

Introduction to Video Compression Techniques

Slides courtesy of Tay Vaughan - "Making Multimedia Work"

Agenda

- Video Compression Overview
- Motivation for creating standards
- What do the standards specify
- Brief review of video compression
- Current video compression standards H.261, H.263, MPEG-1-2-4
- Advanced Video Compression Standards,
 - H.264, VC1, AVS

Video Compression Overview

● Problem:

- Raw video contains an immense amount of data.
- Communication and storage capabilities are limited and expensive.

● Example HDTV video signal:

- 720x1280 pixels/frame, progressive scanning at 60 frames/s:

$$\left(\frac{720 \times 1280 \text{ pixels}}{\text{frame}} \right) \left(\frac{60 \text{ frames}}{\text{sec}} \right) \left(\frac{3 \text{ colors}}{\text{pixel}} \right) \left(\frac{8 \text{ bits}}{\text{color}} \right) = 1.3 \text{ Gb/s}$$

- 20 Mb/s HDTV channel bandwidth

→ Requires compression by a factor of 70 (equivalent to .35 bits/pixel)

Video Compression: Why?

● Bandwidth Reduction

Application	Data Rate	
	Uncompressed	Compressed
Video Conference <i>352 x 240 @ 15 fps</i>	30.4 Mbps	64 - 768 kbps
CD-ROM Digital Video <i>352 x 240 @ 30 fps</i>	60.8 Mbps	1.5 - 4 Mbps
Broadcast Video <i>720 x 480 @ 30 fps</i>	248.8 Mbps	3 - 8 Mbps
HDTV <i>1280 x 720 @ 60 fps</i>	1.33 Gbps	20 Mbps

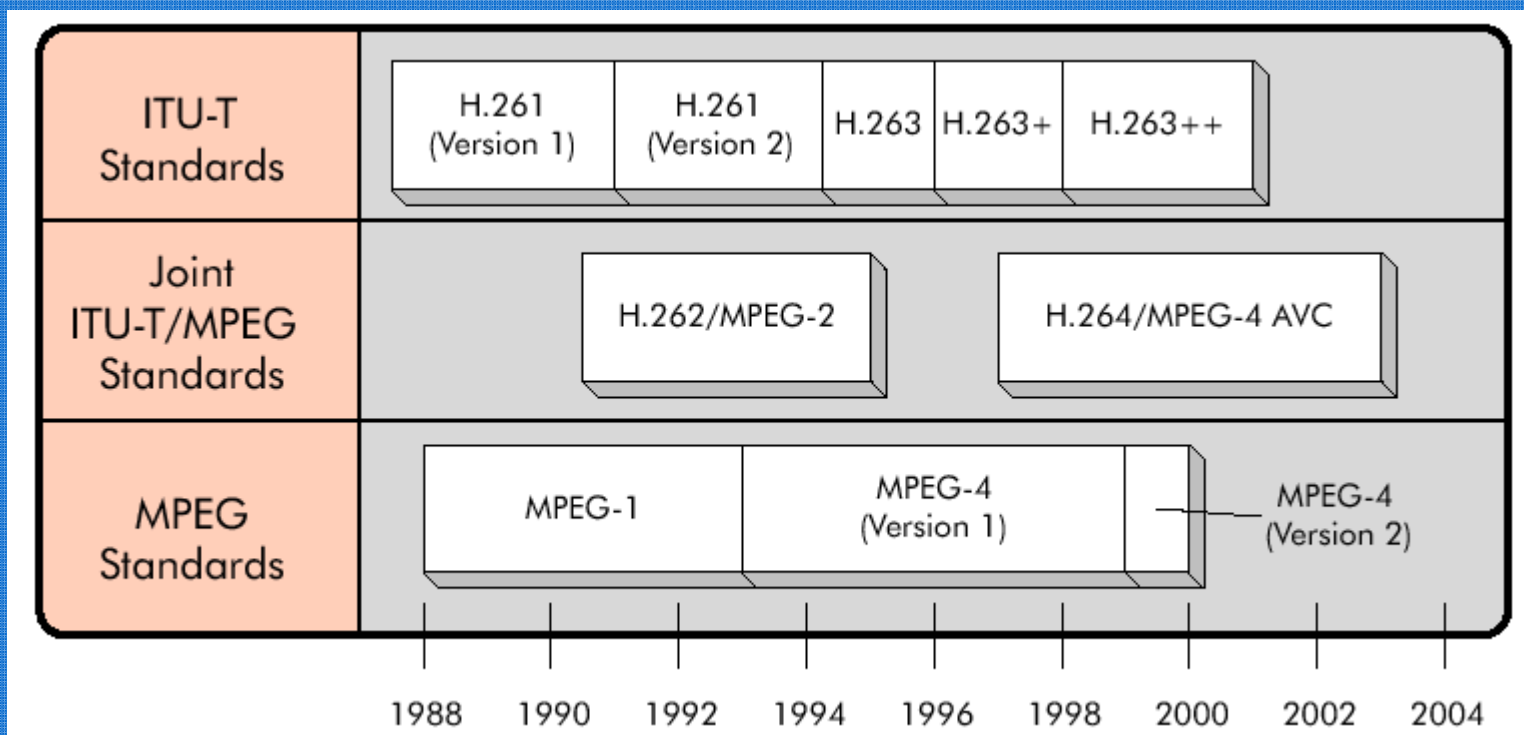
Video Compression Standards

STANDARD	APPLICATION	BIT RATE
JPEG	Continuous-tone still-image compression	Variable
H.261	Video telephony and teleconferencing over ISDN	p x 64 kb/s
MPEG-1	Video on digital storage media (CD-ROM)	1.5 Mb/s
MPEG-2	Digital Television	> 2 Mb/s
H.263	Video telephony over PSTN	< 33.6 kb/s
MPEG-4	Object-based coding, synthetic content, interactivity	Variable
H.264	From Low bitrate coding to HD encoding, HD-DVD, Surveillance, Video conferencing.	Variable

Motivation for Standards

- Goal of standards:
 - Ensuring interoperability – Enabling communication between devices made by different manufacturers.
 - Promoting a technology or industry.
 - Reducing costs.

History of Video Standards

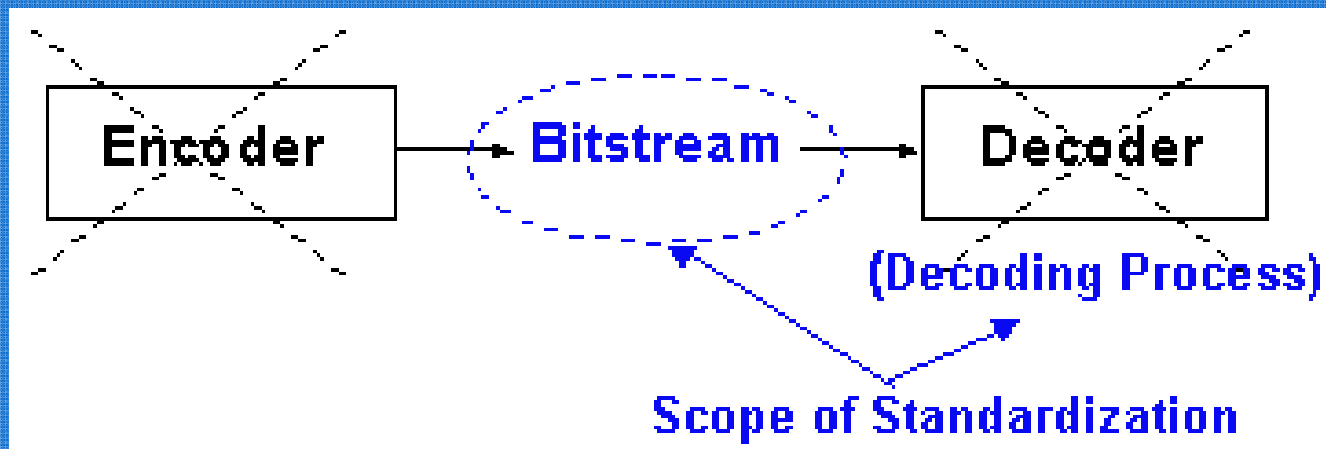


What Do the Standards Specify?

- A video compression system consists of the following:
 - An encoder
 - Compressed bit-streams
 - A decoder
- What parts of the system do the standards specify?

What Do the Standards Specify?

- Not the encoder, not the decoder.



What Do the Standards Specify?

- Just the bit-stream syntax and the decoding process, for example it tells to use IDCT, but not how to implement the IDCT.
- Enables improved encoding and decoding strategies to be employed in a standard-compatible manner.

Achieving Compression

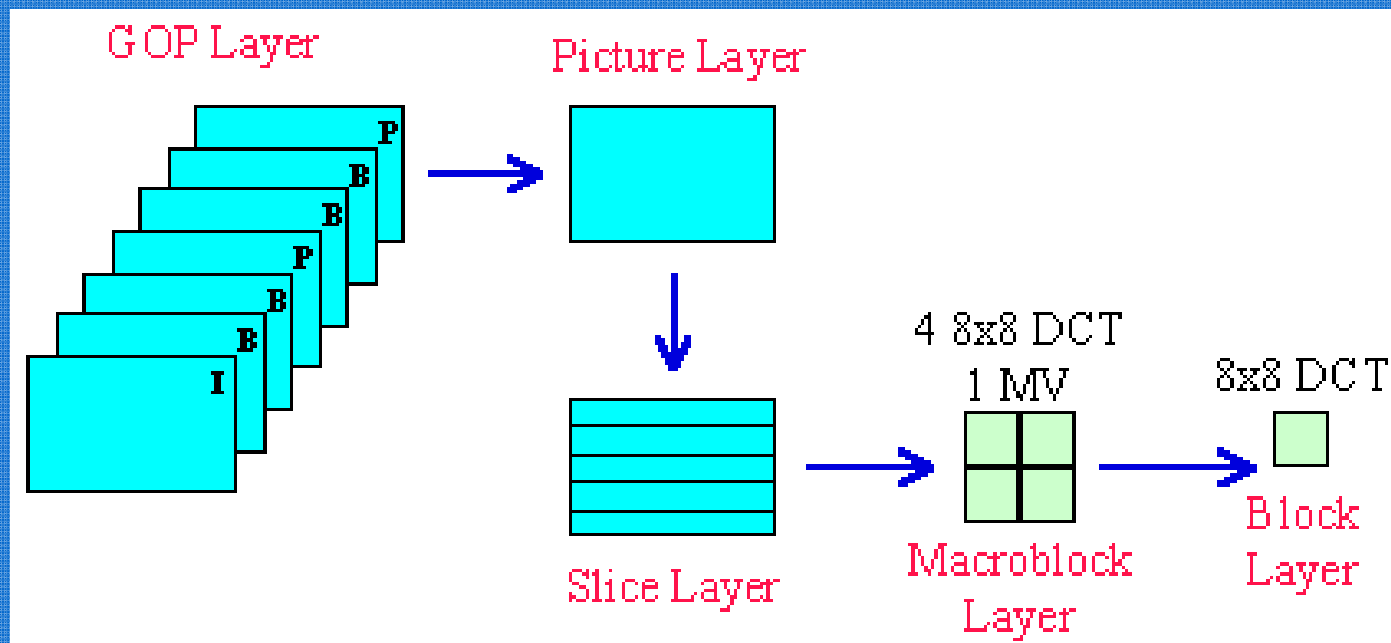
- Reduce redundancy and irrelevancy.
- Sources of redundancy:
 - Temporal – Adjacent frames highly correlated.
 - Spatial – Nearby pixels are often correlated with each other.
 - Color space – RGB components are correlated among themselves.
- Irrelevancy – Perceptually unimportant information.

