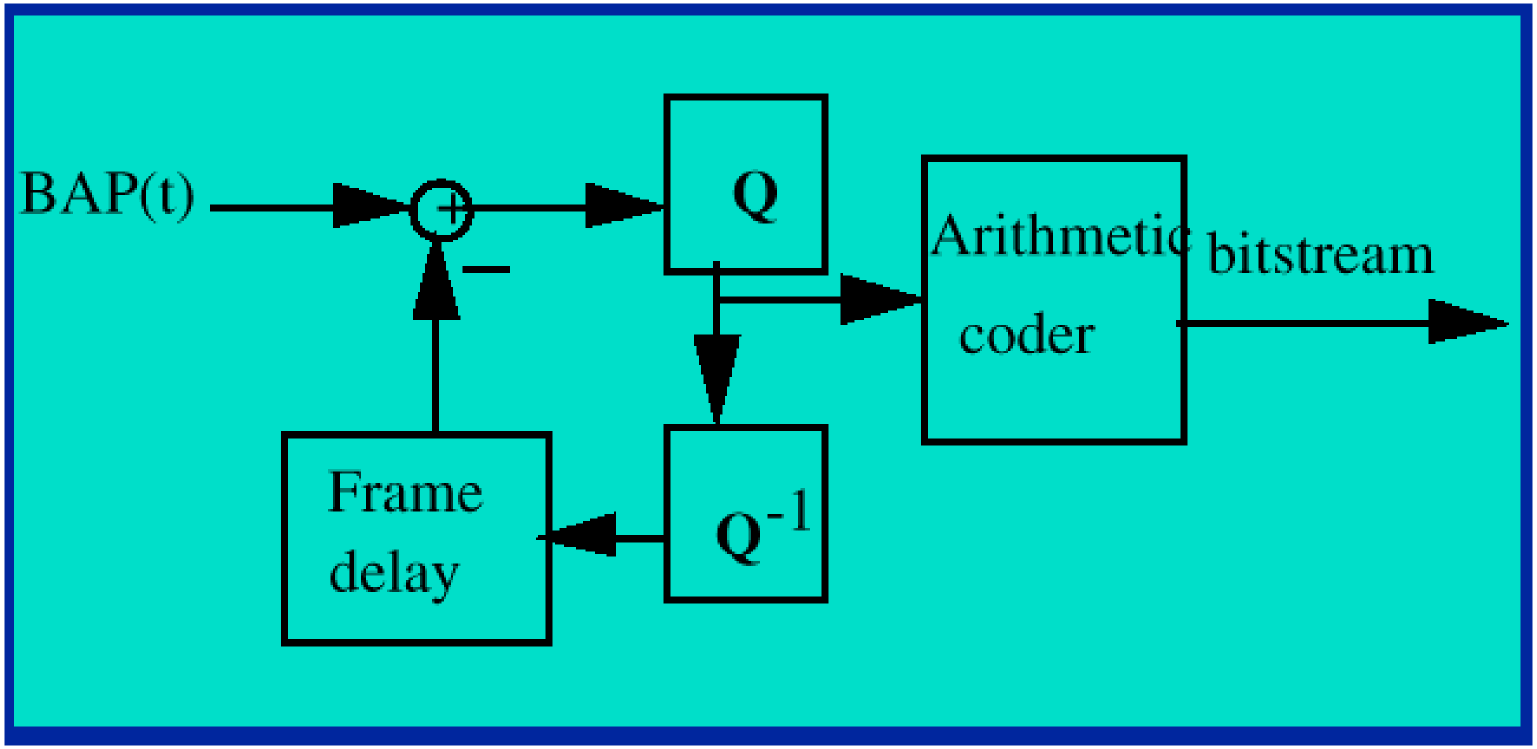
Body Animation - Upper Body



Spine Animatin

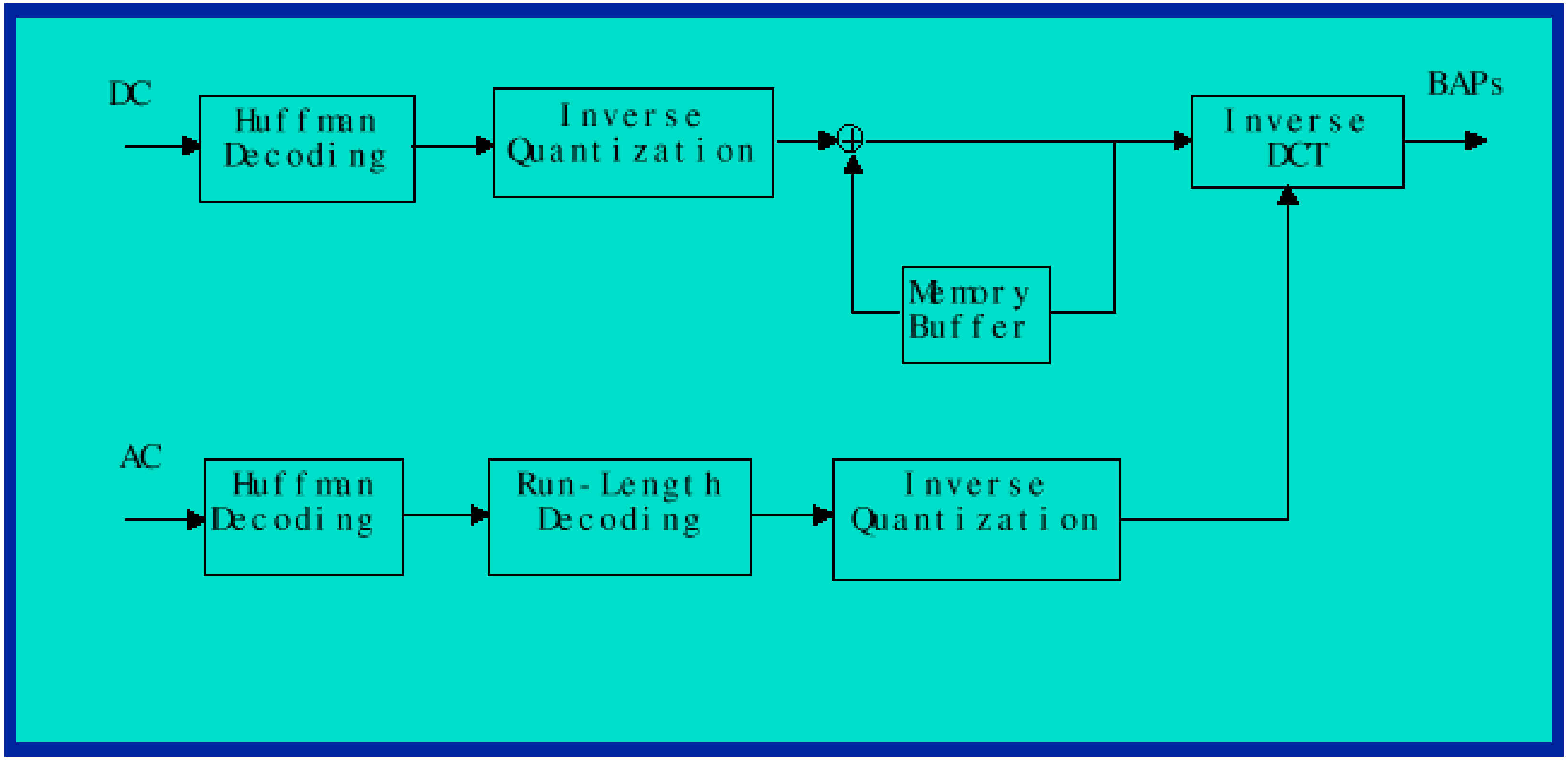


BAP Coding



- Quantization, and predictive coding using arithmetic coding
- DCT coding on 16 frames BAPs

