AY 2024/2025 S1

EE6427 Video Signal Processing

Part 3 Video Compression & Standards

Dr Yap Kim Hui

Room: S2-B2b-53
Tel: 6790 4339
Email: ekhyap@ntu.edu.sg

References

- Gonzalez and Woods Digital Image Processing, 4th edition, 2018.
- Fundamentals of Multimedia Ze-Nian Li, Mark S. Drew, Jiangchuan Liu, 3rd Edition, 2021.
- https://mpeg-miv.org/index.php/home/

Part 3 Outline

- Video Coding & Standards
- MPEG Standards
- H.26x Standards
- New / Emerging Coding & Standards

Video Coding & Standards

Video Compression Basics

Video

Recap: Spatial Redundancy

- Spatial redundancy refers to the statistical correlation between pixels within an image (or more specifically, within a small image neighborhood).
- It is also called intraframe redundancy.

Source: Adobe

Recap: Temporal Redundancy

- Temporal redundancy refers to the statistical correlation between pixels from successive frames in a video sequence.
- Therefore, it is also called interframe redundancy.



Recap: Psychovisual Redundancy

- Frequency masking
  - Human is less sensitive to noise or distortion in high frequency components, and vice versa.
- Color masking
  - Human is more sensitive to luminance (brightness) component than chrominance (color) components.

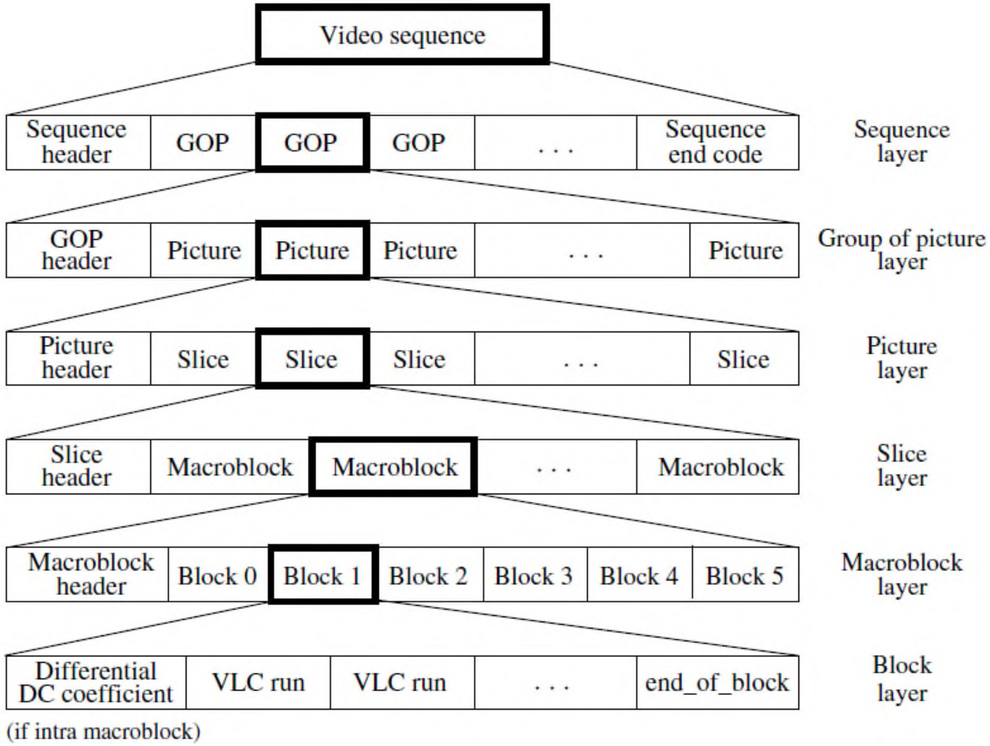
Source: https://www.istockphoto.com/photos/natural-texture
https://www.wallpaperflare.com/lake-digital-wallpaper-river-smooth-surface-trees-hills-water-wallpaper-tgzo

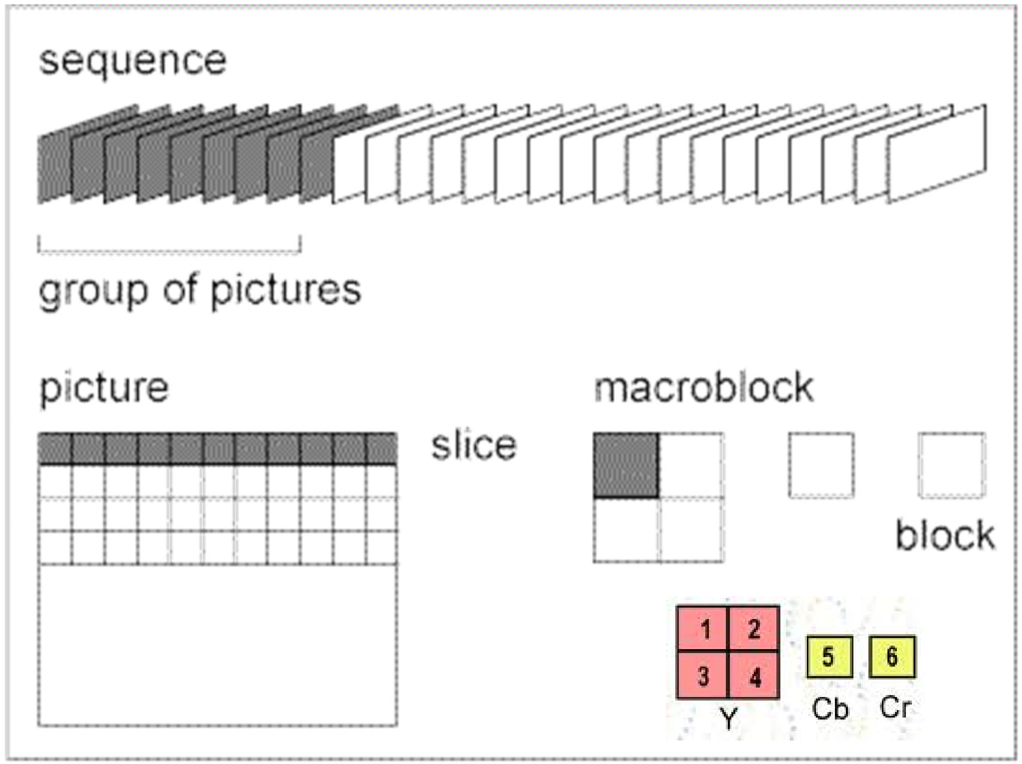
How to Compress a Video?

- By exploiting redundancies in a video.
  - Spatial redundancy.
  - Temporal redundancy.
  - Psychovisual Redundancy.

- By improving coding efficiency:
  - Huffman coding.
  - Predictive / differential coding.

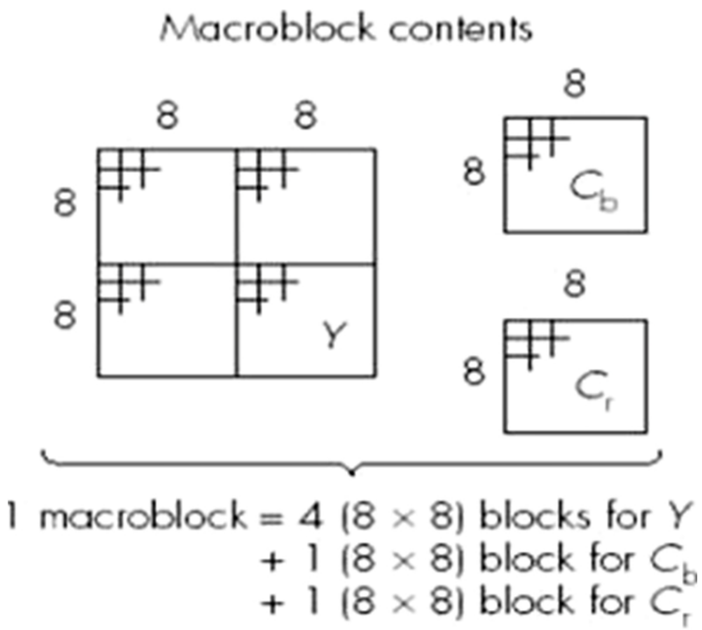
Video Structure





Macroblock

- A macroblock contains a 16x16 Y component and spatially corresponding 8x8 C_b and C_r components.
- A macroblock has four luminance blocks and two chrominance blocks, and it is the basic unit for DCT operation and motion compensation.



Video Compression Basics

- Main idea in video compression:
  - Motion estimation:
    prediction of motion vector for macroblocks.
  - Motion compensation:
    encode small difference between predicted and actual macroblocks.

Motion Estimation & Compensation

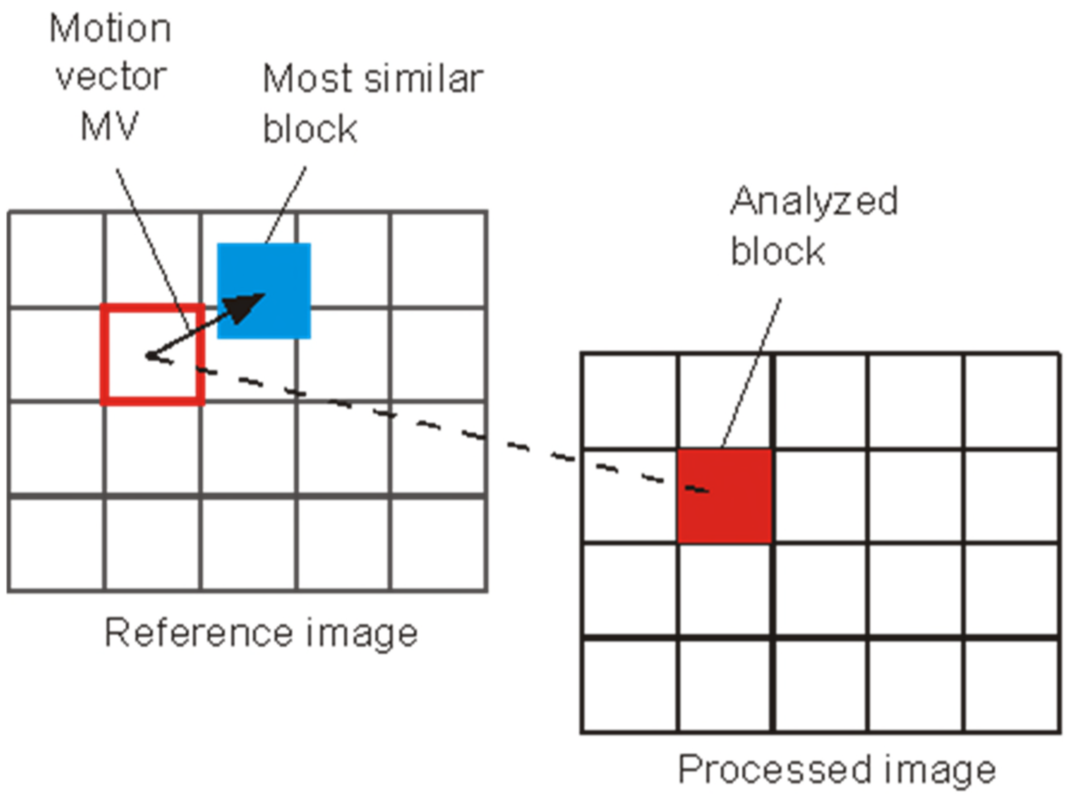
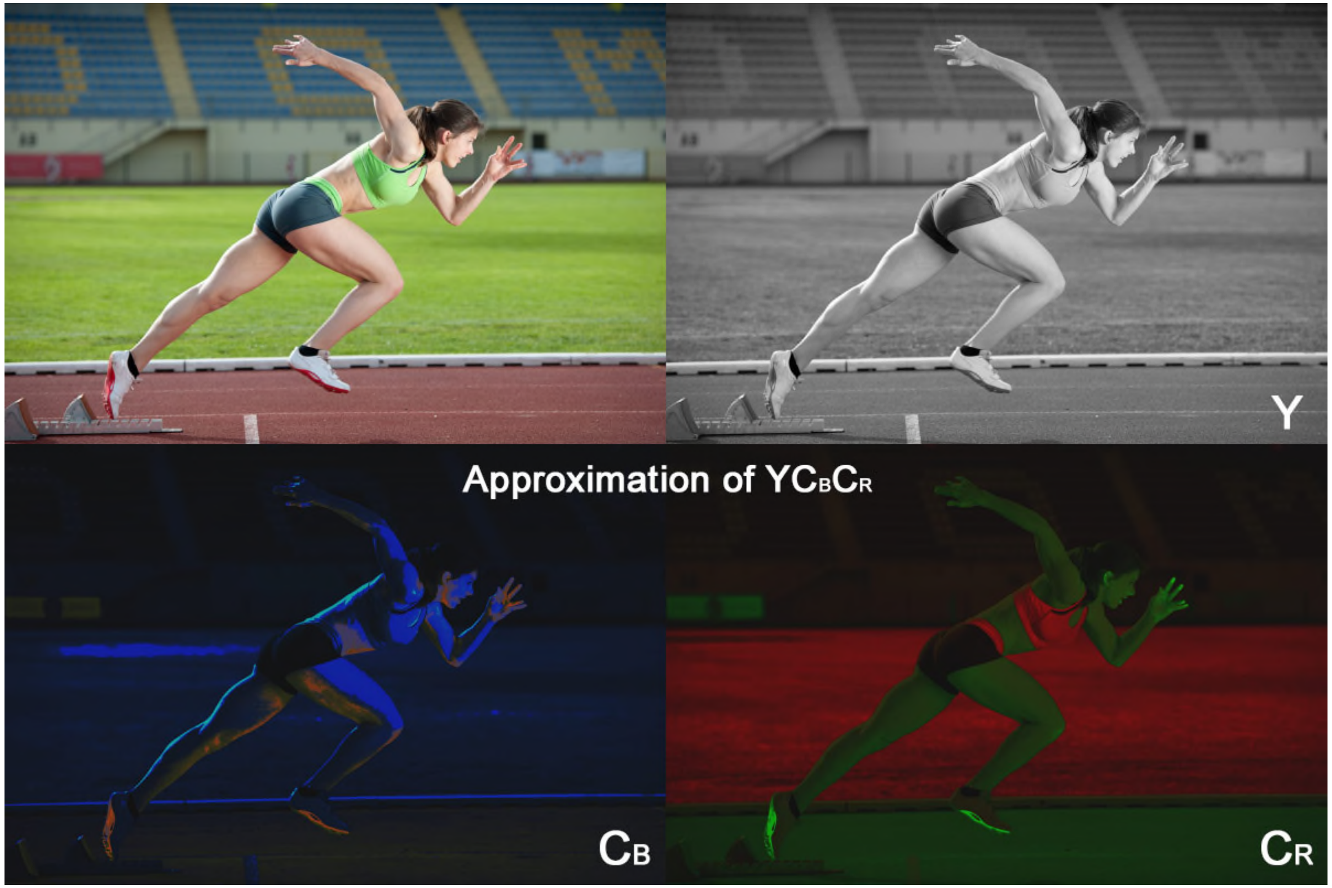


Figure source: http://www.multimedia.edu.pl

Intraframe & Interframe Compression

- Many video compression methods such as MPEG use both intraframe and interframe compression.
- Intraframe compression
  - Consider each video frame as still image (e.g., I-frames in MPEG).
  - Reduce spatial redundancy.
  - Employ JPEG-based compression.
- Interframe compression
  - Utilize correlation along temporal domain (e.g., P & B-frames in MPEG).
  - Reduce temporal redundancy.
  - Employ motion estimation and compensation.

Luminance (Luma) & Chrominance (Chroma)



Source: https://blog.video.ibm.com/streaming-video-tips/what-is-video-encoding-codecs-compression-techniques

Chroma Subsampling

- Subsampling formats:
  - 4:4:4 - No subsampling
  - 4:2:2, 4:1:1 - horizontal subsampling
  - 4:2:0 - horizontal and vertical subsampling

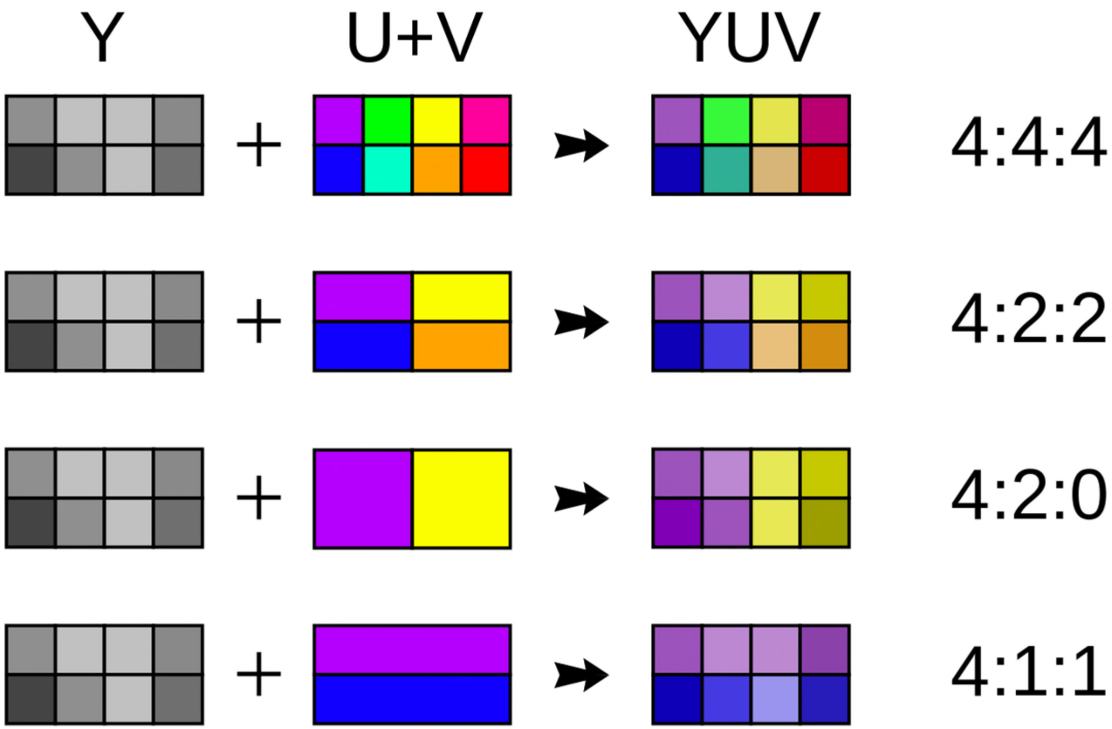


Figure source: https://en.wikipedia.org/wiki/Chroma_subsampling#/media/File:Common_chroma_subsampling_ratios.svg

Chroma Subsampling

- Human is more sensitive to luminance component than chrominance components.
- Hence, to achieve compression, chrominance components are subsampled.
- 4:2:2 sub-sampling

  For every two horizontal Y samples, 1 C_b and 1 C_r are sampled.
- 4:2:0 sub-sampling

  For every 2x2 Y samples, 1 C_b and 1 C_r are sampled.

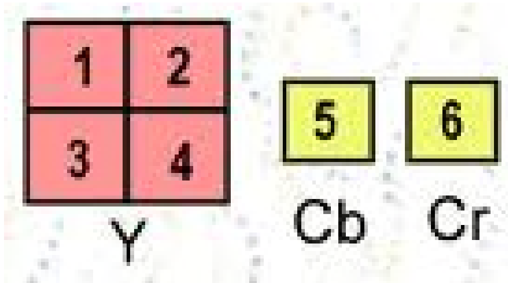
Basic of Video Coding / Compression

https://www.youtube.com/watch?v=7YQ1mikDhIo

MPEG Basics

MPEG

- Objective: to encode/compress video signals.
- MPEG standard is based on DCT coding and motion estimation & compensation.
  - DCT coding: remove intraframe redundancy.
  - Motion estimation & compensation: remove interframe redundancy.
- A color video source has three components
  - a luminance component (Y)
  - two chrominance components (C_b *a*nd C_r) in usually 4:2:0 subsampling format.