Basic Video Compression Architecture

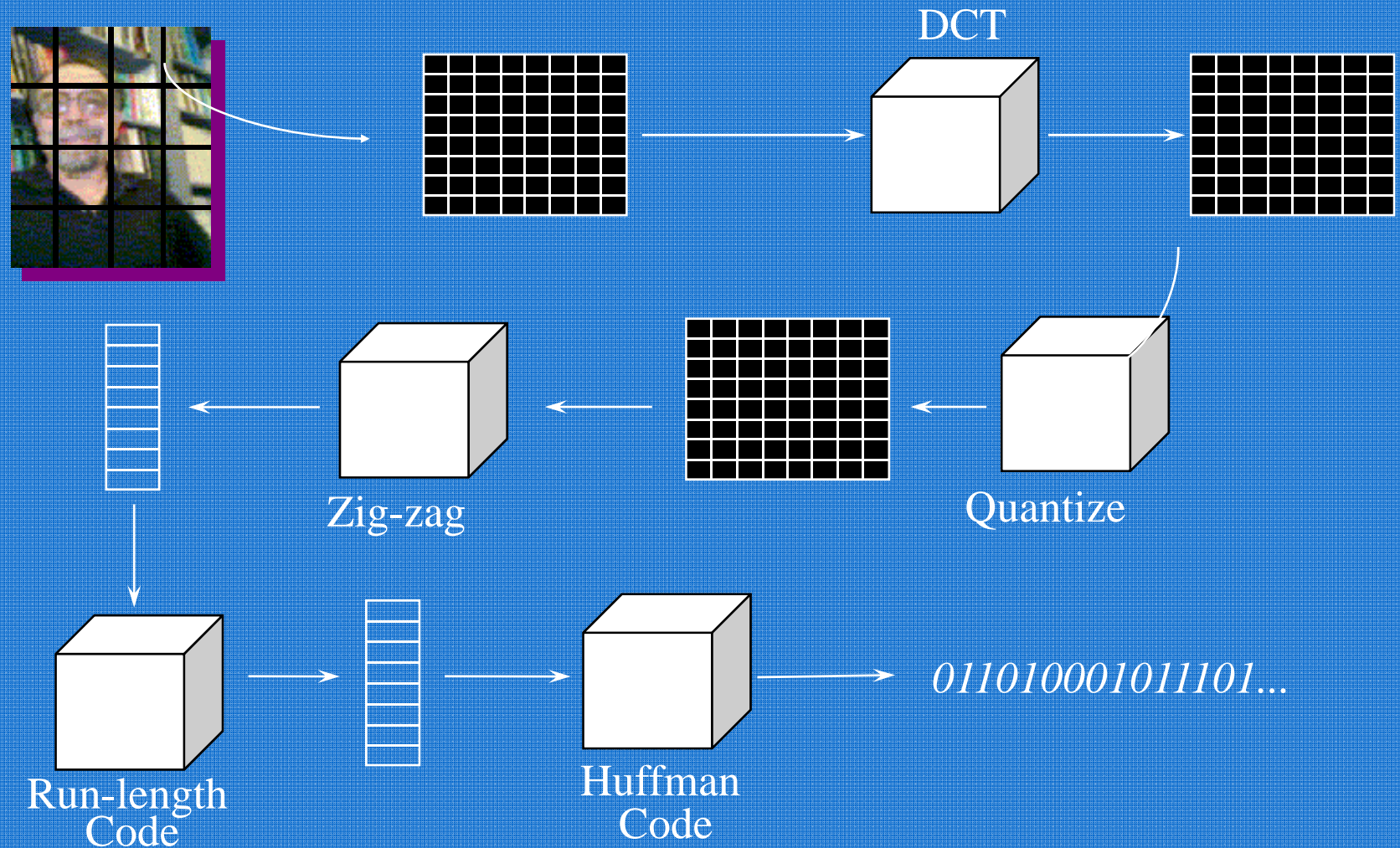
- Exploiting the redundancies
 - Temporal – MC-prediction and MC-interpolation
 - Spatial – Block DCT
 - Color – Color space conversion
- Scalar quantization of DCT coefficients
- Run-length and Huffman coding of the non-zero quantized DCT coefficients

Video Structure

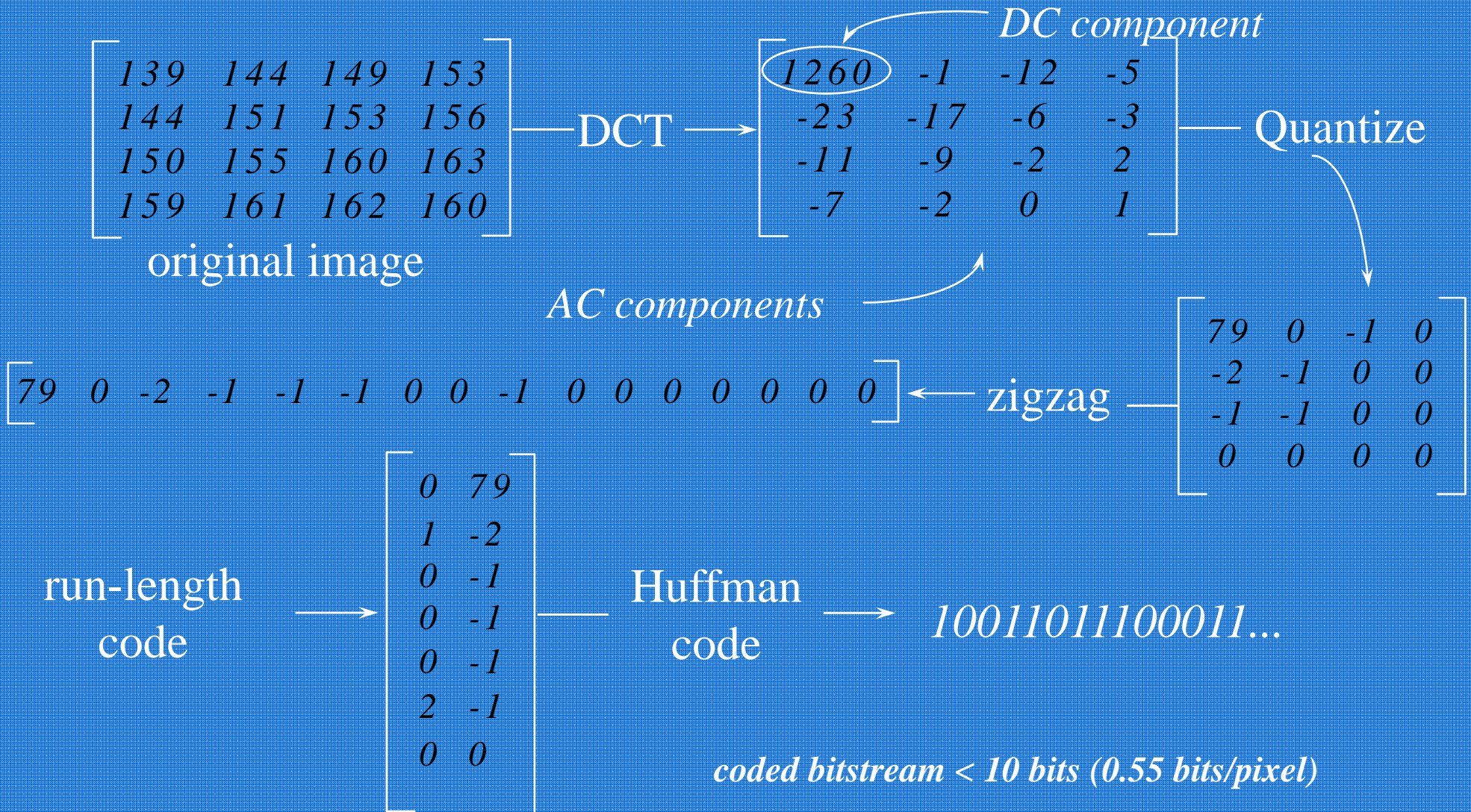
● MPEG Structure



Block Transform Encoding



Block Encoding



Result of Coding/decoding

$$\begin{bmatrix} 139 & 144 & 149 & 153 \\ 144 & 151 & 153 & 156 \\ 150 & 155 & 160 & 163 \\ 159 & 161 & 162 & 160 \end{bmatrix}$$

original block

$$\begin{bmatrix} 144 & 146 & 149 & 152 \\ 156 & 150 & 152 & 154 \\ 155 & 156 & 157 & 158 \\ 160 & 161 & 161 & 162 \end{bmatrix}$$

reconstructed block

$$\begin{bmatrix} -5 & -2 & 0 & 1 \\ -4 & 1 & 1 & 2 \\ -5 & -1 & 3 & 5 \\ -1 & 0 & 1 & -2 \end{bmatrix}$$

errors

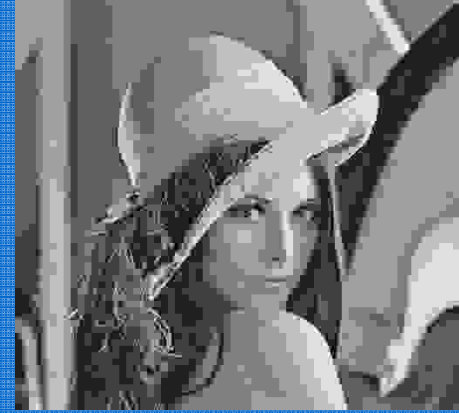
Examples



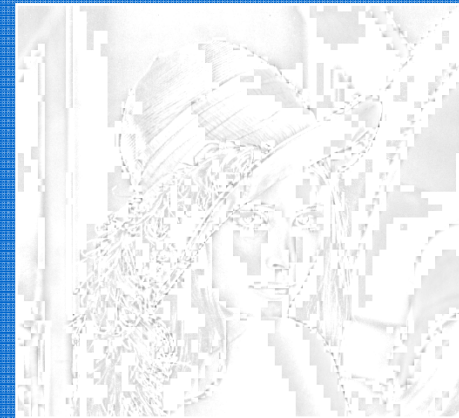
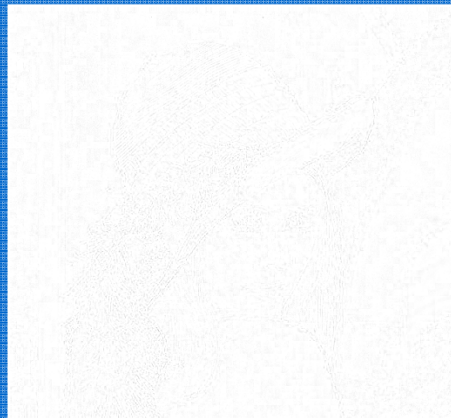
Uncompressed
(262 KB)



Compressed (50)
(22 KB, 12:1)

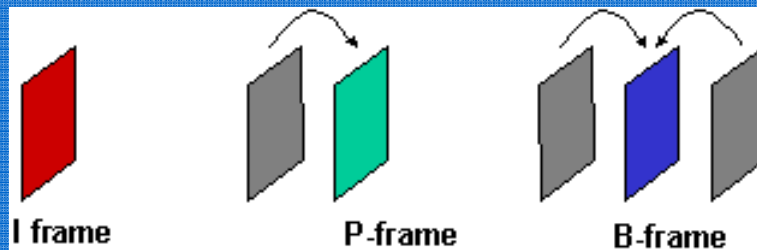


Compressed (1)
(6 KB, 43:1)



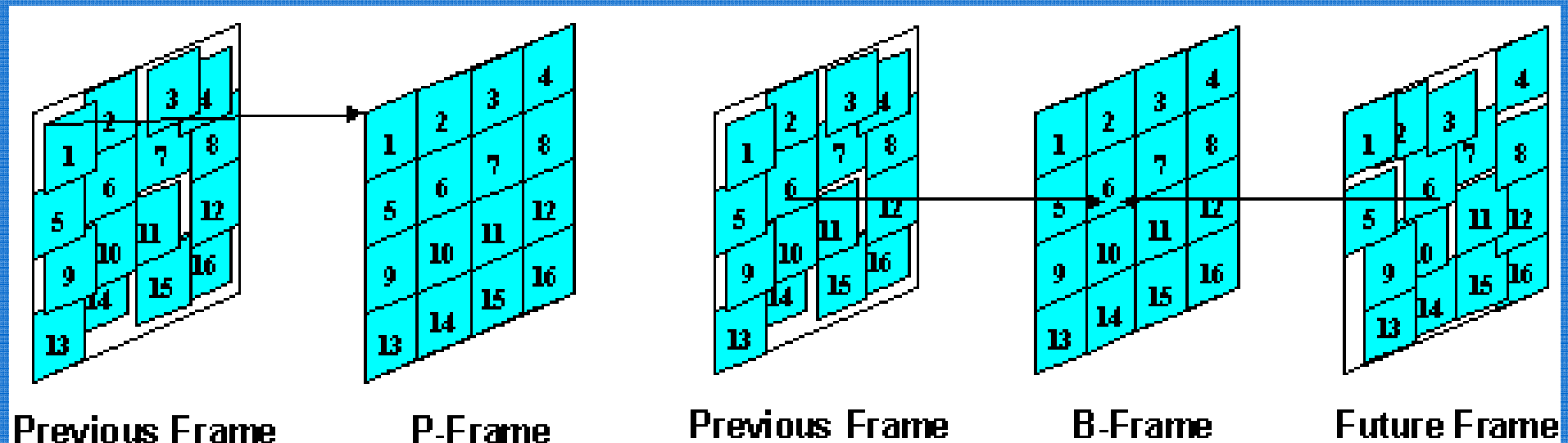
Video Compression

- Main addition over image compression
 - Exploit the temporal redundancy
- Predict current frame based on previously coded frames
- Types of coded frames:
 - I-frame – Intra-coded frame, coded independently of all other frames
 - P-frame – Predictively coded frame, coded based on previously coded frame
 - B-frame – Bi-directionally predicted frame, coded based on both previous and future coded frames



