Fundamentals of Multimedia

MPEG Video Coding II

Lecturer：Jun Xiao
(肖俊）
College of Software and Technology

Content

1. MPEG-4

1.1 Overview

- **MPEG-4**: a newer standard. Besides compression, pays great attention to issues about user interactivities.
- MPEG-4 departs from its predecessors in adopting a new **object-based coding**:
  - Offering higher compression ratio, also beneficial for digital video composition, manipulation, indexing, and retrieval.
  - Arbitrary Shape Coding
  - Static texture coding
  - Face object coding and Animation
  - Body object coding and Animation
- The bit-rate for MPEG-4 video now covers a large range between 5 kbps to 10 Mbps.

1.1 Overview

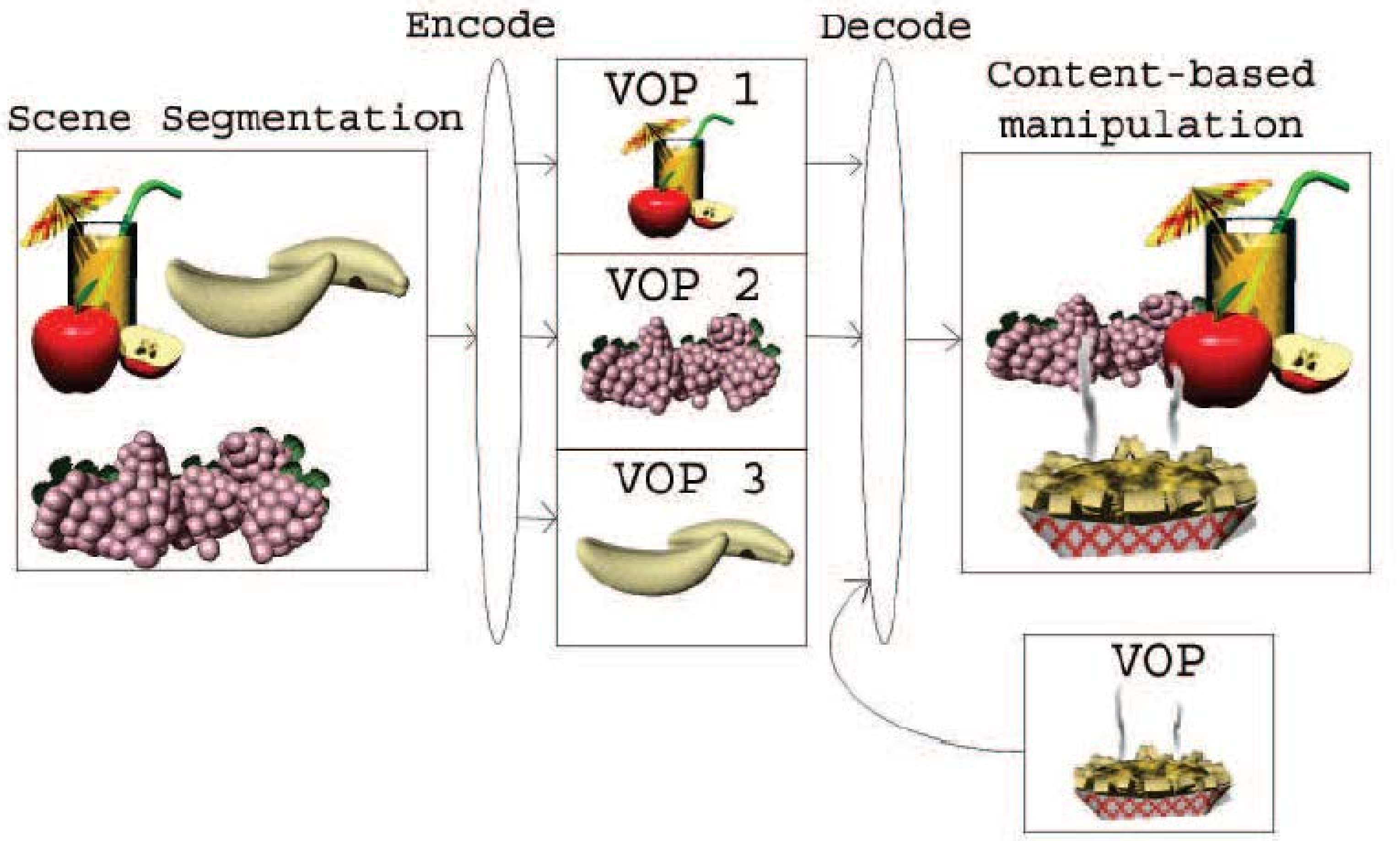
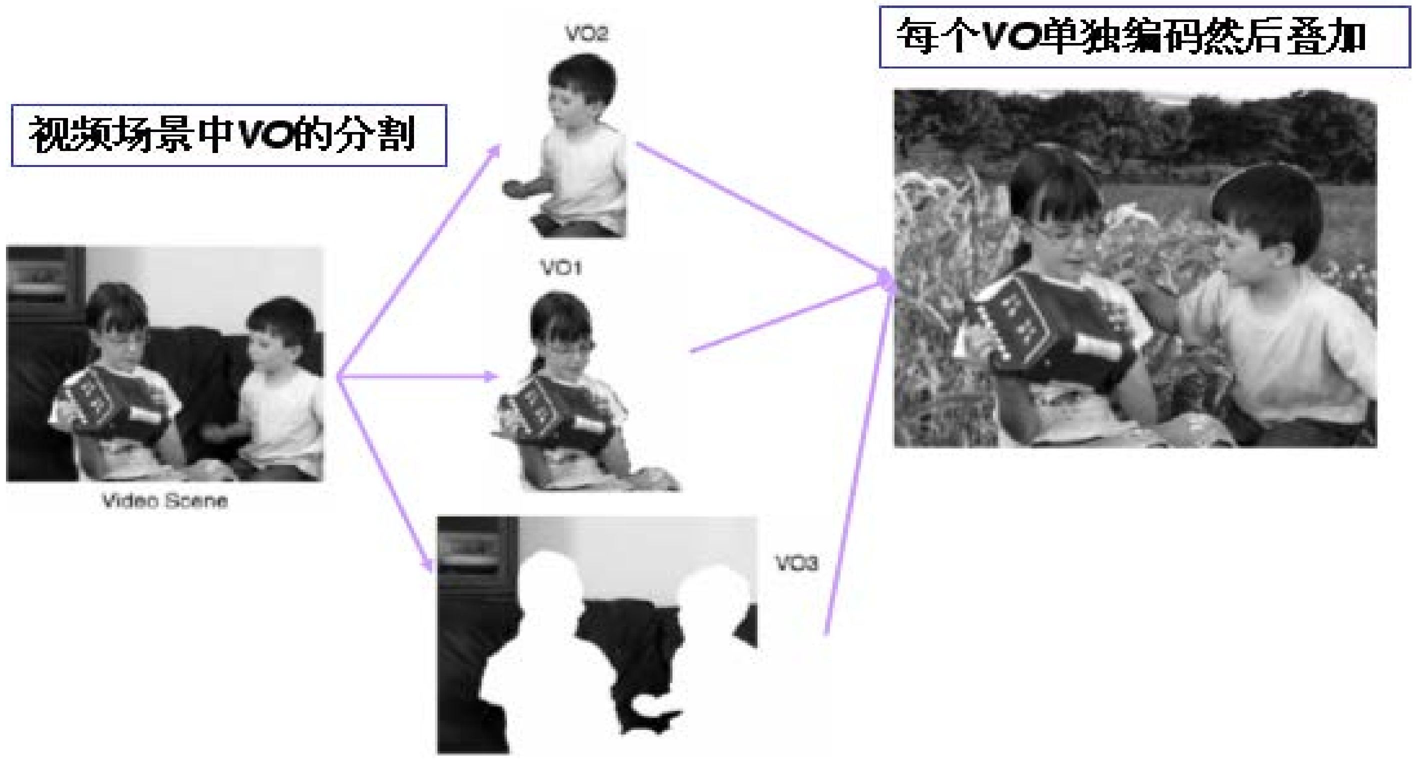


Fig. 12.1: Composition and Manipulation of MPEG-4 Videos.

1.1 Overview



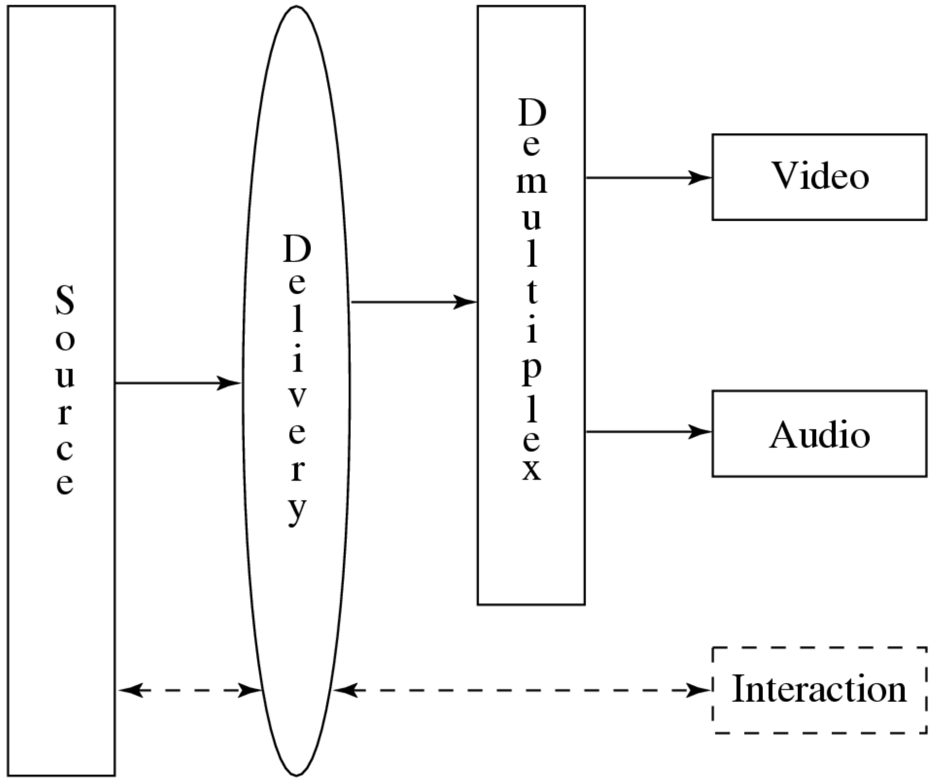
MPEG-4 object based processing

1.1 Overview

- MPEG-4 (Fig. 12.2(b)) 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.