MPEG Video Structure

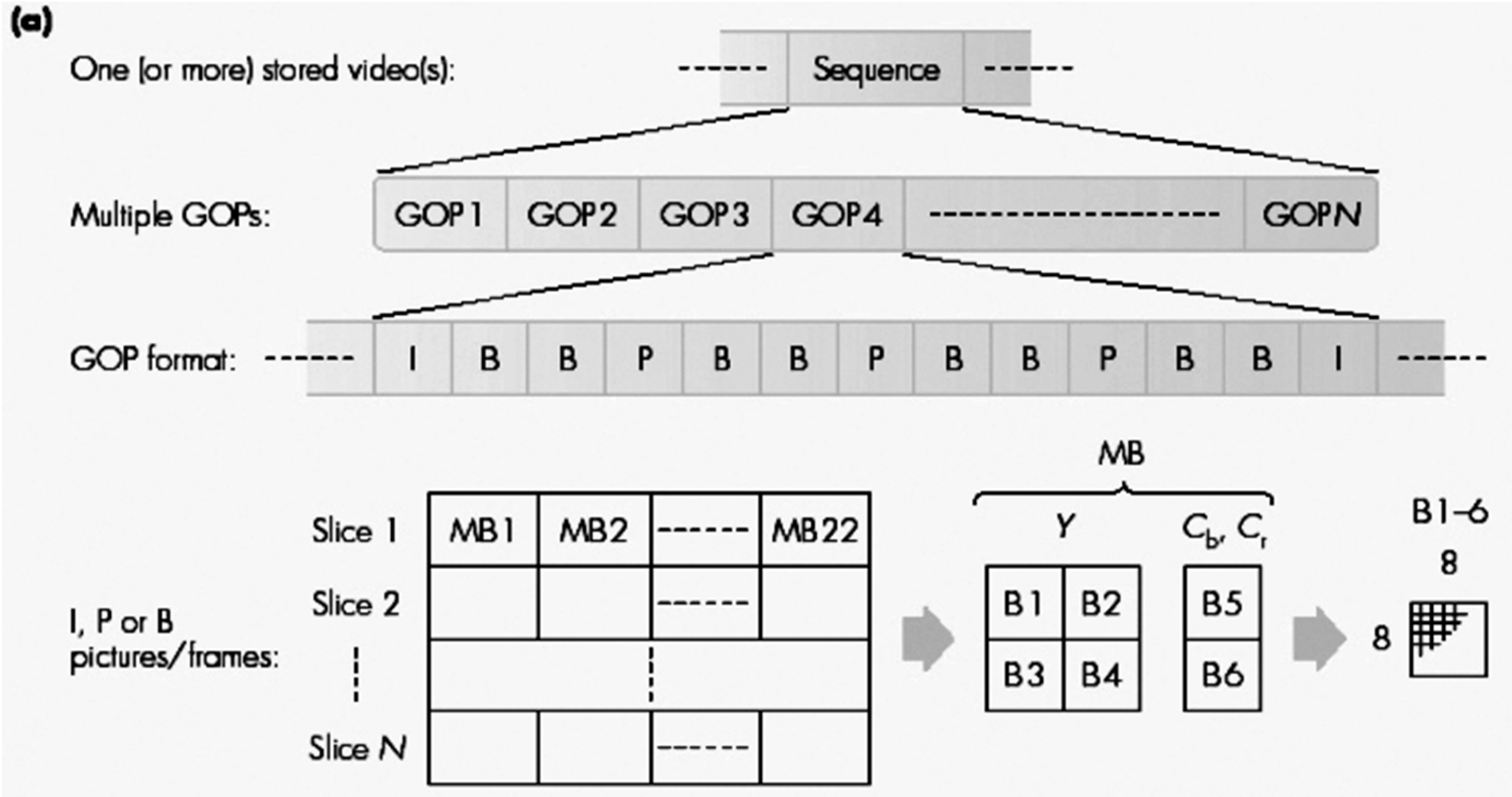


Figure source:
Fred Halsall, Multimedia communications: applications, networks, protocols and standards, Addison-Wesley, 2001

Group of Pictures (GOPs)

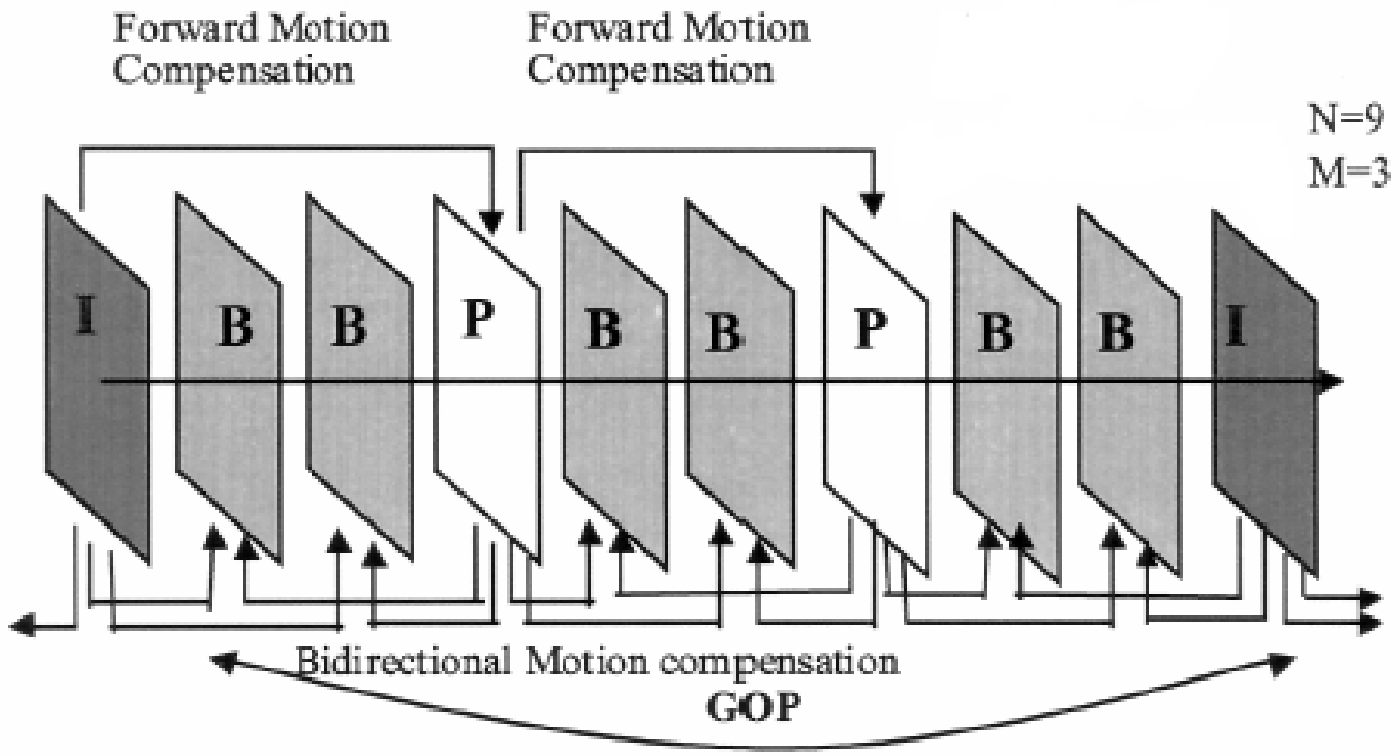


FIGURE 16.1 A group of pictures of video sequence in display order.

GOP

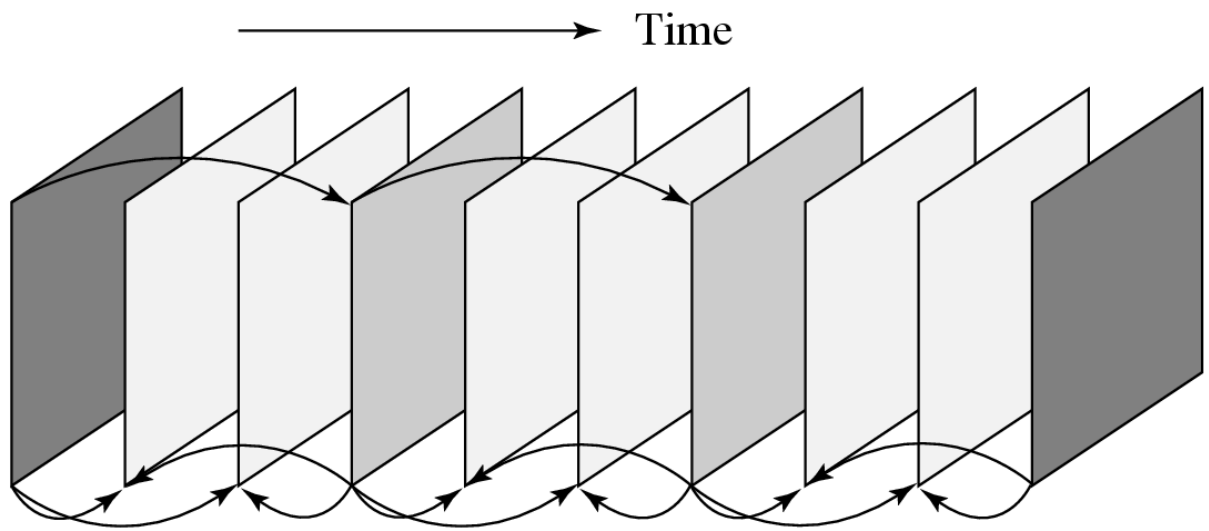
- The video sequence is divided into group of pictures (GOPs).
- Each GOP may include three types of pictures:
  - I-frames (Intra-coded frames)
  - P-frame (Predictive-coded frame)
  - B-frame (Bidirectional-coded frame)

I-frames, P-frames, B-frames

- I-frames
  - coded without reference to any other frames.
  - Y, C_b, C_r blocks are encoded independently using JPEG algorithm.

- P-frames and B-frames
  - known as intercoded or interpolation frames.
  - P-frames are coded using motion estimation and compensation from previous anchor I-frame or P-frame.
  - B-frames are coded using predictions from either past, future, or both anchor frames.

MPEG-1 Display & Coding Order



Display order	I	B	B	P	B	B	P	B	B	I
Coding and transmission order	I	P	B	B	P	B	B	I	B	B

GOP Issues

- Which frame types in a GOP has the largest and smallest compression ratio? Why?
- How do you choose a suitable GOP size?

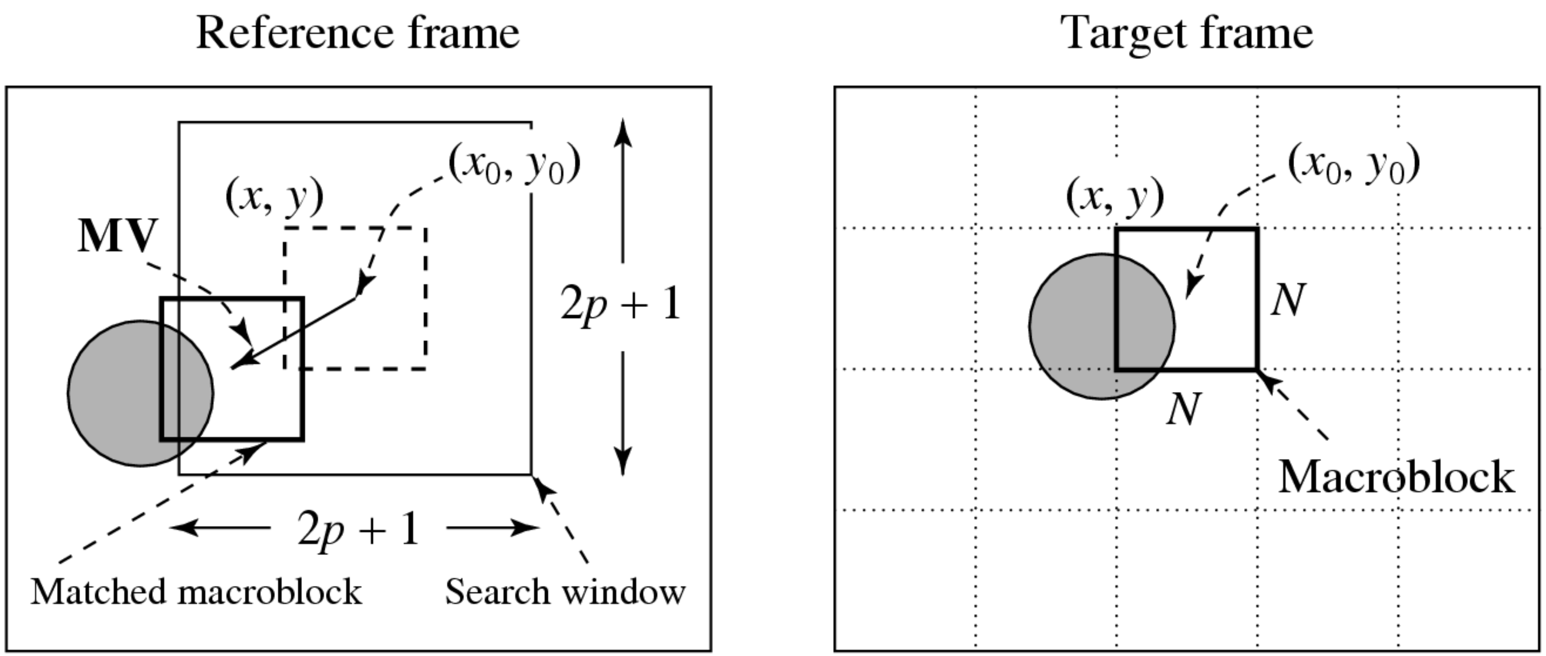
Motion Estimation & Compensation

Motion Estimation & Compensation

- Temporal redundancy: consecutive frames in a video are similar.
- Not every frame of the video needs to be coded independently.
- Idea: difference between the current frame and other frames in the sequence is coded.
- Difference has small values, hence good for compression.
- Video compression:
  - Motion Estimation: motion vector search.
  - Motion Compensation: compute the prediction error.

Motion Estimation (1)

- MV search is usually limited to a small immediate neighborhood
- Both horizontal and vertical displacements in the range [$-p$, p]. This makes a search window of size ($2p + 1$) x ($2p + 1$).



Motion Estimation (2)

- The difference between two macroblocks can be measured by Mean Absolute Difference (MAD):

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

N: size of the macroblock,

k and l : indices for pixels in the macroblock,

i and j : horizontal and vertical displacements,

$C(x + k, y + l)$: pixels in macroblock of Target frame,

$R(x + i + k, y + j + l)$: pixels in macroblock of Reference frame.

- Goal: find motion vector *MV* = (*u*, *v*) such that *MAD(i, j)* is minimum:

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

Motion Vector Visualization

https://www.youtube.com/watch?v=jdxWxBW5cd0&ab_channel=RadimKuncicky

Motion Estimation Methods

- What are the key consideration in the design of motion estimation methods?

Motion Estimation Methods

- Full Search
- Three-step Search
- 2D-Log Search
- Hierarchical Search
- Diamond Search
- Cross Search
- Etc.

Full Search

- Search every point within the search window.
- Pros: accurate, best compression ratio.
- Cons: slow, computationally intensive.

Three Step Search

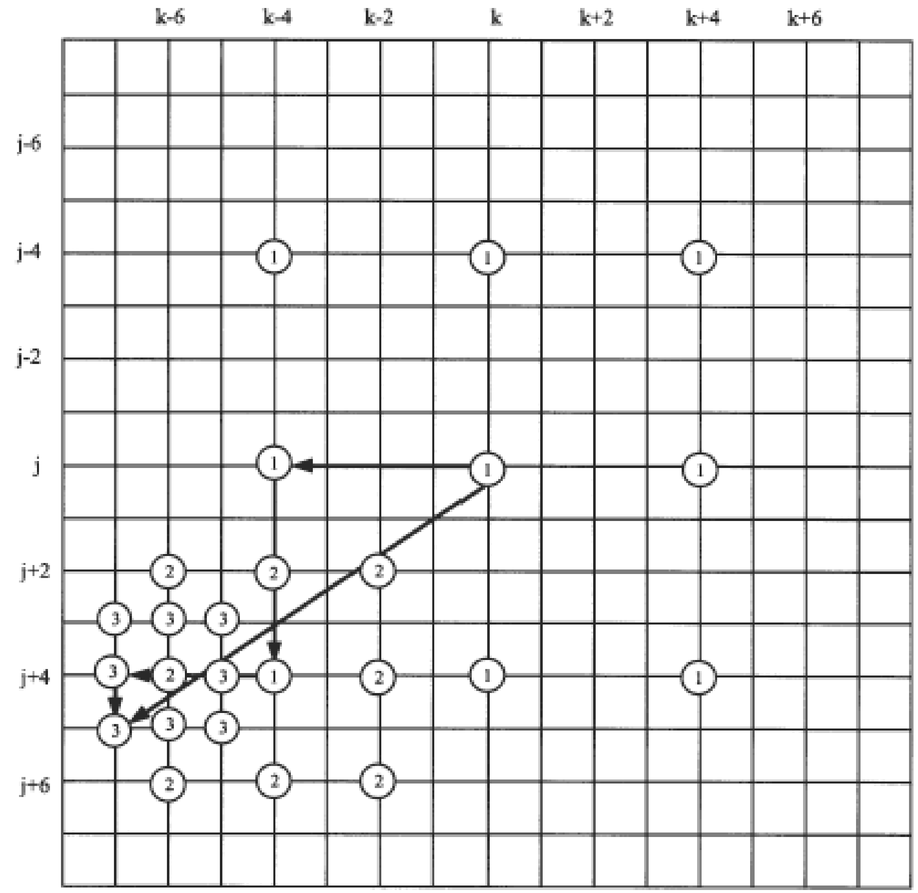


FIGURE 11.4 Three-step search procedure. Points (j+4, k-4), (j+4, k-6), and (j+5, k-7) give the minimum dissimilarity in steps 1, 2, and 3, respectively.

Three Step Search

- Consist of 3 steps only.
- Step 1: Search 9 points for minimum prediction error.
- Step 2: Centered on the found position in Step 1, halve the search width, search 9 points for minimum error.
- Step 3: Centered on the found position in Step 2, halve the search width, search 9 points for minimum error. The final position is the best position found.
- Pros: Fast
- Cons: less accurate, lower compression ratio.