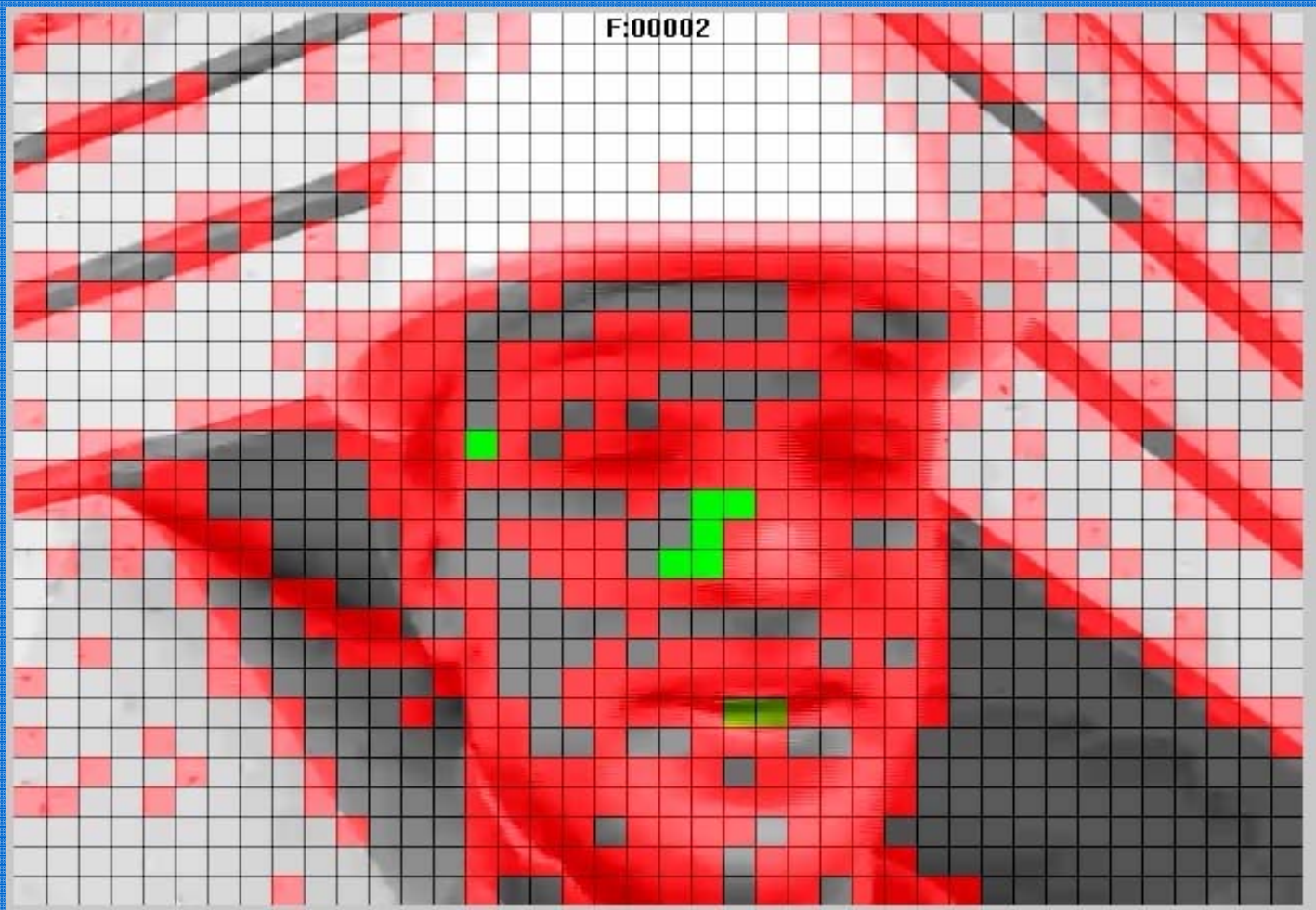
Motion Compensated Prediction (P and B Frames)

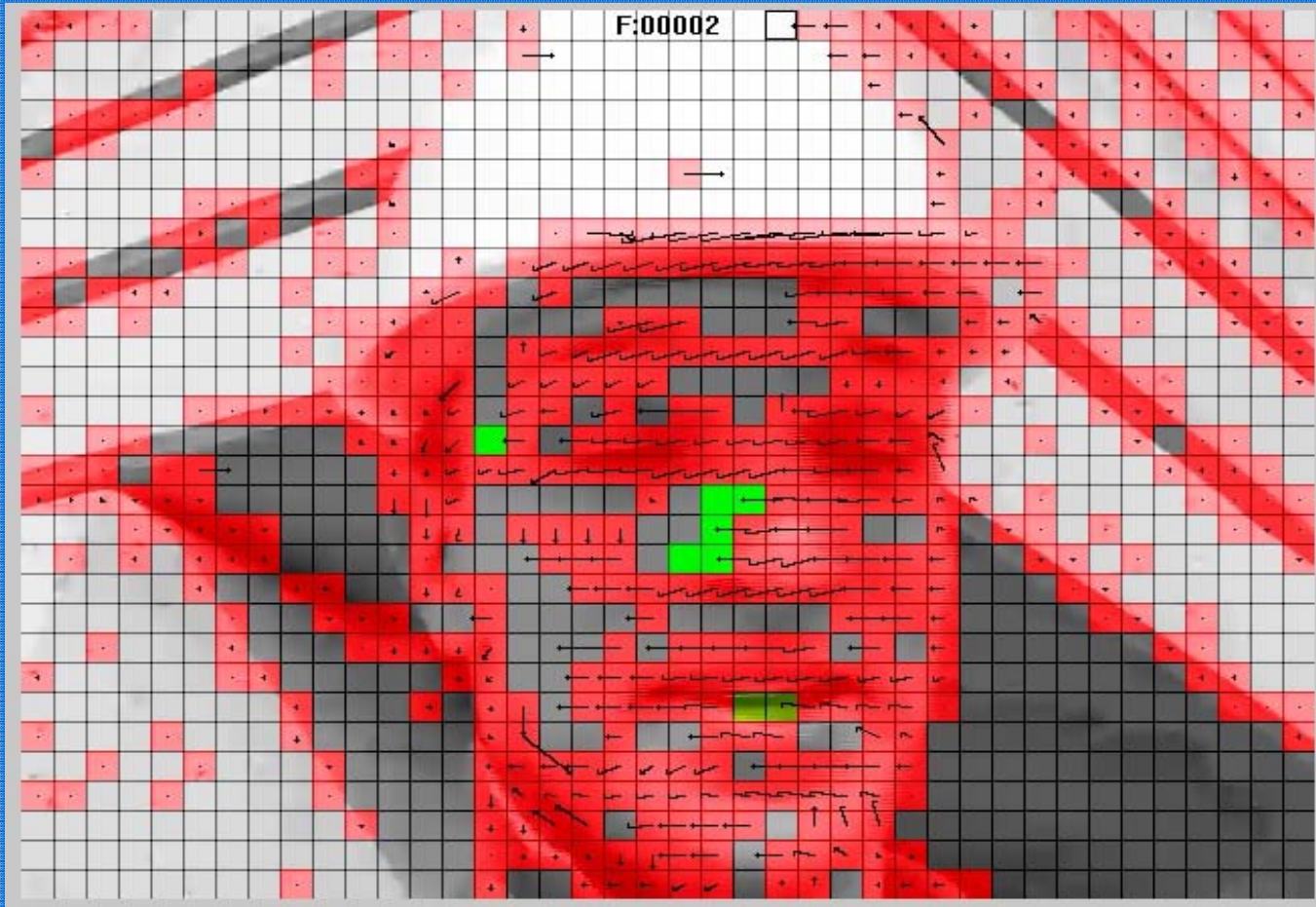
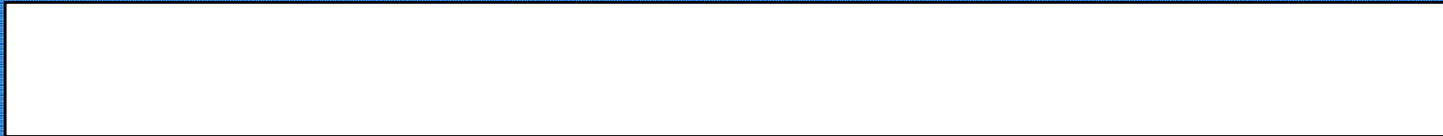
- Motion compensated prediction – predict the current frame based on a reference frame while compensating for the motion
- Examples of block-based motion-compensated prediction for P-frames and B-frames.



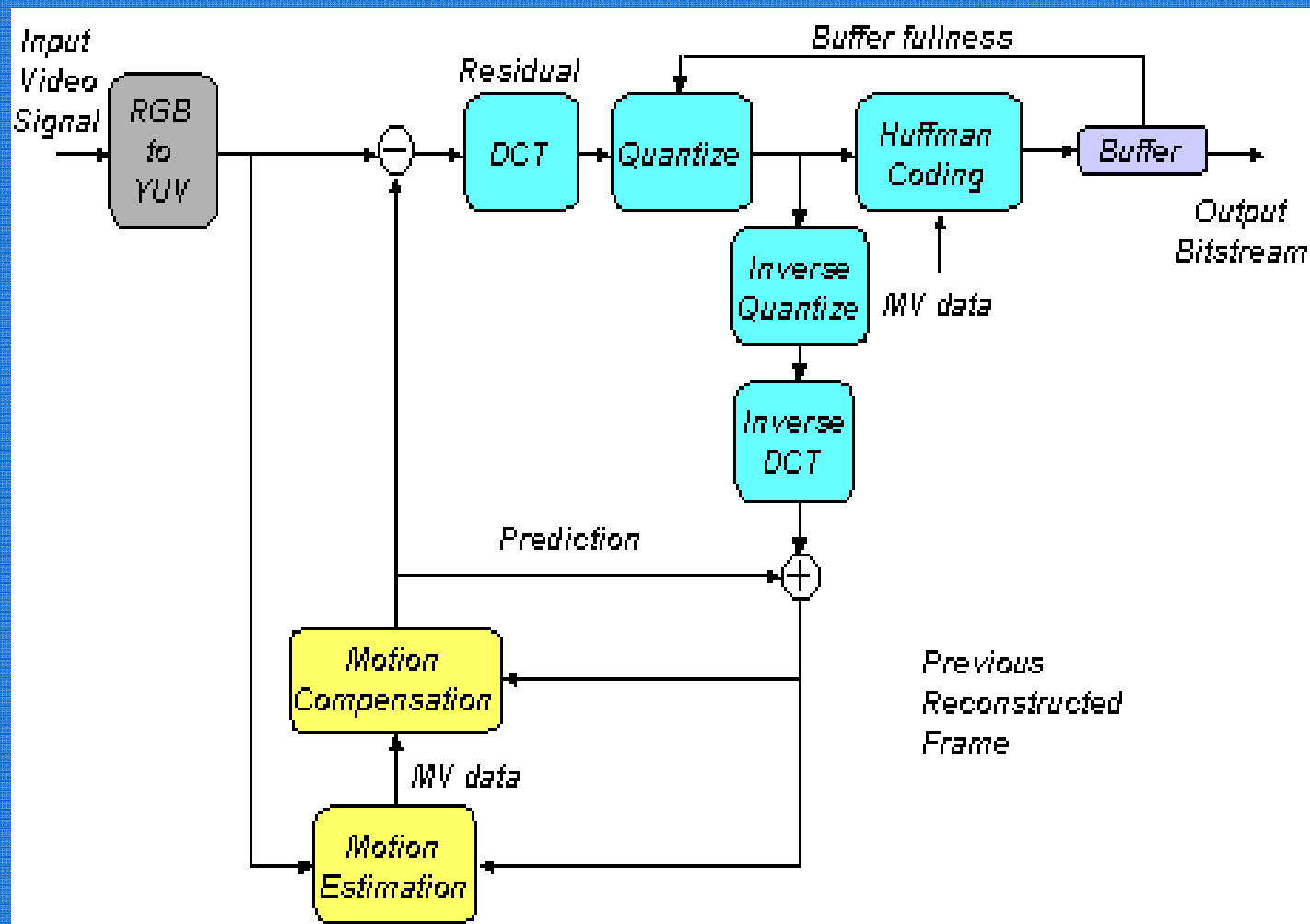
- Video coding is fun !!