1.1 Overview

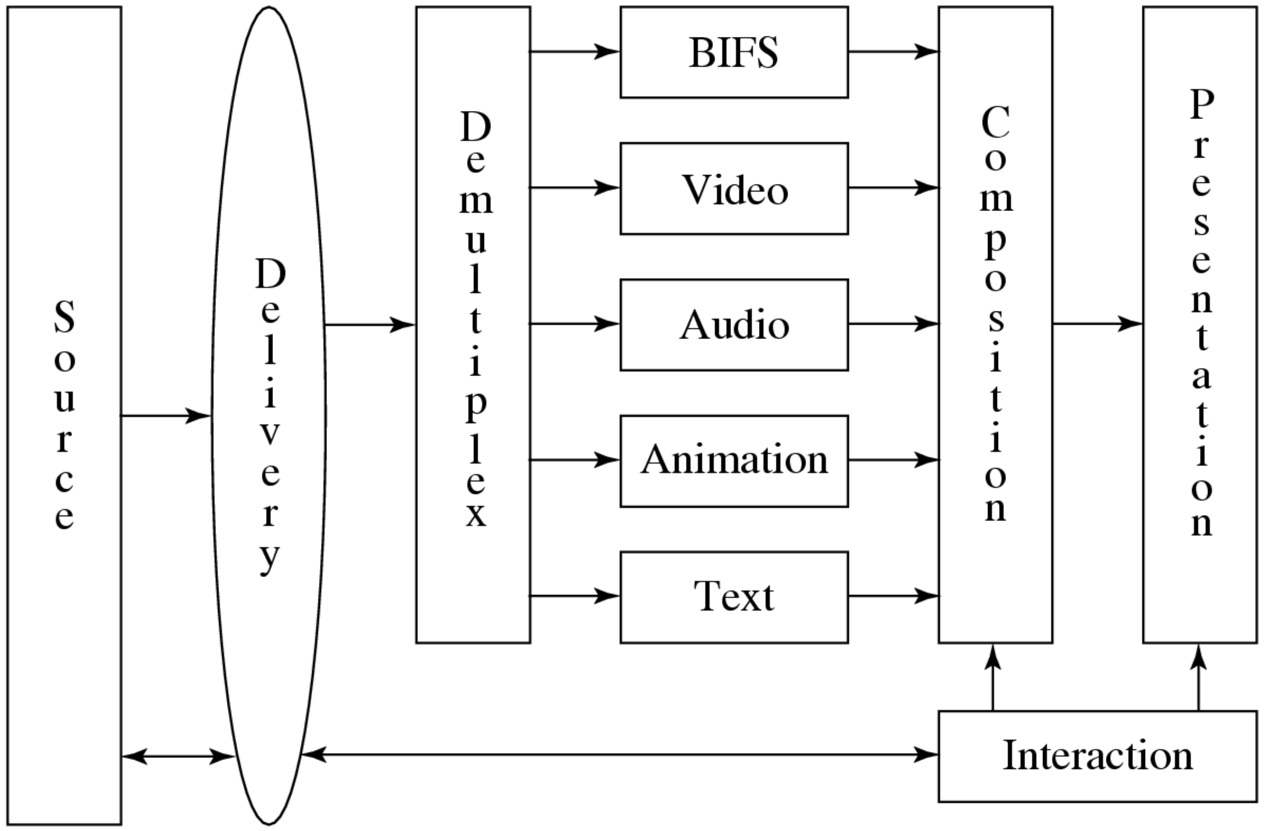
- Comparison of interactivities in MPEG standards.
  - Reference models in MPEG-1 and 2



Interaction in dashed lines supported only by MPEG-2

1.1 Overview

- Comparison of interactivities in MPEG standards.
  - MPEG-4 reference model



1.1 Overview

- The hierarchical structure of MPEG-4 visual bitstreams is very different from that of MPEG-1 and -2, it is very much video object-oriented.

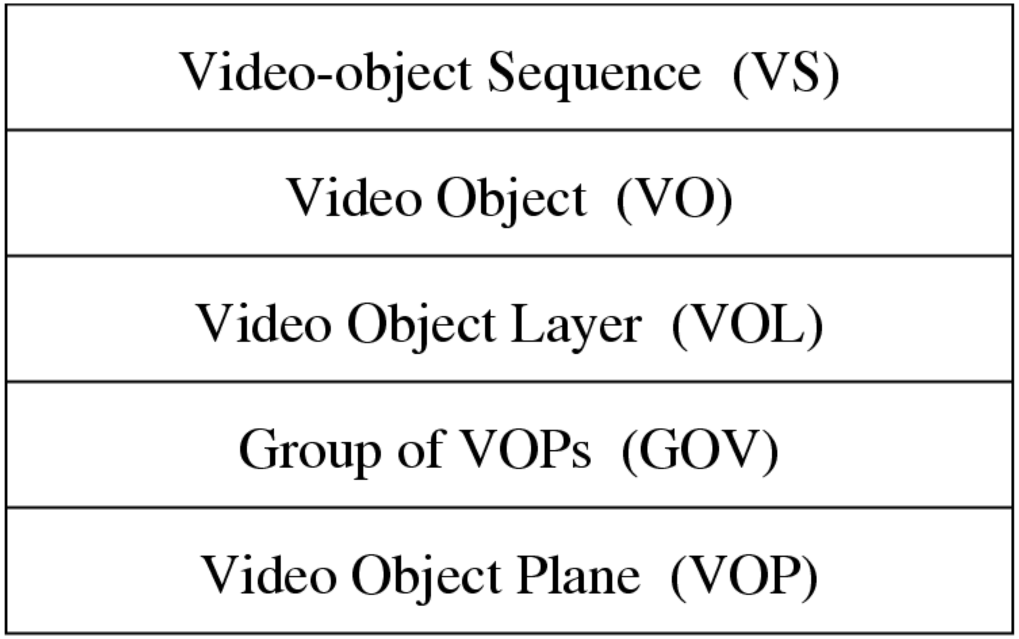


Fig. 12.3: Video Object Oriented Hierarchical Description of a Scene in MPEG-4 Visual Bitstreams.

1.1 Overview

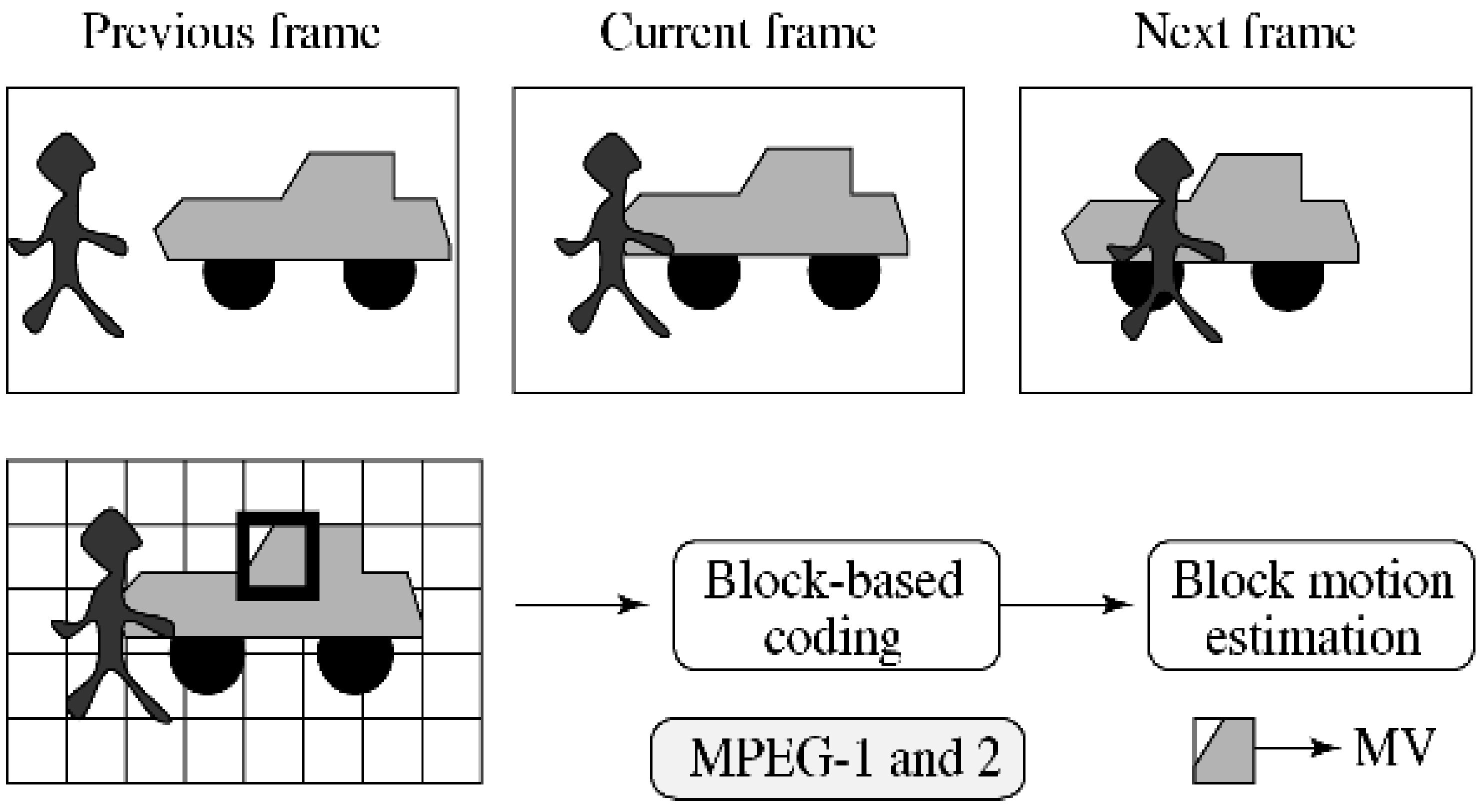
1. **Video-object Sequence** (VS)—delivers the complete MPEG-4 visual scene, which may contain 2-D or 3-D natural or synthetic objects.
2. **Video Object** (VO) — a particular object in the scene, which can be of arbitrary (non-rectangular) shape corresponding to an object or background of the scene.
3. **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.
4. **Group of Video Object Planes** (GOV) — groups Video Object Planes together (optional level).
5. **Video Object Plane** (VOP) — a snapshot of a VO at a particular moment.

1.2 Object-based visual coding

- VOP-Based Coding vs. Frame-Based Coding
- Motion Compensation
- Texture Coding
- Shape Coding
- Static Texture Coding
- Sprite Coding
- Global Motion Compensation

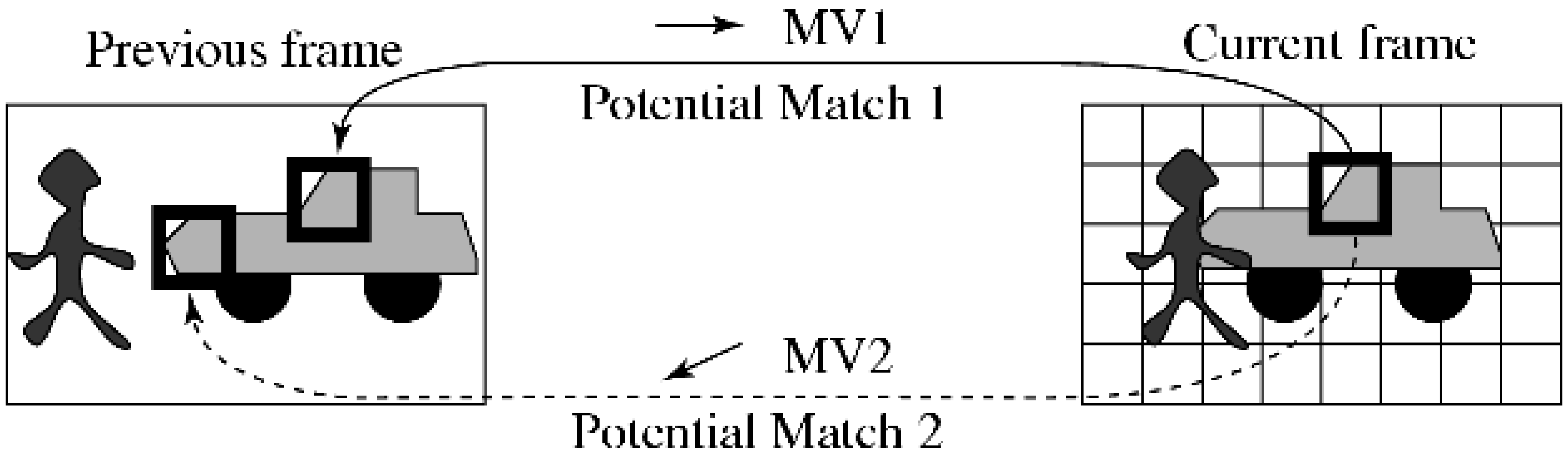
1.2.1 VOP-Based Coding vs. Frame-Based Coding

- 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**).