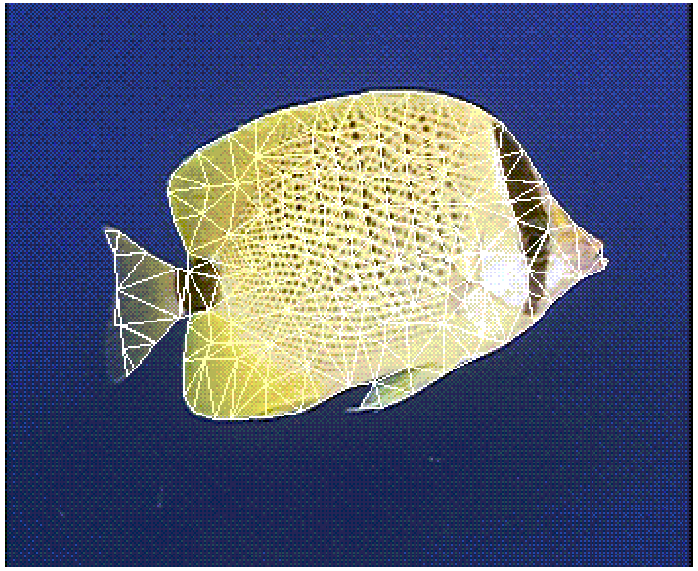
BAP Decoding



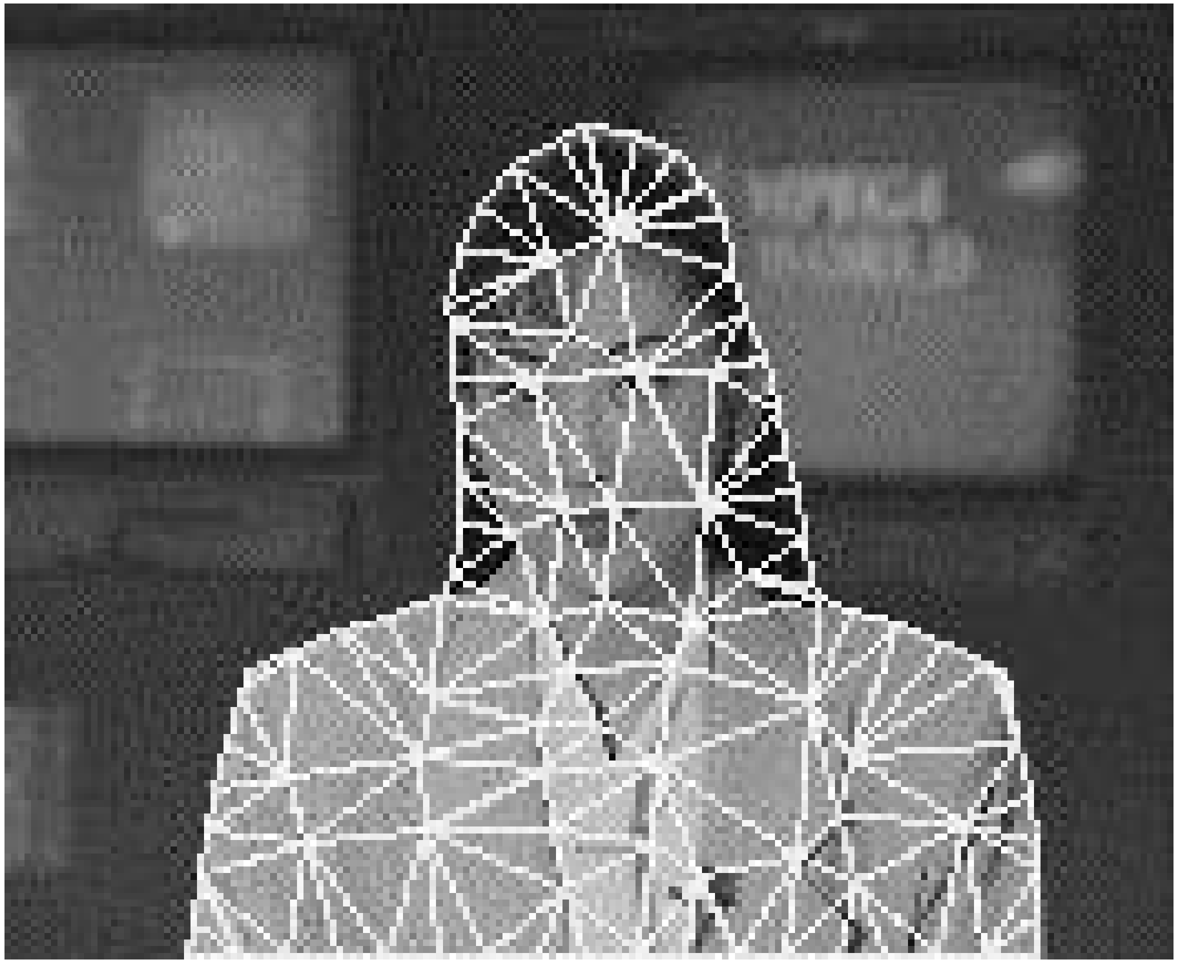
Animated Meshes

- A 2D mesh is a tessellation (Mosaic like) of a 2D planar region into polygonal patches
- MPEG4 considers only triangular meshes where the patches are triangles
- A 2D dynamic mesh refers to 2D mesh geometry and motion

information of all mesh node points within a temporal segment of interest.