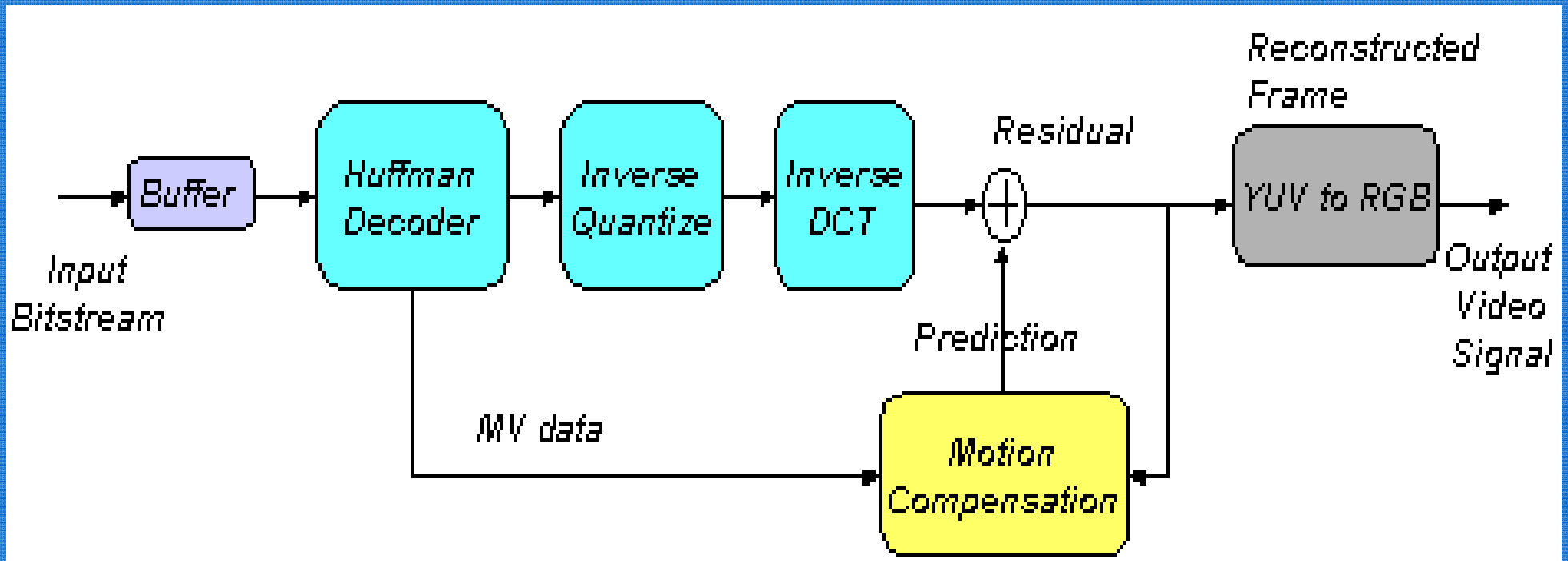




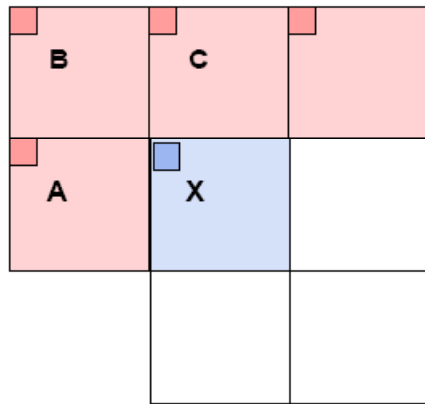
Example Video Encoder



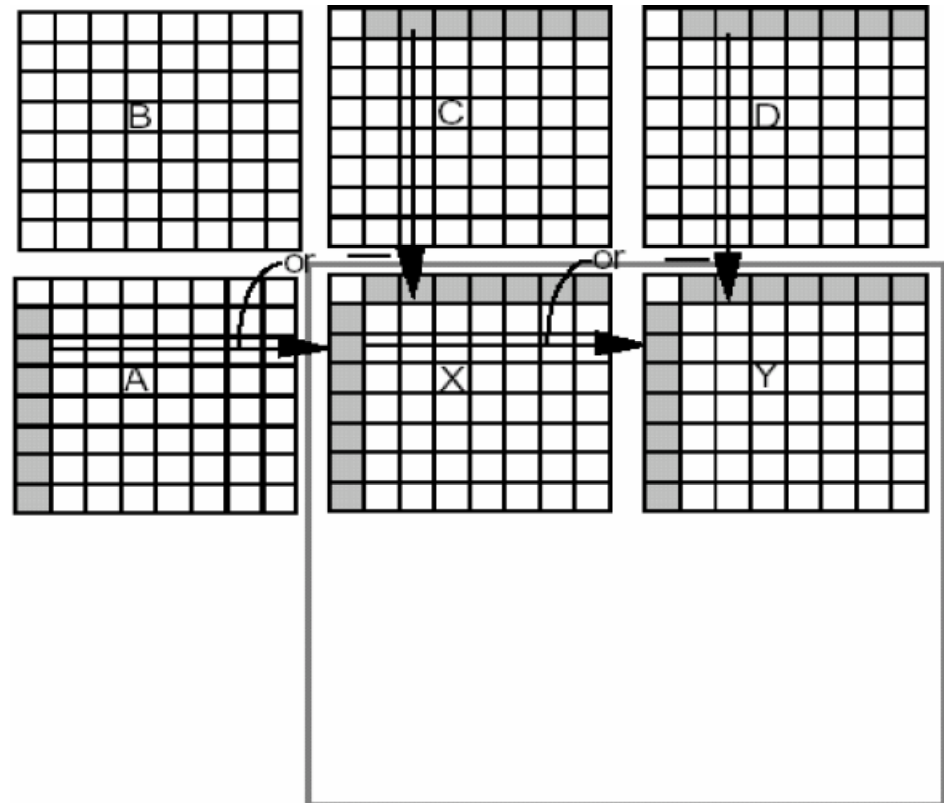
Example Video Decoder



AC/DC prediction for Intra Coding



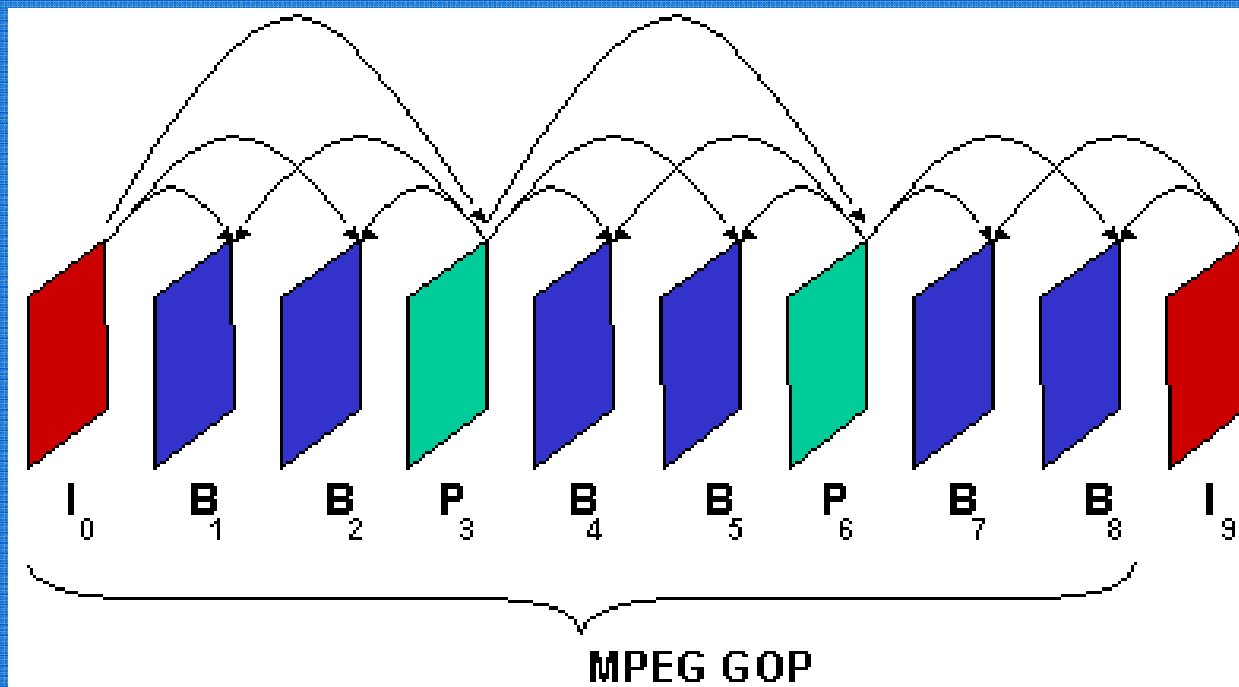
DC-Prediction



AC-Prediction

Group of Pictures (GOP) Structure

- Enables random access into the coded bit-stream.
- Number of B frames and impact on search range.



Current Video Compression Standards

- Classification & Characterization of different standards
 - Based on the same fundamental building blocks
 - Motion-compensated prediction and interpolation
 - 2-D Discrete Cosine Transform (DCT)
 - Color space conversion
 - Scalar quantization, run-length, and Huffman coding
 - Other tools added for different applications
 - Progressive or interlaced video
 - Improved compression, error resilience, scalability, etc

H.261 (1990)

- Goal: real-time, two-way video communication
- Key features
 - Low delay (150 ms)
 - Low bit rates (p x 64 kb/s)
- Technical details
 - Uses I- and P-frames (no B-frames)
 - Full-pixel motion estimation
 - Search range +/- 15 pixels
 - Low-pass filter in the feedback loop

H.263 (1995)

- Goal: communication over conventional analog telephone lines (< 33.6 kb/s)
- Enhancements to H.261
 - Reduced overhead information
 - Improved error resilience features
 - Algorithmic enhancements
 - Half-pixel motion estimation with larger motion search range
 - Four advanced coding modes
 - Unrestricted motion vector mode
 - Advanced prediction mode (median MV predictor using 3 neighbors)
 - PB-frame mode
 - OBMC

MPEG-1 and MPEG-2

● MPEG-1 (1991)

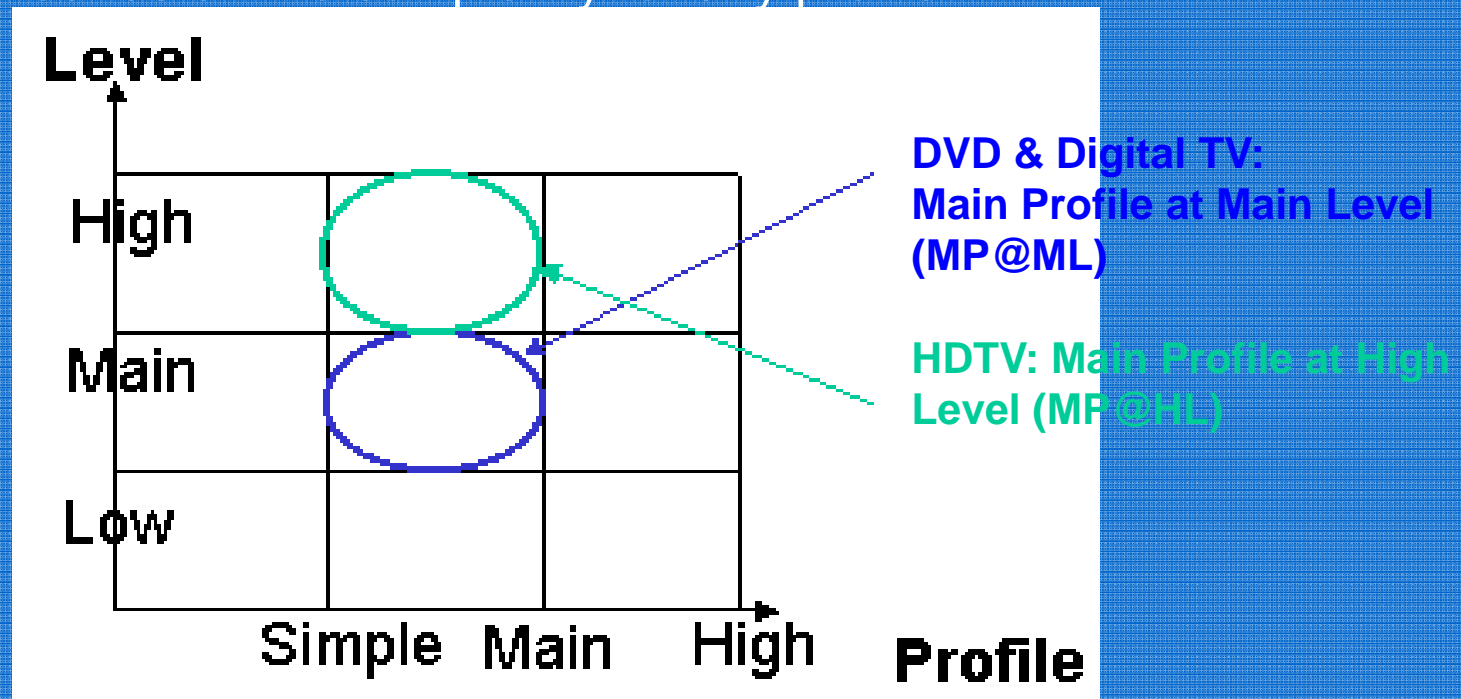
- Goal is compression for digital storage media, CD-ROM
- Achieves VHS quality video and audio at ~1.5 Mb/sec ??

● MPEG-2 (1993)

- Superset of MPEG-1 to support higher bit rates, higher resolutions, and interlaced pictures
- Original goal to support interlaced video from conventional television. Eventually extended to support HDTV
- Provides field-based coding and scalability tools

MPEG-2 Profiles and Levels

- Goal: to enable more efficient implementations for different applications.
- Profile – Subset of the tools applicable for a family of applications.
- Level – Bounds on the complexity for any profile.



MPEG-4 (1993)

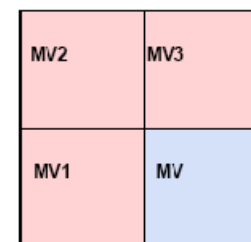
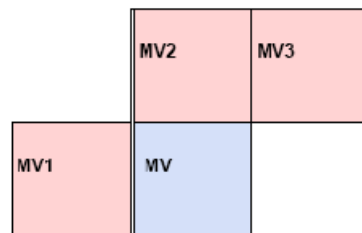
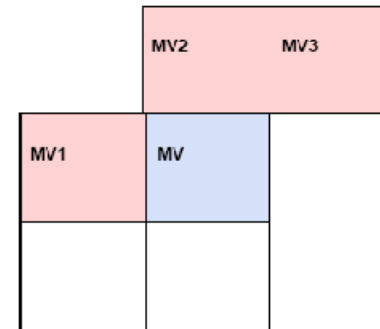
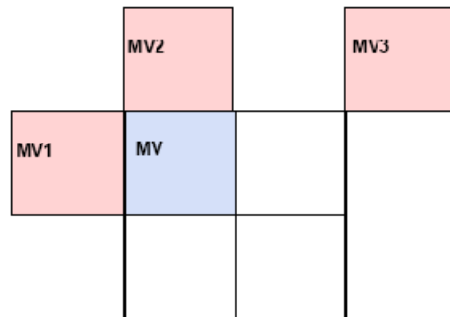
- Primary goals: new functionalities, **not better compression**
 - Object-based or content-based representation –
 - Separate coding of individual visual objects
 - Content-based access and manipulation
 - Integration of natural and synthetic objects
 - Interactivity
 - Communication over error-prone environments
 - Includes frame-based coding techniques from earlier standards

MV Prediction- MPEG-4

● The predicted MV (P_x, P_y) for example in MPEG-4 is computed as follows

● $P_x = \text{Median}(MV1_x, MV2_x, MV3_x)$

● $P_y = \text{Median}(MV1_y, MV2_y, MV3_y)$



Comparing MPEG-1/2 and H.261/3 With MPEG-4

- MPEG-1/2 and H.261/H.263 – Algorithms for compression –
 - Basically describe a pipe for storage or transmission
 - Frame-based
 - Emphasis on hardware implementation
- MPEG-4 – Set of tools for a variety of applications –
 - Define tools and glue to put them together
 - Object-based and frame-based
 - Emphasis on software
 - Downloadable algorithms, not encoders or decoders