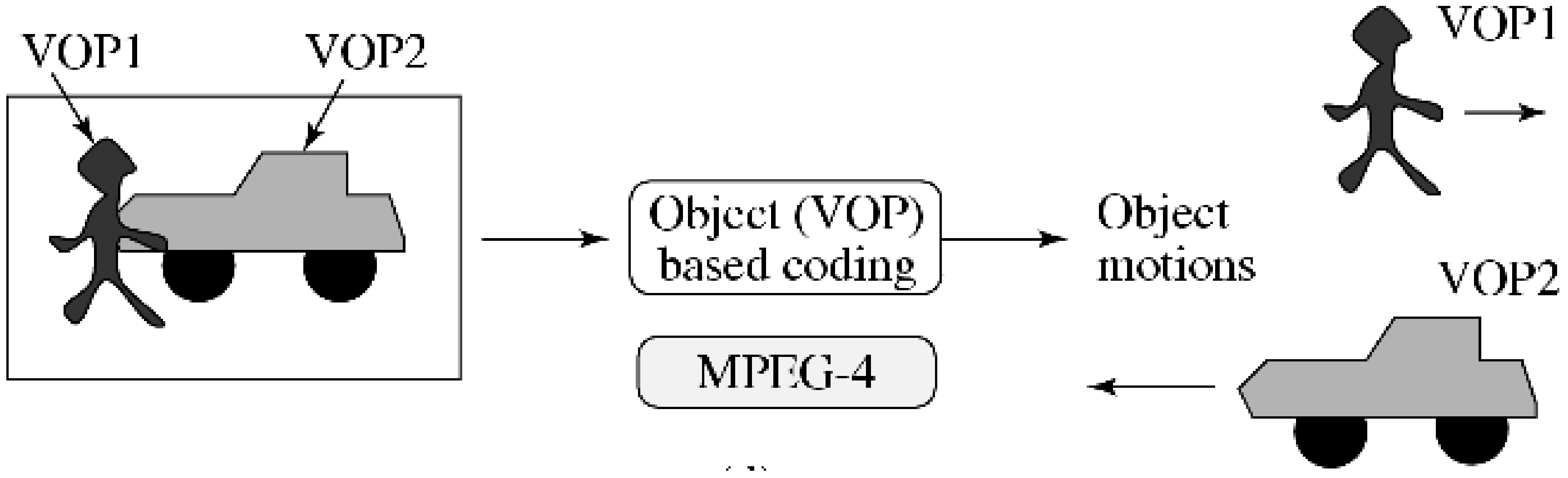
1.2.1 VOP-Based Coding vs. Frame-Based Coding

Fig. 12.4 (c) illustrates a possible example in which both potential matches yield small prediction errors for block-based coding.



1.2.1 VOP-Based Coding vs. Frame-Based Coding

Fig. 12.4 (d) shows that each VOP is of arbitrary shape and ideally 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-VOPs* if only forward prediction is employed, or *B-VOPs* 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.

1.2.2 Motion Compensation

- 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 16×16 in luminance images and 8×8 in chrominance images.

1.2.2 Motion Compensation

- MPEG-4 defines a rectangular *bounding box* for each VOP (see Fig. 12.5 for details).
- The macroblocks that are entirely within the VOP are referred to as **Interior Macroblocks**.

  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.

1.2.2 Motion Compensation

- Motion compensation for interior macroblocks is carried out in the same manner as in MPEG-1 and 2

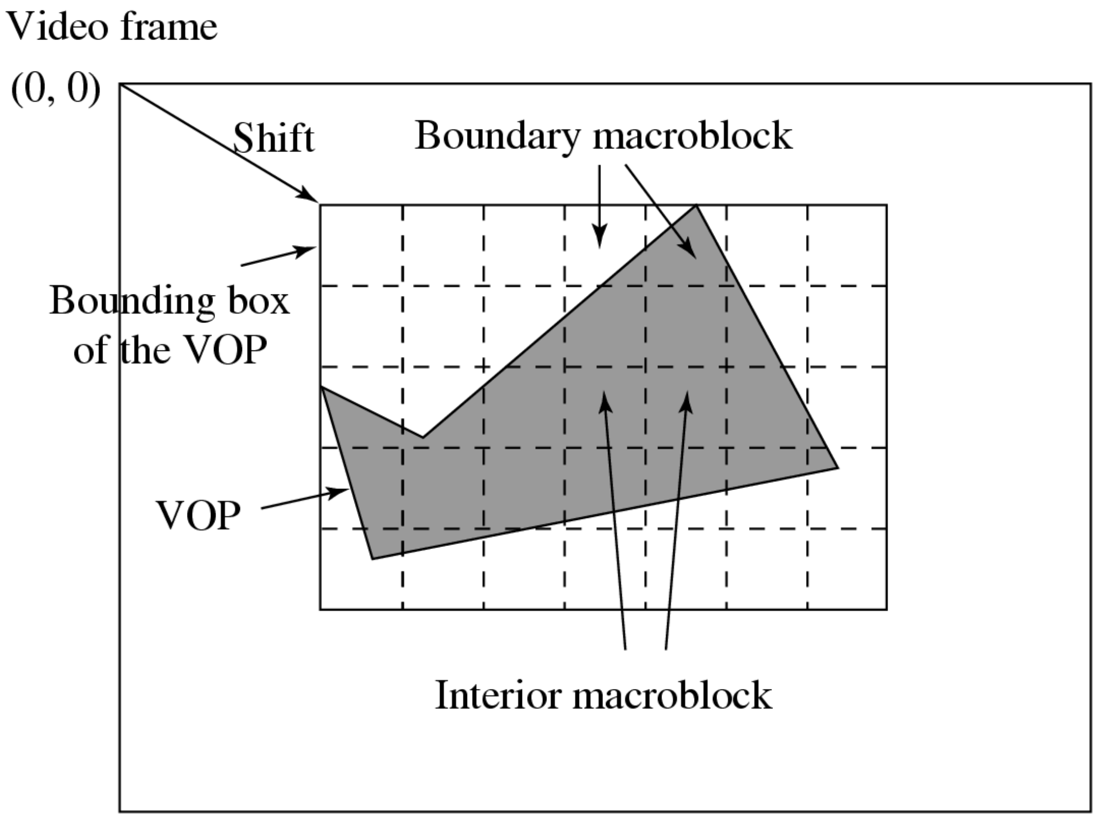
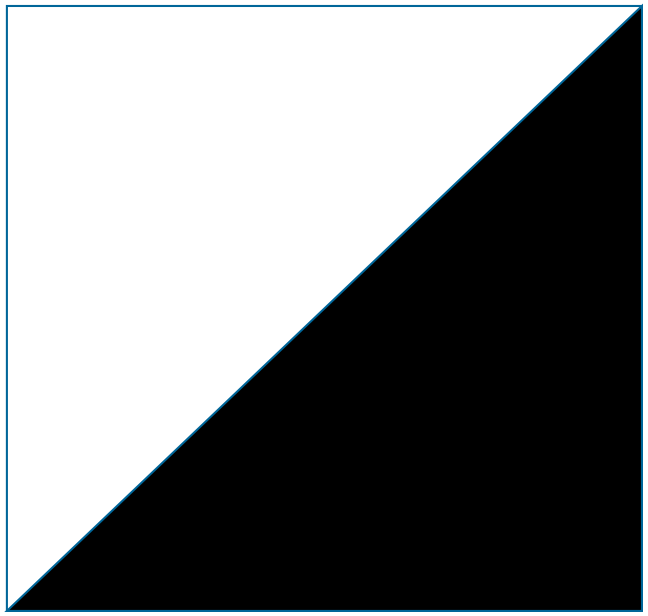
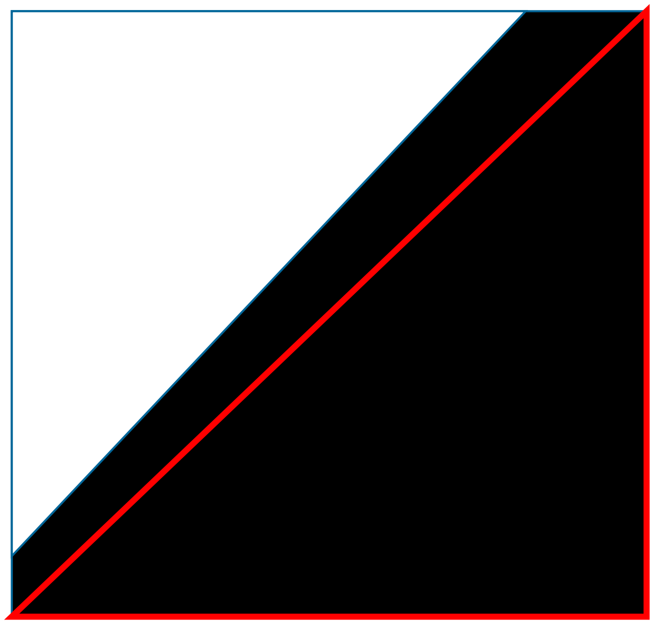
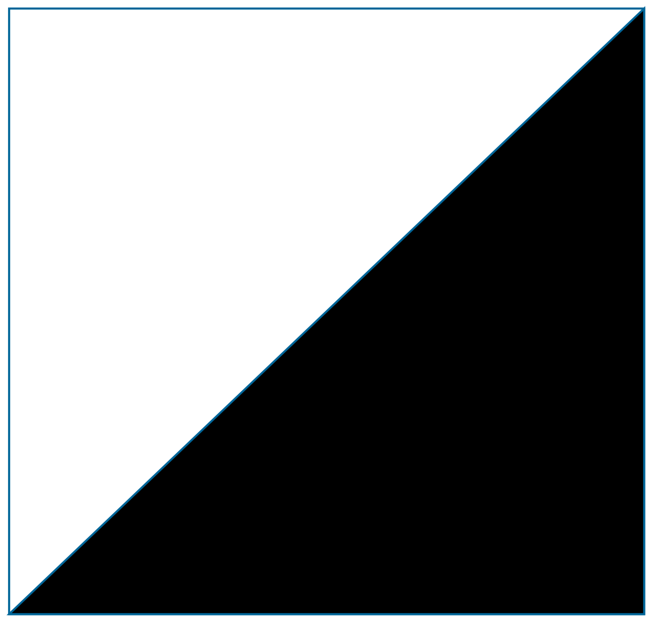


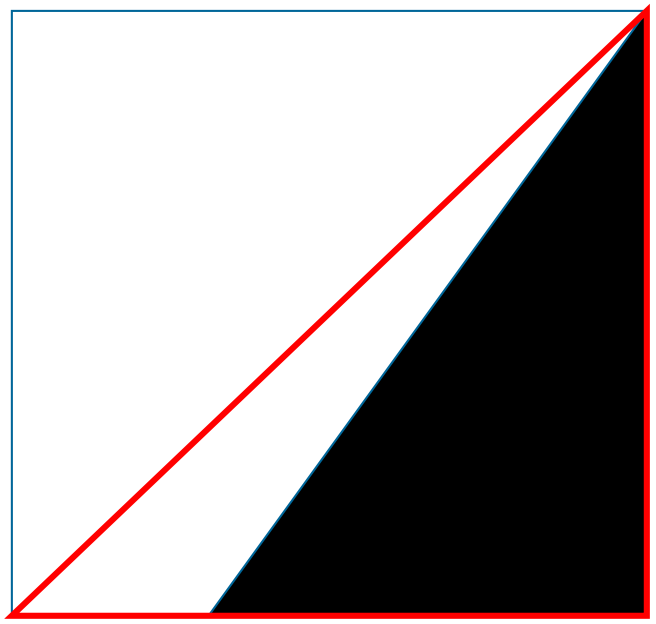
Fig. 12.5: Bounding Box and Boundary Macroblocks of VOP.