2D Mesh Modeling (“akiyo” VO)



Animated Meshes (Cont'd)

- 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
- the affine mapping is a common choice

Video Object Manipulation

- Augmented reality: Merging virtual (computer generated) images with real moving images (video) to create enhanced display information
- Synthetic-object-transfiguration/animation: Replacing a natural video object in a video clip by another video object
- Spatio-temporal interpolation: Mesh motion modeling provides more robust motion-compensated temporal interpolation

Fishy ?

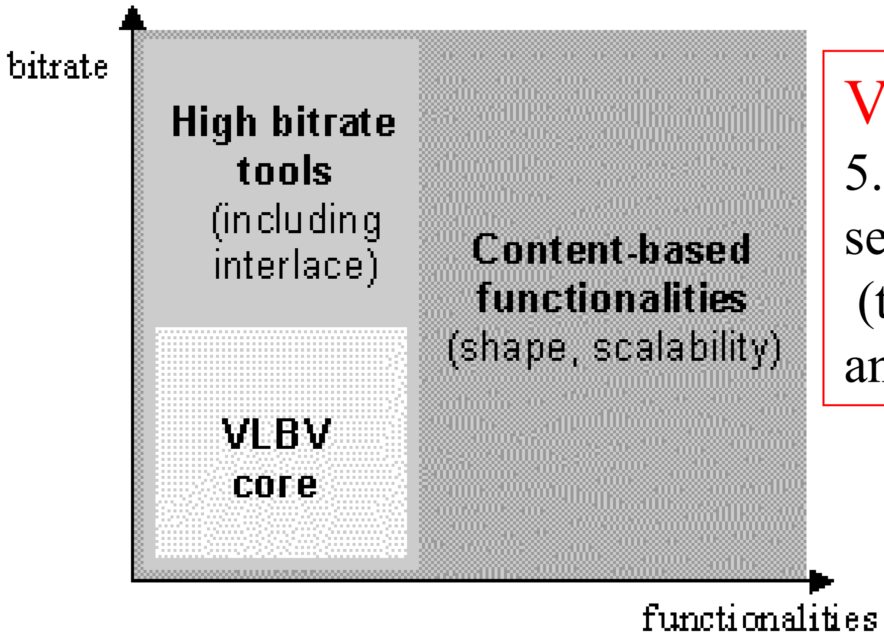
Video Object Compression

- 2D mesh modeling may be used for compression if one chooses to transmit texture maps only at selected key frames and animate these texture maps (without sending any prediction error image) for the intermediate frames
- Called: self-transfiguration

Content-Based Video Indexing

- Mesh representation enables animated key snapshots for a moving visual synopsis of objects.
- Mesh representation provides accurate object trajectory information that can be used to retrieve visual objects with specific motion.
- Mesh representation provides vertex-based object shape representation which is more efficient than the bitmap representation for shape-based object retrieval.