MPEG-1 video vs H.261

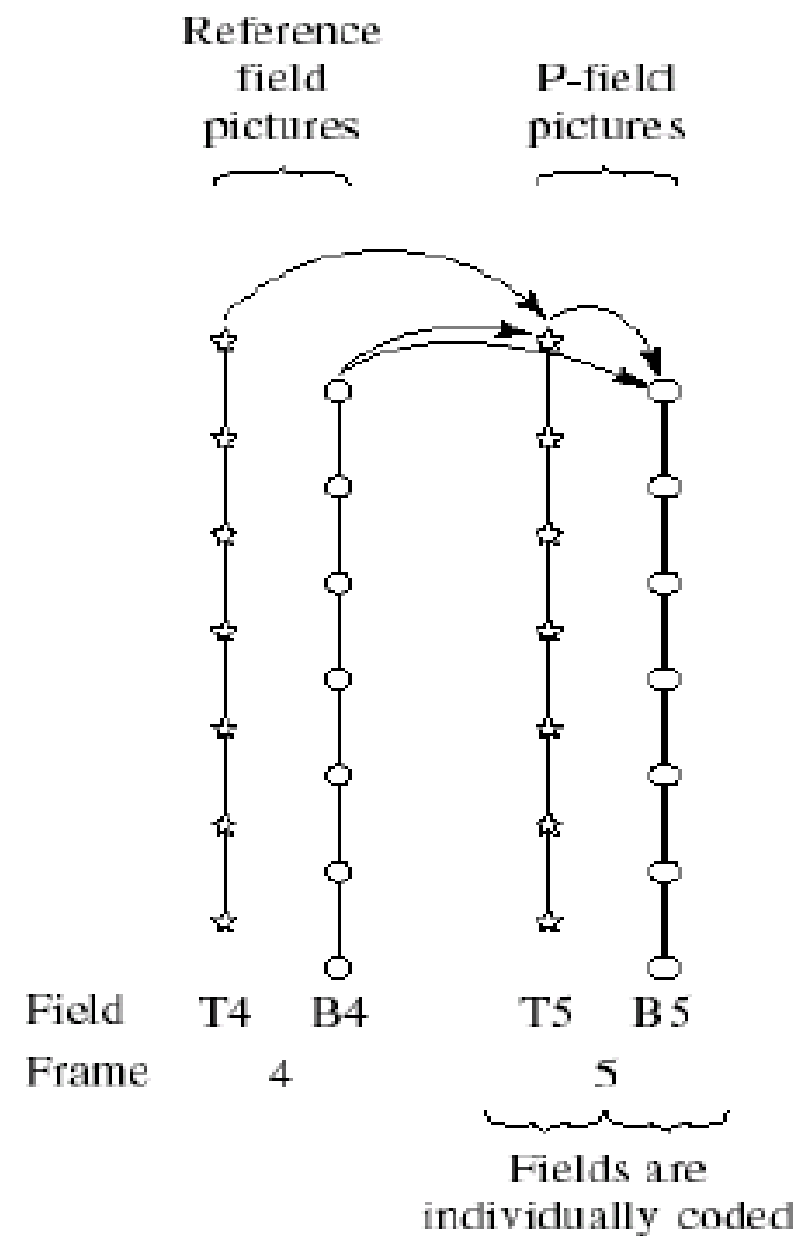
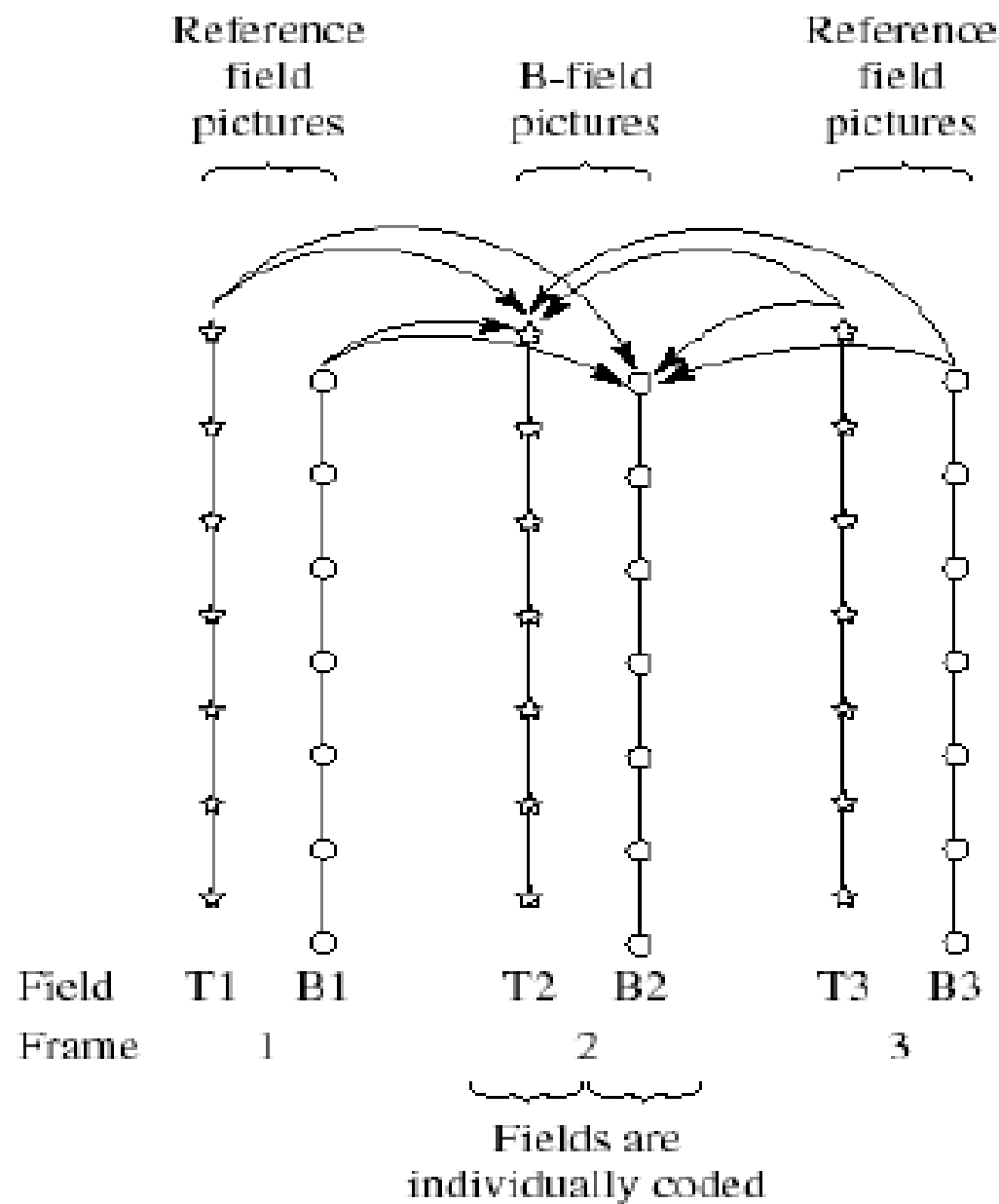
- Half-pel accuracy motion estimation, range up to +/- 64
- Using bi-directional temporal prediction
- Important for handling uncovered regions
- Using perceptual-based quantization matrix for I-blocks (same as JPEG)
- DC coefficients coded predictively

MPEG-2 : MC for Interlaced Video

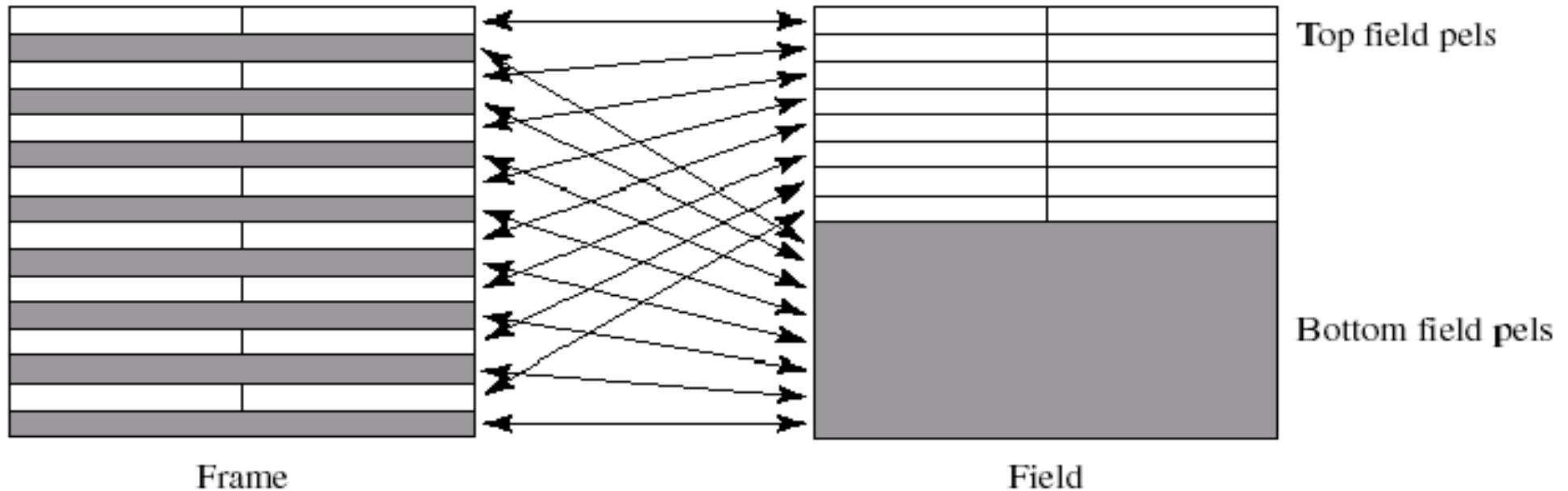
- Field prediction for field pictures
- Field prediction for frame pictures
- Dual prime for P-pictures
- 16x8 MC for field pictures

Field prediction for field pictures

- Each field is predicted individually from the reference fields
- A P-field is predicted from one previous field
- A B-field is predicted from two fields chosen from two reference pictures



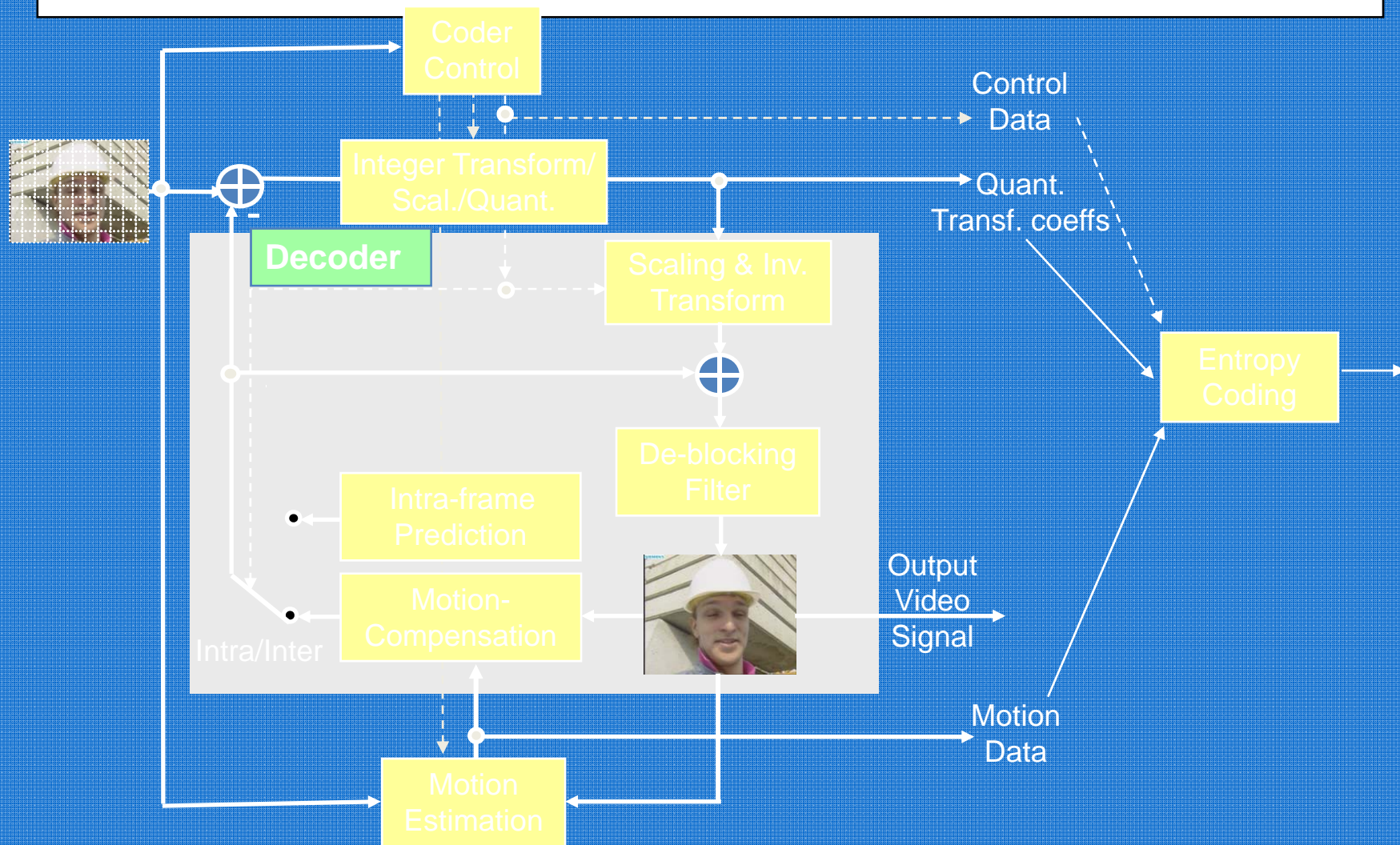
Field Prediction for Frame Pictures



- Field prediction for frame pictures : the MB to be predicted is split into top field pels and bottom field pels. Each 16x8 field block is predicted separately with its own motion vectors (P-frame) or two motion vectors (B-frame)