Macroblock in Target Frame


Macroblock in Reference Frame



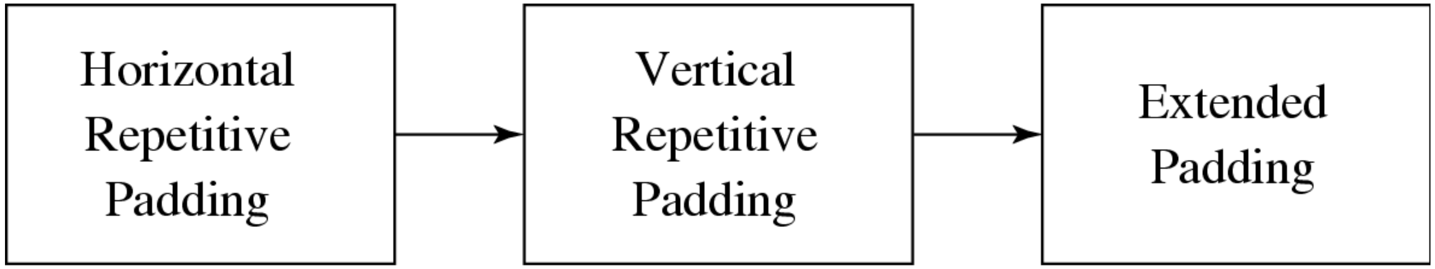
Macroblock in Target Frame



Macroblock in Reference Frame

1.2.2 Motion Compensation

- Padding is applied prior to motion compensation for boundary macroblocks.



- The horizontal padding examines each row and every boundary pixel is replicated to the left and/or right to fill the values out side the VOP.
- If the interval is bounded by two boundary pixel, their average is adopted ;
- The vertical padding works similarly.

1.2.2 Motion Compensation

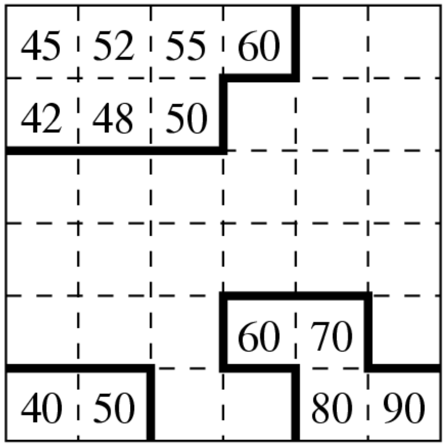
Algorithm 12.1 Horizontal Repetitive Padding:

```
begin
    for all rows in Boundary MBs in the Reference VOP
        if ∃ (boundary pixel) in the row
        for all $interval$ outside of VOP
        if $interval$ is bounded by only one boundary pixel $b$
        assign the value of $b$ to all pixels in $interval$
        else // $interval$ is bounded by two boundary pixels $b_1$ and $b_2$
        assign the value of $(b_1 + b_2)/2$ to all pixels in $interval$
end
```

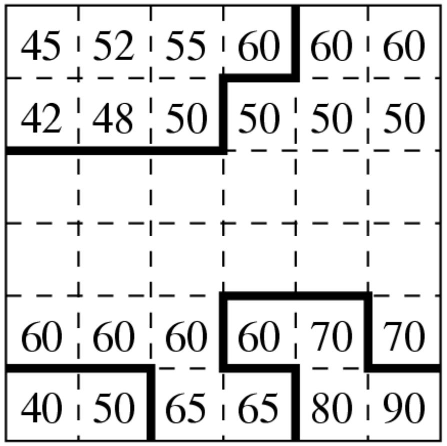
- The subsequent Vertical Repetitive Padding algorithm works in a similar manner.

1.2.2 Motion Compensation

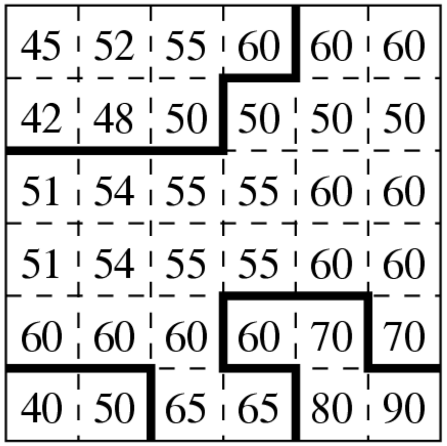
- **Padding (an example)**



(a)



(b)



(c)

- Extended Padding: exterior macroblocks immediately next to boundary macroblocks are filled by replicating the values of the border pixels of the boundary macroblock.
- The macroblocks to use follows a priority list: left, top, right, bottom

1.2.2 Motion Compensation

- Motion Vector Coding
  - Motion estimation

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

$Map(p,q)=1$ if $C(p,q)$ is pixel in VOP

else $Map(p,q)=0$

motion vector $MV:(u,v)=\{(i,j)\,|\,SAD(i,j)$ is minimum

$i,\ j\in[-p,p]\}$, p is maximum of u and v

  - Allows quarter-pixel precision in the luminance components.
  - MV can point beyond the boundaries of the reference VOP, pixel outside the VOP is defined in padding step.

1.2.3 Texture Coding

- I-VOP coded like JPEG
- For P-VOP and B-VOP
  - The prediction error is sent to DCT and VLC
- Texture coding based on DCT
  - For portions of the boundary macroblocks outside the VOP, zeros are padded
  - Quantization step_size for the DC component is 8
  - Two methods can be employed for the AC coefficients
    - H.263 method, all coefficients receive the same quantizer
    - MPEG-2 method, DCT coefficients in the same macroblock can have different quantizers.

1.2.3 Texture Coding

II. SA-DCT based coding for Boundary MBs

- Shape Adaptive DCT (SA-DCT) is another texture coding method for boundary MBs.
- SA-DCT is a 2D DCT and it is computed as a separable 2D transform in two iterations of 1D DCT-N.

$$1D \quad (DCT-N)$$

$$F(u)=\sqrt{\frac{2}{N}}C(u)\sum_{i=0}^{N-1}\cos\frac{(2i+1)u\pi}{2N}f(i)$$

$$1D \quad (IDCT-N)$$

$$\tilde{f}(i)=\sum_{u=0}^{N-1}\sqrt{\frac{2}{N}}C(u)\cos\frac{(2i+1)u\pi}{2N}F(u)$$

$$\text{where} \quad i=0,1,\cdots,N-1;u=0,1,\cdots,N-1$$

$$C(u)=\begin{cases}\frac{\sqrt{2}}{2} & if\ u=0\\ 1 & otherwise\end{cases}$$

1.2.3 Texture Coding

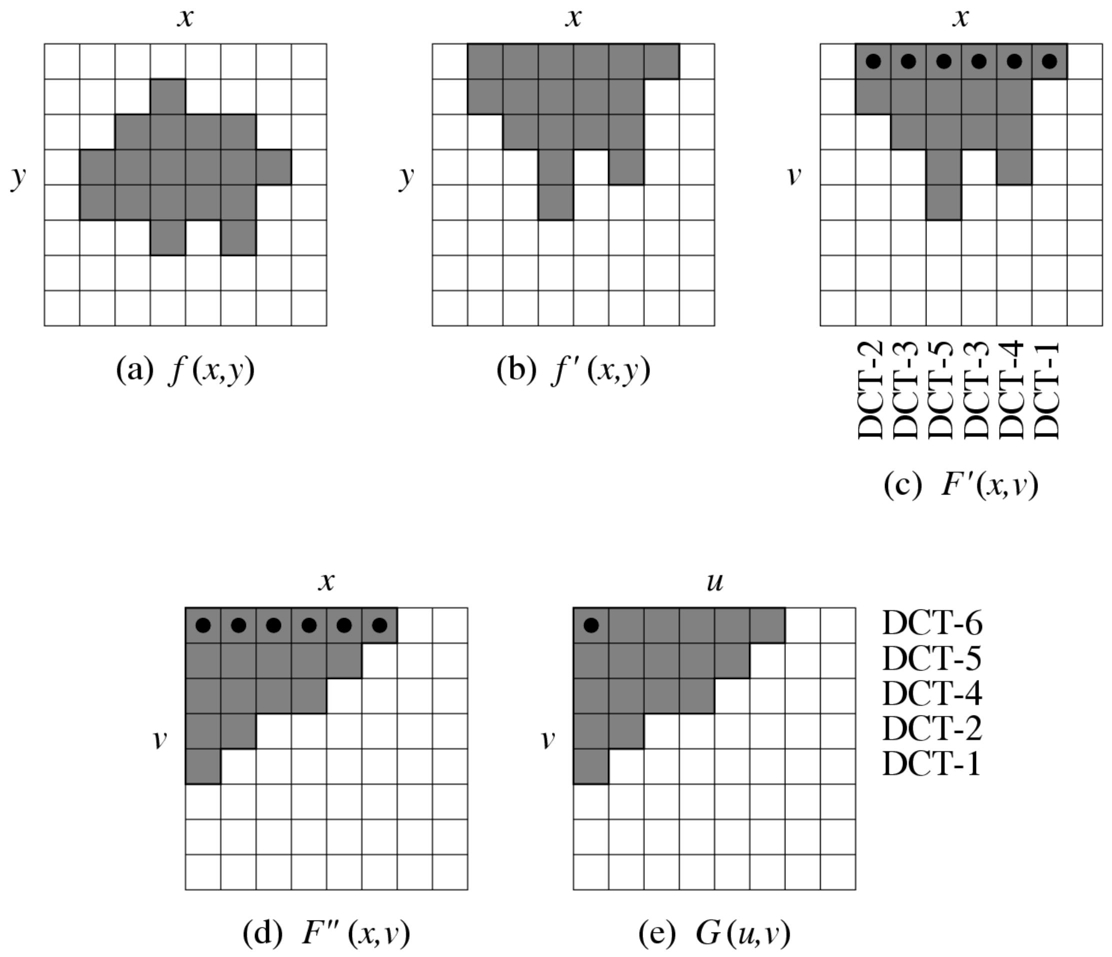


Fig. 12.8: Texture Coding for Boundary MBs Using the Shape Adaptive DCT (SA-DCT).

