Network & Media Laboratory

Fundamentals of Multimedia

Authored by Ze-Nian Li Mark S. Drew

Lecturer：Lu Dongming

Network & Media Laboratory

9、Basic Video Compression Techniques

Content

- Introduction to video compression
- Video compression based on motion compensation
- Search for motion vectors
- H.261
- H.263

1. Introduction to Video Compression

1 Introduction

- **Requirement** of video compression
  - Uncompressed video data could be **extremely large**
  - Pose some problems for **network communication**

- **Feasibility** of video compression
  - Frames in the same scene are very similar, so video data have **temporal redundancy**
  - Even static images can be compressed at large compression ratio, say nothing of video

1 Introduction

- Video is a sequence of images stacked in the **temporal dimension**
- The most **simple** method: **Predictive Coding**
  - Subtract images **in time order**
  - Code the **residual error**
- **Better** methods
  - **Search for the right parts** of the image to subtract from the previous frame.
  - **Motion Estimation**
  - **Motion Compensation**

2. Video Compression Based on Motion Compensation

2.1 Temporal Redundancy

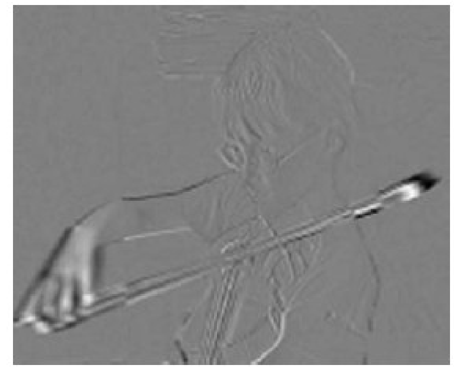
- A video: a sequence of images in temporal dimension
- Consecutive frames are usually similar
  - The video has significant temporal redundancy
  - Not every frame coded independently
  - Difference between adjacent frames are coded
- Main cause of difference between frames
  - Camera or object motion
- Motion generators can be compensated
  - Detecting the displacement of corresponding pixels or regions
  - Measuring their differences (motion compensation MC)

2.1 Temporal Redundancy

- **Both spatial and temporal redundancy** exist in moving sequential pictures

−

=

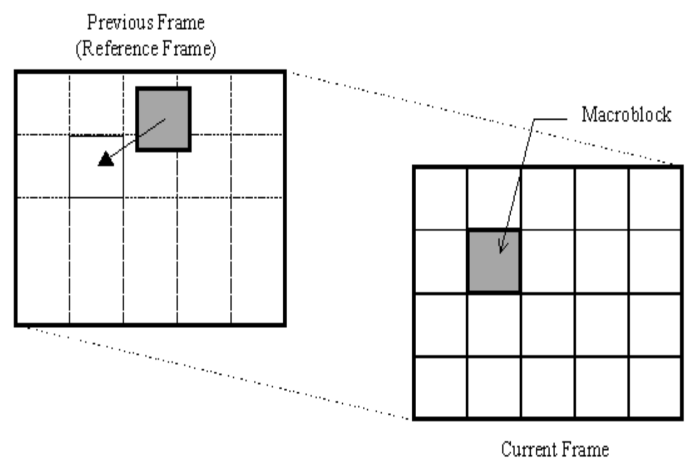


- Principles of moving pictures encoding： reduce spatial redundancy and temporal redundancy
  - **Intra-Frame**: similar as JPEG
  - **Inter-Frame**: based on motion prediction and compensation
    - P frame、B frame
    - Multi-frame references（H.264）

2.2 Motion Compensation

- The three main steps
  - Motion estimation: motion vector search
  - Motion-compensation-based prediction
  - Derivation of the prediction error
- Unit of motion compensation
  - Macroblocks of size N×N
- MV (Motion Vector):
  - Displacement of the reference block to the target macroblock
- Video compression based on motion compensation
  - Only motion vectors and difference macroblocks need to be coded except the first frame

2.2 Motion Compensation



Motion prediction and compensation

3. Search For Motion Vectors

3.1 Criteria of matching

- **Motion Vector（MV）search: a matching problem, called correspondence problem**
- **Horizontal and vertical displacement i, j are in the range [-p, p], a search window of size （2p+1）*（2p+1）**
- **The goal: find (i, j) minimize the distance between two macroblocks**