2D Logarithm / Logarithmic Search

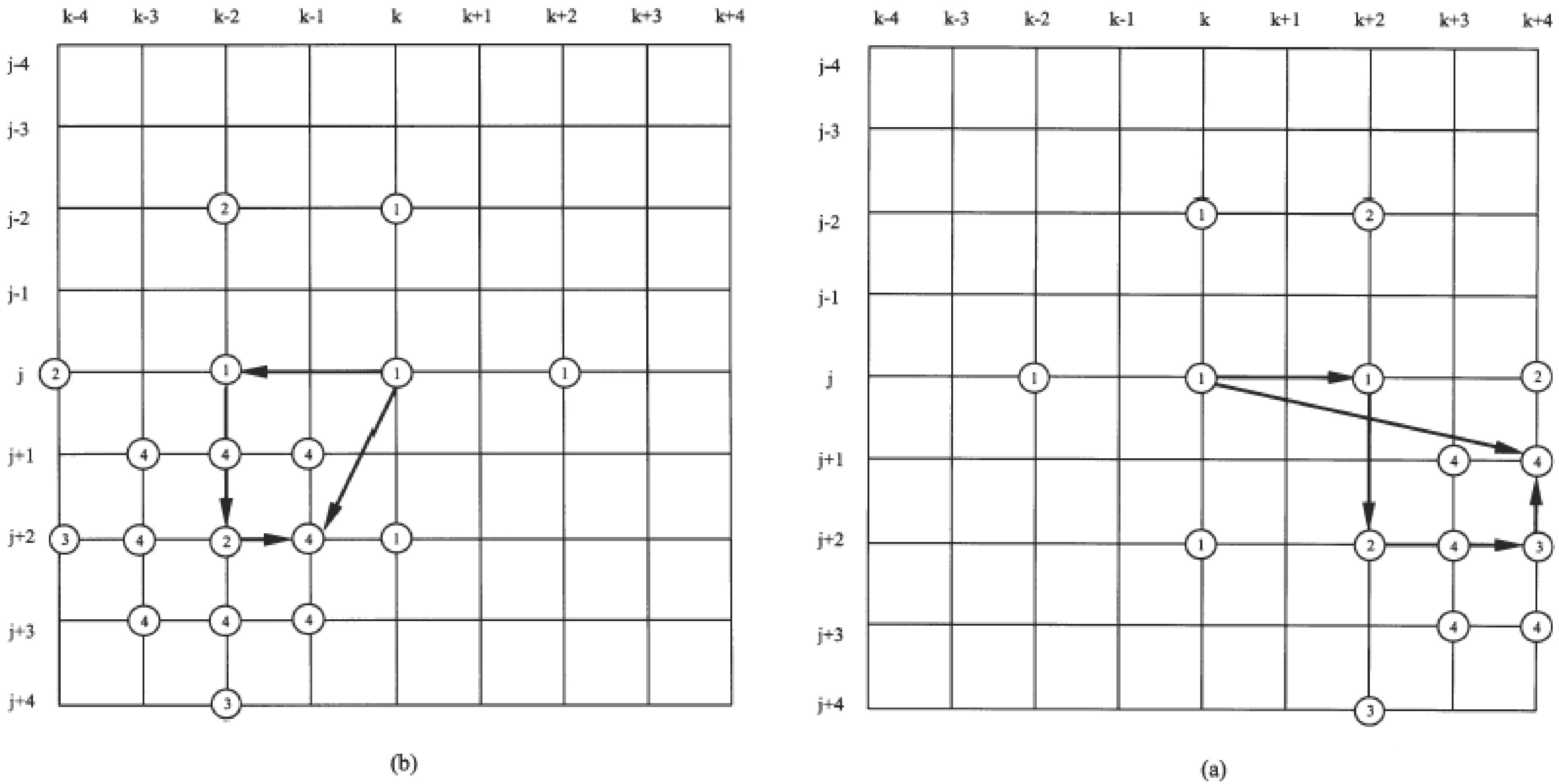


FIGURE 11.3 (a) A 2-D logarithmic search procedure. Points at (j, k+2), (j+2, k+2), (j+2, k+4), and (j+1, k+4) are found to give the minimum dissimilarity in steps 1, 2, 3, and 4, respectively. (b) A 2-D logarithmic search procedure. Points at (j, k-2), (j+2, k-2), and (j+2, k-1) are found to give the minimum dissimilarity in steps 1, 2, 3, and 4, respectively.

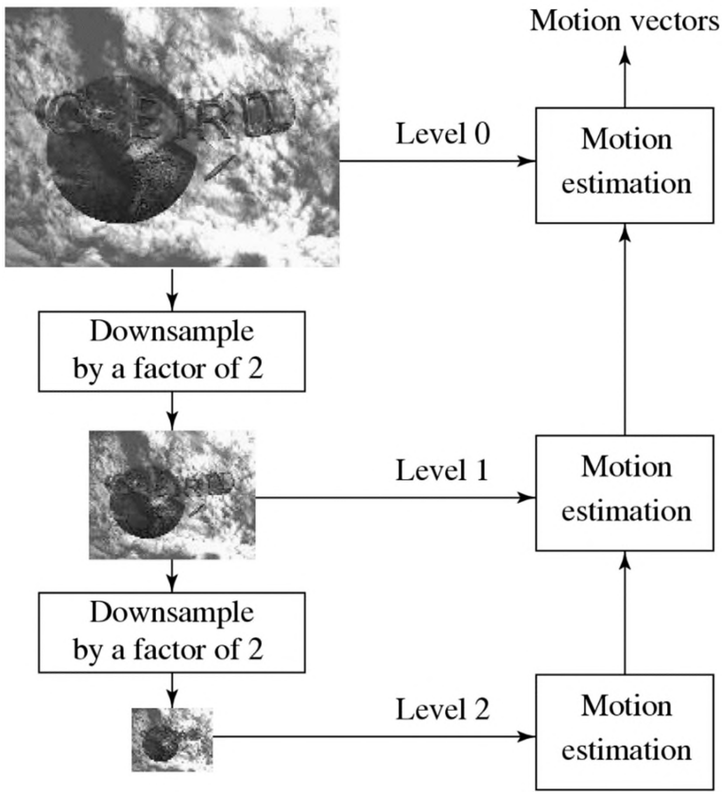
2D Logarithmic Search

- Step 1: Search 5 points for minimum prediction error.
- Step 2: Centered on the found position in previous iteration,
  - (i) If the found position is the central position or located at the border, halve the search width; else, retain the search width.
  - (ii) (a) If the search window is not 3x3 yet, search 5 points for minimum error and repeat Step 2.

    (b) Else, if the search window is 3x3, search 9 points for minimum error and return best found position.
- Speed: Between full search and 3 step search.
- Accuracy: Between full search and 3 step search.

Hierarchical Search

- A three-level hierarchical search, the original image is at Level 0.
- Images at Levels 1 and 2 are obtained by down-sampling from the previous levels by a factor of 2, and the initial search is conducted at Level 2.



A Three-level Hierarchical Search

Hierarchical Search

- Search can benefit from hierarchical/multiresolution approach.
- Initial estimation of motion vector can be obtained from images with a significantly reduced resolution.
- The size of the macroblock is smaller.
- The search range *p* can also be proportionally reduced.
- Hence, the number of operations required is greatly reduced.

History & Milestones

Video Coding Standardization Organization

- ISO and IEC created Moving Picture Experts Group (MPEG) that published MPEG standards.
- ITU published H.26x coding standards, e.g., H.261, H.263.
- ITU worked with ISO/IEC to publish H.262, H.264, H.265, H.266.

Abbreviation	IEC	ITU	ISO
Official Name	International Electrotechnical Commission	International Telecommunication Union	International Organization for Standardization
Target Fields	Electrical engineering and electronic technology	Wireless and electrical communication	Industrial products

Source: https://www.contec.com/support/blog/2022/220524_safetystandard

Key Coding Standards & Milestones

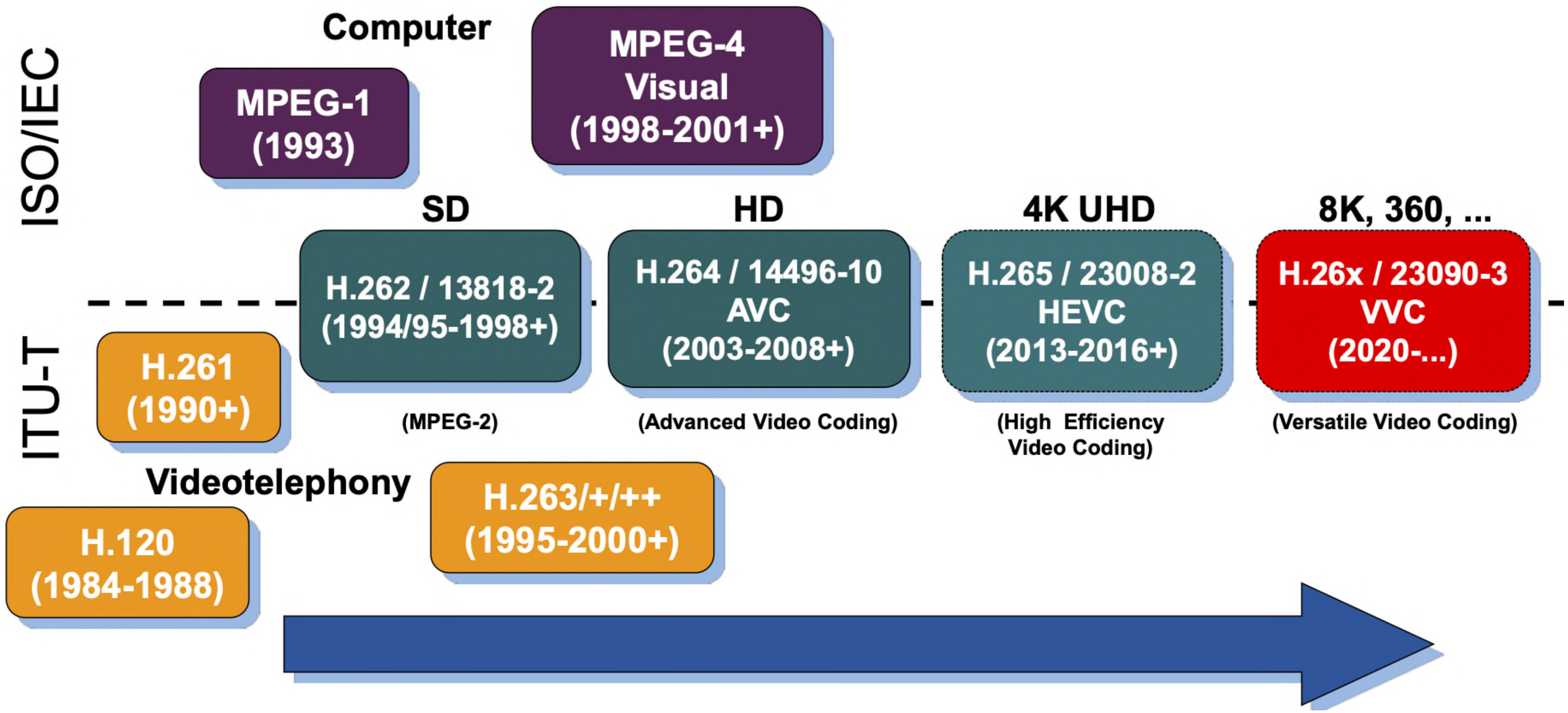


Figure source: https://www.linkedin.com/pulse/video-coding-standards-comparison-sraas/

MPEG Summary (1)

Standard	Key Features
MPEG-1	• Standardised in 1993. • Developed for video and audio storage on CD-ROMs. • Support progressive scan. • Support YUV 4:2:0 at Common Intermediate Format (CIF) resolution (352x288). • Group of pictures include I-, P- and B-frames.
MPEG-2	• Standardised in 1995. • Support videos on DVDs, standard definition TVs and high-definition TVs. • Support YUV 4:2:0 and YUV 4:2:2 formats. • Support interlaced and progressive scans. • Introduce profiles and levels to define the various capabilities. • Support scalable coding.

Source: https://www.vcodex.com/historical-timeline-of-video-coding-standards-and-formats/

MPEG Summary (2)

Standard	Key Features
MPEG-4 Part-2	• Standardised in 1999. • Support video on low bitrate multimedia applications on mobile platforms and the Internet. • Support object-based or content-based coding where a video scene is coded as a set of foreground and background objects. • Support coding of synthetic video and audio including animation.
MPEG-4 Part-10 (AVC)	• Standardised in 2003 as Advanced Video Coding (AVC). • Co-published as H.264 / AVC. • To improve compression performance, use Integer transform, variable block-size motion compensation, directional spatial prediction for intra frames, in-loop deblocking filtering, etc.

Source: https://www.vcodex.com/historical-timeline-of-video-coding-standards-and-formats/

H.26x Summary (1)

Standard	Key Features
H.261	• Standardised in 1988. • Developed for video conferencing. • Support for CIF and QCIF resolutions in YUV 4:2:0 format. • Uses block-based hybrid coding with integer pixel motion compensation.
H.262	• Standardised as MPEG-2 Part-2 (Video) in 1995. • Include bidirectionally-predictive-coded frames (B-frames) compared with H.261. • Support interlaced video.
H.263/ H.263+	• H.263 was standardized in 1996. • H.263+ was standardized in 1998. • Improved quality compared to H.261 at lower bit rate to enable video conferencing/telephony.

Source: https://www.vcodex.com/historical-timeline-of-video-coding-standards-and-formats/

H.26x Summary (2)

Standard	Key Features
H.264 / MPEG AVC	• Standardised in 2003. • Supports video on the Internet, computers, mobile and HDTVs. • Significantly improved picture quality compared to H.263. • Use Integer transform to reduce prediction shift of DCT. • Improved motion compensation with variable block-size, multiple reference frames. • In-loop deblocking filter to reduce block discontinuities.
H.265 / HEVC	• Standardised in 2013. • Similar to H.264/AVC but with improvements such as: – Support UltraHD video up to 8K resolutions with frame rates up to 120 fps. – Greater flexibility in prediction modes and transform block sizes. – More sophisticated interpolation and deblocking filters. – Support parallel processing. – More efficient than H.264 in terms of bitrate savings for same picture quality.

Source: https://www.vcodex.com/historical-timeline-of-video-coding-standards-and-formats/

Other Video Coding Standards

Standard	Key Features
VP8	• Open source and royalty-free video compression format. • Support Web video format YUV4:2:0, 8-bit color depth, progressive scan and 4K resolution. • Motion vector accuracy of 1/4-th pixel for luma & 1/8-th pixel for chroma. • Adaptive in-loop deblocking filter to reduce block discontinuities.
VP9	• Support up to 8K resolution and 120 fps frame rate. • Use superblocks (32x32, 64x64) to exploit high correlation over larger areas in HD video. • Enhanced interpolation for motion compensation with 8-tap filters to achieve 1/8-th pixel accuracy in motion vectors.

Source: https://www.vcodex.com/historical-timeline-of-video-coding-standards-and-formats/

MPEG Standards

MPEG-1

MPEG-1: Overview

- Developed by ISO/IEC in Nov. 1992.
- Standard for lossy video/audio compression.
- Provide coding of video and its associated audio with speed 1.5 Mbps for digital storage media (VCD).
- Extension of JPEG, H.261.
- Supports only progressive scan.
- Supported picture resolution:
  - 352 × 240 for NTSC video at 30 fps.
  - 352 × 288 for PAL video at 25 fps.
  - Use 4:2:0 chroma subsampling.

MPEG-1: Parts

- MPEG-1 is also referred to as ISO/IEC 11172.
- It has five parts:
  - Part 1 - MPEG-1 Systems
  - Part 2 - MPEG-1 Video
  - Part 3 - MPEG-1 Audio
  - Part 4 - Conformance
  - Part 5 - Reference Software

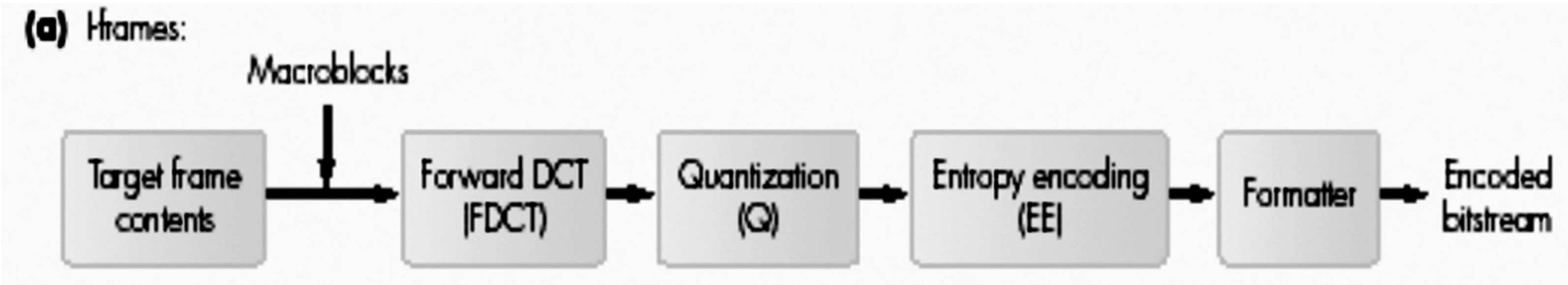
MPEG-1: Limitations

- Support only progressive scanning.
- Low picture quality.
- Low compression ratio.

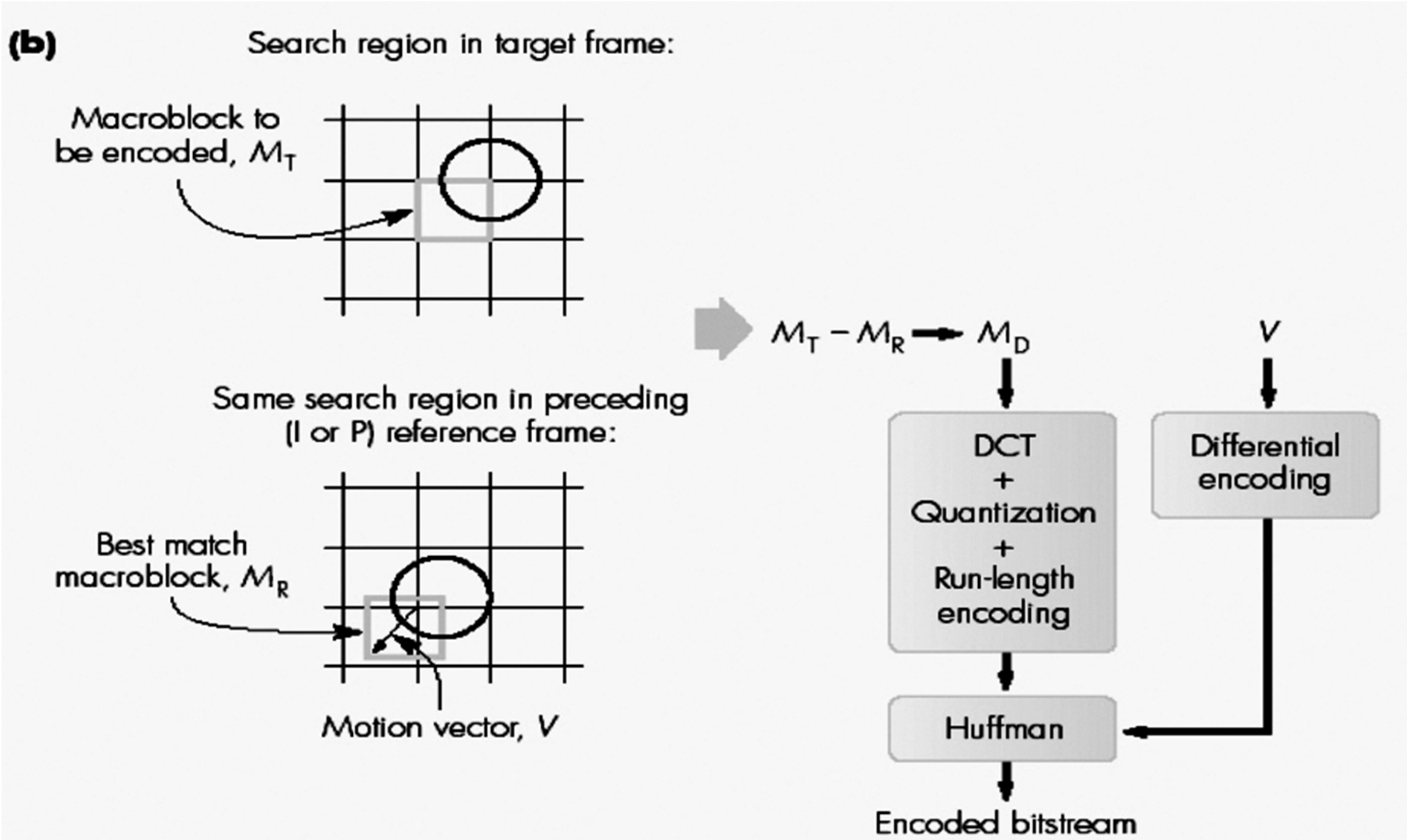
source: https://www.youtube.com/watch?v=Wso9MRuMTZ4&ab_channel=kvulliamy
https://www.reddit.com/r/dvdcollection/comments/yebfc2/built_up_a_tiny_vcd_collection_recently_the/