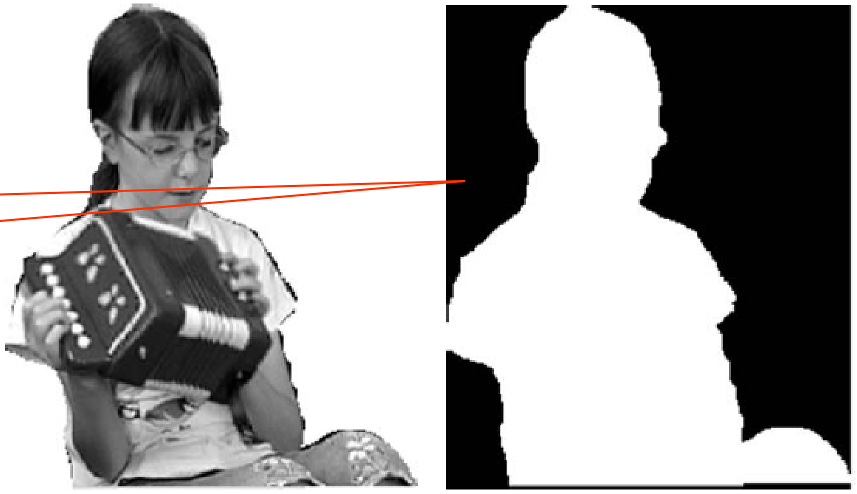
1.2.4 Shape Coding

- 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).

1.2.4 Shape Coding

- **Binary shape coding**

Pixels not in VOP are represented as transparent



(Context-based binary Arithmetic Coding)

1.2.4 Shape Coding

- Binary shape coding

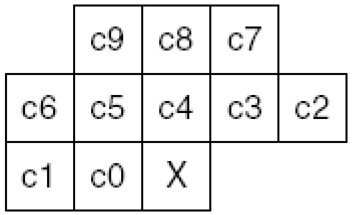
1. Calculate the current context value X

Each pixel is binary, context X is calculated according to 10 pixels already coded.

X has 10bits: C9C8C7C6C5C4C3C2C1C0

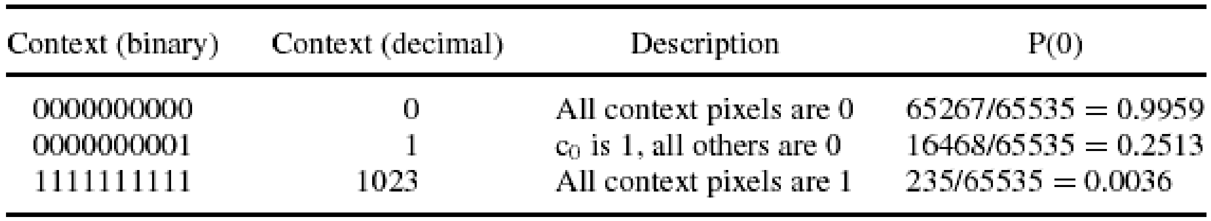
X is range in 0~1024

Lookup tables (MPEG-4 standard) to get the corresponding value.



2. Arithmetic Coding

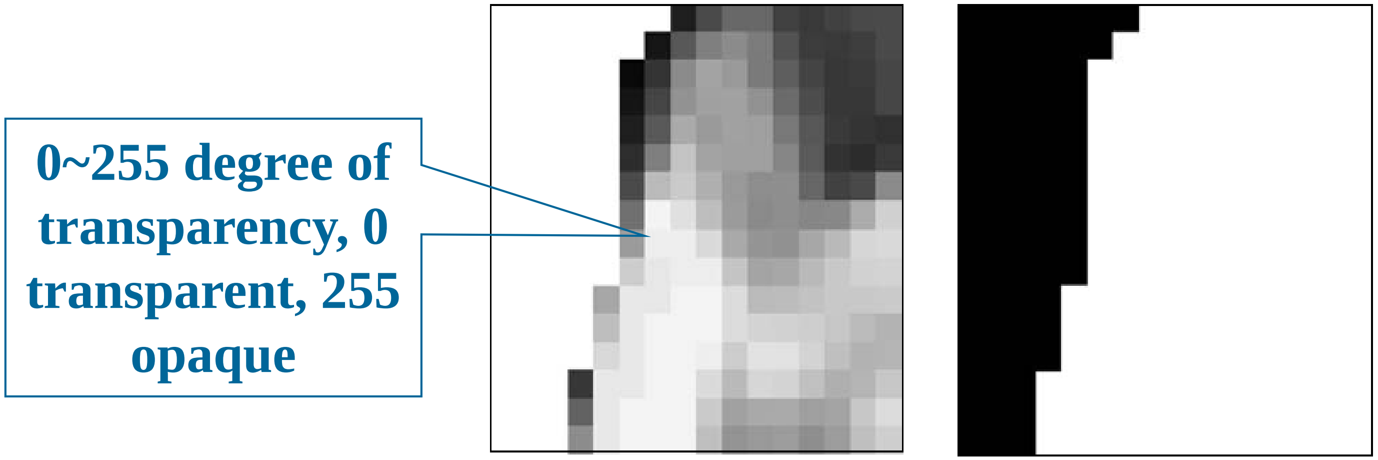
The value in the lookup table indicate the probability of occurrence for each of the 1024 contexts.



Context (binary)	Context (decimal)	Description	P(0)
0000000000	0	All context pixels are 0	65267/65535 = 0.9959
0000000001	1	c_0 is 1, all others are 0	16468/65535 = 0.2513
1111111111	1023	All context pixels are 1	235/65535 = 0.0036

1.2.4 Shape Coding

- Grayscale shape coding
  - Grayscale is used to describe the transparency of the shape, not the texture.



  - Raster graphics uses extra bitplanes for an alpha map.
  - Grayscale shape coding is lossy, while binary shape coding is lossless

1.2.5 Static Texture Coding

Wavelet coding for the texture of static objects

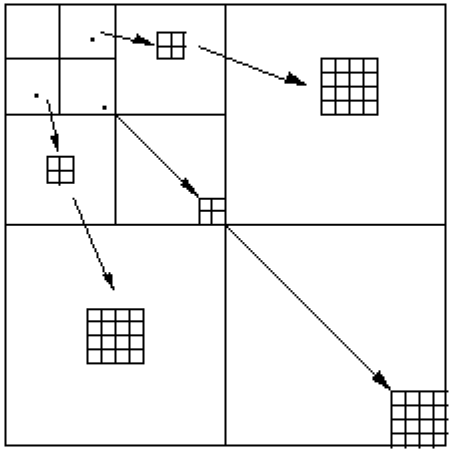
- The sub-bands with the lowest frequency are coded using DPCM
  - Prediction of each coefficient is based on three neighbors
- Other sub-bands are based on a multiscale zero-tree wavelet coding.
- The multiscale zero-tree has a parent-child relation tree
  - The location information of all coefficients is better used.
- A large quantizer is used and the most significant coefficients are coded using arithmetic coding.
  - Difference is coded in the next iteration in which a smaller quantizer is employed.

1.2.5 Static Texture Coding

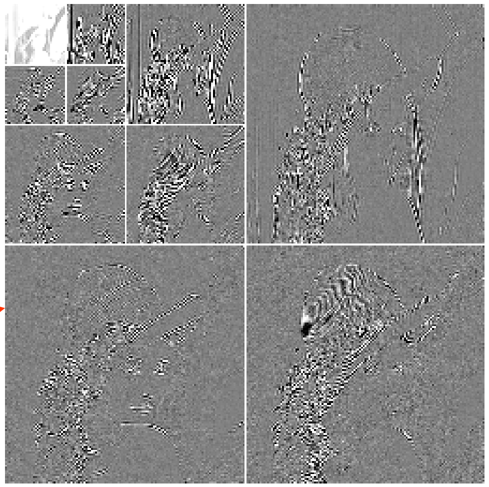
- Zerotree

original image

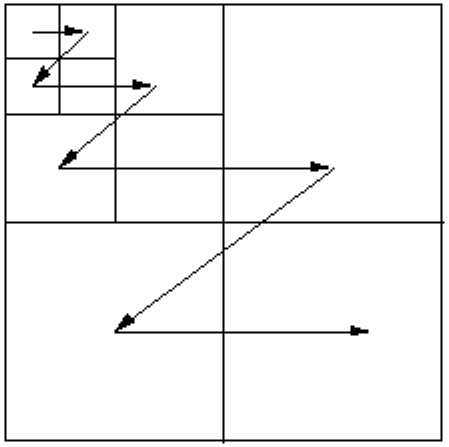
structure



Three level decomposition

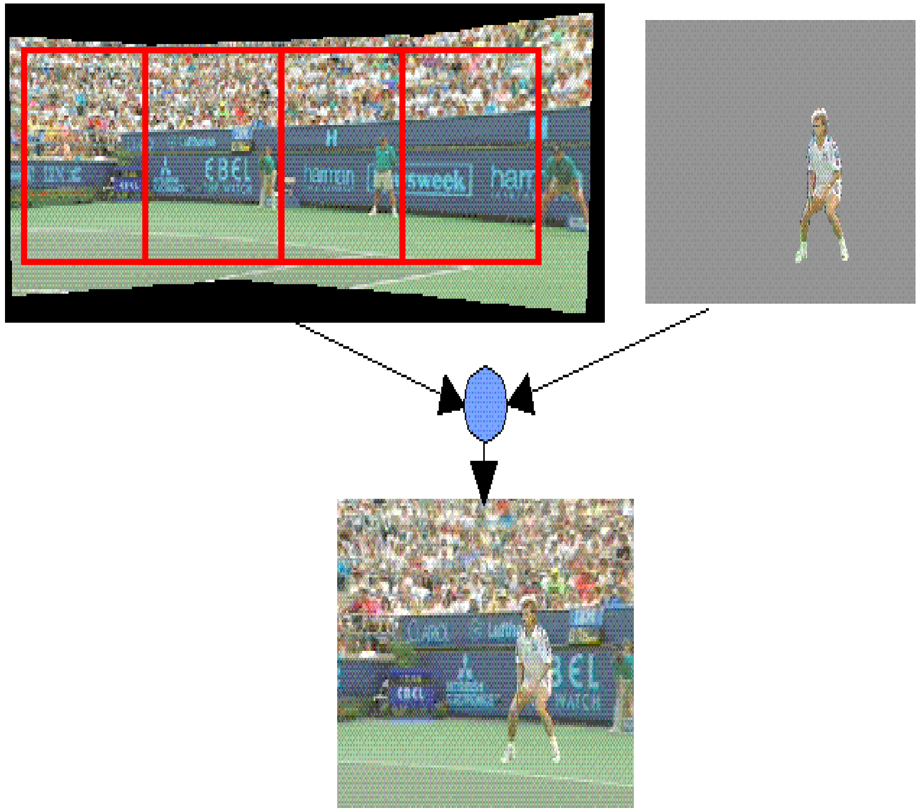


Scan order



1.2.6 Sprite Coding

- Some background can be treated as static image
- Background is effected by camera movement
- Background can coded separately, foreground objects can be used to create flexible object-based composition of MPEG-4 video