$$C(x+k, y+l): \textit{pixel in target frame macroblock}$$

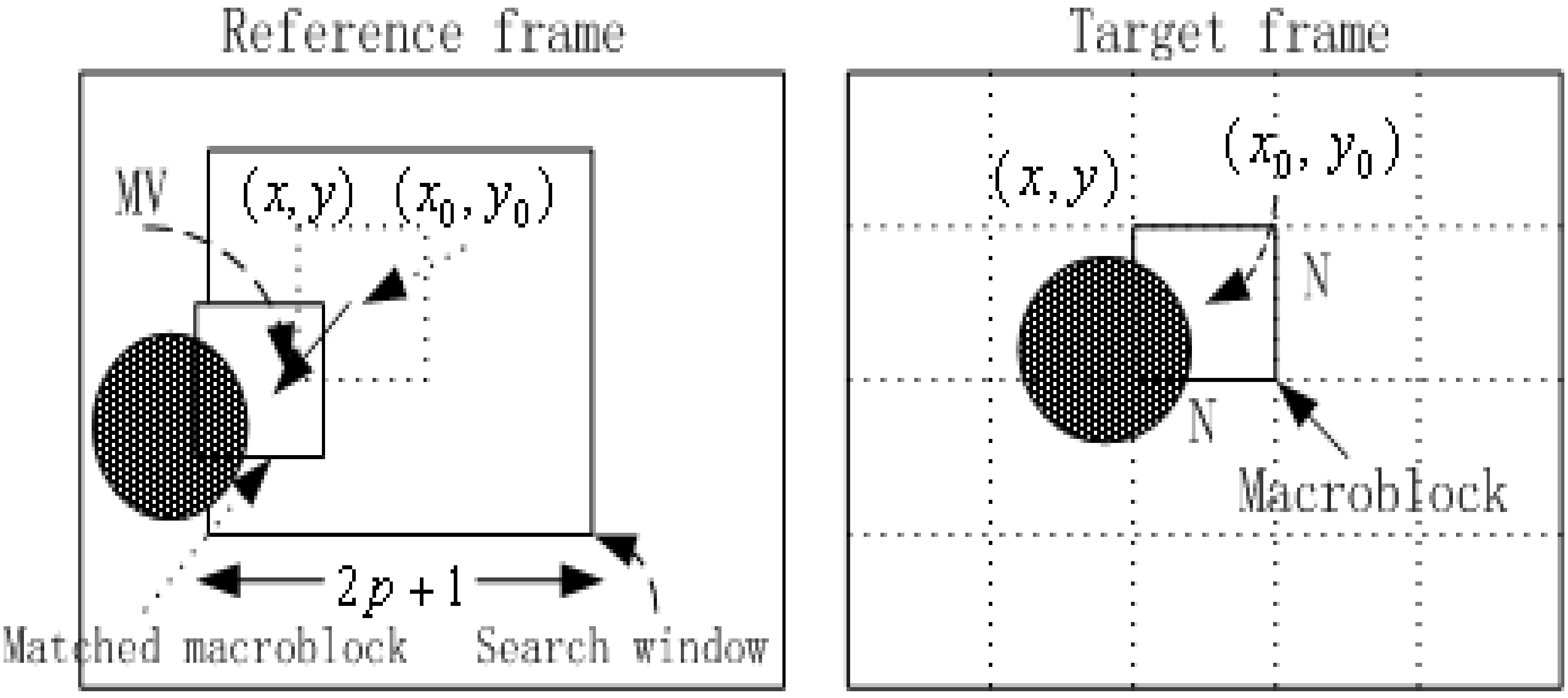
$$R(x+i+k, y+j+l): \textit{pixel in reference macroblock}$$

$$\textit{when motion vector is}\,(i, j)$$

$$MAD(i, j) = \frac{1}{N^2}\sum_{k=0}^{N-1}\sum_{l=0}^{N-1}|C(x+k, y+l) - R(x+i+k, y+j+l)|$$

$$(u, v) = [(i, j) \mid MAD(i, j) \quad \textit{is minimum}, \quad i \in [-p, p], j \in [-p, p]]$$

3.1 Criteria of matching



3.2 Sequential Search

- The simplest MV search method, search the whole (2p+1)*(2p+1) window in reference frame, referred to as Full Search
- Sequential search：

```
Begin
Min_MAD=Large_Number;
        for i= -p to p
                    for j=-p to p
                    {   cur_MAD = MAD(i,j);
                        If(cur_MAD<Min_MAD
                       {  min_MAD = cur_MAD;
                          u=i;
                          v=j;
                       }
                    }
End
```

- The sequential search is very costly;
  - the cost for a macroblock：(2p+1）*（2p+1）*N*N*3 => O(p^2N^2)

3.3 2D-Logarithmic-search

- A cheaper version : suboptimal but effective
- Logarithmic Search procedure:

Begin

offset= $\left\lceil \frac{p}{2} \right\rceil$

Specify nine macroblocks within the search window, they are centered at (x0, y0) and separated by offset horizontally and/or vertically;

while last!=TRUE

{ find the macroblocks yields minimum MAD;

if offset=1 then last=TRUE;

offset = $\left\lceil \frac{offset}{2} \right\rceil$

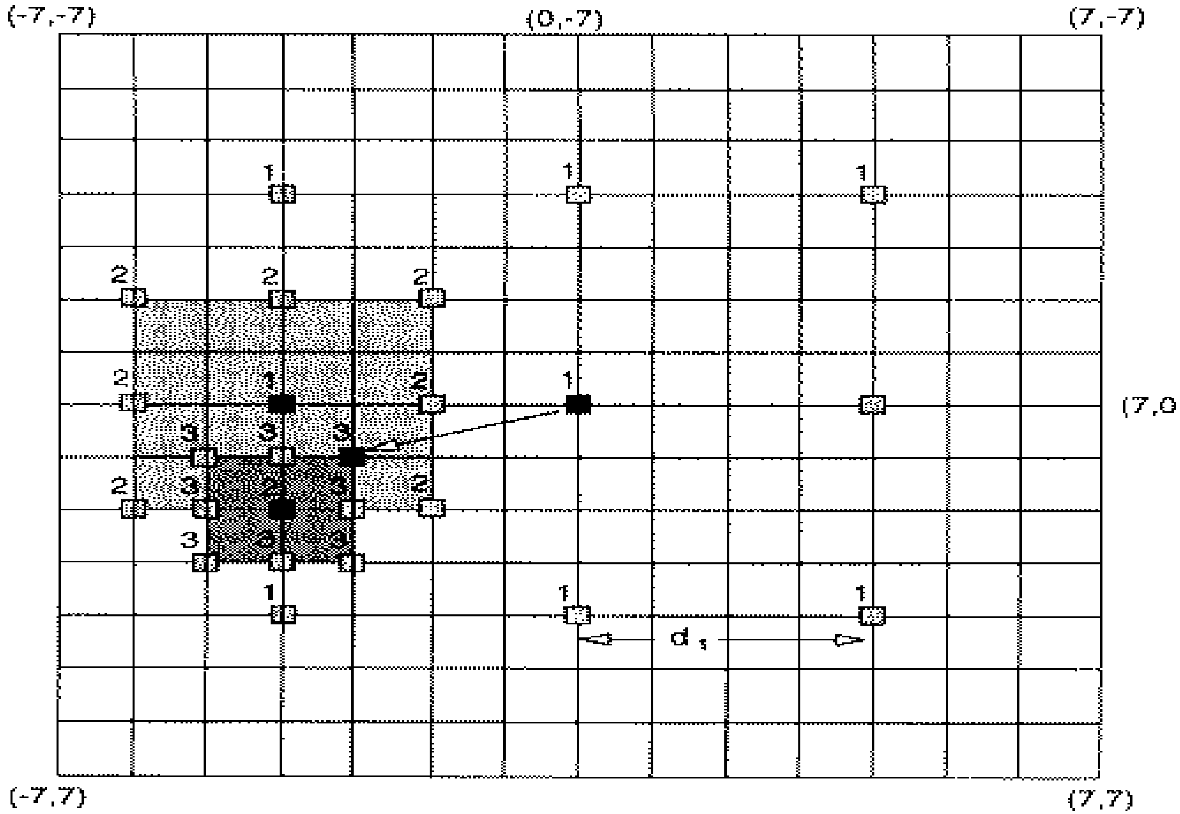
Form a search region with the new offset and new center found;