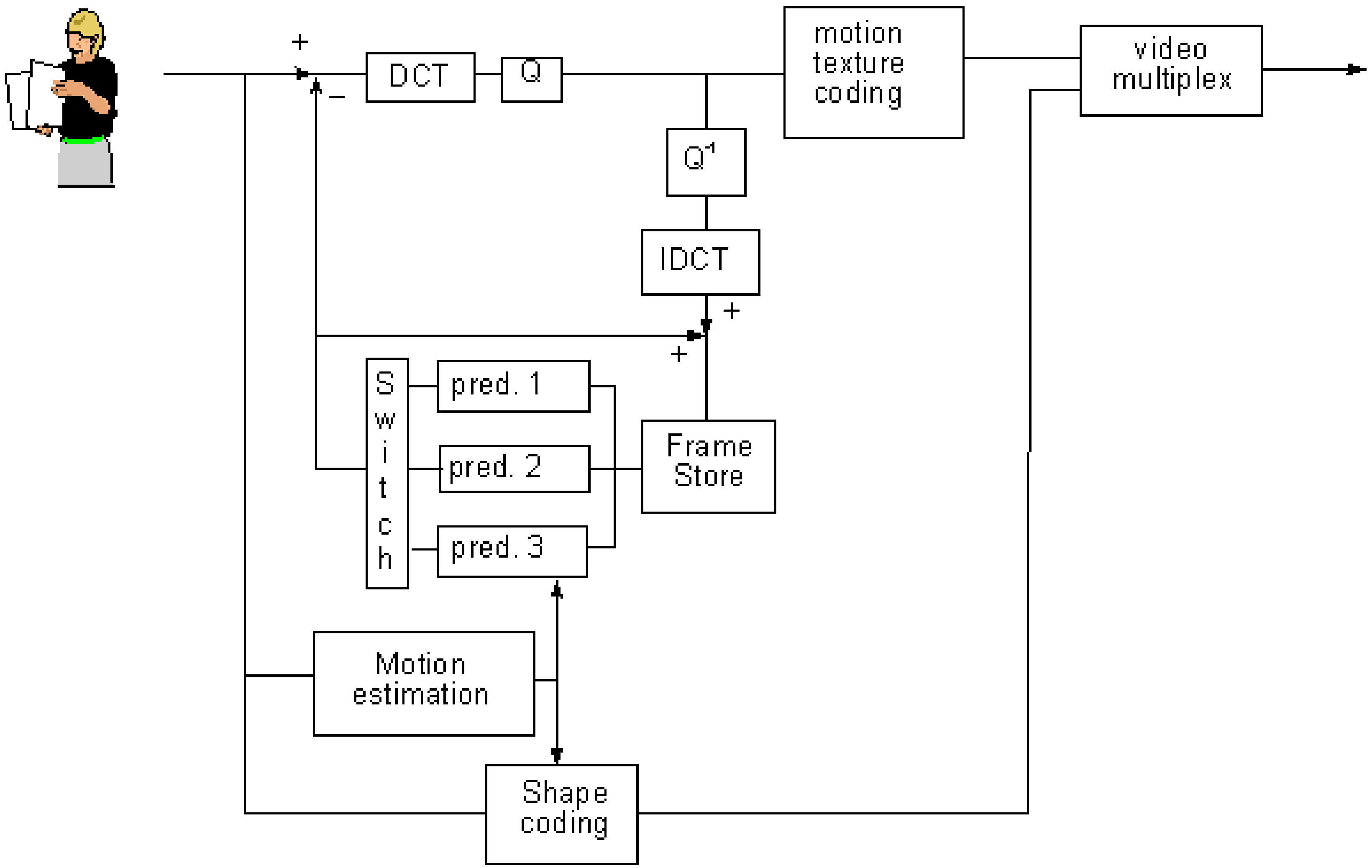
Representing Natural Video


bitrate
High bitrate tools (including interlace)
VLBV core
Content-based functionalities (shape, scalability)
functionalities

VLBV: Very Low Bit-rate Video:
5...64 kbits/s, supporting image sequences with low spatial resolution (typically up to CIF resolution) and low frame rates (typically upto 15 Hz).

Content-based functionalities support the separate encoding and decoding of content (i.e. physical objects in a scene, VOs). This MPEG-4 feature provides the most elementary mechanism for interactivity; flexible representation and manipulation with/of VO content of images or video in the compressed domain, without the need for further segmentation or transcoding at the receiver.