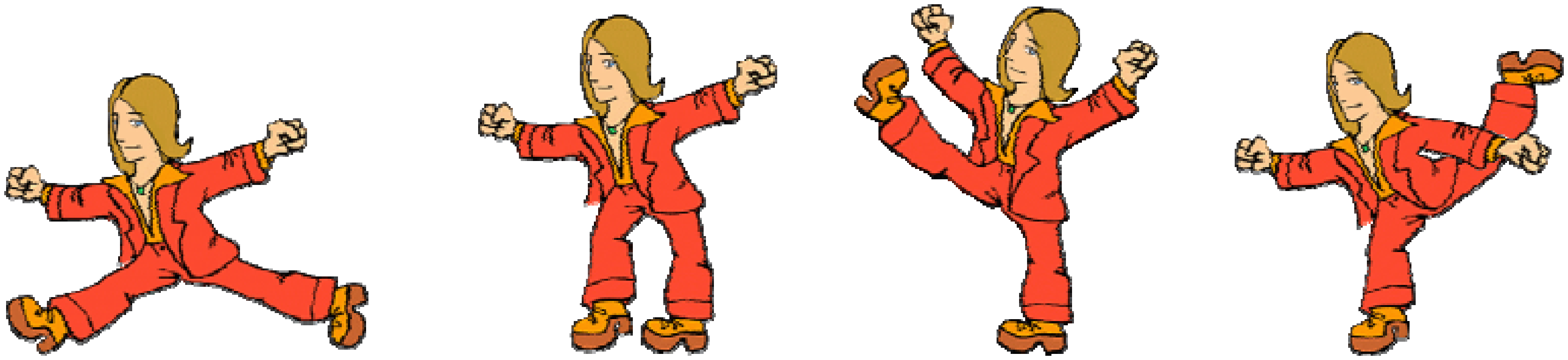


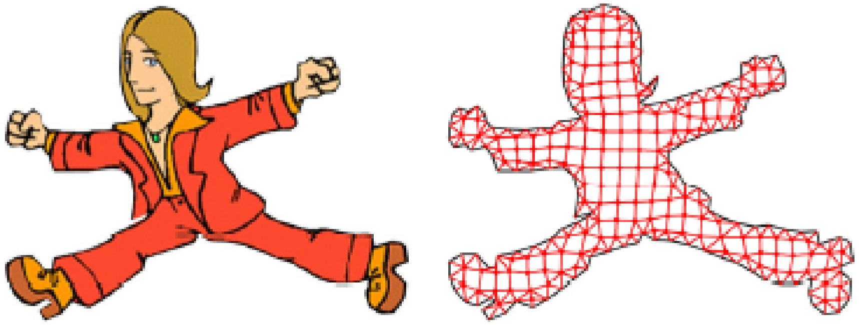
1.2.7 Global Motion Compensation

- Camera motion such as pan, tilt, rotation, and zoom often cause rapid content change between successive frames, block-based motion compensation is not a efficient method for this situation
- GMC is a better choice
- Global Motion Compensation has four major components
  - Global motion estimation
  - Warping and blending
  - Motion trajectory coding
  - Choice of local motion compensation (LMC) or GMC

1.3 Synthetic Object Coding

- Synthetic object: objects are created using computer
- 2D Mesh Object Coding: a tessellation (or partition) of a 2D planar region using polygonal patches:





(a) (b)

1.3 Synthetic Object Coding

– 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).

1.3 Synthetic Object Coding

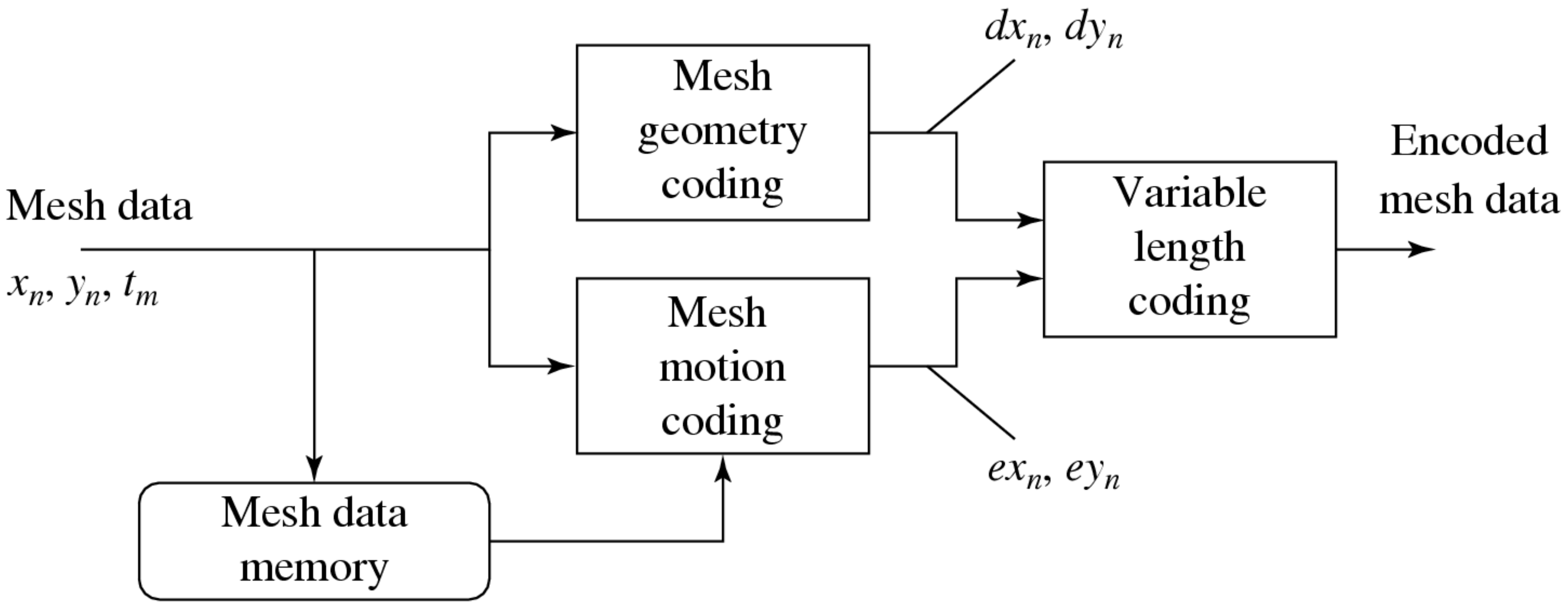
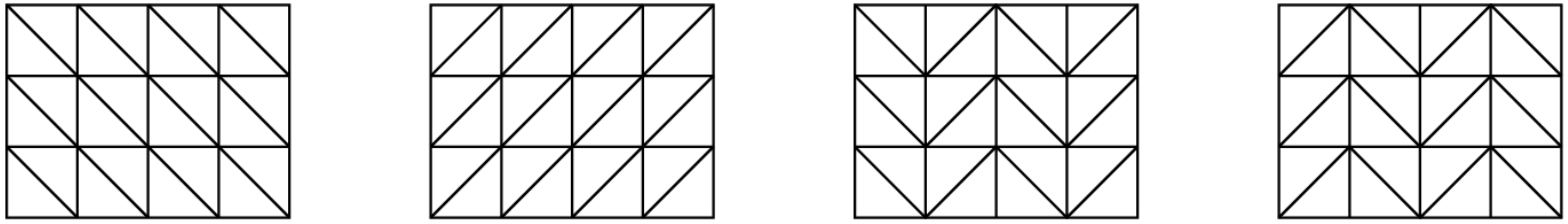


Fig. 12.11: 2D Mesh Object Plane (MOP) Encoding Process

1.3 Synthetic Object Coding

- 2D Mesh Object Coding
  - 2D Mesh Geometry Coding


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

  - Four types of uniform meshes
  - Delaunay mesh is a better object mesh representation
    - Select boundary nodes of the mesh
    - Choose interior nodes
    - Perform Delaunay Triangulation

1.3 Synthetic Object Coding

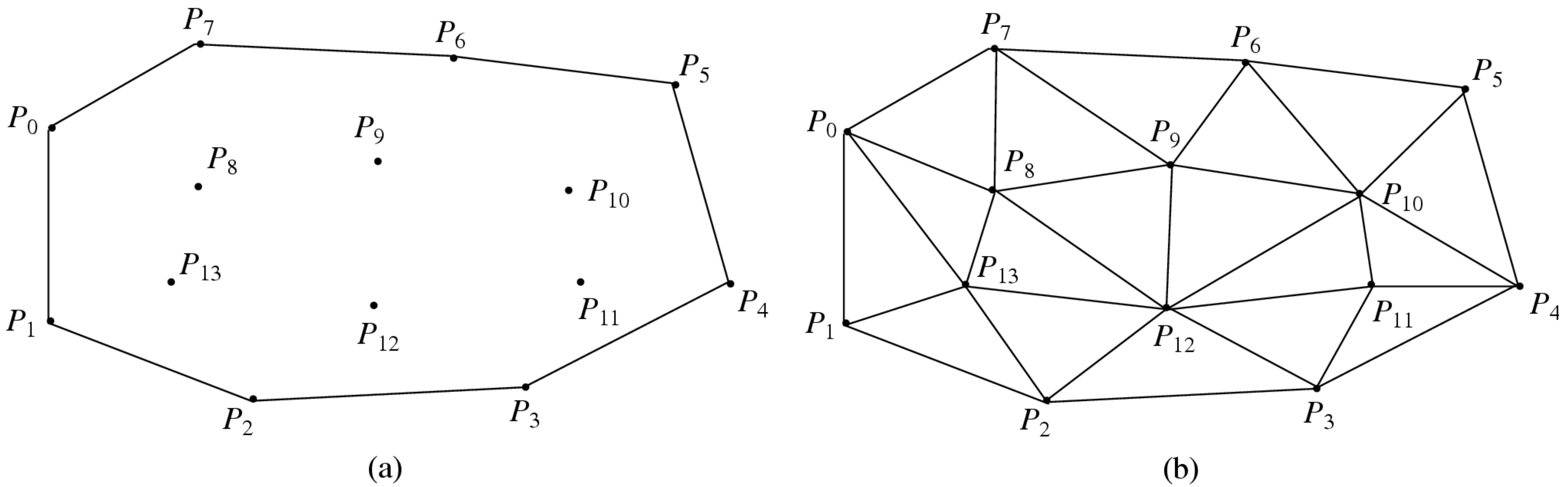


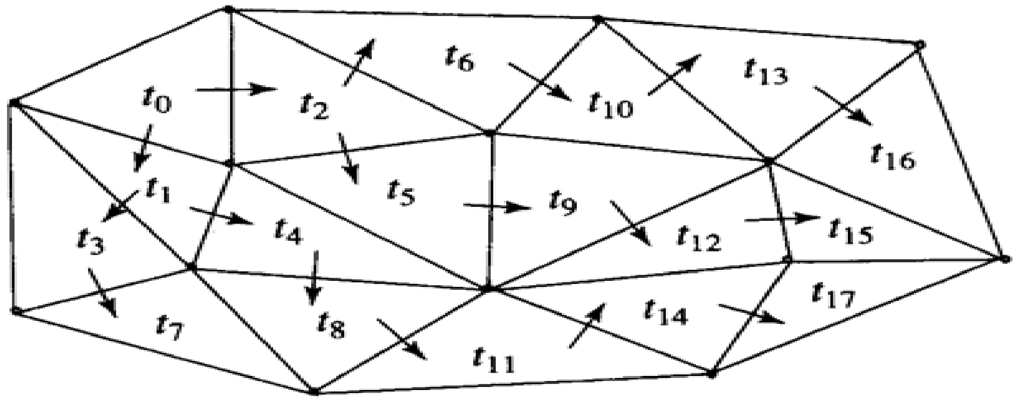
Fig. 12.13: Delaunay Mesh: (a) Boundary nodes (P_0 to P_7) and Interior nodes (P_8 to P_{13}). (b) Triangular mesh obtained by Constrained Delaunay Triangulation.