MPEG-1: I, P & B Frame Encoding

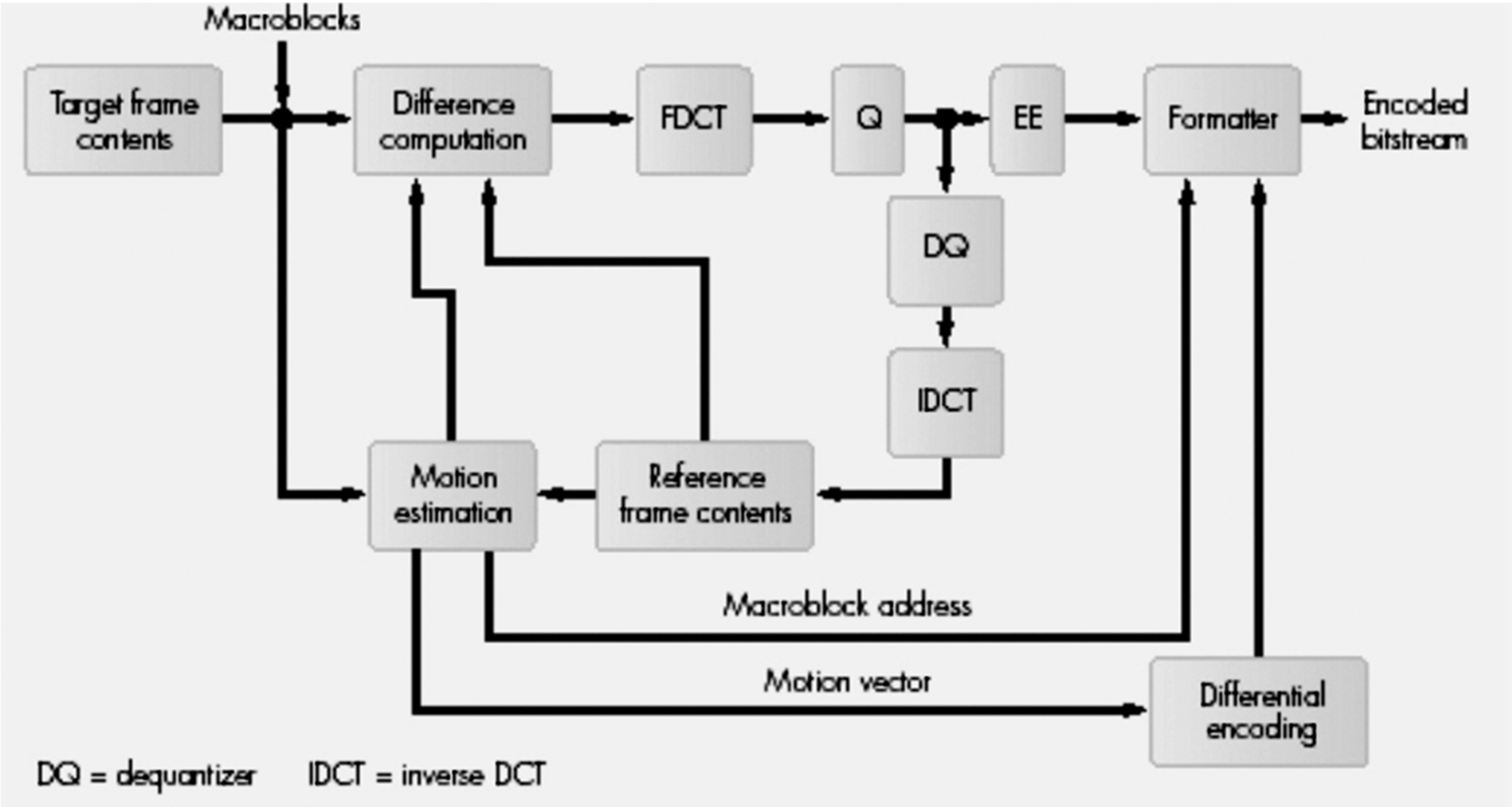
MPEG-1: I-Frame Encoding



MPEG-1: P-Frame Encoding Schematics



MPEG-1: P-Frame Encoding Flowchart



MPEG-1: P-Frame Encoding (I)

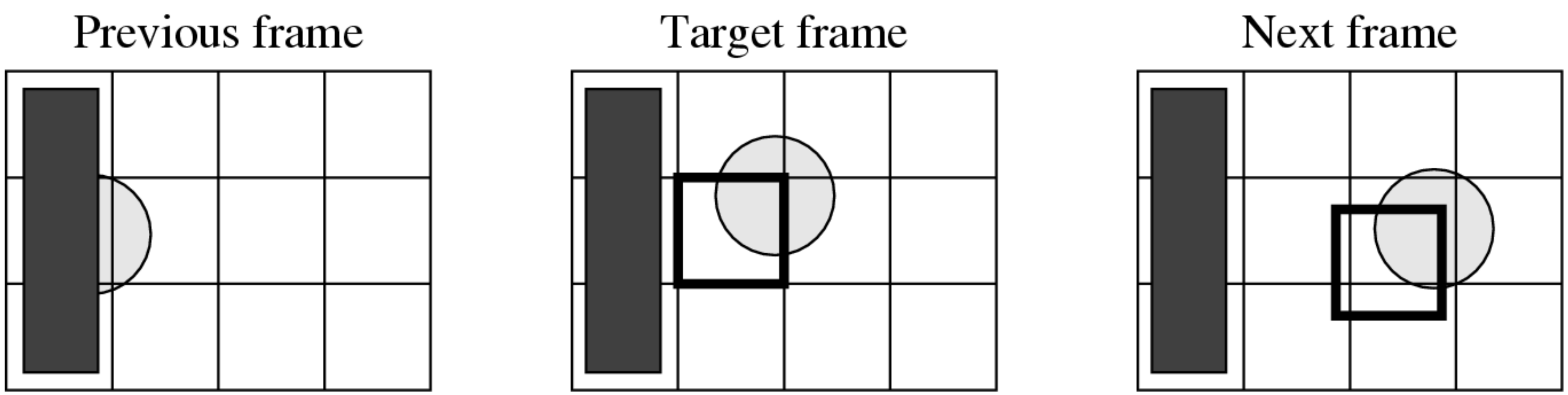
- Perform motion estimation and compensation for each macroblock in the target frame.
- Each macroblock in the target frame is compared with the preceding I or P reference (anchor) frame.
- For each macroblock, a search is performed centered on a search window in the reference frame.
- A match is found if the Sum of Absolute Difference (SAD) is less than a threshold.
- If a match is found, two parameters are determined: motion vector and prediction error (matrix).

MPEG-1: P-Frame Encoding (II)

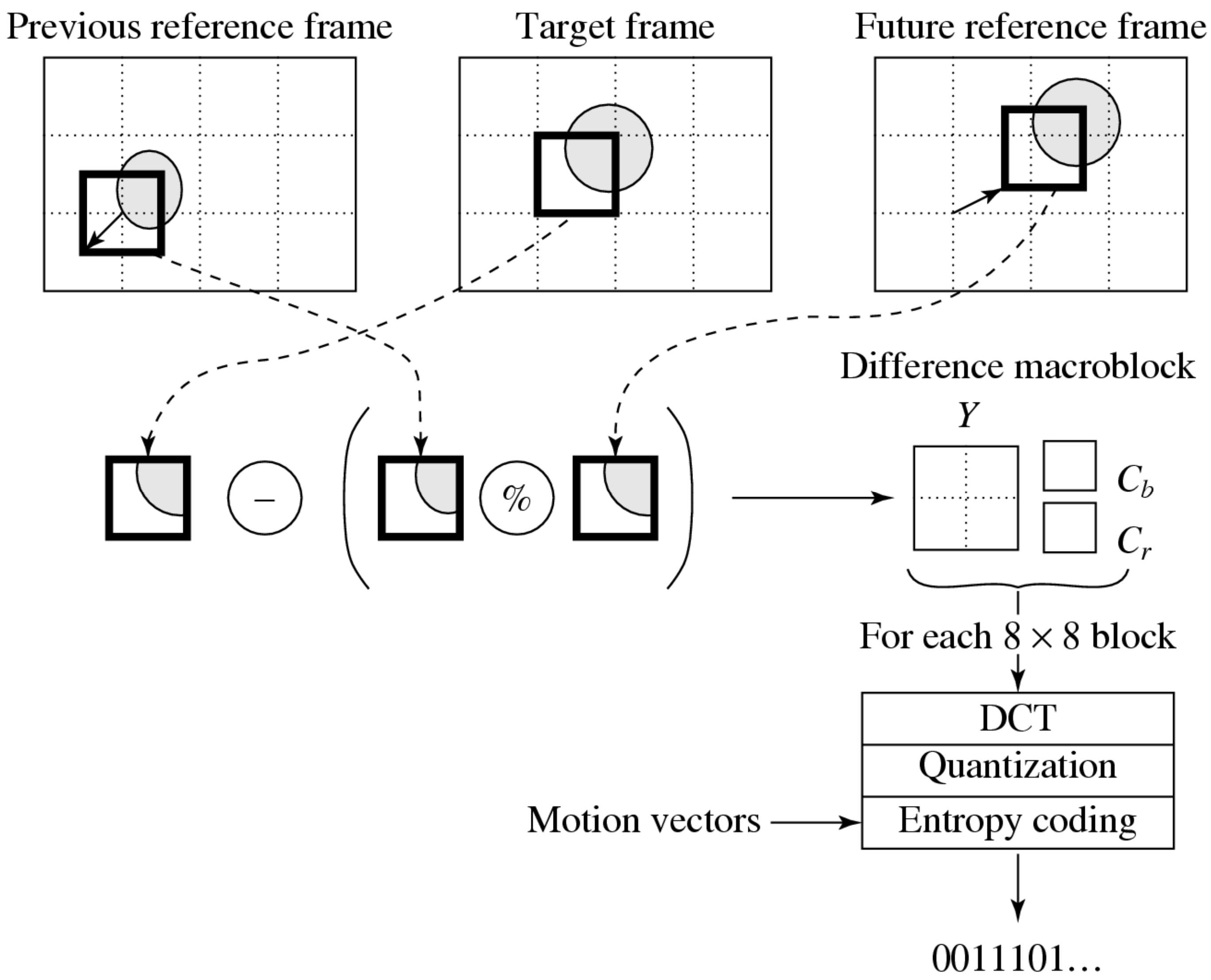
- Motion vector measures the offset or displacement vector between the location of targeted macroblock with the location of the best-matched macroblock in the reference frame.
- Prediction error measures the difference (matrix) between the targeted macroblock and the best-matched macroblock in the reference frame.
- The motion vectors are encoded using differential encoding, and the resulting symbols are Huffman coded.
- The difference matrices for prediction error (Y, C_r, C_b) are encoded using the same steps as for I-frames.
- If no match is found, the macroblock is encoded independently as macroblocks in an I-frame.

MPEG-1: Why Bidirectional Search?

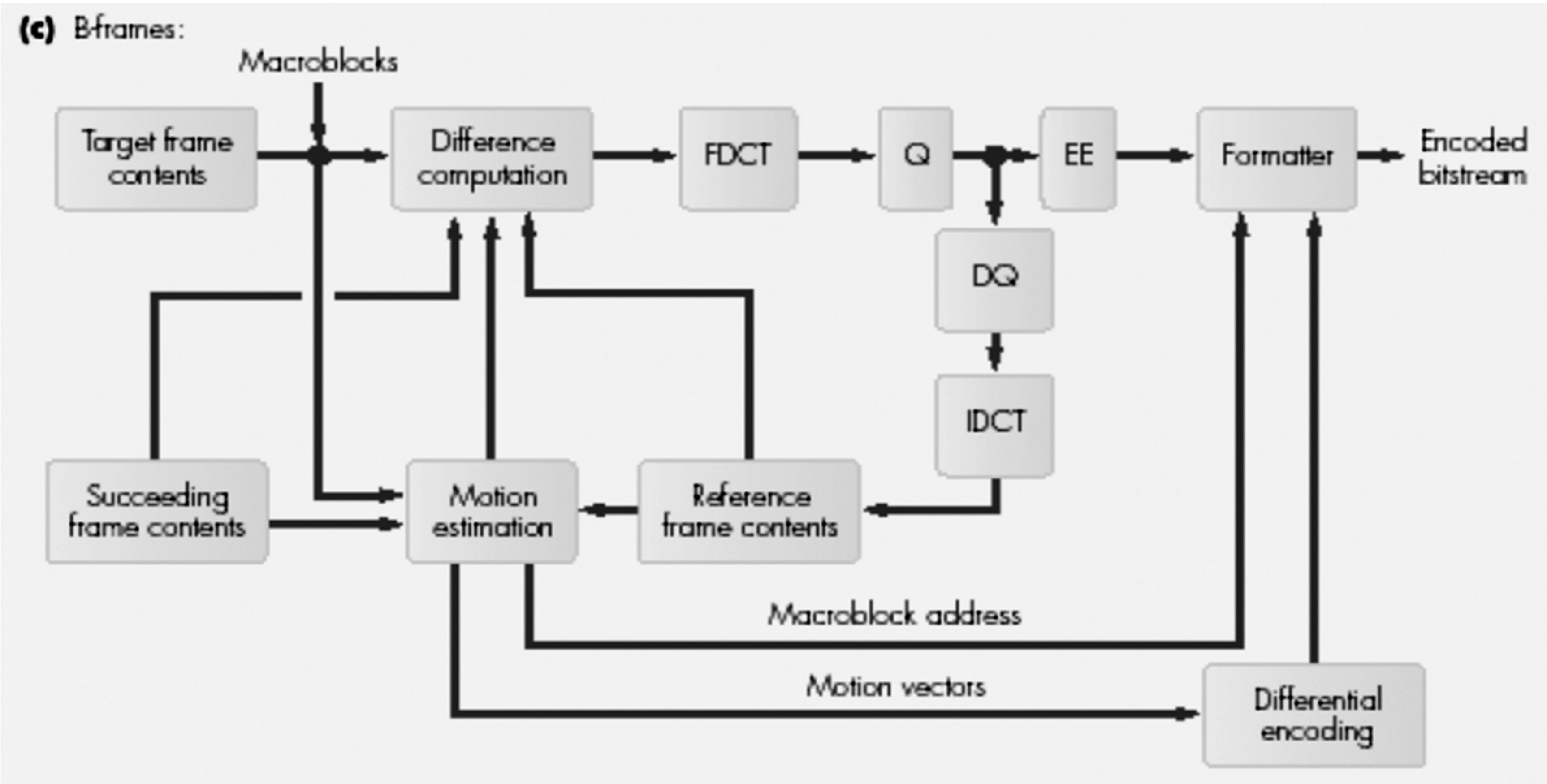
- A higher chance of finding a good match for the Macroblock (MB) of the Target frame in the previous & next reference frames.
- E.g., the MB containing part of a ball in the Target frame cannot find a good matching MB in the previous frame.
- A match however can readily be obtained from the next frame.



MPEG-1: B-Frame Coding



MPEG-1: B-Frame Encoding Flowchart



MPEG-1: B-Frame Encoding

- For each macroblock in the target frame, motion estimation and compensation is performed with respect to both preceding (previous) and succeeding (following) reference (anchor) I- or P-frame.
- This produces 2 sets of (motion vector, difference matrix) pairs, one based on preceding anchor frame, the other based on succeeding anchor frame.
- A third difference matrix is computed using the target macroblock and the mean of preceding and succeeding best-matched macroblocks.
- The set with the smallest prediction error is chosen, and it is encoded in the same way as P-frame.

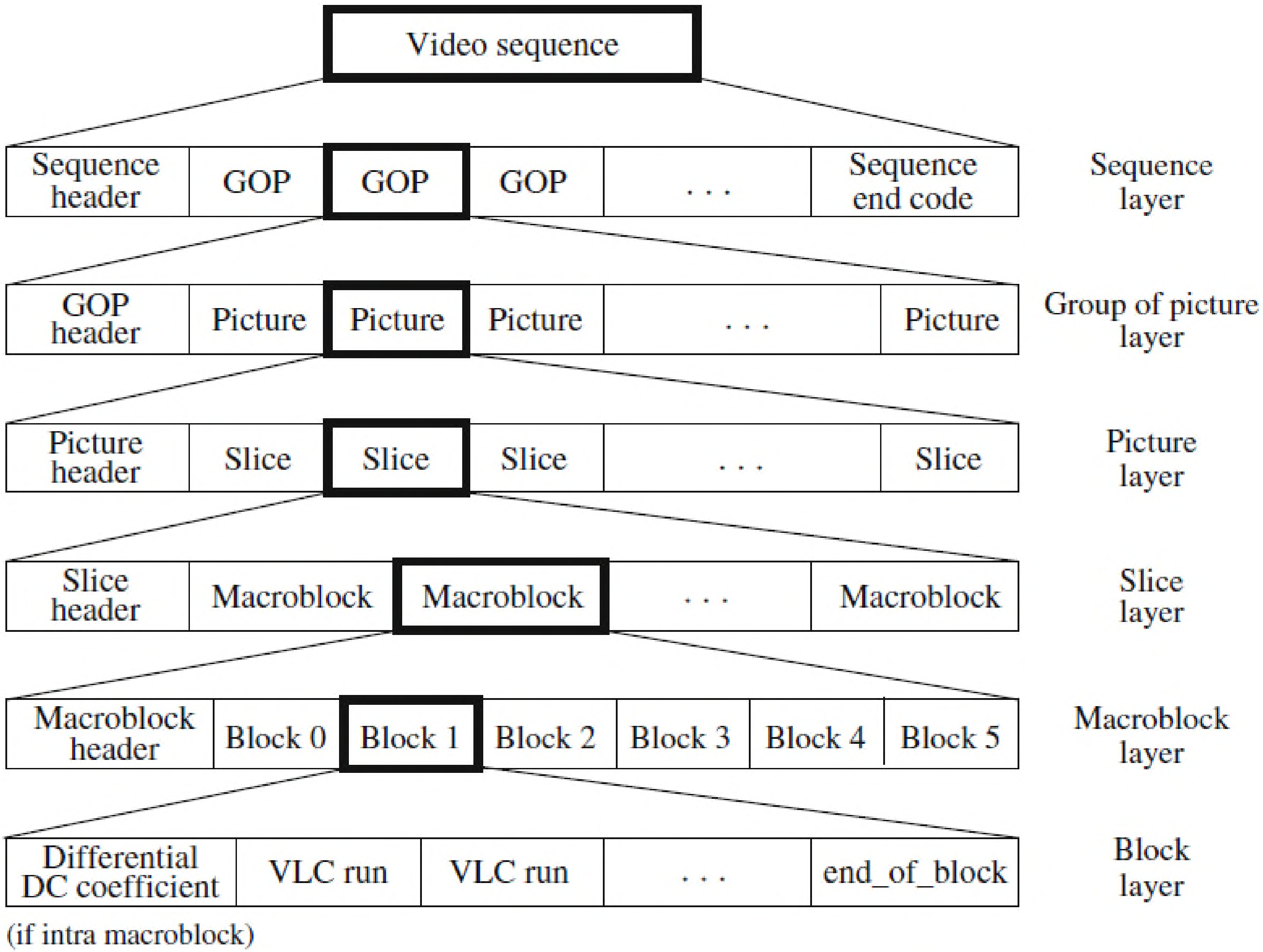
MPEG-1: Frame Size

- P-frames is significantly smaller than I-frames, because temporal redundancy is exploited.
- B-frames are even smaller than P-frames, because of the advantage of bi-directional prediction.