}

End

3.3 2D-Logarithmic-search

- **2D Logarithmic search for motion vectors:**



3.4 Hierarchical Search

Motion vector obtained from images with significantly reduced resolution

```
Begin
    // Get macroblock center position at lowest resolution level k
```

$$x_0^k = x_0^0 / 2^k; y_0^k = y_0^0 / 2^k;$$

Use sequential (or 2D logarithmic) serch to get initial estimated $MV(u^k, v^k)$ at level k

WHILE last != TRUE

{

Find one of the nine macroblocks that yields minimum MAD

$$(2(x_0^k + u^k) - 1 \le x \le 2(x_0^k + u^k) + 1, \quad 2(y_0^k + v^k) - 1 \le y \le 2(y_0^k + v^k) + 1;$$

if k=1 then last = TRUE;

k=k-1;

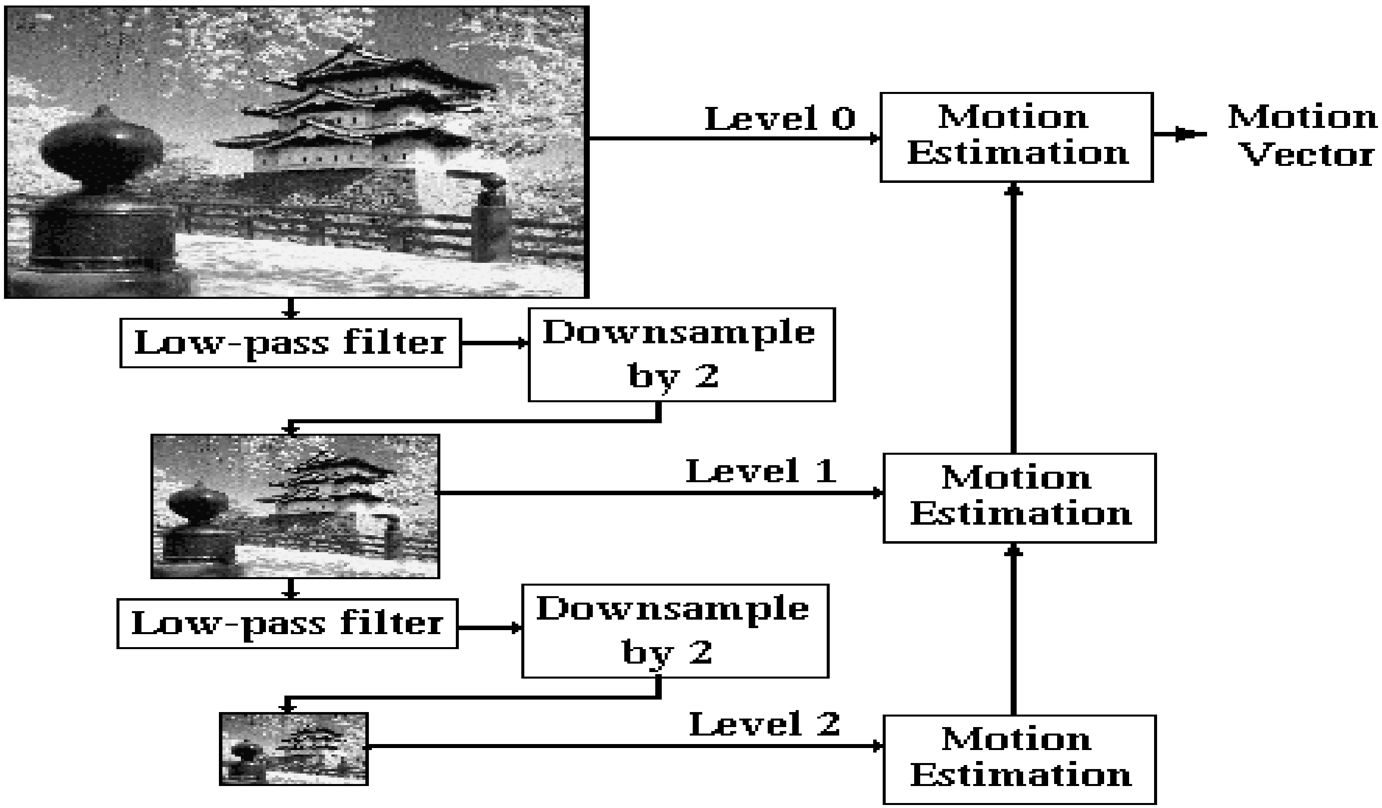
Assign (x_0^k, y_0^k) and (u^k, v^k) with the new center location and motion vectors

}

End

3.4 Hierarchical Search

- Hierarchical search for motion vectors



3.5 Other methods

- Search of motion vector is one of major steps in video compression, many methods are proposed to improve the efficiency.
- Some methods are listed as follows：
  - Three Step Search： TSS
  - Conjugate Direction Search： CDS
  - Cross Search Algorithm： CSA
  - New TSS： NTSS
  - Four-Step Search： FSS
  - Diamond Search Algorithm： DS
  - Adaptive Block Matching Algorithm： ABMA

4. H.261

4.1 Overview of H.261

- **H.261 is an earlier digital video compression standard. Initiated in 1988 by CCITT， formed in 1990**
- **Designed for videophone, videoconferencing and other audiovisual services over ISDN**
- **Requires the video encoders delay less than 150msec.**