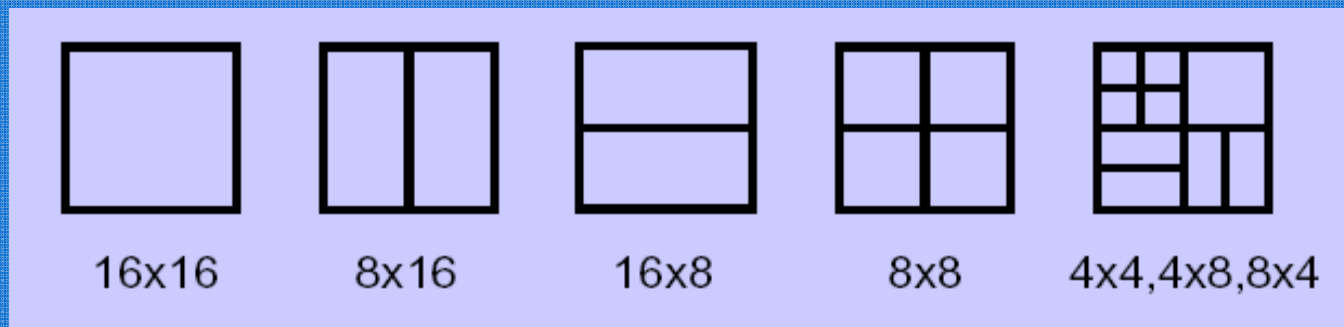
Advanced Video Coding Standard, H.264

- Common elements with other standards
- Macroblocks: 16x16 luma + 2 x 8x8 chroma samples
- Input: association of luma and chroma and conventional
- Sub-sampling of chroma (4:2:0)
- Block motion displacement
- Motion vectors over picture boundaries
- Variable block-size motion
- Block transforms
- Scalar quantization
- I, P and B picture coding types



H.264

- New elements introduced
- Every macroblock is split in one of 7 ways
 - Up to 16 mini-blocks (and as many MVs)



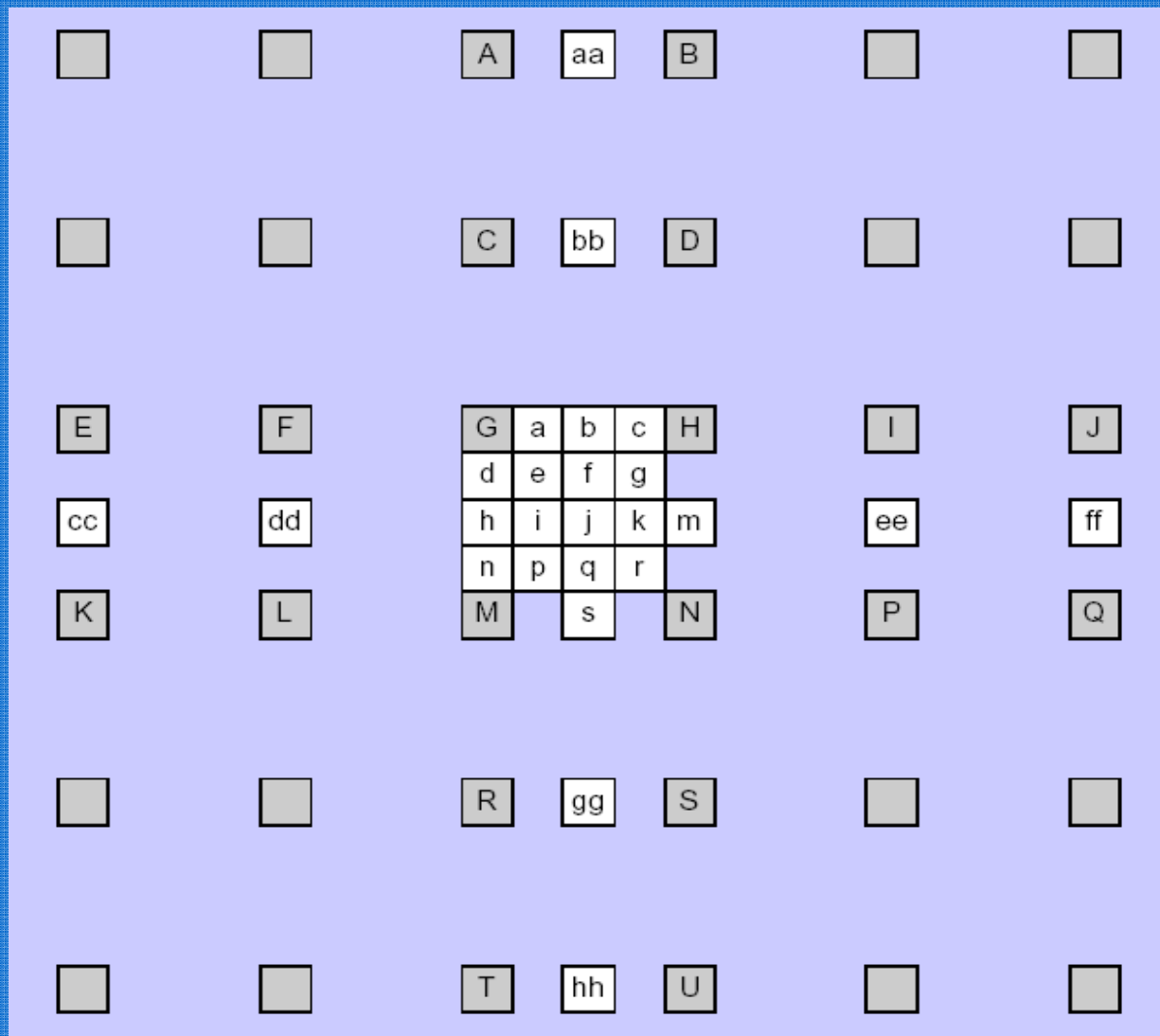
- Accuracy of motion compensation = $1/4$ pixel
- Multiple reference frames

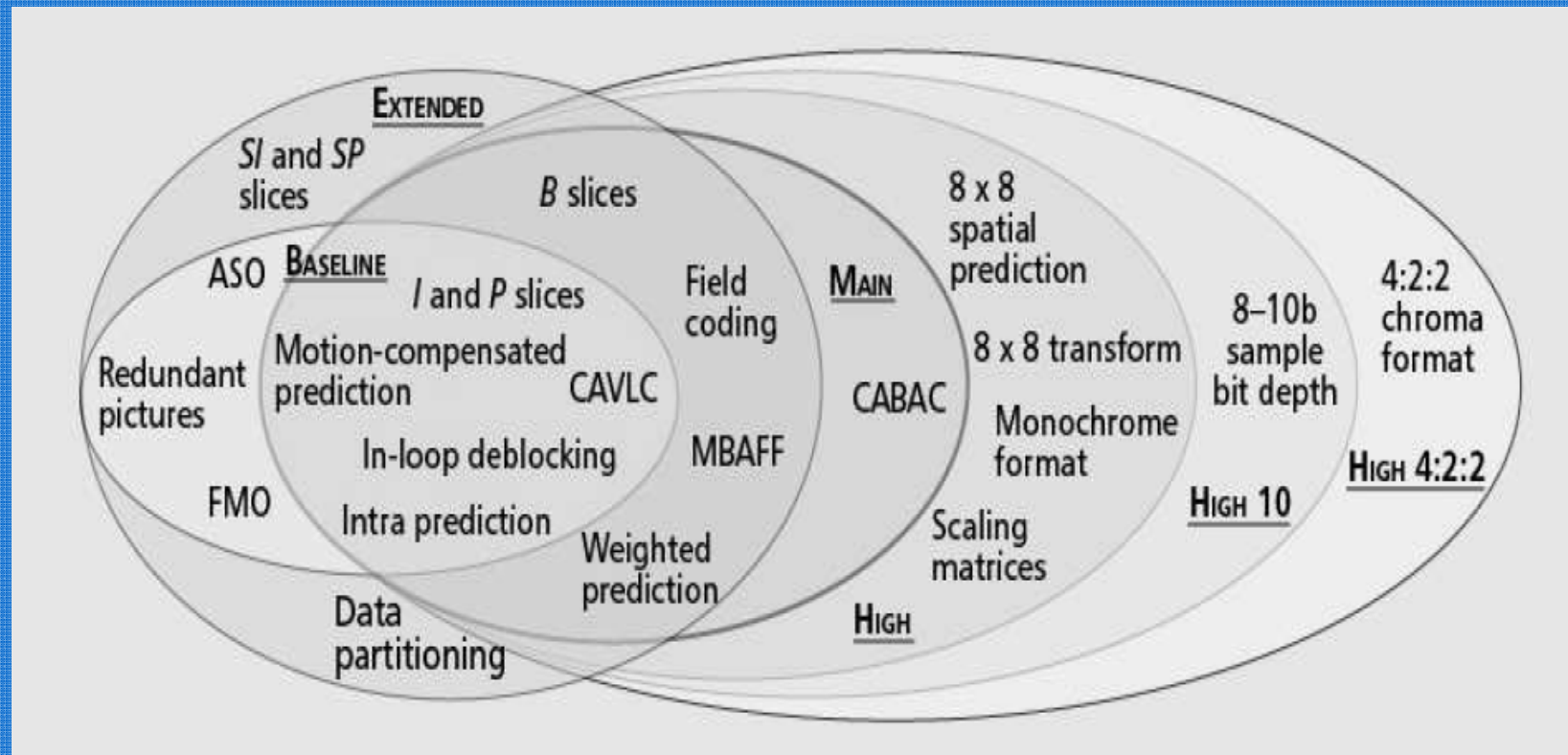
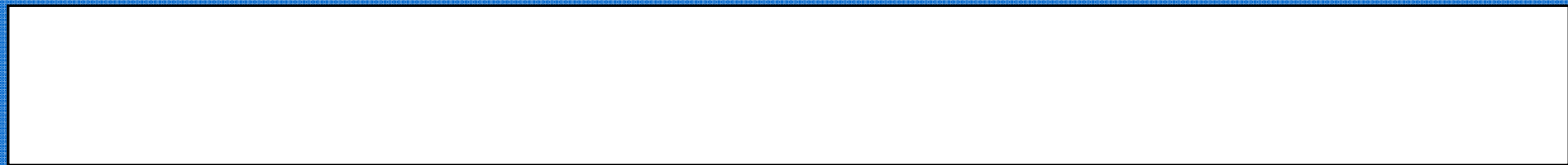
H.264

- Improved motion estimation
- De-blocking filter at estimation
- Integer 4x4 DCT approximation
 - Eliminates
 - Problem of mismatch between different implementation.
 - Problem of encoder/decoder drift.
- Arithmetic coding for MVs & coefficients.
- Compute SATD (Sum of Absolute Transformed Differences) instead of SAD.
 - Cost of transformed differences (i.e. residual coefficients) for 4x4 block using 4 x 4 Hadamard-Transformation

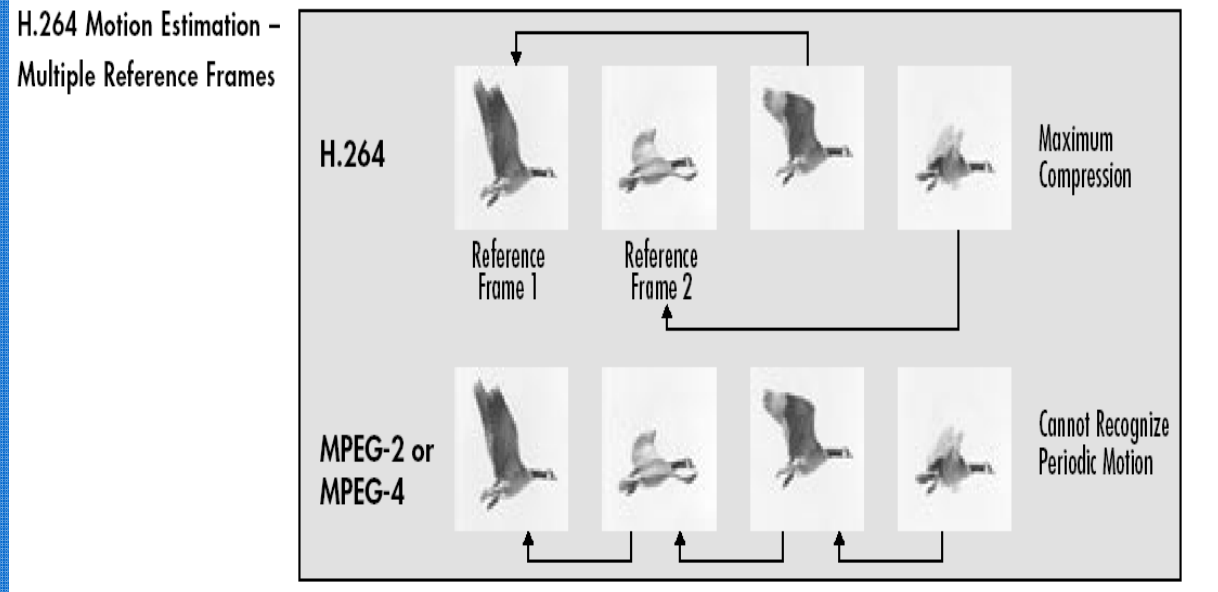
H.264/AVC

- Half sample positions are obtained by applying a 6-tap filter .
(1,-5,20,20,-5,1)
- Quarter sample positions are obtained by averaging samples at integer and half sample positions





Support for multiple reference pictures. It gives significant compression when motion is periodic in nature.