MPEG-4 Video & Image Coding Scheme



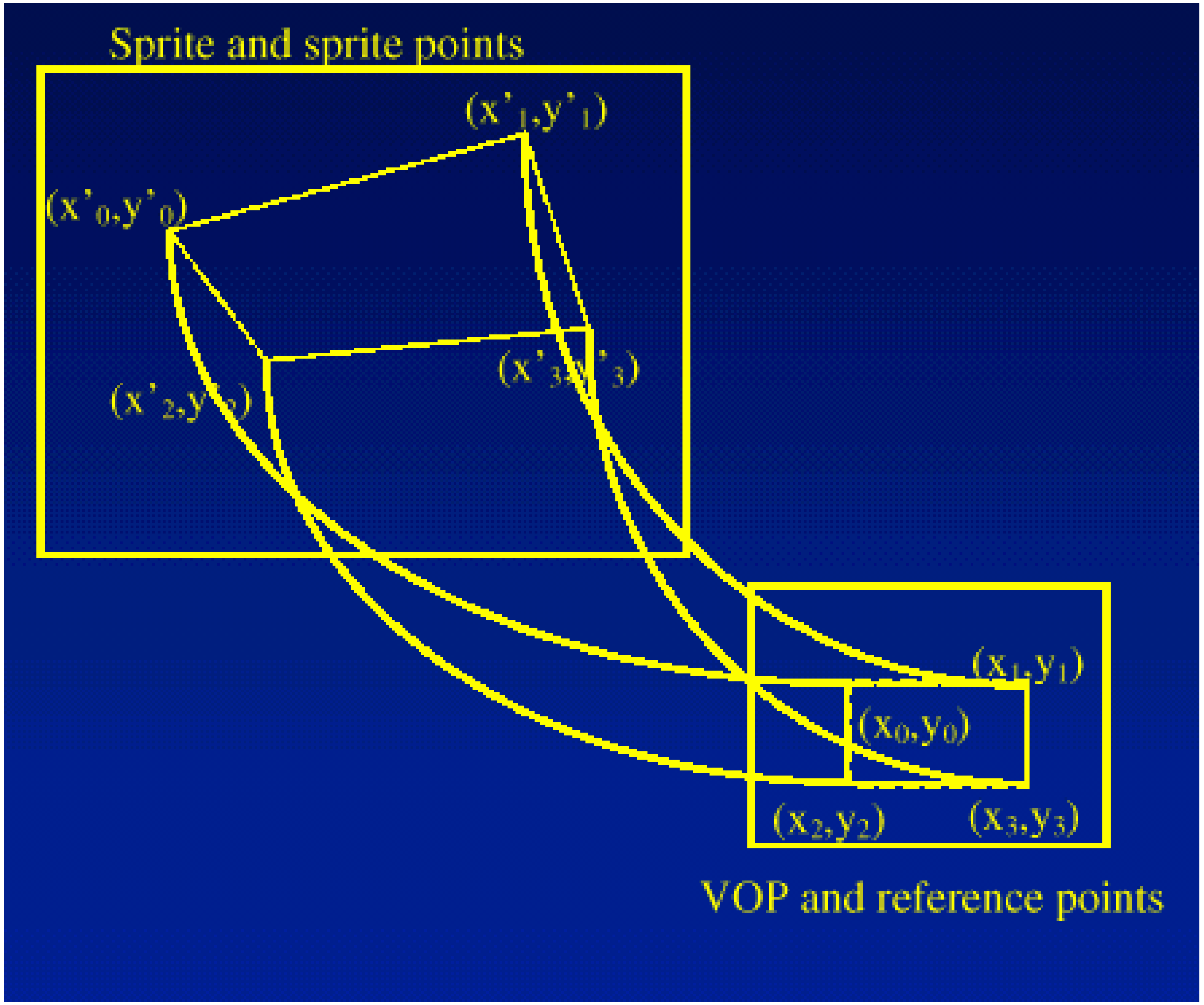
Motion Prediction Techniques

- Standard 8x8 or 16x16 pixel block-based motion estimation and compensation.
- Global motion compensation based on the transmission of a static "sprite". A static sprite is a possibly large still image, describing panoramic background. For each consecutive image in a sequence, only 8 global motion parameters describing camera motion are coded to reconstruct the object.
- These parameters represent the appropriate Affine transform of the sprite transmitted in the first frame.