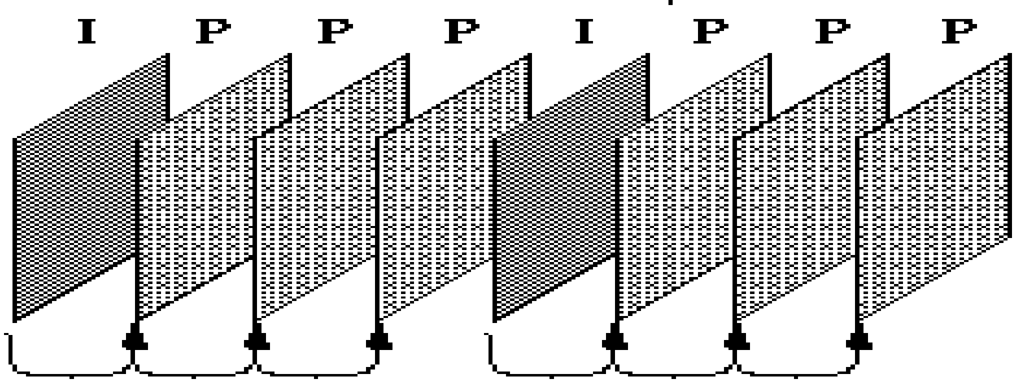
Video formats supported by H.261

Video format	Luminance image resolution	Chrominance image Resolution	Bitrate (Mbps) (if 30 fps and Uncompressed)	H.261 support
QCIF	176×144	88×72	9.1	Required
CIF	352×288	176×144	36.5	Optional

4.1 Overview of H.261

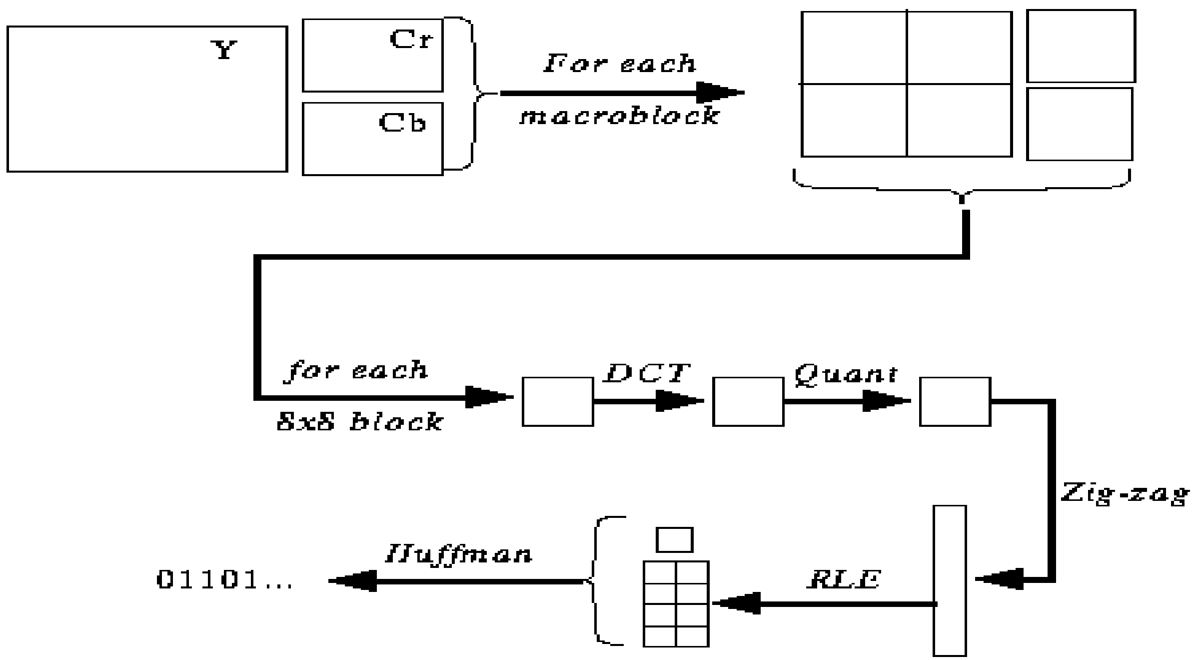
- **H.261 has two types of image frames:**
  - **Intra-fames (I-frames) and inter-frames (P-frames)**
- **I-frame is independent image**
- **P-frame depend on previous I-frame or P-frame**
  - **Remove temporal redundancy**