1.3 Synthetic Object Coding

- 2D Mesh Object Coding
  - 2D Mesh Motion Coding
    - For any MOP triangle i,j,k .if motion vectors for i,j are known, then motion vector for k can be predicted as

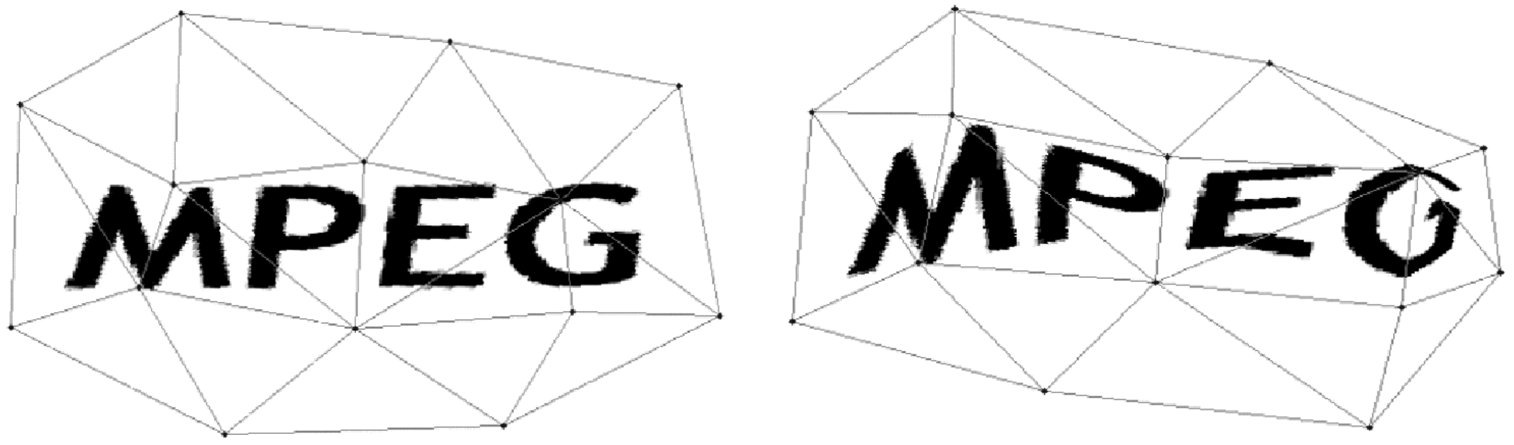
$$Pred_k=0.5(Pred_i+Pred_j)$$

    - When motion vectors of a triangle is coded, uncoded vertex of the neighboring MOP triangle share an edge with the previous triangle is coded, and so on, until all the triangles are coded.



1.3 Synthetic Object Coding

- 2D Mesh Object Coding
  - 2D Object Animation
    - The previous step established a one-to-one mapping between the mesh triangle in the reference MOP and the target MOP
    - Affine transform is used to achieve animated sequence



(a) (b)

1.3 Synthetic Object 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.

1.3 Synthetic Object Coding

– Face Object Coding and Animation

- MPEG-4 has adopted a generic default face model, which was 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.
- Fig. 12.16 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.

1.3 Synthetic Object Coding

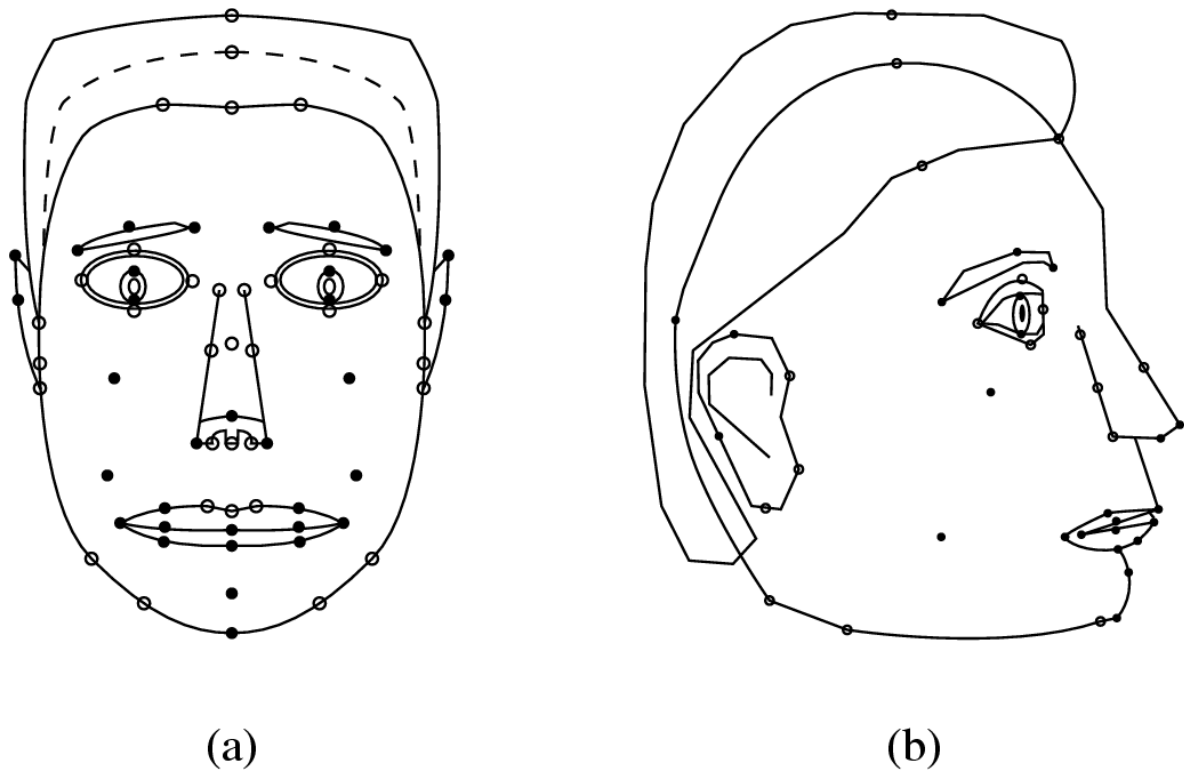


Fig. 12.16: Feature Points for Face Definition Parameters (FDPs). (Feature points for teeth and tongue not shown.)

1.3 Synthetic Object Coding

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.

1.3 Synthetic Object Coding

- 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.

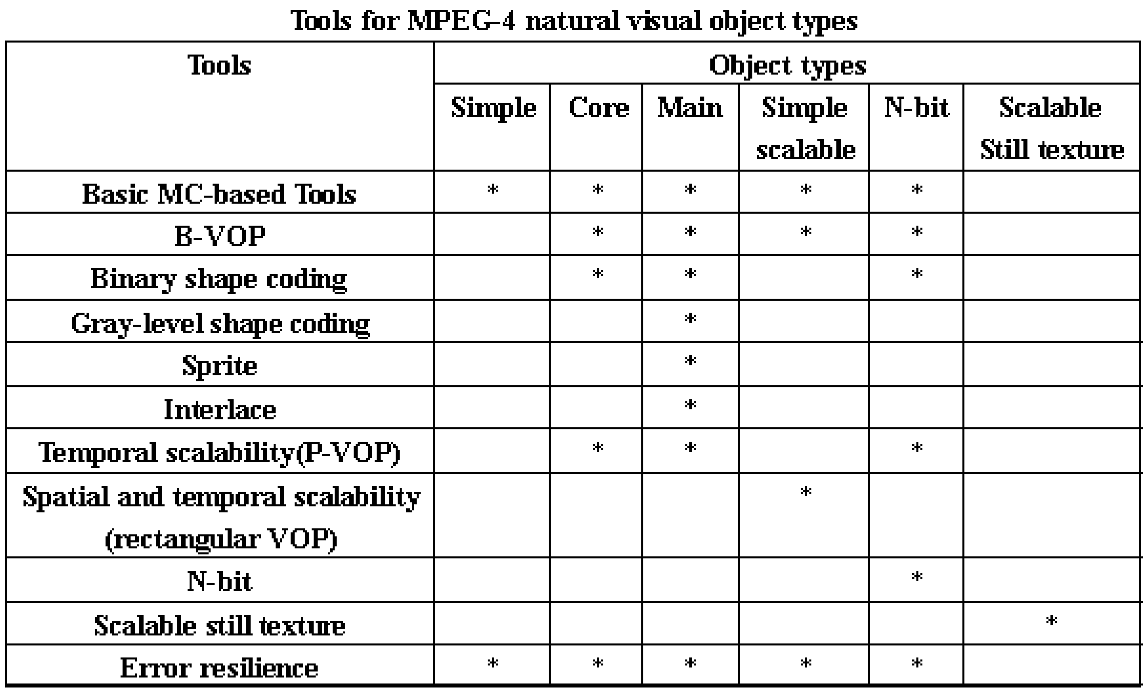
• For specific bodies, **Body Definition Parameters (BDPs)** can be specified for body dimensions, body surface geometry, and optionally, texture.

• The coding of BAPs is similar to that of FAPs: quantization and predictive coding are used, and the prediction errors are further compressed by arithmetic coding.

1.4 MPEG4 Object Types, Profiles and Levels

- Like MPEG2, MPEG4 defines many profiles and levels: Visual profiles, Audio profiles, Graphics profiles, Scene description profiles, Object descriptor profiles

MPEG4 defines the tools needed to create video objects and the ways they can be combined in a scene



Tools for MPEG-4 natural visual object types

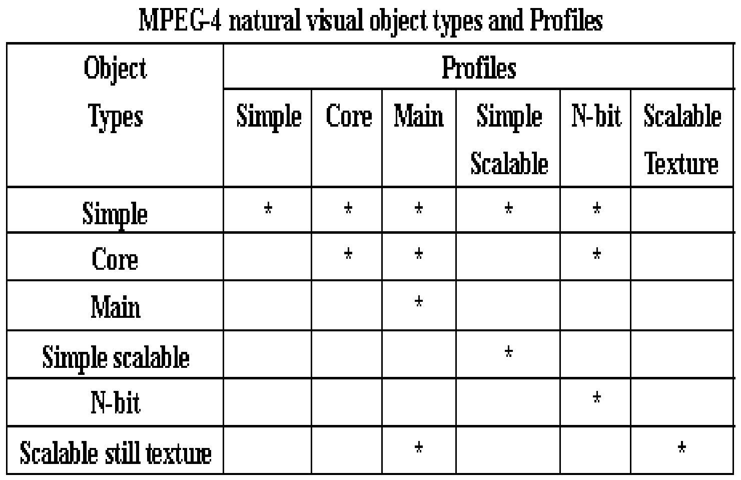
Tools	Object types					
	Simple	Core	Main	Simple scalable	N-bit	Scalable Still texture
Basic MC-based Tools	*	*	*	*	*	
B-VOP		*	*	*	*	
Binary shape coding		*	*		*	
Gray-level shape coding			*			
Sprite			*			
Interlace			*			
Temporal scalability(P-VOP)		*	*		*	
Spatial and temporal scalability (rectangular VOP)				*		
N-bit					*	
Scalable still texture						*
Error resilience	*	*	*	*	*	