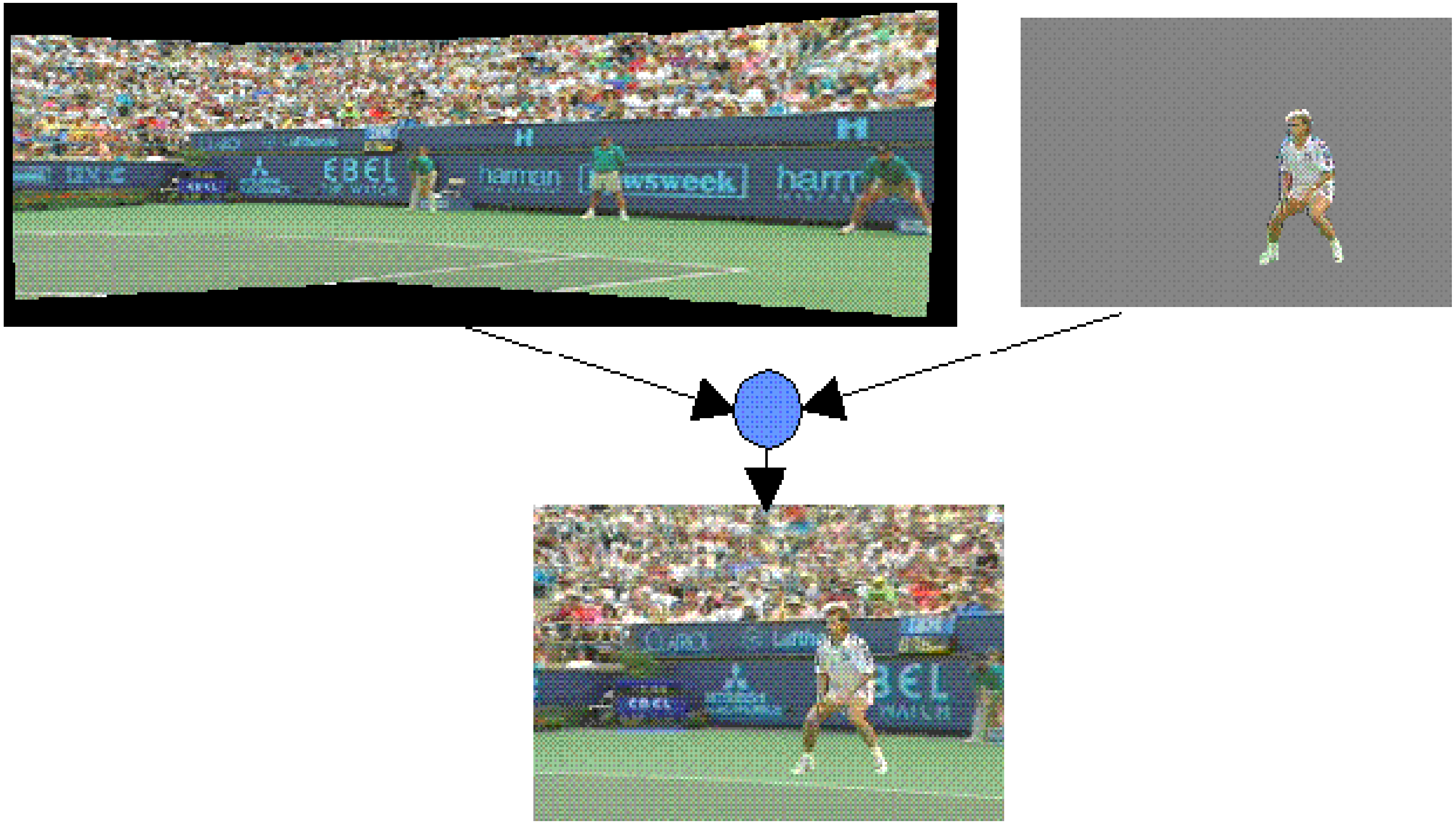
Sprite Panorama Concept

- Assumed that the foreground object (tennis player) can be segmented from the background and that the sprite panorama image can be extracted from the sequence prior to coding.
- The large panorama is transmitted to the receiver only once as first frame of the sequence to describe the background, and is stored in a "sprite buffer".