- PAFF (Picture adaptive frame/field)
 - Combine the two fields together and to code them as one single coded frame (frame mode).
 - Not combine the two fields and to code them as separate coded fields (field mode).
- MBAFF (Macro block adaptive frame/field)
 - The decision of field/frame happens at macro block pair level.

- Flexible macro block ordering
 - Picture can be partitioned into regions (slices)
 - Each region can be decoded independently.

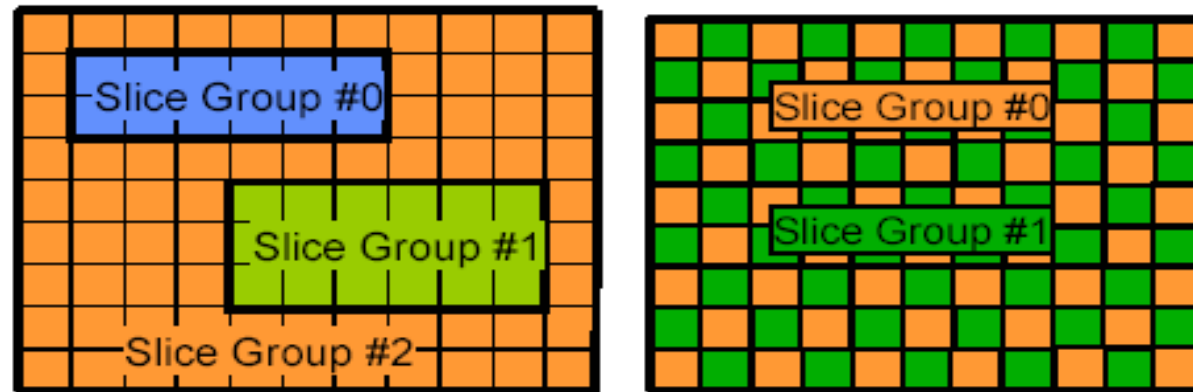

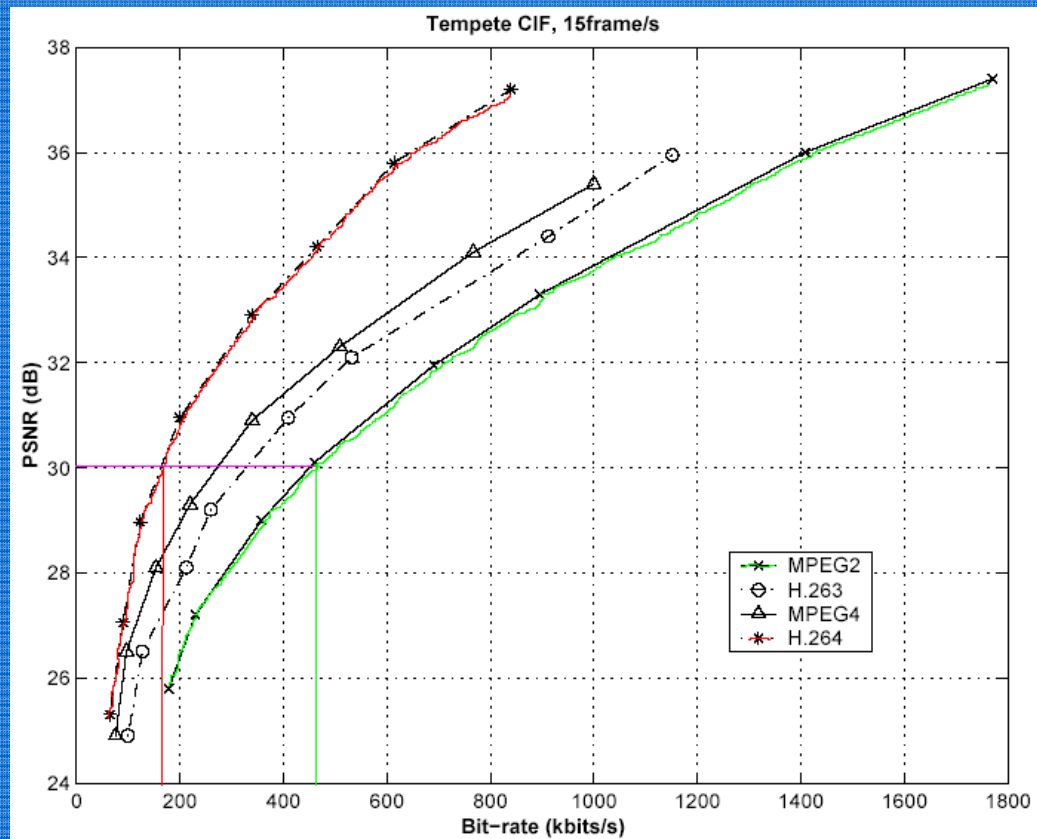


Fig. 7: Subdivision of a QCIF frame into slices when utilizing FMO.

- 
- Arbitrary slice ordering.
 - Since each slice can be decoded independently. It can be sent out of order
 - Redundant pictures
 - Encoder has the flexibility to send redundant pictures. These pictures can be used during loss of data.

Feature	MPEG4	WMV9	H.264
Prediction Block size	16*16, 8*8	16*16, 16*8, 8*8 , 4*4	4*4,4*8,8*4,8*8, 8*16,16*8,16*16
Intra Prediction	Ac Prediction (Transform Domain)	Ac Prediction (Transform Domain)	Intra Prediction (Spatial Domain)
Entropy coding	VLC	VLC	CAVLC,CABAC
Reference frame	One picture	Two (Interlace)	Multiple pictures
Weighted Prediction	No	No	Yes
De-blocking Filter	No (Optional)	Yes	Yes
Transform	8*8 DCT	4*4,4*8,8*4,8*8	4*4,8*8(High Profile) Integer DCT



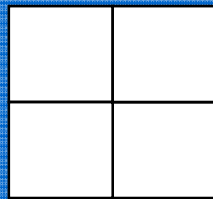
Baseline Profile

Spatial Domain Intra Prediction

- ◆ What is Spatial Domain Intra Prediction?
- ◆ New Approach to Prediction...
- ◆ Advantages of the spatial domain prediction...
- ◆ The Big Picture...
- ◆ Intra-Prediction Modes
- ◆ Implementation Challenges for Intra-Prediction

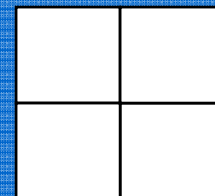
- Intra Prediction is a process of using the pixel data predicted from the neighboring blocks for the purpose of sending information regarding the current macro-block instead of the actual pixel data.

Current Block



Transform Engine

Top Neighbor



Transform Engine

Current Block

New approach to Prediction...

- The H.264/AVC uses a new approach to the prediction of intra blocks by doing the prediction in the spatial domain rather than in frequency domain like other codecs.
- The H.264 /AVC uses the reconstructed but unfiltered macroblock data from the neighboring macroblocks to predict the current macroblock coefficients.

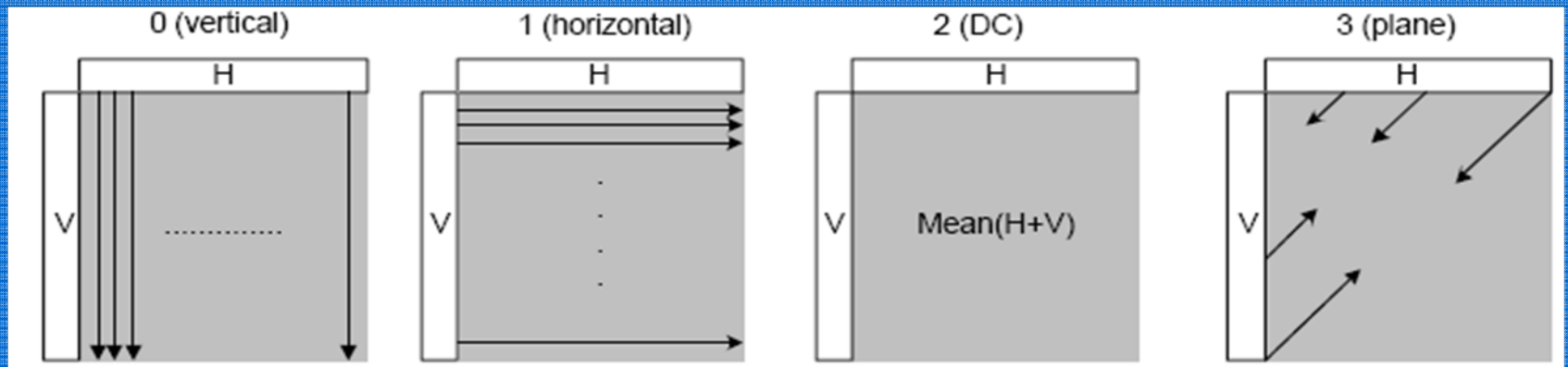
Advantages of spatial domain predictions...

- Intuitively, the prediction of pixels from the neighbouring pixels (top/left) of macro-blocks would be more efficient as compared to the prediction of the transform domain values.
- Predicting from samples in the pixel domain helps in better compression for intra blocks in a inter frame.
- Allows to better compression and hence a flexible bit-rate control by providing the flexibility to eliminate redundancies across multiple directions.

Intra Prediction Modes

- H.264/AVC supports intra-prediction for blocks of 4 x 4 to help achieve better compression for high motion areas.
 - Supports 9 prediction modes.
 - Supported only for luminance blocks
- H.264/AVC also has a 16 x 16 mode, which is aimed to provide better compression for flat regions of a picture at a lower computational costs.
 - Supports 4 direction modes.
 - Supported for 16x16 luminance blocks and 8x8 chrominance blocks

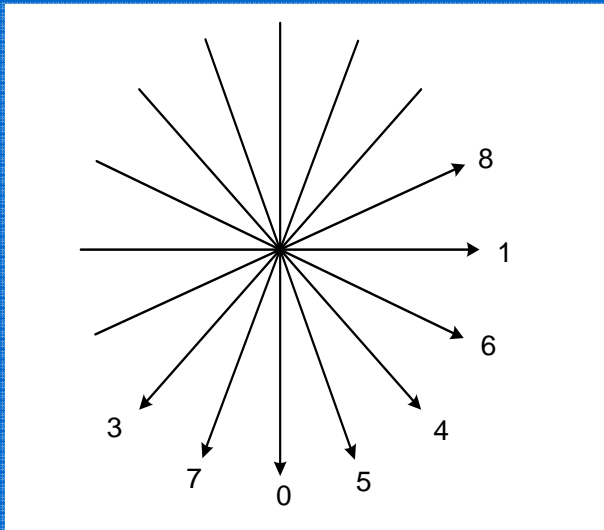
LUMA 16x16 / CHROMA Intra-Prediction Modes explained...



Intra16x16PredMode	Name of Intra16x16PredMode
0	Vertical (prediction mode)
1	Horizontal (prediction mode)
2	DC (prediction mode)
3	Plane (prediction mode)

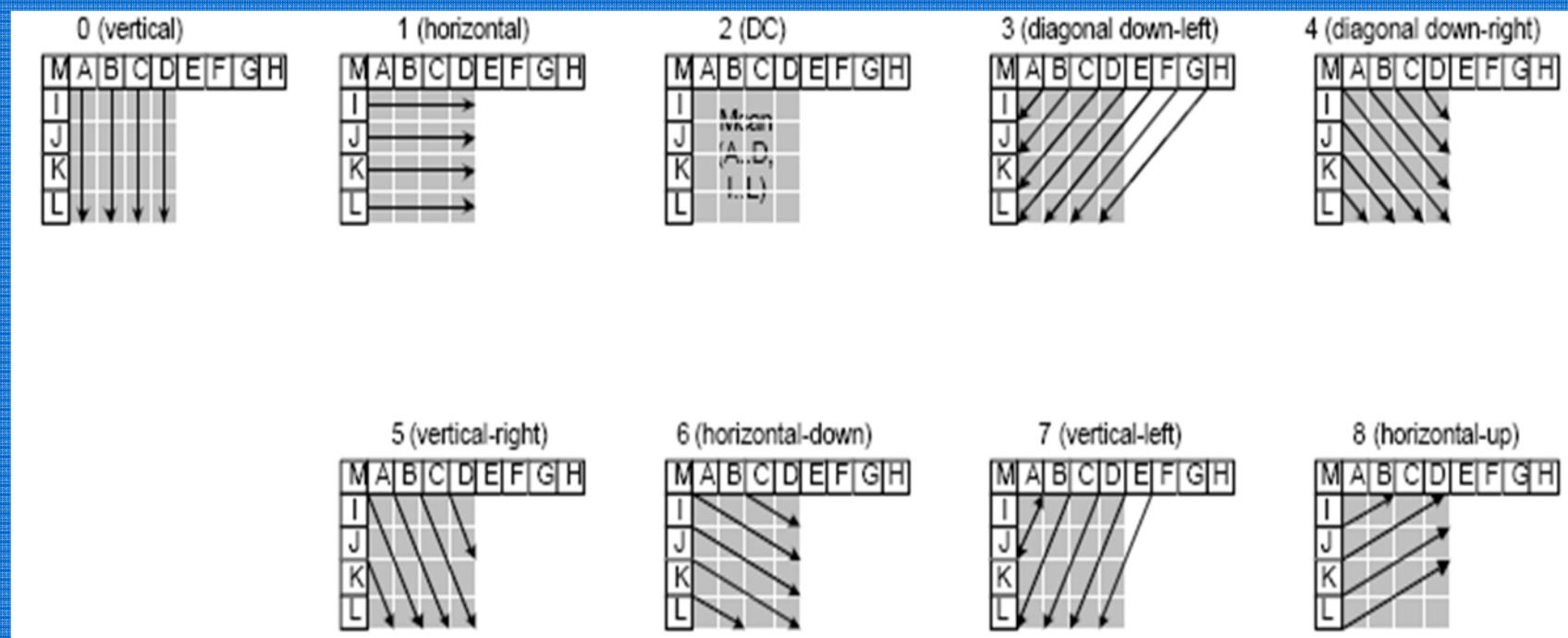
explained...