Table 11.4: Typical Compression Performance of MPEG-1 Frames

Type	Size	Compression
I	18kB	7:1
P	6kB	20:1
B	2.5kB	50:1
Average	4.8kB	27:1

MPEG-1: Bitstream



Exercise: MPEG-1

(a) Briefly describe the purpose of chroma subsampling in the MPEG-1 video compression.

(4 Marks)

(b) Draw a flowchart of the P-frame encoding in the MPEG-1 standard. Clearly label all the key steps in the flowchart. Briefly explain the role of entropy encoding in the process.

(9 Marks)

Solution

MPEG-1 Summary

- Intraframe coding:
  - JPEG-based.
- Interframe coding:
  - DCT-based.
  - Fixed block-size motion compensation.
- Features:
  - 1 or 2 reference frames for P or B frames, respectively.
  - Low resolution.
  - Low bitrate.

MPEG-2

MPEG-2: Overview

- Aim to address limitations of MPEG-1: e.g., low bitrate (1.5 Mbps), progressive-scan only.
- Standardized in 1995.
- Developed for digital broadcast TV (interlaced-scan) at a high bitrate (4 Mbps).
- Defined different profiles for different applications.
- Support scalable coding.

MPEG-2: Profiles & Levels

Table 11.5 Profiles and Levels in MPEG-2

Level	Simple profile	Main profile	SNR scalable profile	Spatially scalable profile	High profile	4:2:2 profile	Multiview profile
High		*			*	*	
High 1440		*		*	*	*	
Main	*	*	*		*	*	*
Low	*	*	*				

Table 11.6 Four levels in the main profile of MPEG-2

Level	Maximum resolution	Maximum fps	Maximum pixels/sec	Maximum coded data rate (Mbps)	Application
High	$1,920 \times 1,152$	60	62.7×10^6	80	Film production
High 1440	$1,440 \times 1,152$	60	47.0×10^6	60	Consumer HDTV
Main	720×576	30	10.4×10^6	15	Studio TV
Low	352×288	30	3.0×10^6	4	Consumer tape equivalent

MPEG-2: Parts

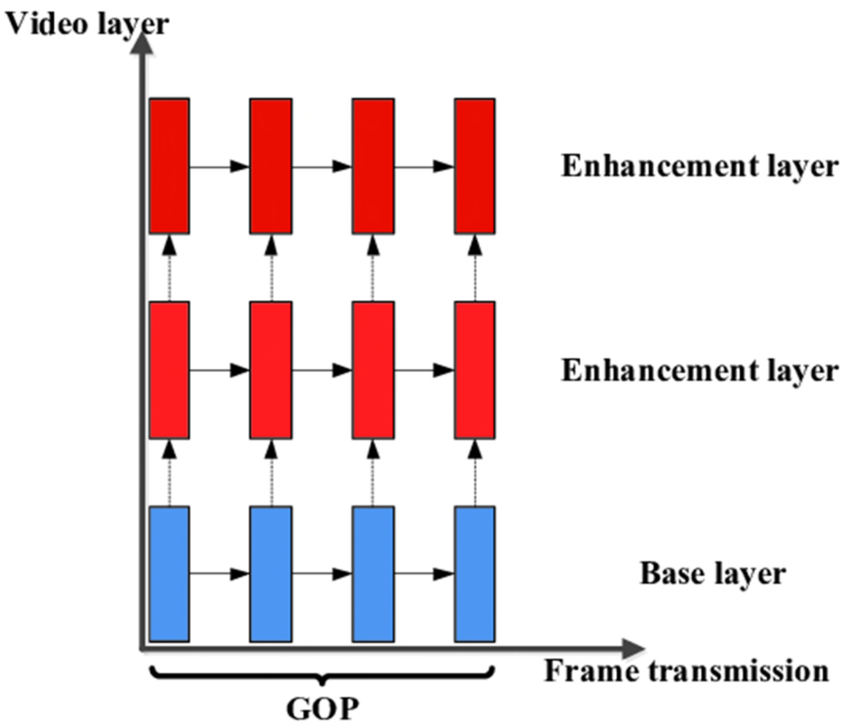
- Part-1 Systems
- Part-2 Video
- Part-3 Audio
- Part-4 Conformance
- Part-5 Reference software
- Part-6 DSM CC
- Part-7 AAC - Advanced Audio Coding
- Part-9 RTI - Real Time Interface

MPEG-2: Scalabilities

- Scalable coding is useful for MPEG-2 video transmitted over networks with following characteristics:
  - Networks with very different bitrates.
  - Networks with variable bit rate (VBR) channels.
  - Networks with noisy connections.

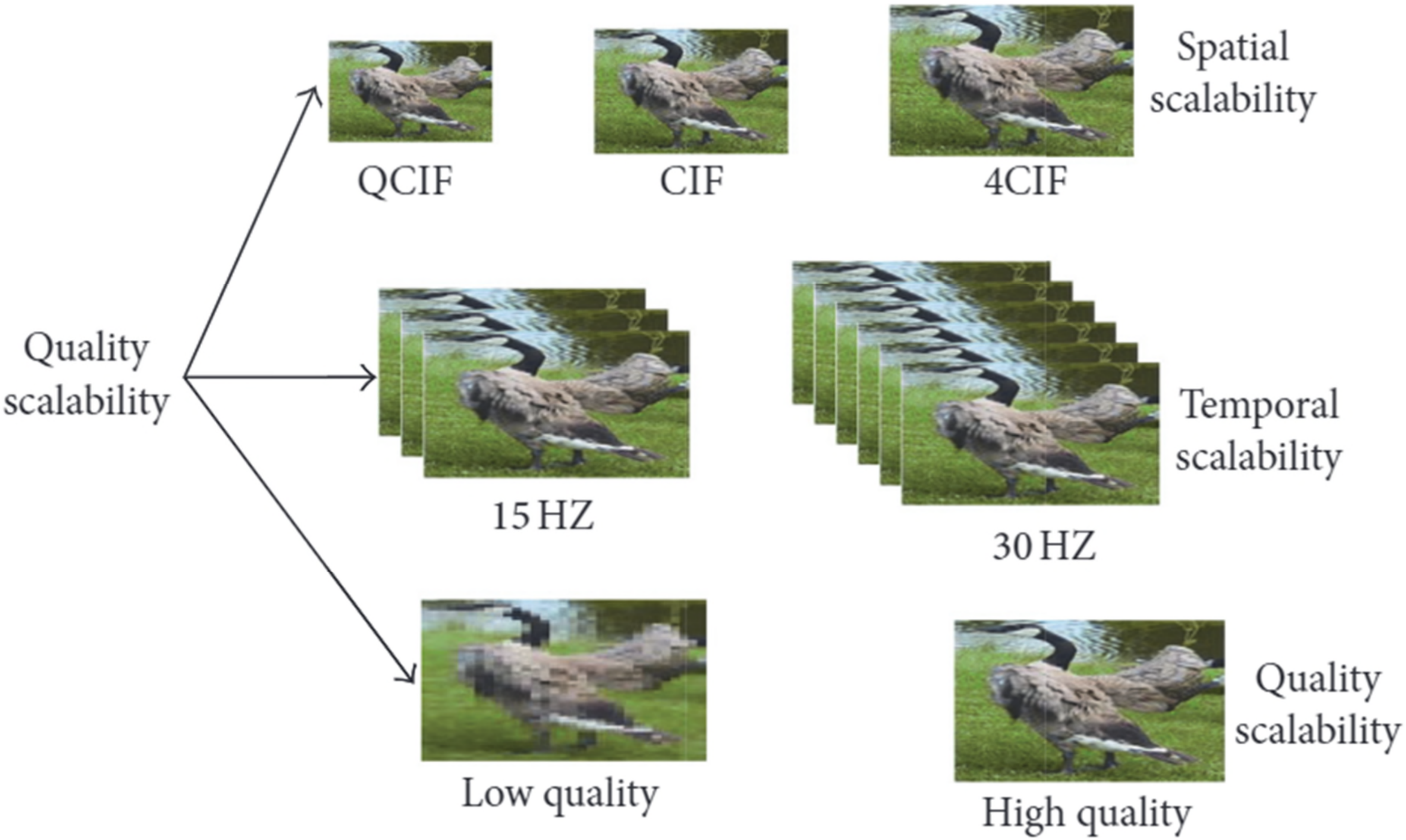
MPEG-2 Scalable Coding

- Layered coding: A base layer and one or more enhancement layers.
- The base layer can be independently encoded, transmitted and decoded to obtain basic video quality.
- Bitstreams of base layer are sent first, to give users a fast and basic view of the video, followed by enhancement layer to improve quality.
- The encoding and decoding of the enhancement layer, however, are dependent on the base layer or the previous enhancement layer.



source: https://www.researchgate.net/figure/Hierarchical-layers-for-fidelity-scalable-video-coding_fig1_334813272

Type of Scalabilities



source: https://www.semanticscholar.org/paper/Efficient-Enhancement-for-Spatial-Scalable-Video-Khairy-Hamdy/d483c84609f393257dc2874aa86604a626242cd1

Type of Scalabilities

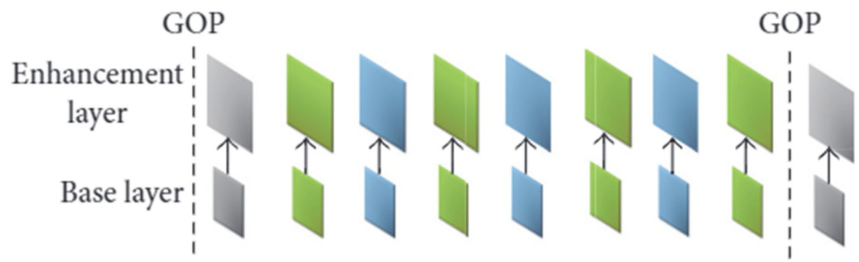


FIGURE 4: Spatial scalability.

FIGURE 5: Quality scalability.

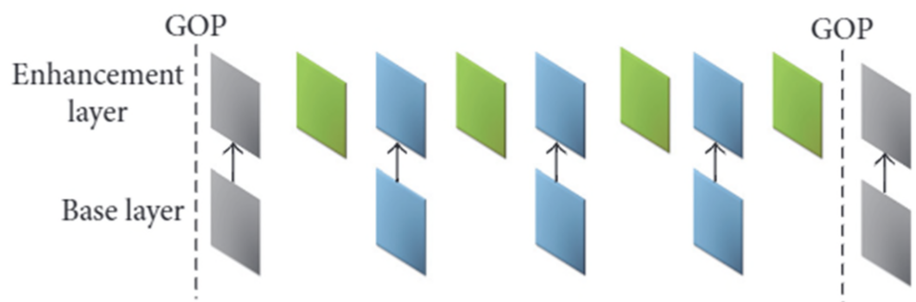


FIGURE 6: Temporal scalability.

source: https://www.semanticscholar.org/paper/Efficient-Enhancement-for-Spatial-Scalable-Video-Khairy-Hamdy/d483c84609f393257dc2874aa86604a626242cd1

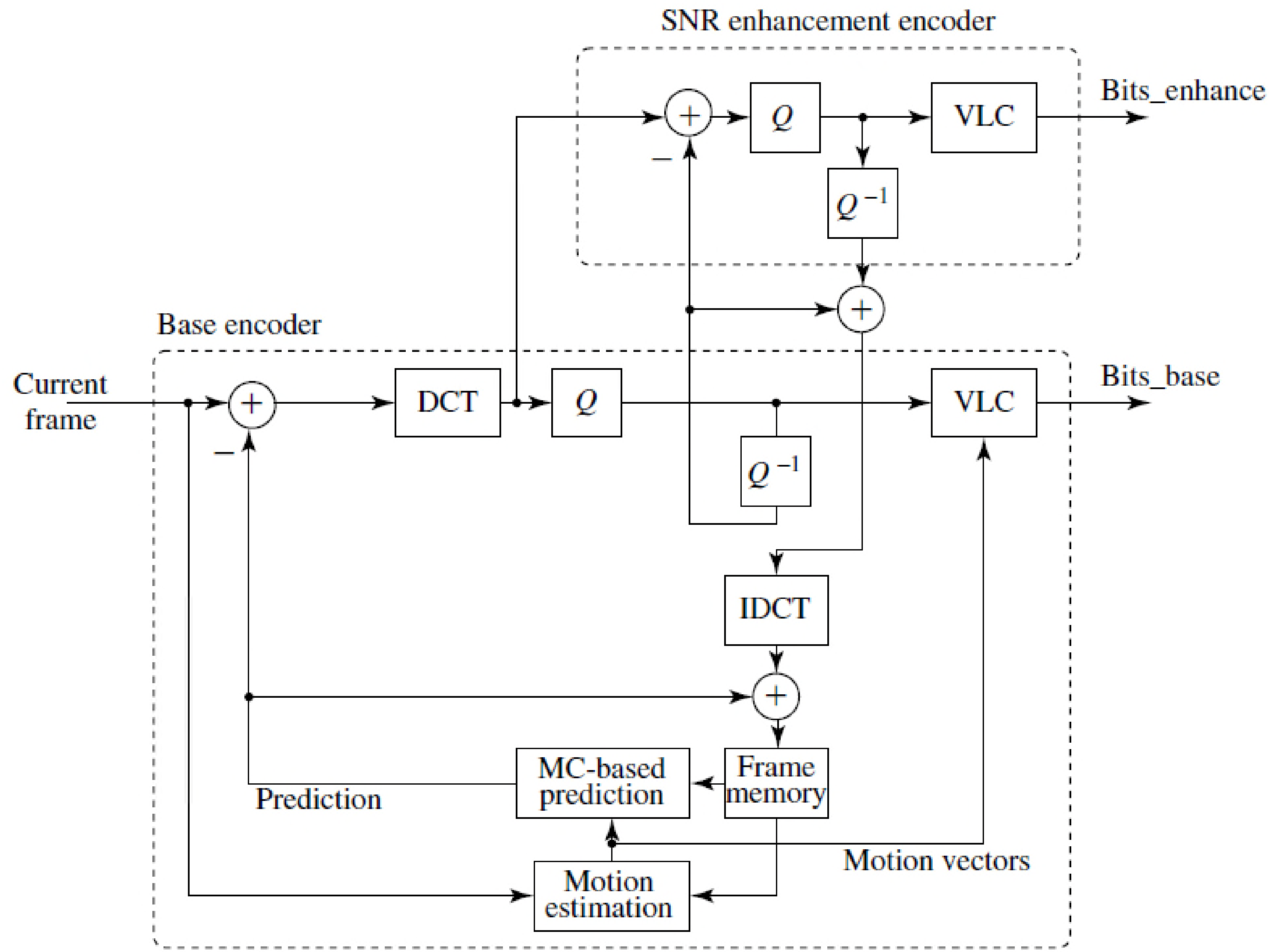
MPEG-2: Type of Scalabilities

- SNR scalability: The enhancement layer provides higher SNR.
- Spatial scalability: The enhancement layer provides higher spatial resolution.
- Temporal scalability: The enhancement layer facilitates higher frame rate.
- Hybrid scalability: This combines any two of the above three scalabilities.
- Data partitioning: Quantized DCT coefficients are split into partitions.

MPEG-2: SNR Scalability

- Enhancement over base layer to improve Signal-Noise-Ratio (SNR).
- Generate output bitstreams Bits_base and Bits_enhance at two layers:
  - Base Layer has a coarse quantization of the DCT coefficients, resulting in fewer bits and a relatively low-quality video.
  - The coarsely quantized DCT coefficients are then inversely quantized (Q^{-1}) and fed to the Enhancement Layer to be compared with the original DCT coefficient.
  - Their difference is finely quantized to generate a DCT coefficient refinement, which, after VLC, becomes the bitstream called Bits_enhance.

MPEG-2: SNR Scalability

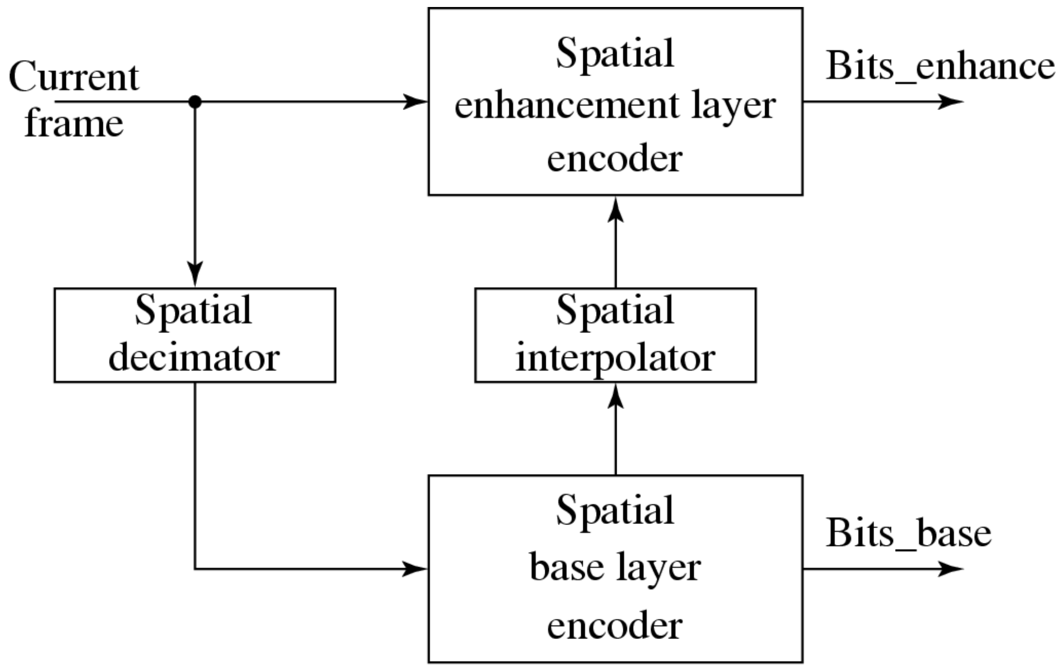


(a) Encoder

MPEG-2: Spatial Scalability

- The base layer is used to generate bitstream of reduced resolution pictures.
- When combined with the enhancement layer, pictures at the original resolution are produced.

MPEG-2: Spatial Scalability

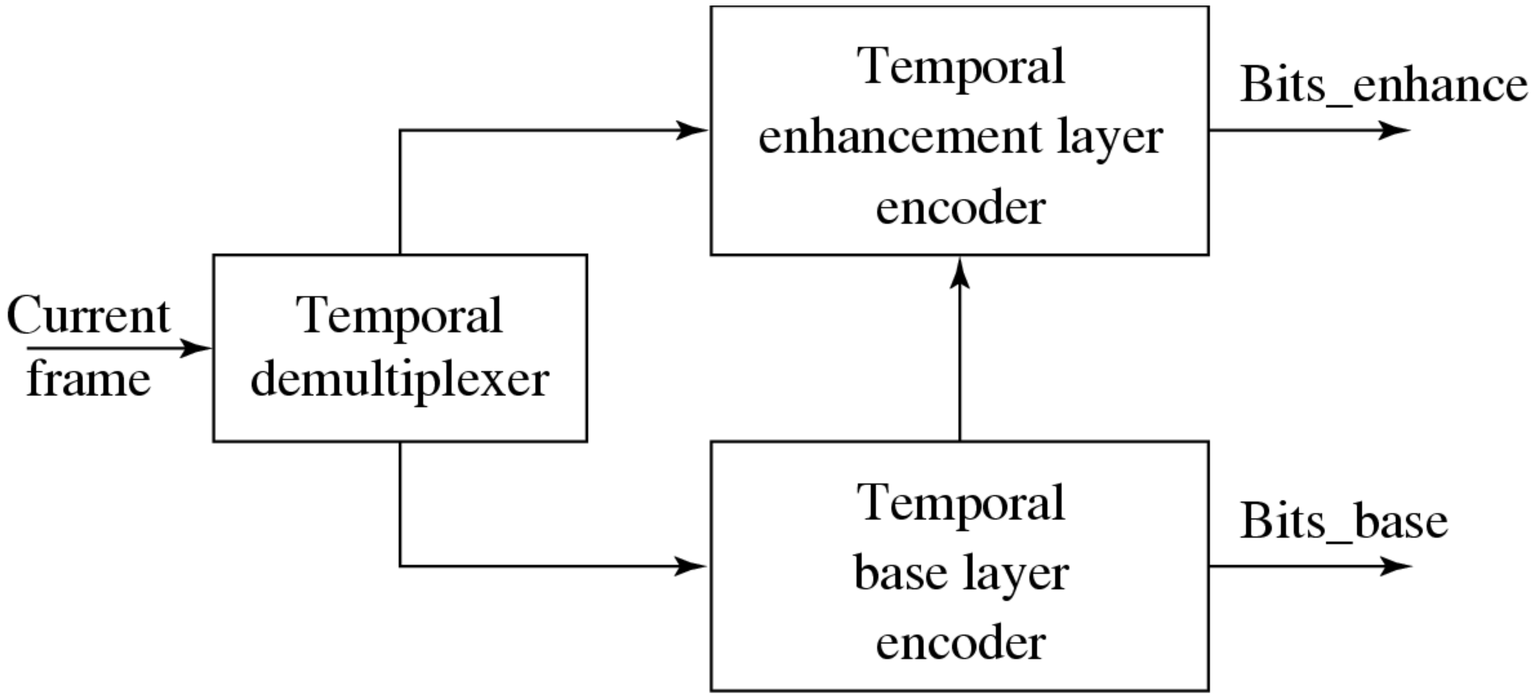


(a)

MPEG-2: Temporal Scalability

- The input video is temporally demultiplexed into two pieces, each carrying half of the original frame rate.
- Base Layer Encoder carries out the normal single-layer coding procedures for its own input video and yields the output bitstream Bits_base.