H.261 Frame sequence



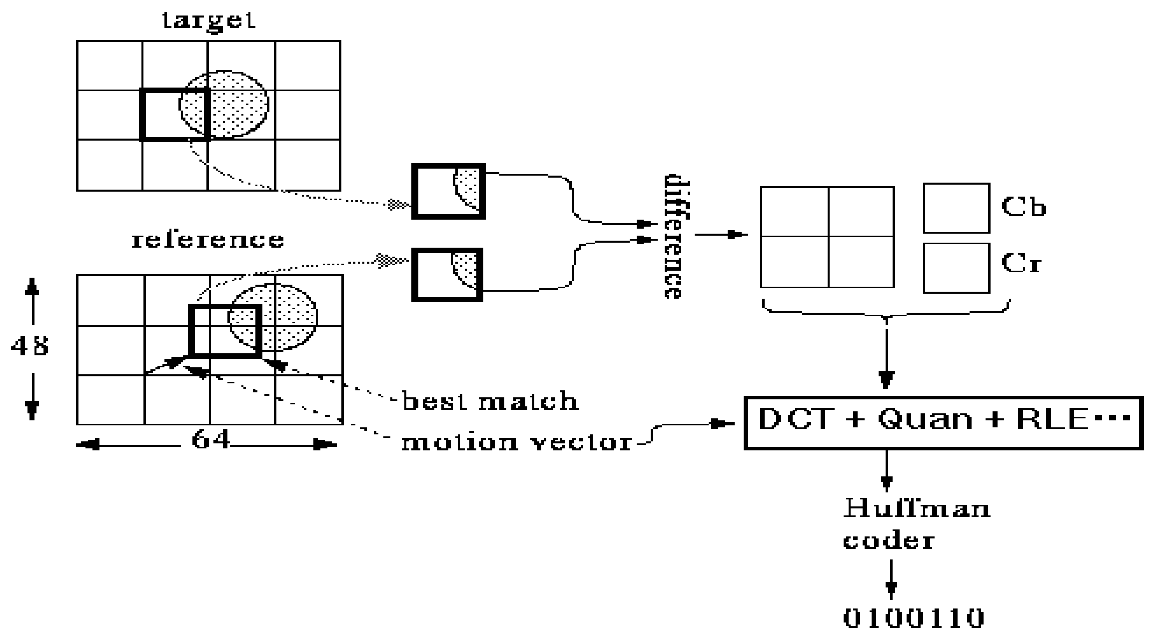
4.2 Intra-Frame Coding

- Macroblock for Y frame is of size 16*16 ;
- Cb,Cr correspond to 8*8 block



4.3 Inter-Frame predictive Coding

- **Encoding based on motion compression.**
  - **Prediction error is measured by difference macroblock;**
- **If prediction error exceed certain level, the macroblock itself is coded. (non-motion-compensated-macroblock)**



4.3 Inter-Frame predictive Coding

- Use reconstructed frame as reference frame, not original frame.
- Reference frame can be previous I frame or P frame.
- The motion vector is not directly coded, predicted error is sent for entropy coding.

$$MVD = MV_{Preceding} - MV_{Current}$$

4.4 Quantization in H.261

- Quantization uses a constant (step_size) for all DCT coefficient in a macroblock, step_size is one of the even value from 2 to 62.
- In intra mode, step_size=8 is always used for DC coefficient.

$$QDCT = round\left(\frac{DCT}{step\_size}\right) = round\left(\frac{DCT}{8}\right)$$

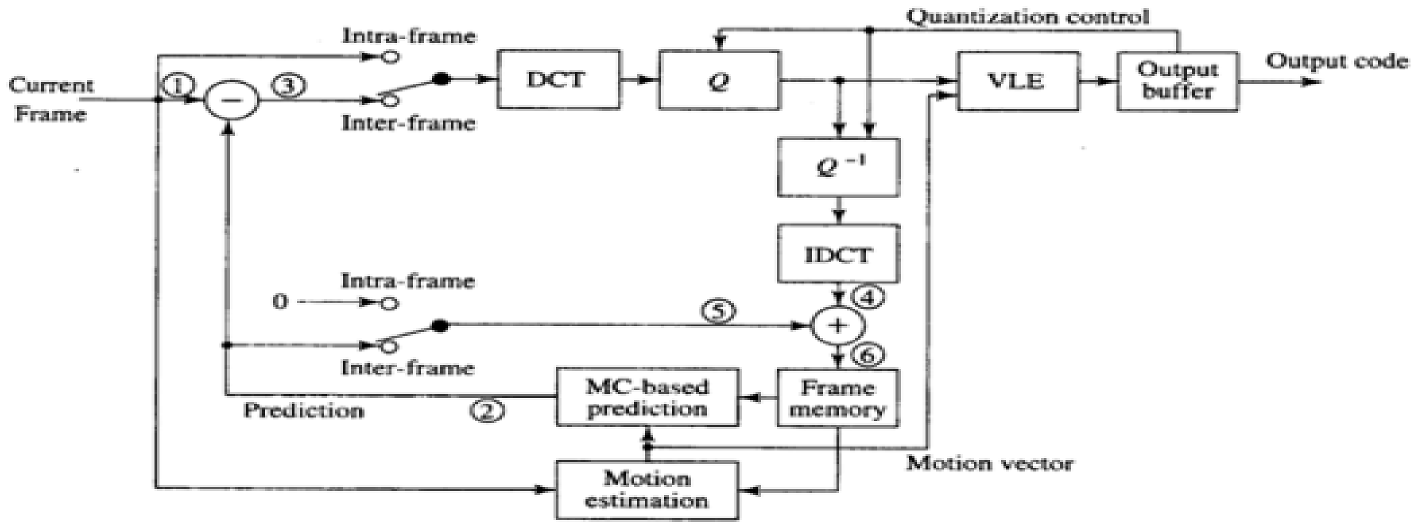
- For all other coefficients

$$QDCT = \left\lfloor \frac{DCT}{step\_size} \right\rfloor = \left\lfloor \frac{DCT}{2 \times scale} \right\rfloor$$

- scale is an integer in the range of [1， 31]