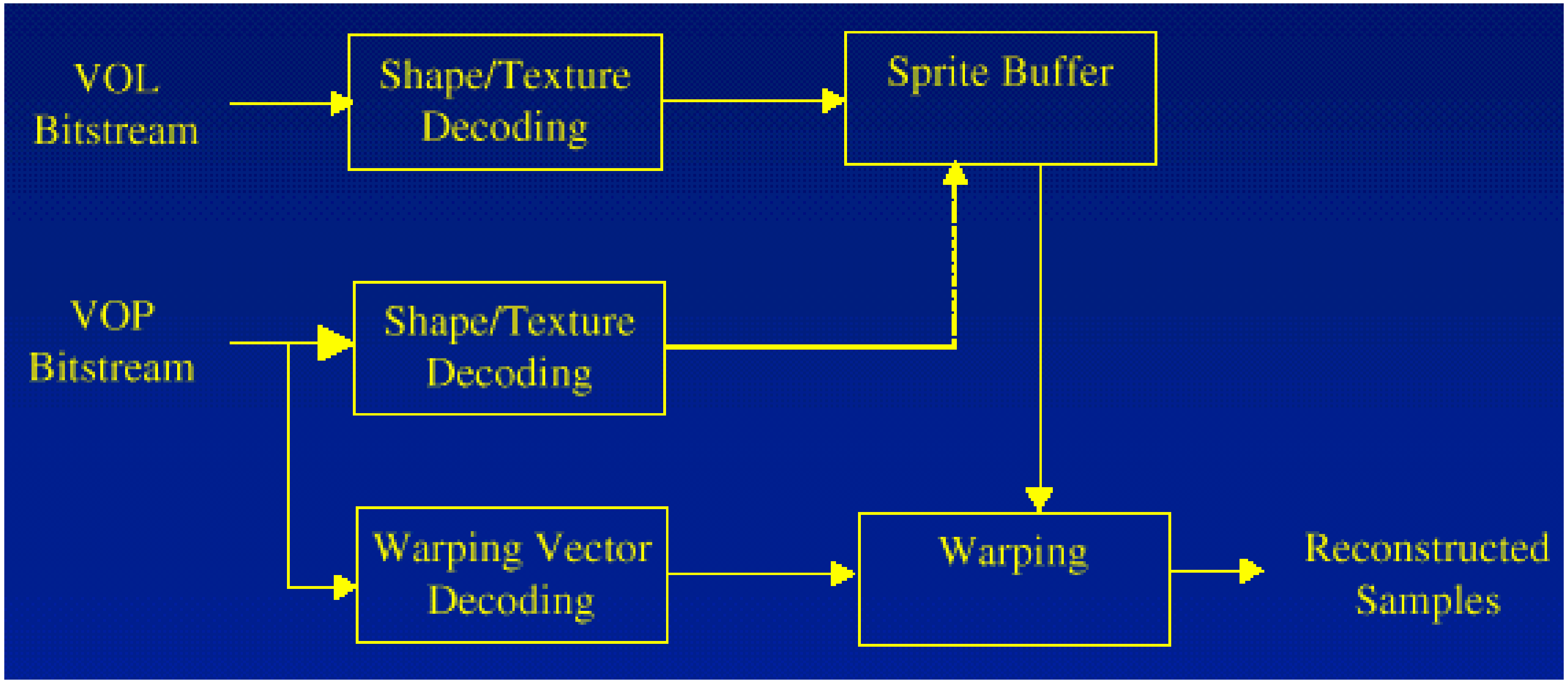
Static Sprite Coding



“Sprite” Coding Example



Sprite Coding: Block Diagram



- Basic sprite coding
- Low latency sprite coding
- Scalable sprite coding

Coding of Textures and Still Images

- Efficient Coding of visual textures and still images (e.g. animated meshes) is supported by the visual texture mode.
- This mode is based on a zero-tree wavelet algorithm that provides very high coding efficiency over a very wide range of bitrates.
- It also provides spatial and quality scalabilities (up to 11 levels of spatial scalability & continuous quality scalability and also Arbitrary-shaped object coding

Scalable Coding of Video Objects

Scalability refers to the ability to only decode a part of a bitstream and reconstruct images or image sequences with:

- reduced decoder complexity and thus reduced quality.
- reduced spatial resolution
- reduced temporal resolution
- equal temporal and spatial resolution but with reduced quality.

Robustness in Error Prone Environments

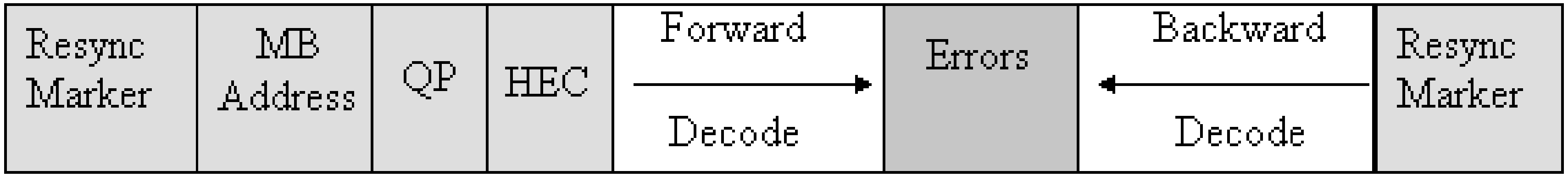
- Due to the rapid growth of mobile communications, it is extremely important that access is available to audio and video information via wireless networks.
- The error resilience tools developed for MPEG-4 can be divided into three major areas:
- resynchronization
- data recovery
- error concealment

Resynchronization

- Enables resynchronization between the decoder and the bitstream after a residual error have been detected.
- Generally, the data between the synchronization point prior to the error and the first point where synchronization is reestablished, is discarded.
- If the resynchronization approach is effective at localizing the amount of data discarded by the decoder, then the ability of other types of tools that recover or conceal data is greatly enhanced.
- The general approach is adopted from H.261/3, and based on GOB header (start code).

Data Recovery

- Attempt to recover data that in general would be lost.
- These tools are not simply error correcting codes, but instead techniques that encode the data in an error resilient manner.
- E.g. ,one particular tool is Reversible Variable Length Codes: the variable length codewords are designed such that they can be read both in the forward as well as the reverse direction:



Error Concealment

- Data partitioning by separating the motion and the texture.
- This approach requires that a second resync' marker be inserted between motion and texture information.
- If the texture information is lost, this approach utilizes the motion information to conceal these errors.
- That is, the texture information is discarded, while the motion is used to motion compensate the previous decoded VOP.

MPEG-4 Demo

- http://wwwam.hhi.de/~kow/obj_video/Test.html