MPEG-2: Temporal Scalability



(a) Block Diagram

MPEG-2: Hybrid Scalability

- Any two of the above three scalabilities can be combined to form hybrid scalability:
  - Spatial and Temporal Hybrid Scalability.
  - SNR and Spatial Hybrid Scalability.
  - SNR and Temporal Hybrid Scalability.
- Usually, a three-layer hybrid coder will be adopted which consists of Base Layer, Enhancement Layer 1, and Enhancement Layer 2.

MPEG-2: Data Partitioning

- Base partition contains lower-frequency DCT coefficients.
- Enhancement partition contains high-frequency DCT coefficients.
- Strictly speaking, data partitioning is not layered coding:
  - Since a single stream of video data is simply divided up.
  - There is no further dependence on the base partition in generating the enhancement partition.
- Useful for transmission over noisy channels and for progressive transmission.

MPEG-2: Advantages over MPEG-1

- Better picture quality than MPEG-1 at comparable bitrate.
- Support both progressive and interlaced scanning whereas MPEG-1 supports only progressive scanning.
- Support scalability but MPEG-1 does not.
- Better resilience to bit errors. MPEG-2 can transmit video on noisy and unreliable networks with bit errors.
- Support 4:2:2 and 4:4:4 chroma subsampling to increase color quality.
- More flexible video formats, support various picture resolutions as defined by DVD and HDTV.

Exercise: MPEG-2 Scalability

(d) Briefly discuss how MPEG-2 scalability can be used in networks with variable bitrate channels.

(6 Marks)

Solution

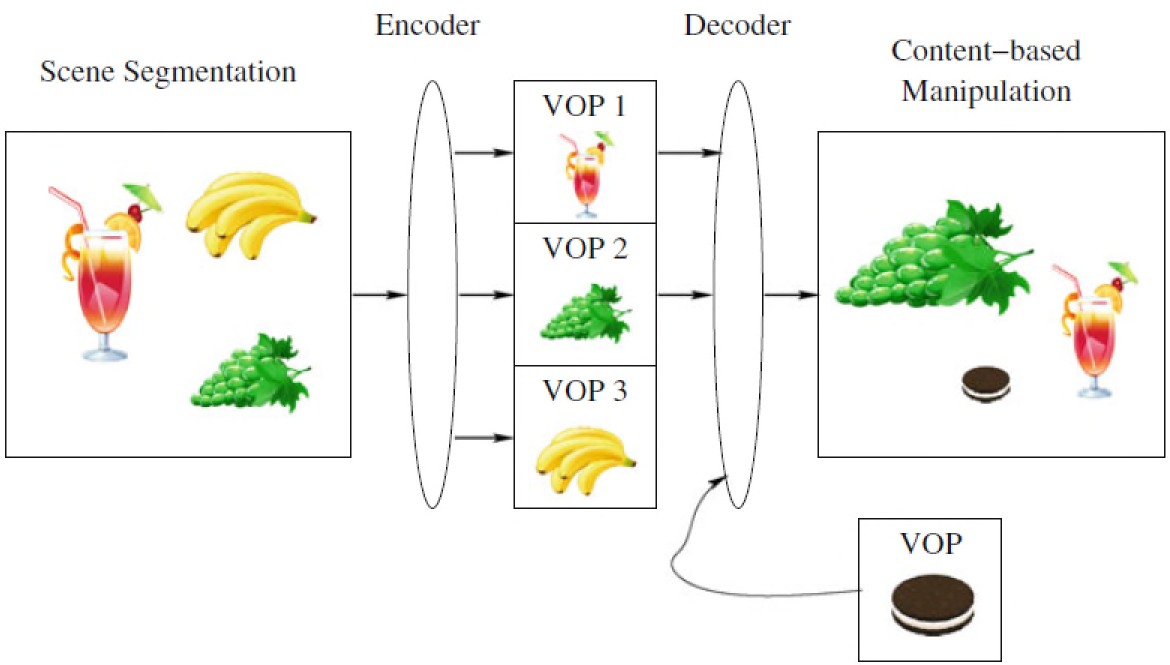
MPEG-2 Summary

- Intraframe coding:
  - JPEG-based.
- Interframe coding:
  - DCT-based.
  - Fixed block-size motion compensation.
- Features:
  - Larger range of resolutions and bitrates.
  - Support different profiles and levels.
  - Scalable coding.
  - More chroma subsampling schemes.

MPEG-4

MPEG-4: Overview

- MPEG-1 and -2 employ block-based coding techniques.
- MPEG-4 adopts object-based coding, beneficial for digital video composition, manipulation, indexing, and retrieval.



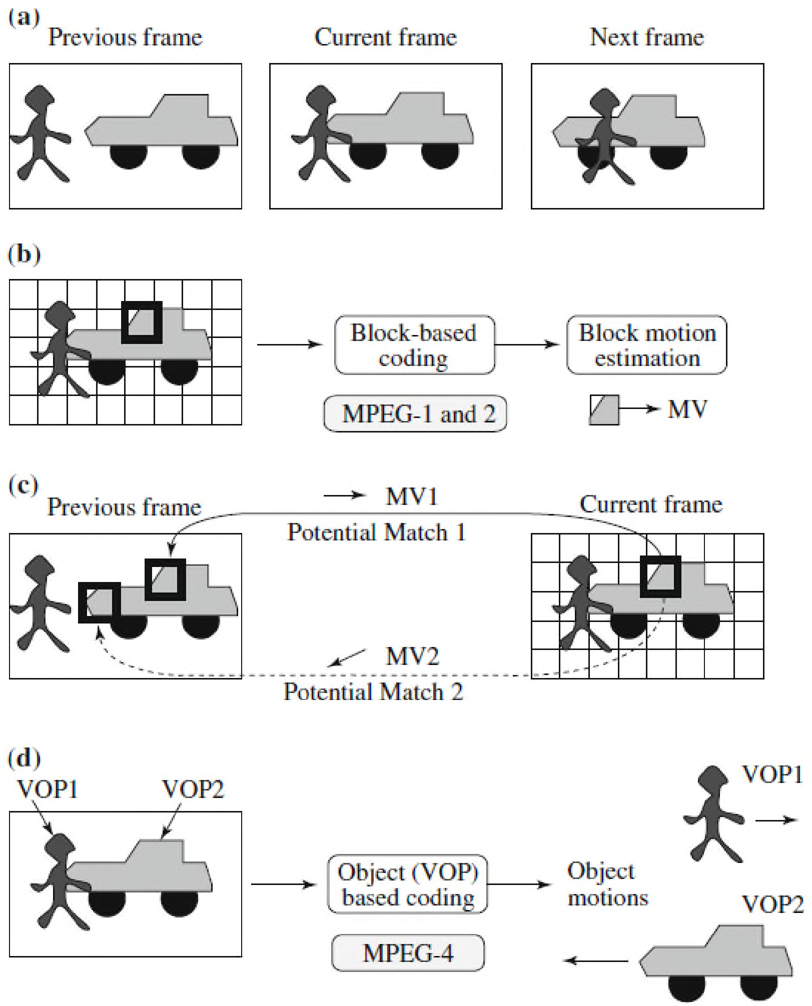
Composition and Manipulation of MPEG-4 Videos. (VOP = Video object plane)

MPEG-4: Overview

- Finalized in October 1998.
- Bitrate: 5 kbps to 10 Mbps.
- Video-object oriented hierarchical description of a scene in MPEG-4 bitstreams.
- Compose media objects to create desirable audiovisual scenes.
- Multiplexing and synchronizing the bitstreams for these media data entities.

Video Object-Based Coding vs Frame-Based Coding

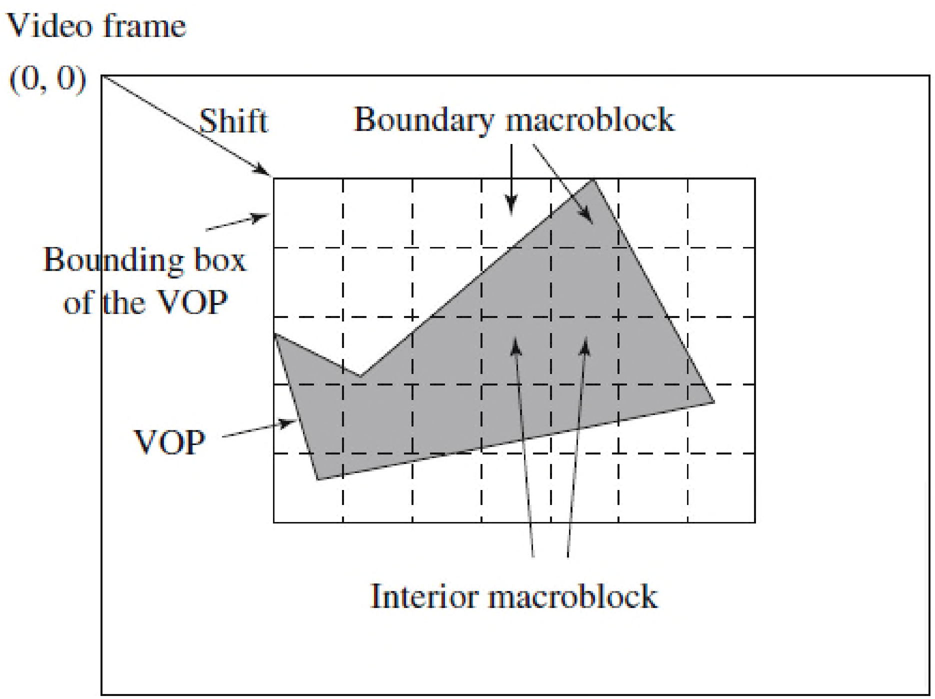
- MPEG-1, MPEG-2: frame-based (block-based) coding.
- Figure (c) illustrates a possible example in which both potential matches yield small prediction errors for block-based coding.
- Figure (d) shows that each VOP is of arbitrary shape and ideally will obtain a unique motion vector consistent with the actual object motion.



Frame block-based coding vs object-based coding:
(a) Video sequence, (b) MPEG-1 and 2 block-based coding , (c) Two potential matches in MPEG-1 and 2, (d) Object-based coding in MPEG-4

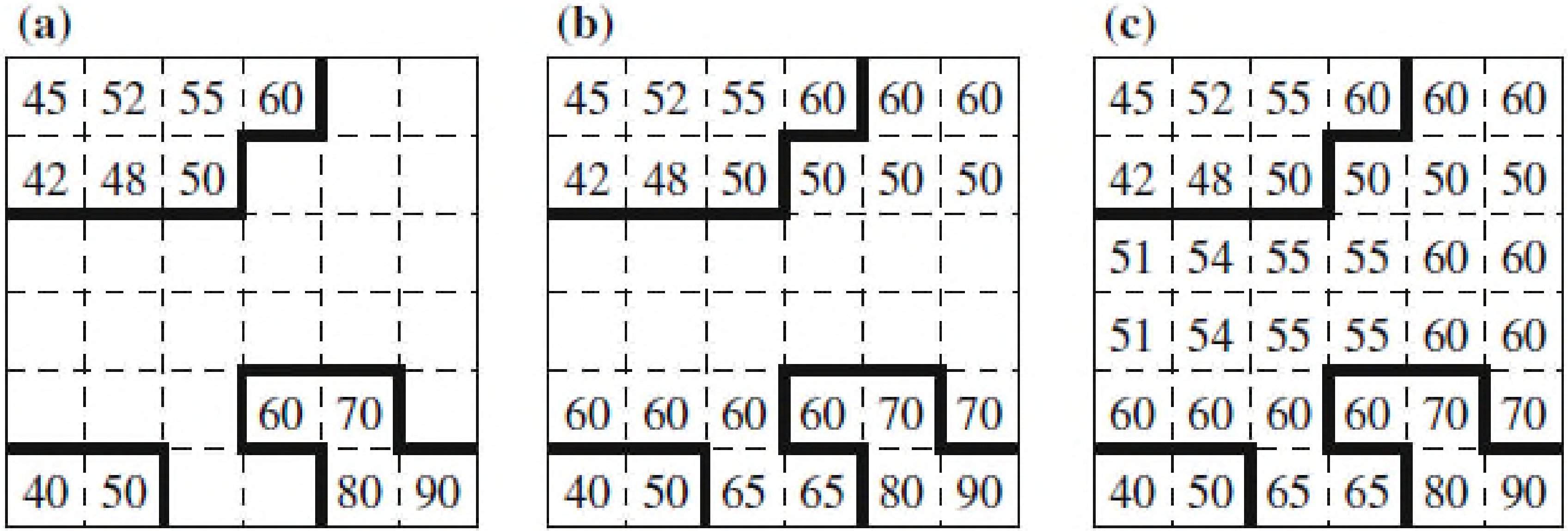
MPEG-4: VOP Coding

- Involve motion estimation, motion compensation, and coding of the prediction error.
- MPEG-4 defines a rectangular bounding box for each VOP, and each VOP is divided into many macroblocks.
- Macroblocks entirely within the VOP are interior macroblocks.
- Macroblocks straddle the boundary called boundary macroblocks.



MPEG-4: VOP Coding

- Interior macroblocks perform motion compensation in the same manner as in MPEG-1 and 2.
- Boundary macroblocks requires a preprocessing step of padding to the reference VOPs prior to motion estimation



An example of Repetitive Padding in a boundary macroblock of a Reference VOP: (a) Original pixels within the VOP. (b) After Horizontal Repetitive Padding. (c) Followed by Vertical Repetitive Padding.

MPEG-4: Advantages

- Integrate natural and synthetic content, in the form of objects.
- Support 2D and 3D contents, and users' interactivity.
- Coding at very low rates (2 Kbit/s for speech, 5 Kbit/s for video) to very high rates (5 Mbit for transparent quality video, 64 Kbit/s per channel for CD quality audio).
- Support for management and protection of intellectual property.

MPEG-4 Summary

- Coding technique: object-based coding
- Features:
  - Integrate natural and synthetic content.
  - Support video composition, manipulation, etc.

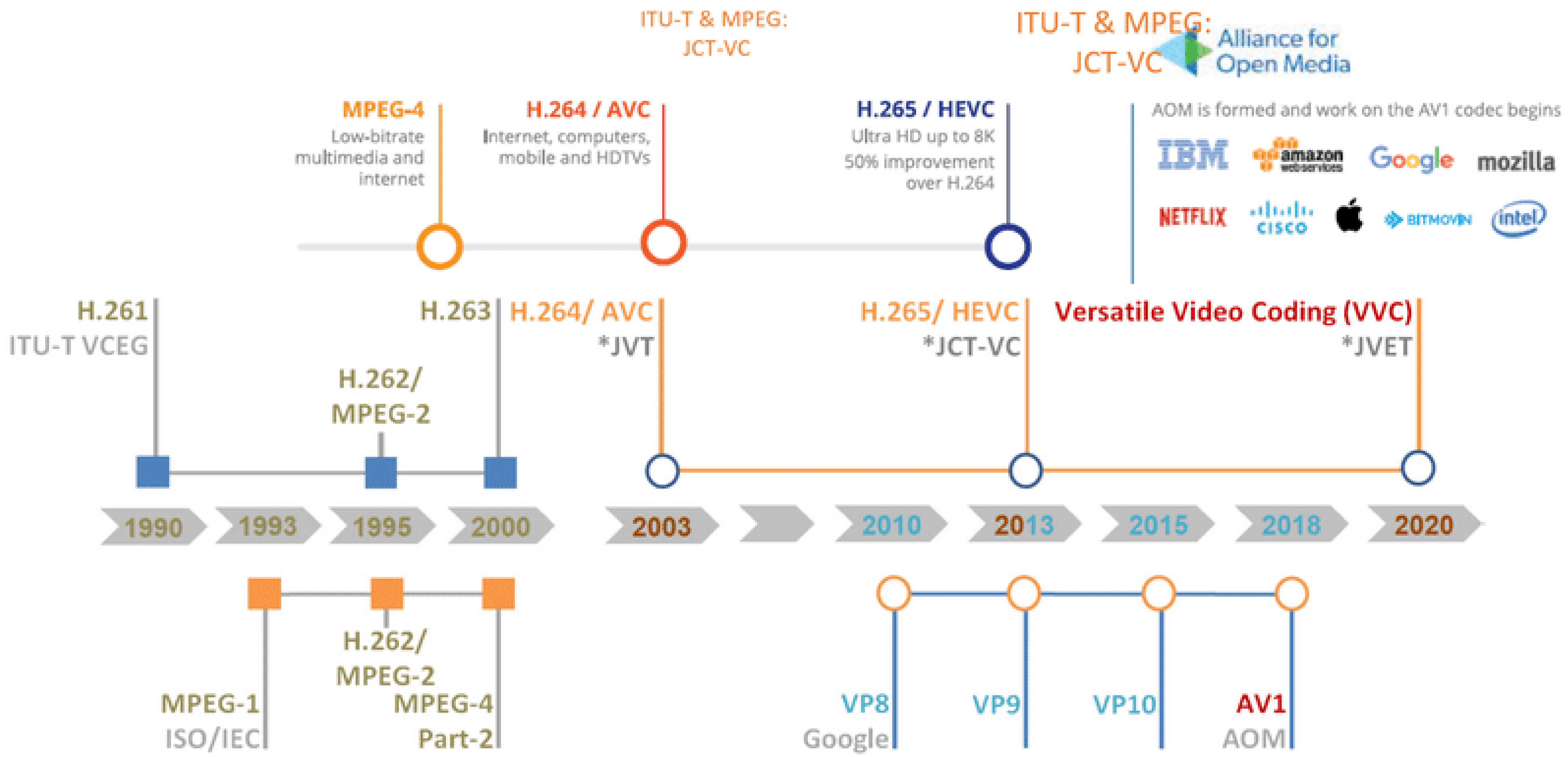
MPEG-7

MPEG-7

- Goal: audiovisual content-based retrieval.
- Target multimedia applications: content creation and consumption of multimedia data.
- Published in September 2001.
- Formal name: Multimedia Content Description Interface.
- Not a video coding standard.