4.5 H.261 encoder and decoder

- Encoder and data flow



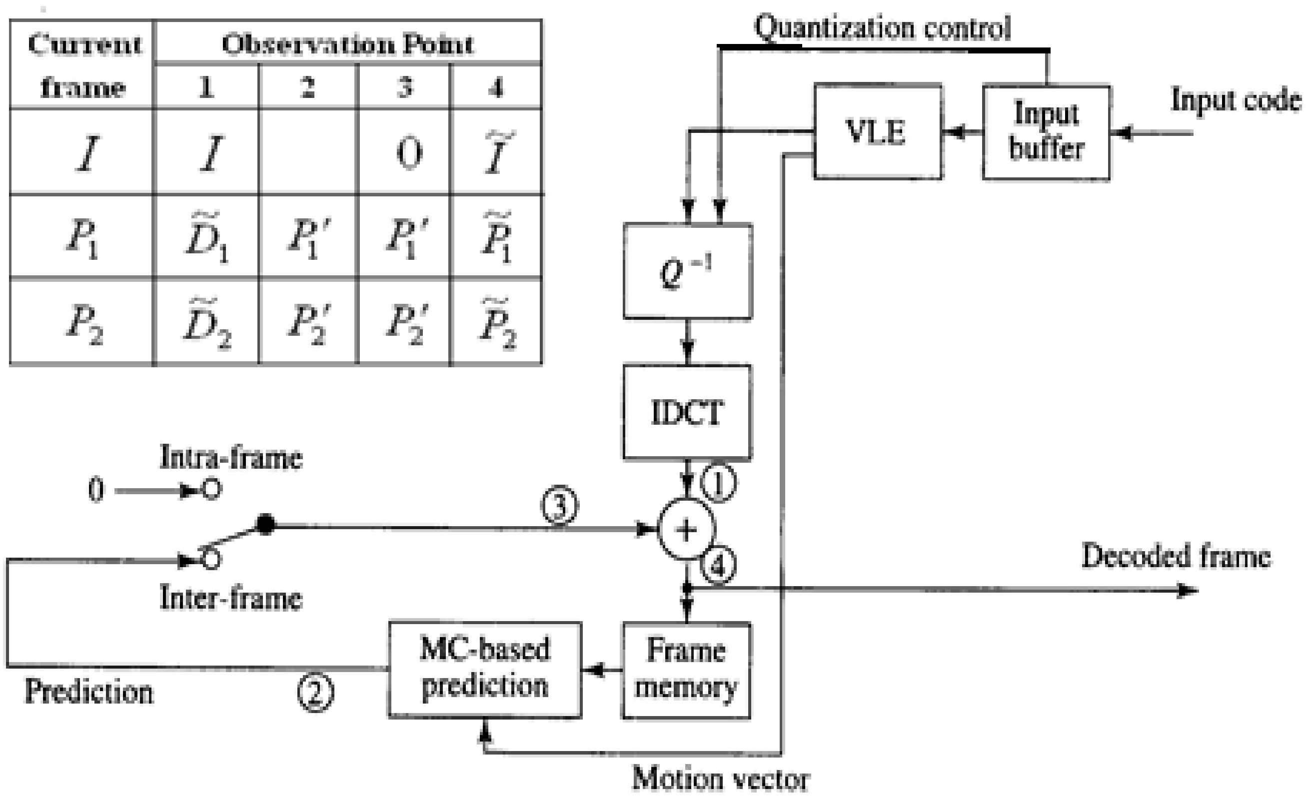
(a) Encoder

Current frame	Observation Point					
	1	2	3	4	5	6
I	I			$\tilde{I}$	0	$\tilde{I}$
P_1	P_1	P_1'	D_1	$\tilde{D}_1$	P_1'	$\tilde{P}_1$
P_2	P_2	P_2'	D_2	$\tilde{D}_2$	P_2'	$\tilde{P}_2$

4.5 H.261 encoder and decoder

- Decoder and data flow

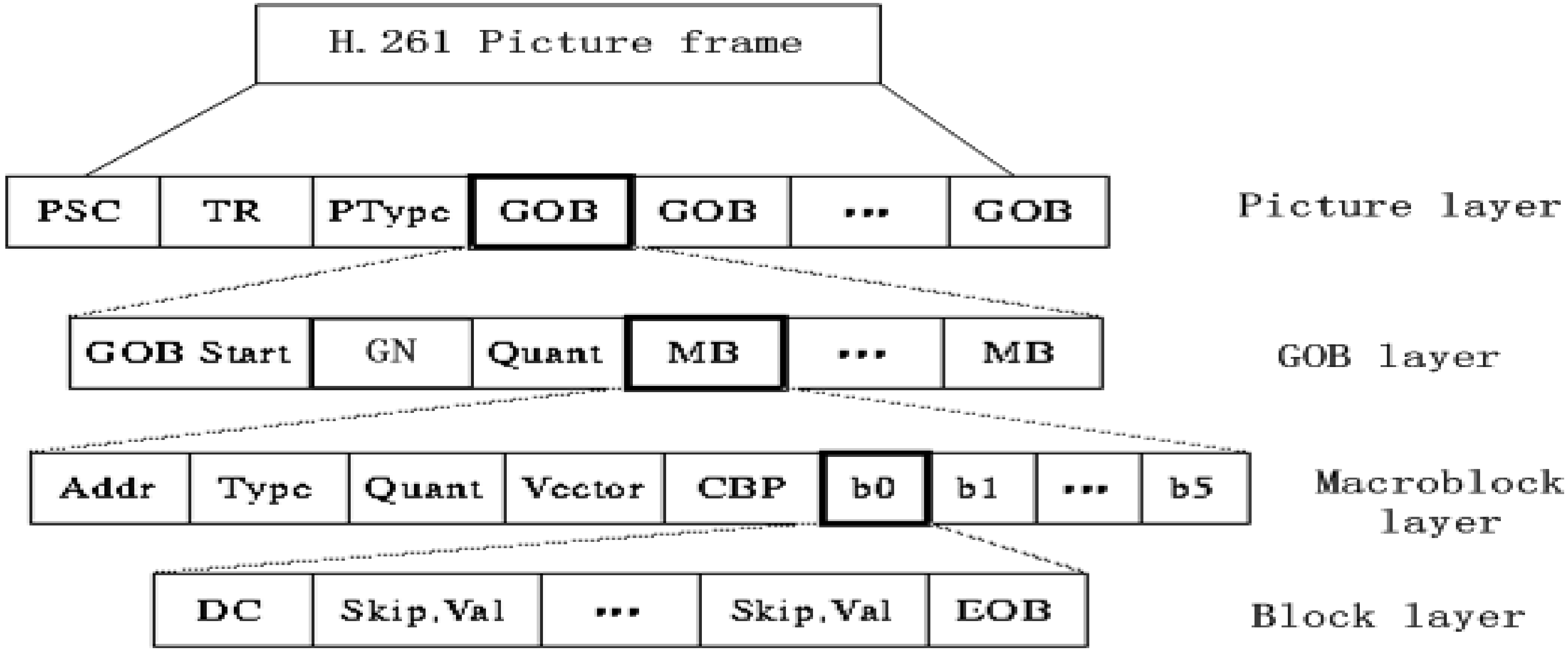
Current frame	Observation Point			
	1	2	3	4
I	I		0	$\tilde{I}$
P_1	$\tilde{D}_1$	P_1'	P_1'	$\tilde{P}_1$
P_2	$\tilde{D}_2$	P_2'	P_2'	$\tilde{P}_2$