- The H264 /MPEG4 AVC provides for eliminating redundancies in almost all directions using the 9 modes as shown below.



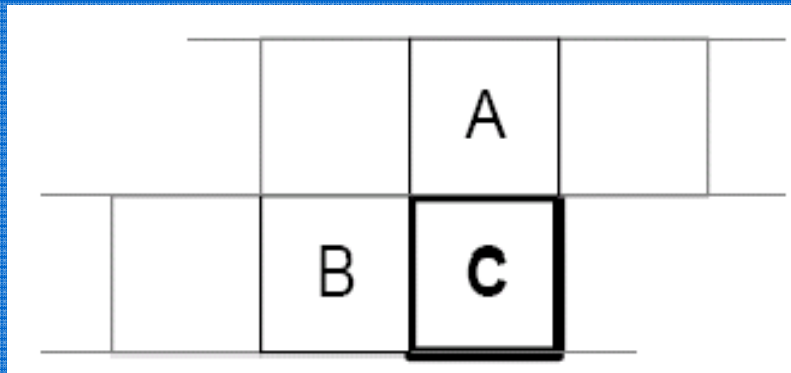
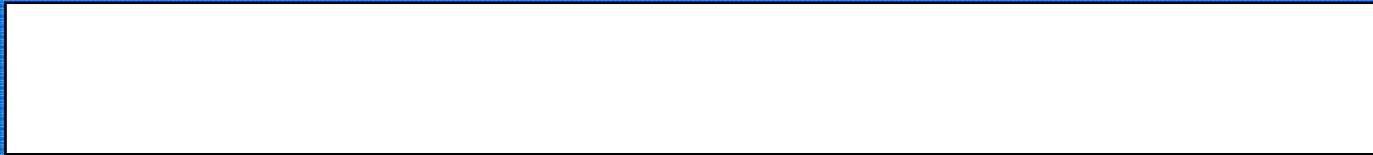
Intra4x4PredMode	Name of Prediction Mode
0	Vertical
1	Horizontal
2	DC
3	Diagonal_Down_Left
4	Diagonal_Down_Right
5	Vertical_Right
6	Horizontal_Down
7	Vertical_Left
8	Horizontal_Up

Luma 4x4 Intra-Prediction Modes explained...





- 1. Determining the prediction mode
(Only for a 4x4 block size mode).**
2. Determination of samples to predict the block data.
3. Predict the block data.



- Flag in the bit-stream indicates, whether prediction mode is present in the bit-stream or it has to be Implicitly calculated.
- In case of Implicit mode, the prediction mode is the minimum of prediction modes of neighbors 'A' and 'B'.



1. Determining the prediction mode (Only for a 4x4 block size mode).
- 2. Determination of samples to predict the block data.**
3. Predict the block data.

- To Predict a 4x4 block (a-p), a set of 13 samples (A-M) from the neighboring pixels have to be chosen.
- For a 8x8 chrominance block a set of 17 neighboring pixels are chosen as sample values.
- Similarly for predicting a 16x16 luminance block, a set of 33 neighboring pixels are selected as the samples

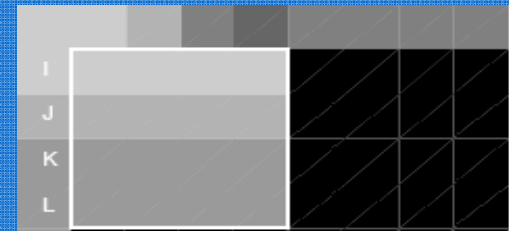
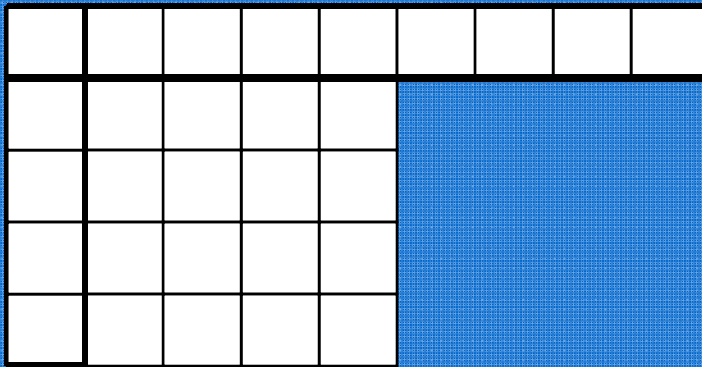
M	A	B	C	D	E	F	G	H
I	a	b	c	d				
J	e	f	g	h				
K	i	j	k	l				
L	m	n	o	p				



1. Determining the prediction mode (Only for a 4x4 block size mode).
2. Determination of samples to predict the block data.
3. **Predict the block data.**

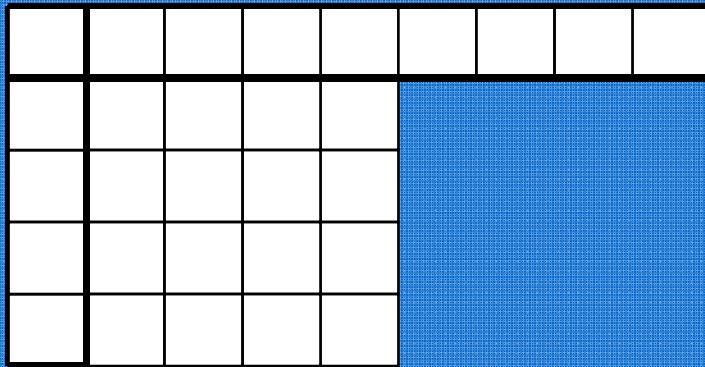


- **Horizontal prediction mode**






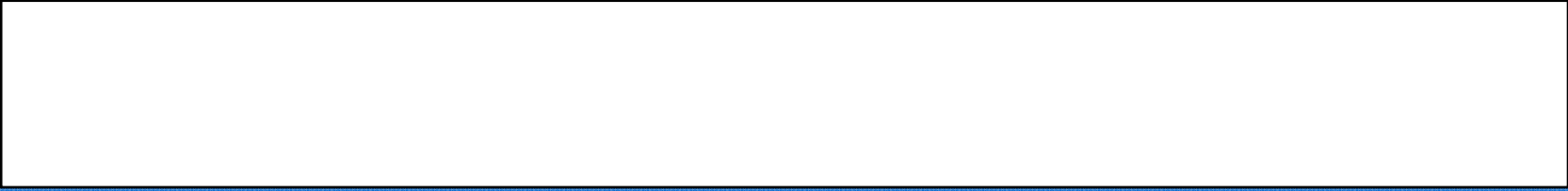
- **DC prediction mode**



	A	B	C	D		
I						
J						
K						
L						

X = Mean

- 
- The dependence of the blocks prediction samples on it's neighbors, which itself may a part of current MB prevent parallel processing of block data.
 - Each of the 16 blocks in a given MB can choose any one of the nine prediction modes, With each mode entire processing changes. Each mode has a totally different mathematical weighting function used for deriving the predicted data from the samples.



❑ Coarse quantization of the block-based image transform produce disturbing blocking artifacts at the block boundaries of the image.

❑ The second source of blocking artifacts is motion compensated prediction. Motion compensated blocks are generated by copying interpolated pixel data from different locations of possibly different reference frames.

❑ When the later P/B frames reference these images having blocky edges, the blocking artifacts further propagates to the interiors of the current blocks block worsening the situation further.



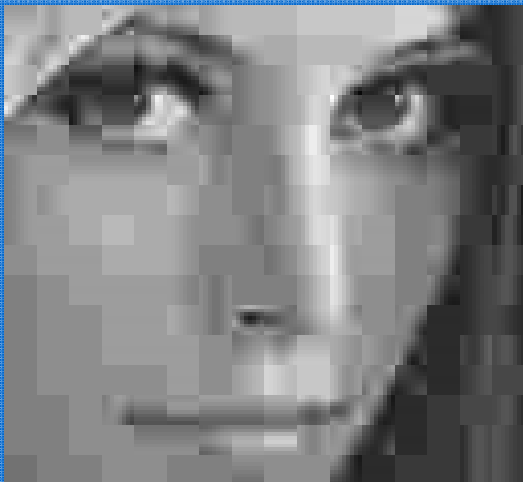
Original Frame



Reference frame



De-blocked Reference frame



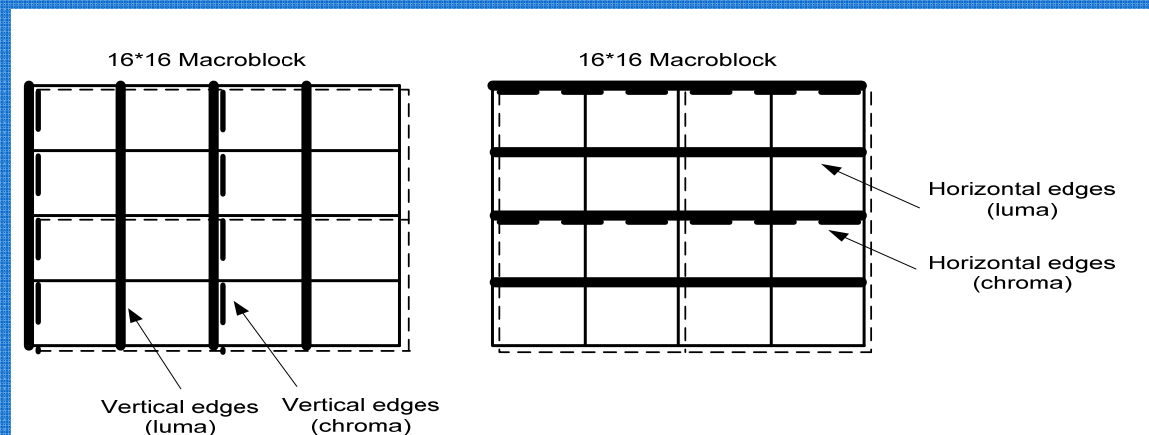
H.264 /AVC adaptive De-blocking filter : Advantage over post-processing approach.

- Ensures a certain level of quality.
- No need for potentially an extra frame buffer at the decoder.
- Improves both objective and subjective quality of video streams. Due to the fact that filtered reference frames offer higher quality prediction for motion compensation.

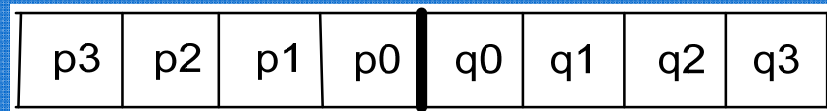
H.264 /AVC adaptive De-blocking filter : Introduction

- The best way to deal with these artifacts is to filter the blocky edges to have a smoothed edge. This filtering process is known as the de-block filtering.
- Till recently, the coding standards, defined the de-blocking filter, but not mandating the use of the same, as the implementation is cycle consuming and is a function of the quality needed at the user end.
- But it was soon figured out that if the de-block filter is not compulsorily implemented the frames suffered from blockiness caused in the past frames used as reference.
- This coupled with the increasing number crunching powers of the modern day DSP's, made it a easier choice for the standards body to make this de-block filter mandatory tool or a block in the decode loop – IN LOOP DEBLOCK FILTER.
- This filter not only smoothened the irritating blocky edges but also helped increase the rate-distortion performance.

- Last process in the frame decoding, which ensures all the top/left neighbors have been fully reconstructed and available as inputs for de-blocking the current MB.
- Applied to all 4x4 blocks except at the boundaries of the picture.
- Filtering for block edges of any slice can be selectively disabled by means of flag.
- Vertical edges filtered first (left to right) followed by horizontal edges (top to bottom)



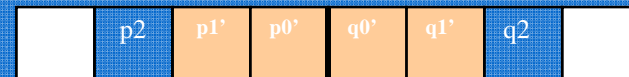
- For de-blocking an edge, 8 pixel samples in all are required in which 4 are from one side of the edge and 4 from the other side.



- Of these 8 pixel samples the de-block filter updates 6 pixels for a luminance block and 4 pixels for a chrominance block.



Luminance pixels after filtering



Chrominance pixels after filtering

continued

- Is it just low pass filter?
- We want to filter only blocking artifacts and not genuine edges!!!
- Content-dependent boundary filtering strength.
- The Boundary strengths are a method of implementing adaptive filtering for a given edge based on certain conditions based on
 - MB type
 - Reference picture ID
 - Motion vector
 - Other MB coding parameters
- The Boundary strengths for a chrominance block is determined from the boundary strength of the corresponding luminance macro block.



continued

- The blocking artifacts are most noticeable in very smooth region where the pixel values do not change much across the block edge.
- Therefore, in addition to the boundary strength, a filtering threshold based on the pixel values are used to determine if de-blocking process should be carried for the current edge.