H.26x Standards

Historical Timeline of H.26x Standards



https://www.behance.net/gallery/89647147/Video-Codec-Timeline

H.261

H.261: Overview

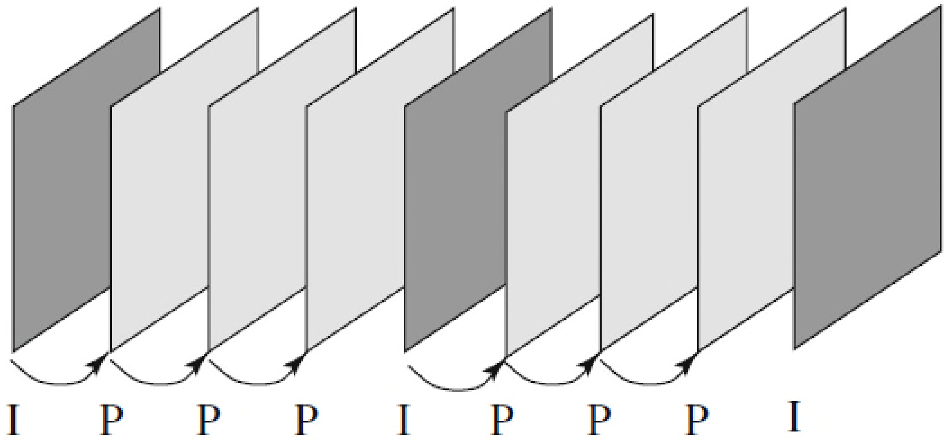
- Developed by ITU-T in 1990.
- Designed for video conferencing & other audiovisual services.
- Its principle of motion compensation (MC)-based compression is retained in all later standards.
- H.261 support YUV 4:2:0 video formats.

Table 10.2 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

H.261: Overview

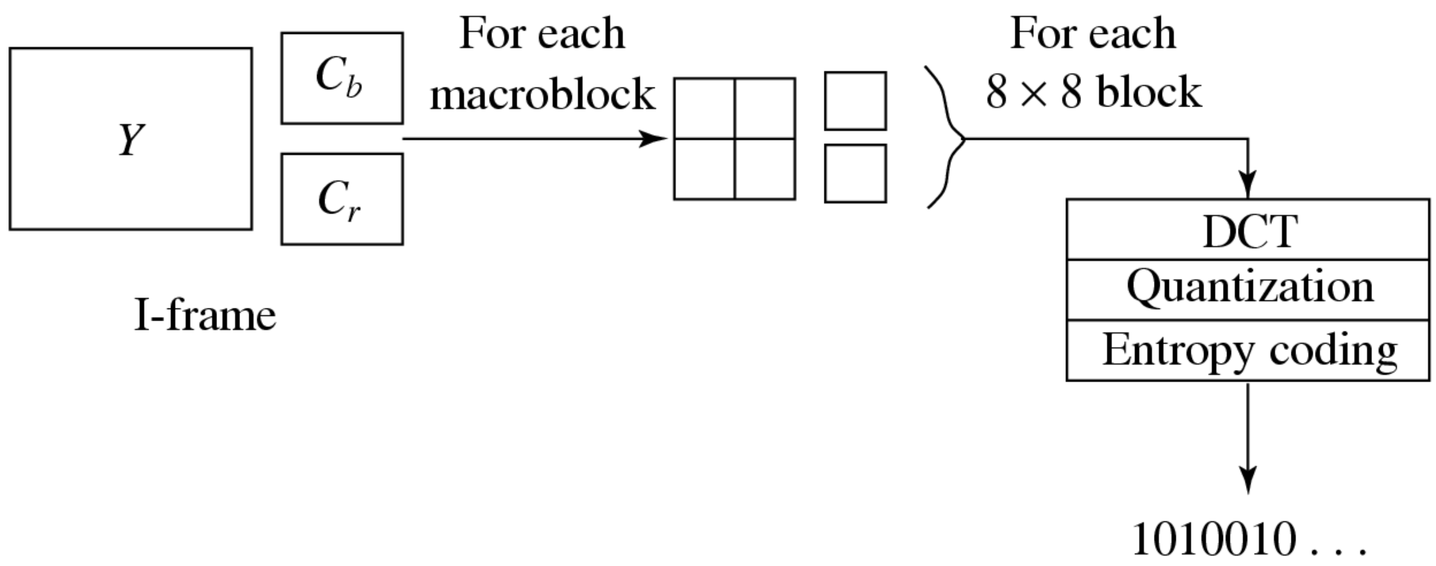
- 2 frame types: Intra-frames (I-frames) and Inter-frames (P-frames).
- I-frames are treated as independent images.
- P-frames refer to macroblocks in the preceding I- or P-frame.



H.261 Frame sequences

H.261: I-frame Coding

- A macroblock contains 4Y, 1Cb, and 1Cr of 8 x 8 pixel blocks for 4:2:0 chroma subsampling.
- Macroblocks are of size 16 x 16 pixels for the Y frame, and 8 x 8 for Cb and Cr frames.
- For each 8 x 8 block, a DCT transform is applied, the DCT coefficients then go through quantization, zigzag scanning, and entropy coding.

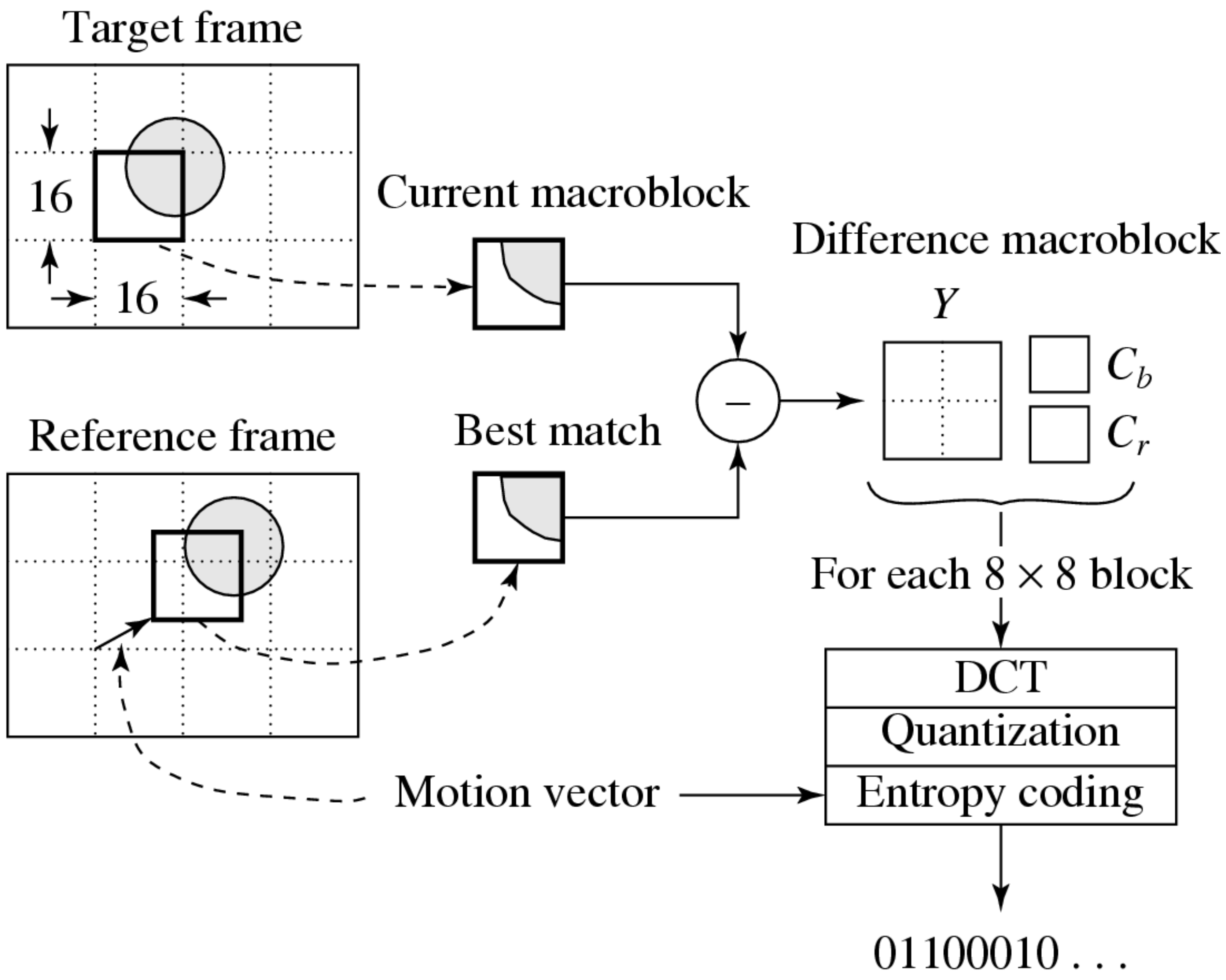


I-frame Coding

H.261: P-frame Coding

- P-frame coding is based on motion compensation
  - For each macroblock in the Target frame, a motion vector is estimated by the search method.
  - After the prediction, a difference macroblock is computed to measure the prediction error.
  - Each of these 8 x 8 blocks goes through DCT, quantization, zigzag scanning and entropy coding.
- P-frame coding encodes the difference macroblock.
- Motion vector difference (MVD) is sent using entropy coding.

H.261: P-frame Coding



H.261 P-frame Coding Based on Motion Compensation

H.261: Quantization

- Use a constant step size for all DCT coefficients within a macroblock.
- Let DCT and QDCT represent DCT coefficients before and after quantization.
- DC coefficient of Intra mode:

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

- Other coefficients:

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

scale: an integer in the range of [1, 31].

H.261 Summary

- Intraframe coding:
  - Similar to JPEG-based.
- Interframe coding:
  - DCT-based.
  - Fixed block-size motion compensation.
- Features:
  - I and P frame types only.
  - Low resolution.
  - Low bitrate.
  - Target video conferencing applications.

H.262

H.262: Overview

- Jointly developed by ITU-T Study Group 16 Video Coding Experts Group (VCEG) and ISO/IEC Moving Picture Experts Group (MPEG).
- Standardized as MPEG-2 Part 2 (ISO/IEC 13818-2) in 1995.
- H.262 /MPEG-2 includes three basic types of coded frames:
  - Intra-coded frames (I-frames)
  - Predictive-coded frames (P-frames)
  - Bidirectionally-predictive-coded frames (B-frames)

H.263 / H.263+

H.263: Overview

- Basis for the development of MPEG-4 Part 2.
- H.263 standardized in 1996, H.263+ in 1998.
- Improved quality compared to H.261 at lower bit rate to enable video conferencing.
- H.263+ enables sub-pixel motion vectors up to 1/8-th pixel accuracy for improved compression.

H.263: Supported Formats

Video Formats Supported by H.263

Video format	Luminance image resolution	Chrominance image resolution	Bit-rate (Mbps) (if 30 fps and uncompressed)	Bit-rate (kbps) BPPmaxKb (compressed)
sub-QCIF	128×96	64×48	4.4	64
QCIF	176×144	88×72	9.1	64
CIF	352×288	176×144	36.5	256
4CIF	704×576	352×288	146.0	512
16CIF	$1,408 \times 1,152$	704×576	583.9	1024

H.263+: Overview

- Additional flexibility in terms of source formats, aspect ratios, etc.
- Support Temporal, SNR, and Spatial scalabilities.
- Include deblocking filters in the coding loop to reduce blocking effects.

Recent Standards: H.264, H.265, H.266

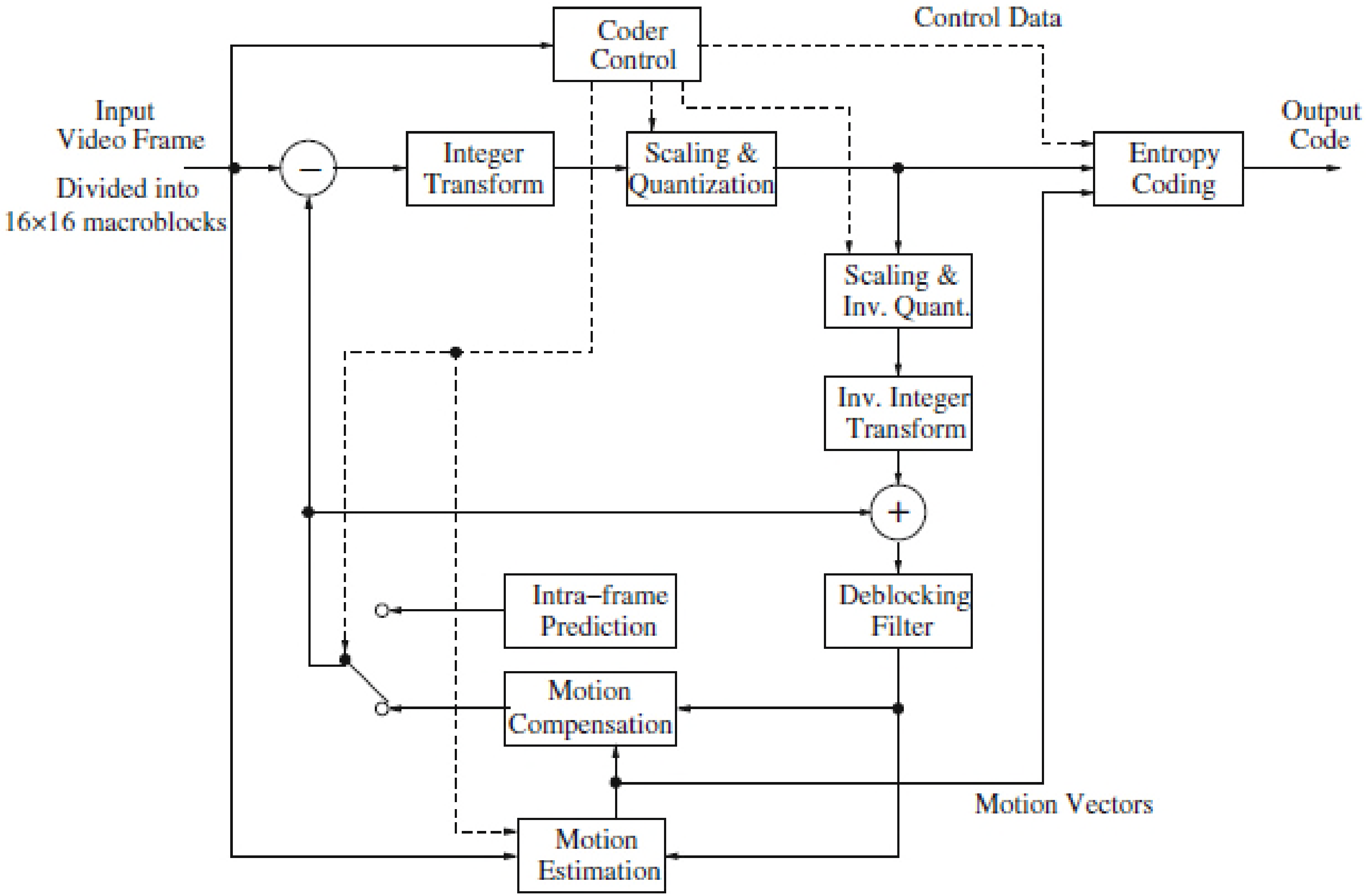
- Jointly developed by ISO/IEC MPEG and ITU-T VCEG (Video Coding Experts Group).
- Objective:
  - Improve coding efficiency.
  - Manage increases in video resolution and frame rate, from HD to UHD video.
  - H.266 was developed to further address emerging applications of high dynamic range (HDR) video, 360◦ video, and others (e.g., panoramic video).

H.264

H.264: Overview

- Standardized by the ITU-T VCEG and the ISO/IEC MPEG in 2003.
- Also known as Advanced Video Coding (AVC) or MPEG-4 Part 10.
- Provide up to 50 % better compression efficiency than H.262 and up to 30% better than H.263.
- Application: Internet video, computers, HDTV broadcast, Blu-ray discs, mobile and portable devices.
- Improve motion compensation with variable block size, multi-reference frames.

H.264: Encoder



H.264: Main Features (1)

- Integer transform in 4×4 blocks. Low complexity, no drifting.
- Variable block-size motion compensation, from 16×16 to 4×4 in luma images.
- Quarter-pixel accuracy in motion vectors, accomplished by interpolations.
- Multiple reference picture motion compensation.
- Directional spatial prediction for intra frames.
- In-loop deblocking filtering.

H.264: Main Features (2)

- Context-Adaptive Variable Length Coding (CAVLC) and Context-Adaptive Binary Arithmetic Coding (CABAC)
- More robust to data errors and data losses.
- Employ hybrid coding
  - Intra-frame spatial prediction.
  - Inter-frame motion prediction.
  - Transform coding on residual errors.

H.264: Motion Compensation

- Variable Block-Size Motion Compensation.
- Two types: 16 × 16 macroblock (M Types) and 8 × 8 partition (8 × 8 Types).
- By default, the macroblocks are of the size 16×16.
- Can be further divided into smaller partitions.

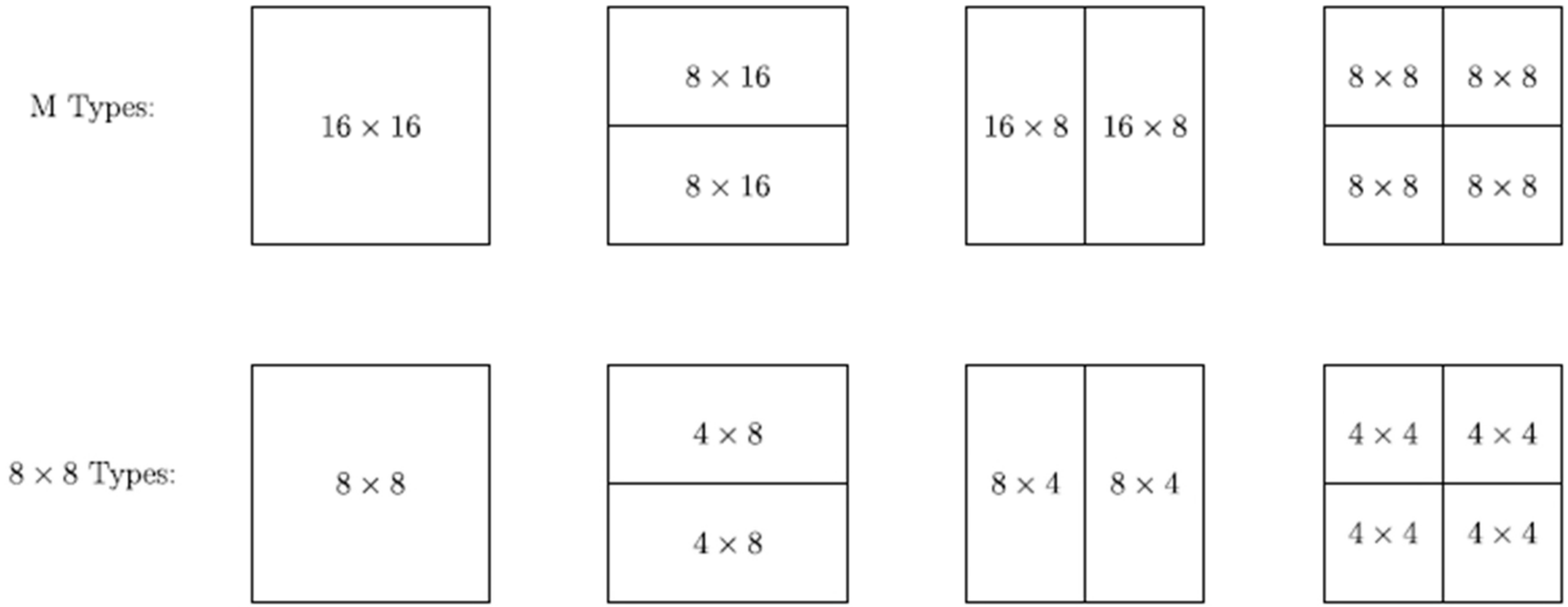


Fig. 12.2 Segmentation of the macroblock for motion estimation in H.264. Top: Segmentation of the macroblock. Bottom: Segmentation of the 8 × 8 partition

H.264: Motion Compensation

- Accuracy of motion compensation is of quarter-pixel precision in luma images.
- Pixel values at half-pixel and quarter-pixel positions can be derived by interpolation.
- Pixel values (*b* and *h*) at half-pixel positions are estimated using 6-tap filters:

$$b_1 = E - 5F + 20G + 20H - 5I + J$$
$$h_1 = A - 5C + 20G + 20M - 5R + T$$

$$b = (b_1 + 16) \gg 5$$
$$h = (h_1 + 16) \gg 5.$$

- Note that $(b_1 + 16) >> 5$ is equivalent to $b = \text{round}(b_1/2^5) = \text{round}(b_1/32)$.
- b and h are clipped to image range, e.g., 0-255.