(b) Decoder

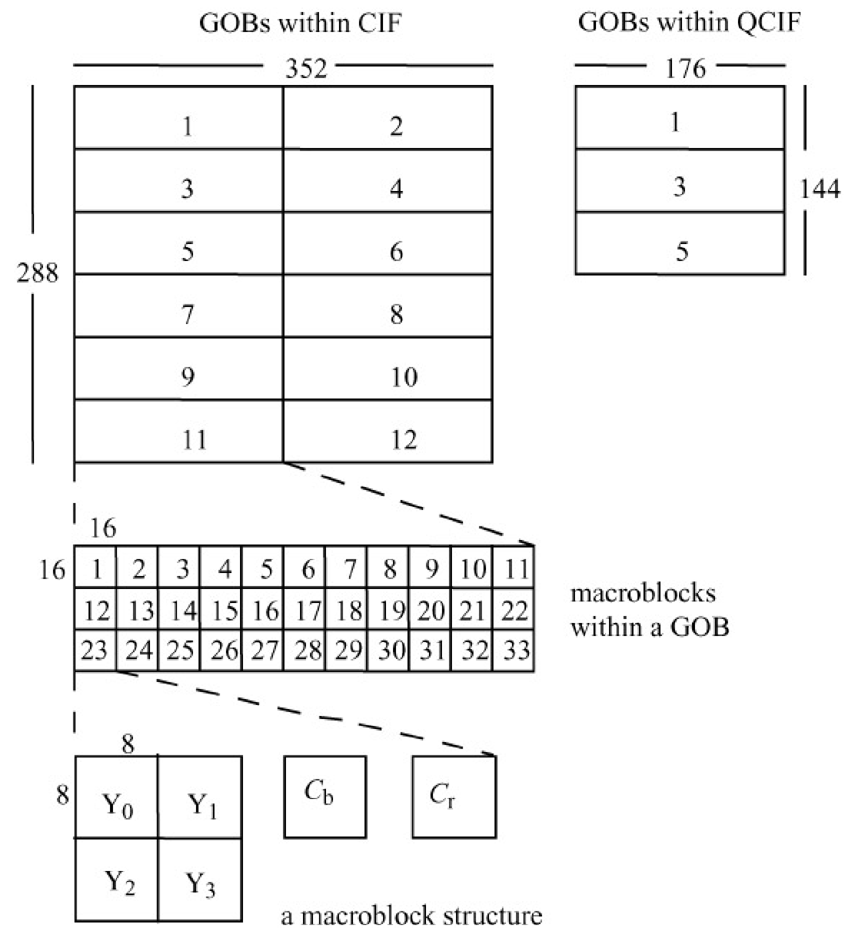
4.6 H.261 Video Bitsteam Syntax

- H.261 video bitsteam syntax： four layers
  - Picture, Group of Blocks, Macroblock and Block



4.6 H.261 Video Bitsteam Syntax

- In H.261, Frame （Picture） is the highest layer



4.6 H.261 Video Bitsteam Syntax

- Syntax of H.261 video bitstream
  - PSC (Picture Start Code)
  - TR (Temporal Reference)
  - Ptype (Picture Type)
  - GOB (Group of Blocks)
  - GBSC (GOB Start Code)
  - GN (Group Number)
  - Gquant (GOB Quantizer)
  - MB (Macroblock)
  - Mquant (MB Quantizer)
  - MVD (Motion Vector Data)
  - CBP (Coded Block Pattern)
  - EOB (End of Block)

5. H.263

5.1 overview of H.263

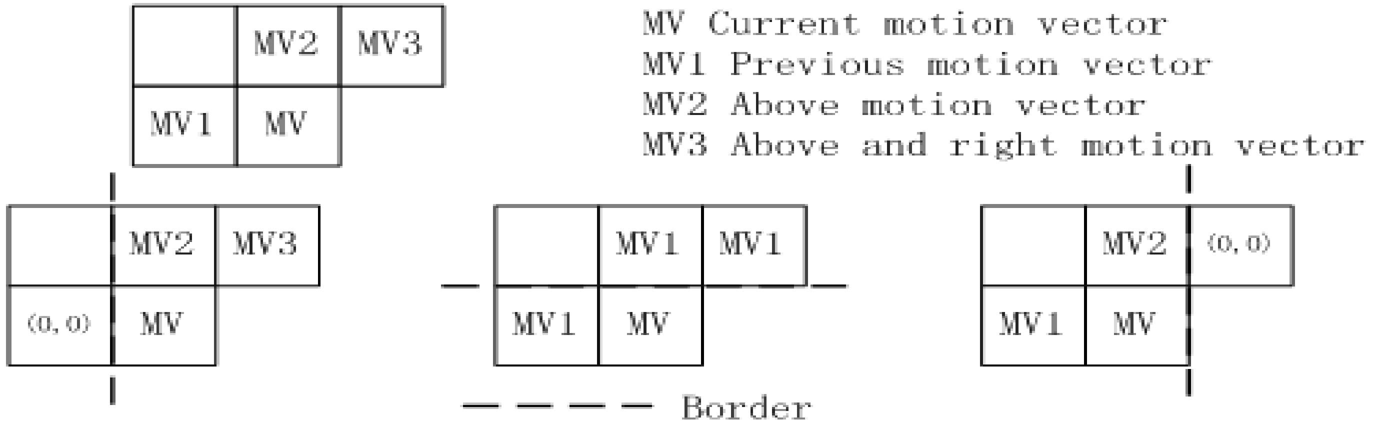
- **An improve standard for videoconferencing and other audiovisual services on PSTN, adopted by ITU-T Study Group 15 in 1995**
- **H.263 supports sub-QCIF, 4CIF和16CIF, GOBs don't have a fixed size.**

Video format	Luminance Image Resolution	Chrominance Image Resolution	H.263 support	Bit-rate (Mbit/s) (if uncompressed, 30 fps)	Bitrate(kbps) BPPmaxKb (compressed)
SQCIF	128 x 96	64 x 48	Required	4.4	64
QCIF	176 x 144	88 x 72	Required	9.1	64
CIF	352 x 288	176 x 144	Optional	36.5	256
4CIF	704 x 576	352 x 288	Optional	146.0	512
16CIF	1408 x 1152	704 x 576	Optional	583.9	1024