1.4 MPEG4 Object Types, Profiles and Levels

- **Object types and levels in different profiles**

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

Levels in Simple, Core, and Main Visual Profiles


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			*			*

MPEG4 Natural Visual Object Types and Profiles

1.5 MPEG-4 Part10 / H.264

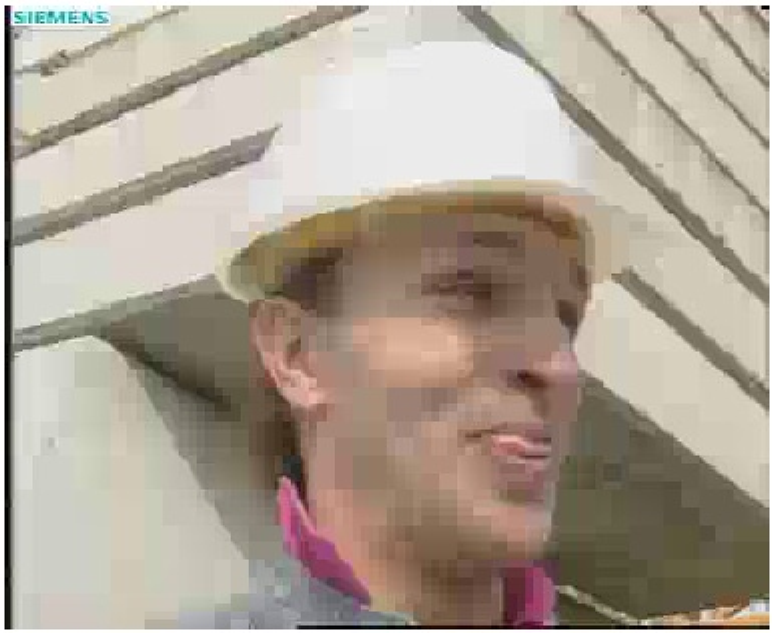
- 2001, MPEG and ITU-T VCEG (Video Coding Experts Group) united JVT (Joint Video Team)
- JVT proposed H.264 draft to ISO in 2003
- H.264 offers up to 50% better compression than MPEG-2 and up to 30% better than H.263+ and MPEG-4 advanced simple profile

1.5 MPEG-4 Part10 / H.264

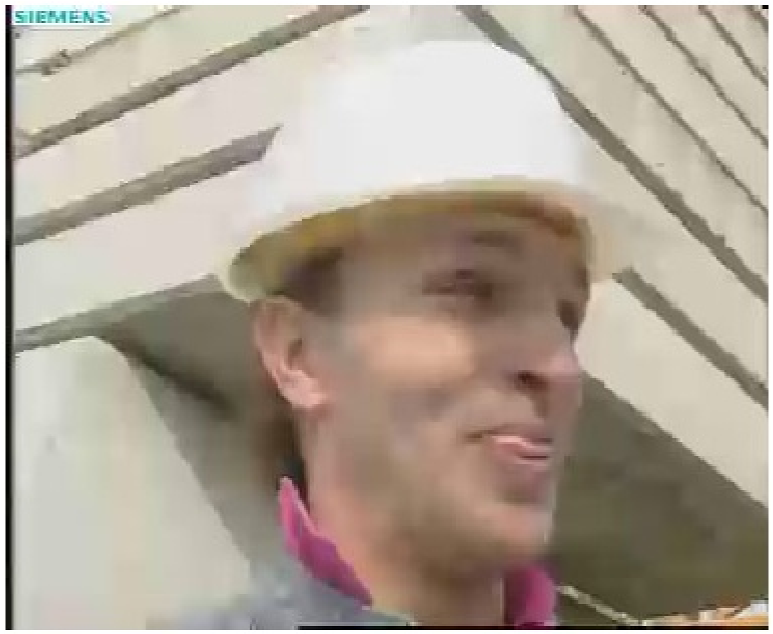
- Core Features
  - Entropy decoding
    - Unified-VLC(UVLC) and Context Adaptive VLC (CAVLC)
  - Motion compensation or intra-prediction
    - Variable block size and more accurate motion compensation.
  - Transform, Scan, Quantization
    - Nonlinear quantization and different quantization scales
  - I-Prediction
    - Intra-coded macroblocks are all predicted using neighboring reconstructed pixels
  - In-loop Deblocking Filters
    - Adopts a sophisticated signal-adaptive deblocking filter.

1.5 MPEG-4 Part10/H.264

- Deblocking Filter in H.264 can obtain pleasing results



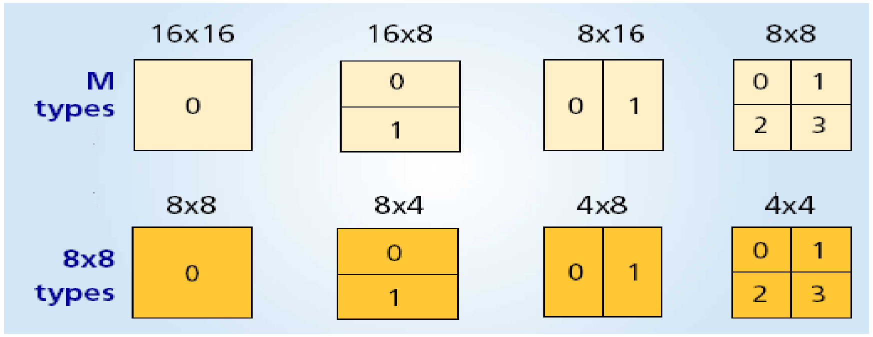
Non Deblocking

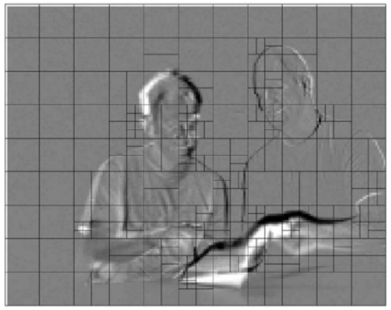


After Deblocking

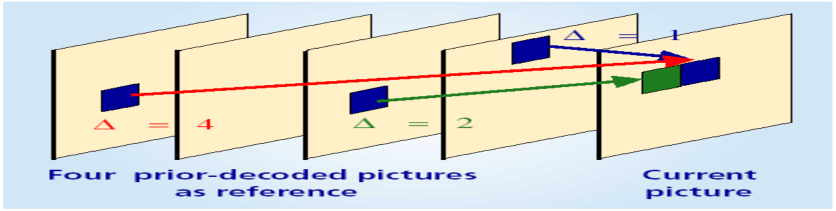
1.5 MPEG-4 Part10 / H.264

- Inter prediction
  - Tree-structured Motion Compensation
  - H.264 supports different block size, block size can down to 4*4





  - Select the optimal block size, minimize the difference between current an reference frame.
  - P frame can use more than one previous frames as reference frames.



2. MPEG-7

Overview of MPEG-7(1)

- More and more multimedia content becomes an integral part of various applications, effective and efficient retrieval becomes a primary concern.

- MPEG7 is to satisfy the need of audiovisual content-based retrieval.

- MPEG7 was initialized in 1998， finished in 2001.
- MPEG7 supports a variety of multimedia applications.
- MPEG7 doesn’t describe any feature extracting methods. Its formal name is “multimedia content description interface”.

Overview of MPEG-7(2)

- MPEG-7
  - Descriptors (D), Description Schemes (DS),
  - Description Schemes (DDL)
- Descriptor (D)
  - Color, Texture, Shape, motion, localization
- Description Schemes (DS)
  - Basic elements, content management, content description, navigation and access
- XML Schema Language and MPEG7 Extensions

Overview of MPEG-7(2)

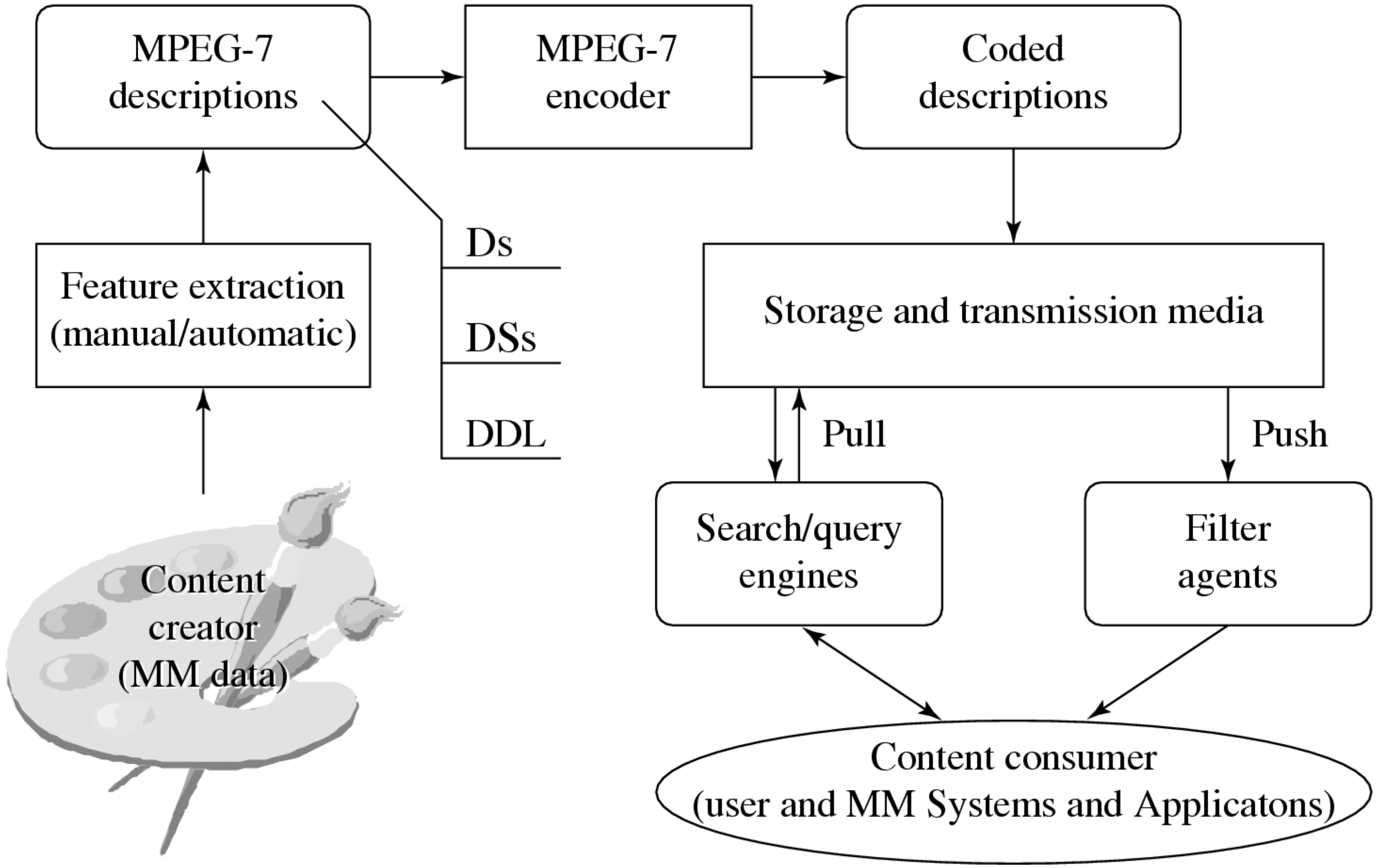

MPEG-7 descriptions
MPEG-7 encoder
Coded descriptions
Feature extraction (manual/automatic)
Ds
DSs
DDL
Storage and transmission media
Pull
Push
Search/query engines
Filter agents
Content creator (MM data)
Content consumer (user and MM Systems and Applicatons)

Fig. 12.17: Possible Applications using MPEG-7.

3. MPEG-21

Overview of MPEG-21

- MPEG-21 is to define a uniform way to define, identify, describe, manage, and protect multimedia data.
- MPEG21 has 7 key parts
  - Digital item declaration
  - Digital item identification and description
  - Content management and usage
  - Intellectual property management and protection
  - Terminal and networks
  - Content representation
  - Event reporting

The End

Thanks！

Email: junx@cs.zju.edu.cn