5.2 Motion Compensation in H.263

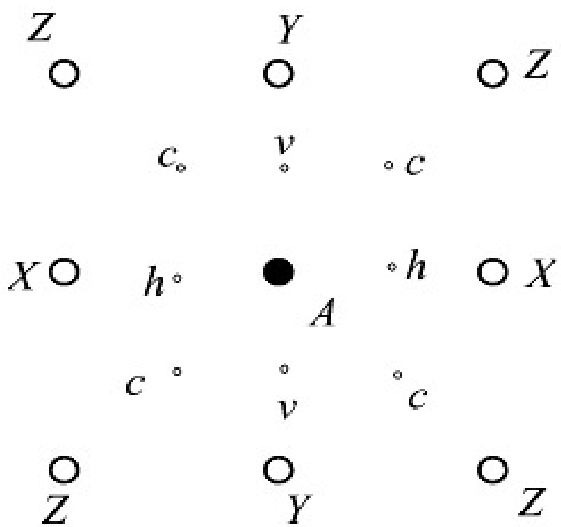
- H.263 supports **half-pixel positions** which are generated by bilinear interpolation method.
- MV are predicted by median values

$$u_p = median(u_1, u_2, u_3),$$
$$v_p = median(v_1, v_2, v_3)$$



5.2 Motion Compensation in H.263

- **Prediction at half pixel precision**
  - **Use inner-interpolation to get half pixel vales**
  - **Can reduce predicted errors**



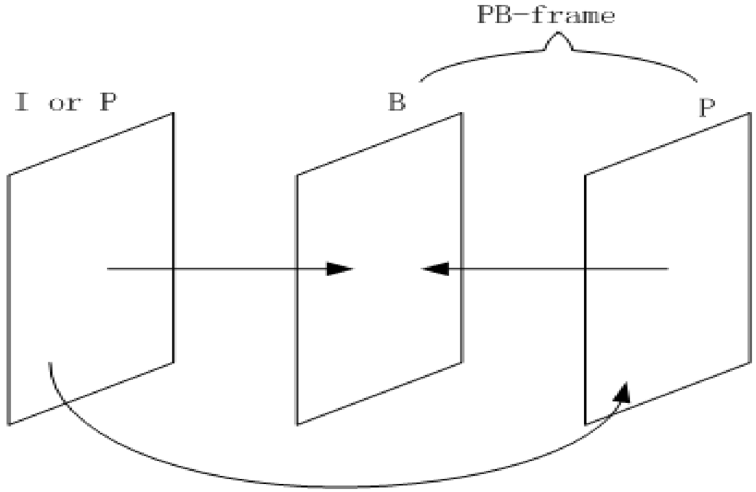
$$h = \frac{A+X}{2}$$

$$v = \frac{A+Y}{2}$$

$$c = \frac{A+X+Y+Z}{4}$$

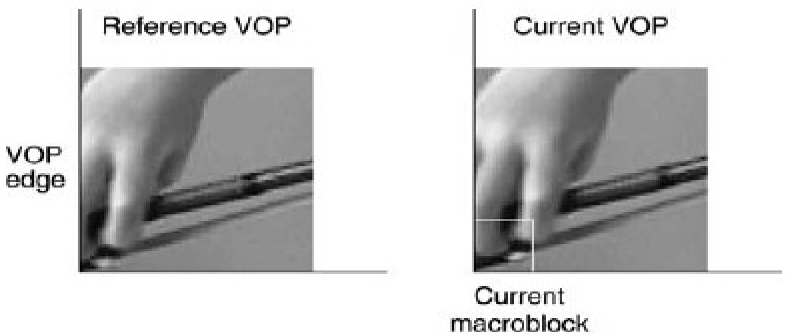
5.3 Optional H.263 Coding Modes

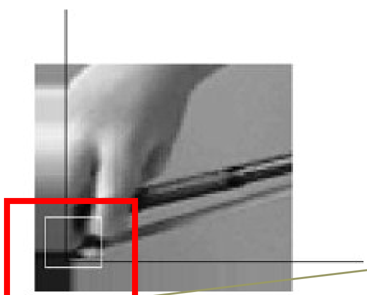
- **Negotiable options** besides its core algorithm:
  - Unrestricted motion vector mode.
  - Syntax-based arithmetic coding mode
  - Advanced prediction mode (4 MV for a macroblock）
  - PB-frames mode

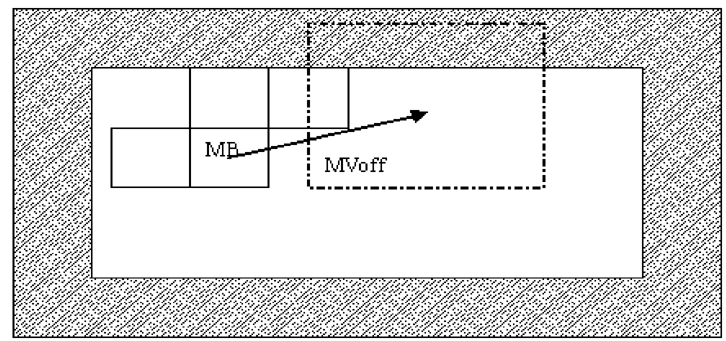


5.3 Optional H.263 Coding Modes

- Unrestricted Motion Vectors
  - Referenced not restricted by image boundary
- Fulfilled by enlarging the coding image edges for a certain size(e.g. macroblock size)







The best matching VOP is outside

5.4 H.263+ and H.263++

- Second version of H.263
  - Redefine the unrestricted motion vector mode
  - Slice structure is used to replace GOB
  - Implements Temporal,SNR,and Spatial scalabilities
  - Supports improved PB-frames mode
  - Use deblocking filtersto reduce blocking effects
- H.263++: the new extension
  - Enhanced reference picture selection（ERPS）
  - Data partition slice (DPS)
  - Additional supplemental enhancement information

Network & Media Laboratory

The end!

Thanks!