H.264: Motion Compensation

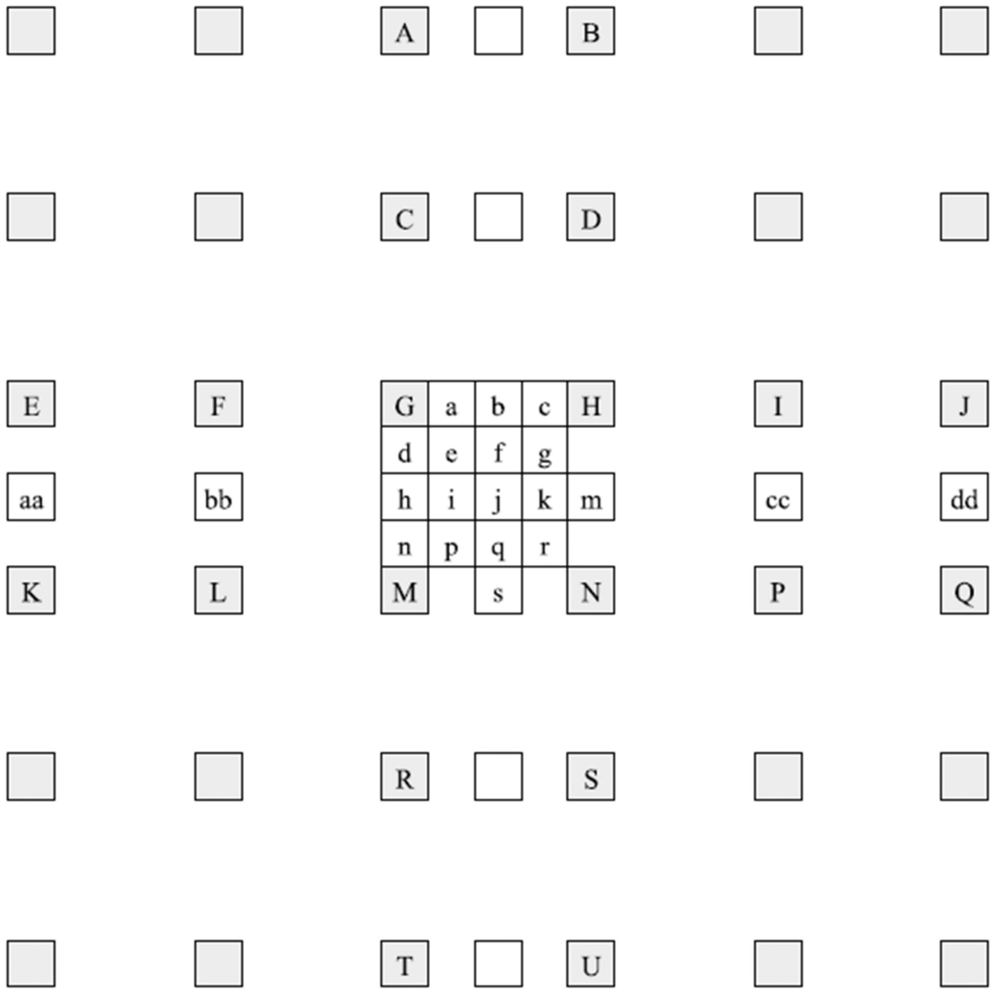


Fig. 12.3 Interpolation for fractional samples in H.264. Upper-case letters indicate pixels on the image grid. Lower-case letters indicate pixels at half-pixel and quarter-pixel positions

Step 1

$$b_1 = E - 5F + 20G + 20H - 5I + J$$
$$h_1 = A - 5C + 20G + 20M - 5R + T$$

$$b = (b_1 + 16) \gg 5$$
$$h = (h_1 + 16) \gg 5.$$

Step 2

$$j_1 = aa_1 - 5bb_1 + 20h_1 + 20m_1 - 5cc_1 + dd_1$$
$$j = (j_1 + 512) \gg 10.$$

Step 3

$$a = (G + b + 1) \gg 1$$

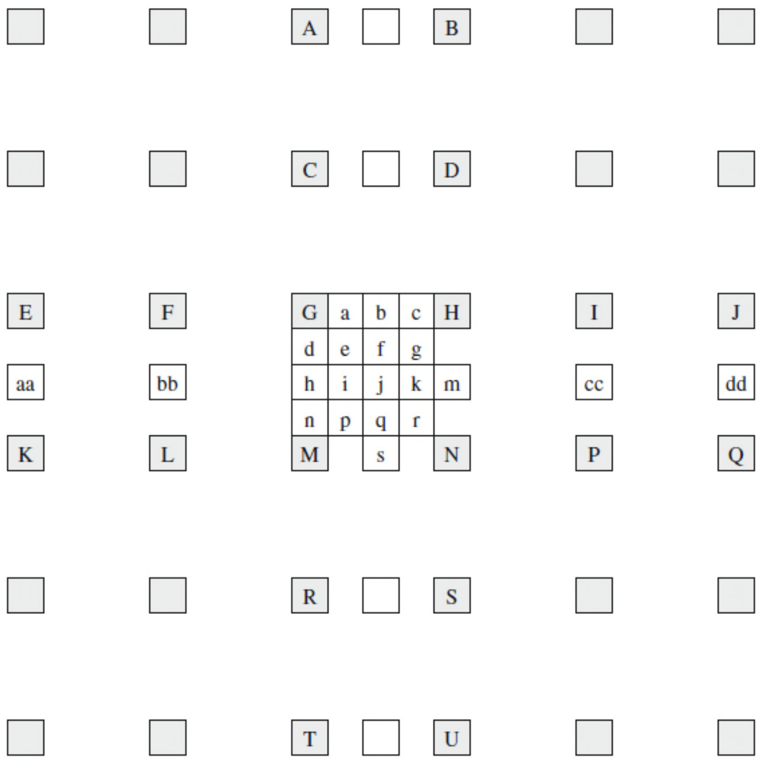
Step 4

$$e = (b + h + 1) \gg 1$$

Exercise: H.264 Motion Compensation

Assume the following pixel values: A=50, B=60, C=70, D=80, E=10, F=20, G=30, H=40, I=50, J=60, K=70, L=80, M=90, N=100, P=120, Q=130, R=10, S=20, T=30, U=40.

Find the half-pixel values b & h, and quarter-pixel values a & e (values range from 0 to 255).



half-pixel

$$b_1 = E - 5F + 20G + 20H - 5I + J \qquad b = (b_1 + 16) \gg 5$$

$$h_1 = A - 5C + 20G + 20M - 5R + T \qquad h = (h_1 + 16) \gg 5.$$

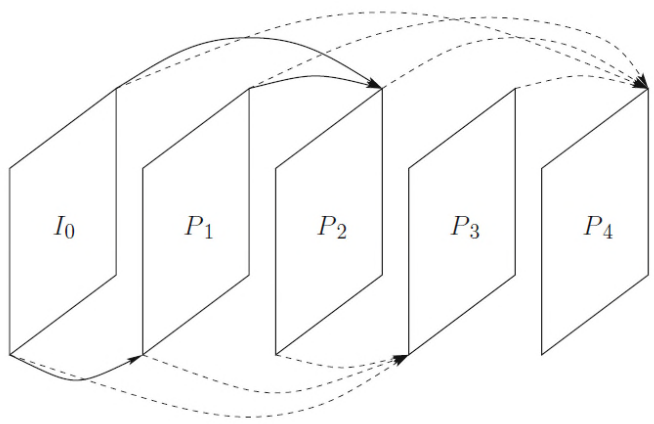
quarter-pixel $\quad a = (G + b + 1) \gg 1$

quarter-pixel $\quad e = (b + h + 1) \gg 1$

Solution

H.264: Additional Options in GOP

- No B-frames
  - Prediction of macroblocks in B-frames incurs more delay and storage.
  - In this option, only I- and P-frames are allowed.
  - Compression efficiency is relatively low.
  - More suitable for certain applications, e.g., videoconferencing.
  - Compatible with Baseline Profile or Constrained Baseline Profile of H.264.
- Multiple reference frames
  - To find the best match for each macroblock in P-frames, H.264 allows up to N reference frames.
  - Improve the compression efficiency.
  - Require much more computation.



H.264: Integer Transform

- DCT causes prediction shift because of floating point calculation and rounding errors in transform and inverse transform.
- It could be accumulated, causing a large error.
- In H.264, integer transform is used to address the issue.
- H.264 provides a quantization scheme with nonlinear step-sizes to obtain accurate rate control.

H.264: Integer Transform

- 2D DCT can be realized by two consecutive 1D transforms.
- Let **f** be the 4 × 4 input matrix, F be the DCT transformed data, and T be the DCT-matrix:

$$\mathbf{F} = \mathbf{T} \times \mathbf{f} \times \mathbf{T}^T$$

- The 4 x 4 DCT matrix is given by:

$$\mathbf{T_4} = \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix}$$

$$a = 1/2,$$

$$b = \sqrt{\tfrac{1}{2}} \cos \tfrac{\pi}{8}$$

$$c = \sqrt{\tfrac{1}{2}} \cos \tfrac{3\pi}{8}$$

H.264: Integer Transform

- To derive a scaled 4 × 4 integer transform, we can simply scale the entries of T_4 up and round them to nearest integers

$$\mathbf{H} = \text{round}(\alpha \cdot \mathbf{T_4})$$

- Let $\alpha = 2.5$, it yields 4 x 4 integer DCT matrix

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}$$

H.264: Integer Transform

- **H** is orthogonal. However, its rows no longer have the same norm.
- Normalization step is postponed into the quantization step.
- Inverse transform:

$$\mathbf{H_{inv}} = \begin{bmatrix} 1 & 1 & 1 & 1/2 \\ 1 & 1/2 & -1 & -1 \\ 1 & -1/2 & -1 & 1 \\ 1 & -1 & 1 & -1/2 \end{bmatrix}$$

H.264: Quantization and Scaling

- Let **f** be 4 × 4 input matrix, and $\hat{\mathbf{F}}$ quantized transform output.
- The forward integer transform, scaled and quantized:

$$\hat{\mathbf{F}} = \text{round}\left[(\mathbf{H} \times \mathbf{f} \times \mathbf{H}^T) \cdot \mathbf{M_f}/2^{15}\right]$$

where "×" denotes matrix multiplication, "·" denotes element-by-element multiplication, and $\mathbf{M_f}$ is the 4 × 4 quantization matrix derived from matrix **m** and quantization parameter QP.

H.264: Quantization and Scaling

For $0 \leq QP < 6$, we have

$$\mathbf{M_f} = \begin{bmatrix} \mathbf{m}(QP,0) & \mathbf{m}(QP,2) & \mathbf{m}(QP,0) & \mathbf{m}(QP,2) \\ \mathbf{m}(QP,2) & \mathbf{m}(QP,1) & \mathbf{m}(QP,2) & \mathbf{m}(QP,1) \\ \mathbf{m}(QP,0) & \mathbf{m}(QP,2) & \mathbf{m}(QP,0) & \mathbf{m}(QP,2) \\ \mathbf{m}(QP,2) & \mathbf{m}(QP,1) & \mathbf{m}(QP,2) & \mathbf{m}(QP,1) \end{bmatrix}$$

For $QP \geq 6$, each element $\mathbf{m}(QP, k)$ is replaced by $\mathbf{m}(QP\%6, k)/2^{\lfloor QP/6 \rfloor}$.

The matrix **m** used to generate $\mathbf{M_f}$

QP	Positions in $\mathbf{M_f}$ (0, 0), (0, 2) (2, 0), (2, 2)	Positions in $\mathbf{M_f}$ (1, 1), (1, 3) (3, 1), (3, 3)	Remaining $\mathbf{M_f}$ positions
0	13107	5243	8066
1	11916	4660	7490
2	10082	4194	6554
3	9362	3647	5825
4	8192	3355	5243
5	7282	2893	4559

H.264: Quantization and Scaling

- Let $\tilde{\mathbf{f}}$ be the dequantized and then inversely transformed result. The scaled, de-quantized, and inverse integer transform are:

$$\tilde{\mathbf{f}} = \text{round}\left[(\mathbf{H_{inv}} \times (\hat{\mathbf{F}} \cdot \mathbf{V_i}) \times \mathbf{H_{inv}}^T)/2^6\right]$$

where $\mathbf{V_i}$ is the 4 × 4 de-quantization matrix derived from matrix **v** and quantization parameter QP.

H.264: Quantization and Scaling

For $0 \le QP < 6$,

$$\mathbf{V_i} = \begin{bmatrix} \mathbf{v}(QP,0) & \mathbf{v}(QP,2) & \mathbf{v}(QP,0) & \mathbf{v}(QP,2) \\ \mathbf{v}(QP,2) & \mathbf{v}(QP,1) & \mathbf{v}(QP,2) & \mathbf{v}(QP,1) \\ \mathbf{v}(QP,0) & \mathbf{v}(QP,2) & \mathbf{v}(QP,0) & \mathbf{v}(QP,2) \\ \mathbf{v}(QP,2) & \mathbf{v}(QP,1) & \mathbf{v}(QP,2) & \mathbf{v}(QP,1) \end{bmatrix}$$

For $QP \ge 6$, each element $\mathbf{v}(QP, k)$ is replaced by $\mathbf{v}(QP\%6, k) \cdot 2^{\lfloor QP/6 \rfloor}$.

The matrix **v** used to generate $\mathbf{V_i}$

QP	Positions in $\mathbf{V_i}$ (0, 0), (0, 2) (2, 0), (2, 2)	Positions in $\mathbf{V_i}$ (1, 1), (1, 3) (3, 1), (3, 3)	Remaining $\mathbf{V_i}$ positions
0	10	16	13
1	11	18	14
2	13	20	16
3	14	23	18
4	16	25	20
5	18	29	23

H.264: Example of Integer Transform (1)

Integer Transform and Quantization

$$\hat{\mathbf{F}} = \text{round}\left[(\mathbf{H} \times \mathbf{f} \times \mathbf{H}^T) \cdot \mathbf{M_f} / 2^{15} \right]$$

$$\mathbf{f} = \begin{matrix} 72 & 82 & 85 & 79 \\ 74 & 75 & 86 & 82 \\ 84 & 73 & 78 & 80 \\ 77 & 81 & 76 & 84 \end{matrix} \qquad \mathbf{M_f} = \begin{matrix} 13107 & 8066 & 13107 & 8066 \\ 8066 & 5243 & 8066 & 5243 \\ 13107 & 8066 & 13107 & 8066 \\ 8066 & 5243 & 8066 & 5243 \end{matrix} \qquad \hat{\mathbf{F}} = \begin{matrix} 507 & -12 & -2 & 2 \\ 0 & -7 & -14 & 5 \\ 2 & 0 & -8 & -11 \\ -1 & 8 & 4 & 3 \end{matrix}$$

Inverse Integer Transform and De-Quantization

$$\tilde{\mathbf{f}} = \text{round}\left[(\mathbf{H_{inv}} \times (\hat{\mathbf{F}} \cdot \mathbf{V_i}) \times \mathbf{H_{inv}}^T) / 2^6 \right]$$

$$\mathbf{V_i} = \begin{matrix} 10 & 13 & 10 & 13 \\ 13 & 16 & 13 & 16 \\ 10 & 13 & 10 & 13 \\ 13 & 16 & 13 & 16 \end{matrix} \qquad \tilde{\mathbf{f}} = \begin{matrix} 72 & 82 & 85 & 79 \\ 74 & 75 & 86 & 82 \\ 84 & 73 & 78 & 80 \\ 77 & 81 & 76 & 84 \end{matrix} \qquad \epsilon = \mathbf{f} - \tilde{\mathbf{f}} = \begin{matrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{matrix}$$

(a) QP = 0

H.264: Example of Integer Transform (2)

$$\begin{matrix} 72 & 82 & 85 & 79 \\ 74 & 75 & 86 & 82 \\ 84 & 73 & 78 & 80 \\ 77 & 81 & 76 & 84 \end{matrix}$$

$\mathbf{f}$

$$\begin{matrix} 6554 & 4033 & 6554 & 4033 \\ 4033 & 2622 & 4033 & 2622 \\ 6554 & 4033 & 6554 & 4033 \\ 4033 & 2622 & 4033 & 2622 \end{matrix}$$

$\mathbf{M_f}$

$$\begin{matrix} 254 & -6 & -1 & 1 \\ 0 & -4 & -7 & 3 \\ 1 & 0 & -4 & -6 \\ 0 & 4 & 2 & 1 \end{matrix}$$

$\hat{\mathbf{F}}$

$$\begin{matrix} 20 & 26 & 20 & 26 \\ 26 & 32 & 26 & 32 \\ 20 & 26 & 20 & 26 \\ 26 & 32 & 26 & 32 \end{matrix}$$

$\mathbf{V_i}$

$$\begin{matrix} 72 & 82 & 85 & 79 \\ 74 & 75 & 86 & 82 \\ 84 & 74 & 78 & 80 \\ 77 & 82 & 76 & 84 \end{matrix}$$

$\tilde{\mathbf{f}}$

$$\begin{matrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{matrix}$$

$\epsilon = \mathbf{f} - \tilde{\mathbf{f}}$

(b) QP = 6.

H.264: Example of Integer Transform (3)

$$\begin{array}{cccc} 72 & 82 & 85 & 79 \\ 74 & 75 & 86 & 82 \\ 84 & 73 & 78 & 80 \\ 77 & 81 & 76 & 84 \end{array}$$
$$\mathbf{f}$$

$$\begin{array}{cccc} 410 & 252 & 410 & 252 \\ 252 & 164 & 252 & 164 \\ 410 & 252 & 410 & 252 \\ 252 & 164 & 252 & 164 \end{array}$$
$$\mathbf{M_f}$$

$$\begin{array}{cccc} 16 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}$$
$$\hat{\mathbf{F}}$$

$$\begin{array}{cccc} 320 & 416 & 320 & 416 \\ 416 & 512 & 416 & 512 \\ 320 & 416 & 320 & 416 \\ 416 & 512 & 416 & 512 \end{array}$$
$$\mathbf{V_i}$$

$$\begin{array}{cccc} 80 & 80 & 80 & 80 \\ 80 & 80 & 80 & 80 \\ 80 & 80 & 80 & 80 \\ 80 & 80 & 80 & 80 \end{array}$$
$$\tilde{\mathbf{f}}$$

$$\begin{array}{cccc} -8 & 2 & 5 & -1 \\ -6 & -5 & 6 & 2 \\ 4 & -7 & -2 & 0 \\ -3 & 1 & -4 & 4 \end{array}$$
$$\epsilon = \mathbf{f} - \tilde{\mathbf{f}}$$

(c) QP = 30

H.264: Intra Coding

- Intra-coded macroblocks are predicted using some of the neighboring reconstructed pixels.
- Different intra prediction block sizes (4×4 or 16×16) can be chosen.
- Nine prediction modes for the 4×4 blocks.
- Four prediction modes for the 16×16 blocks.

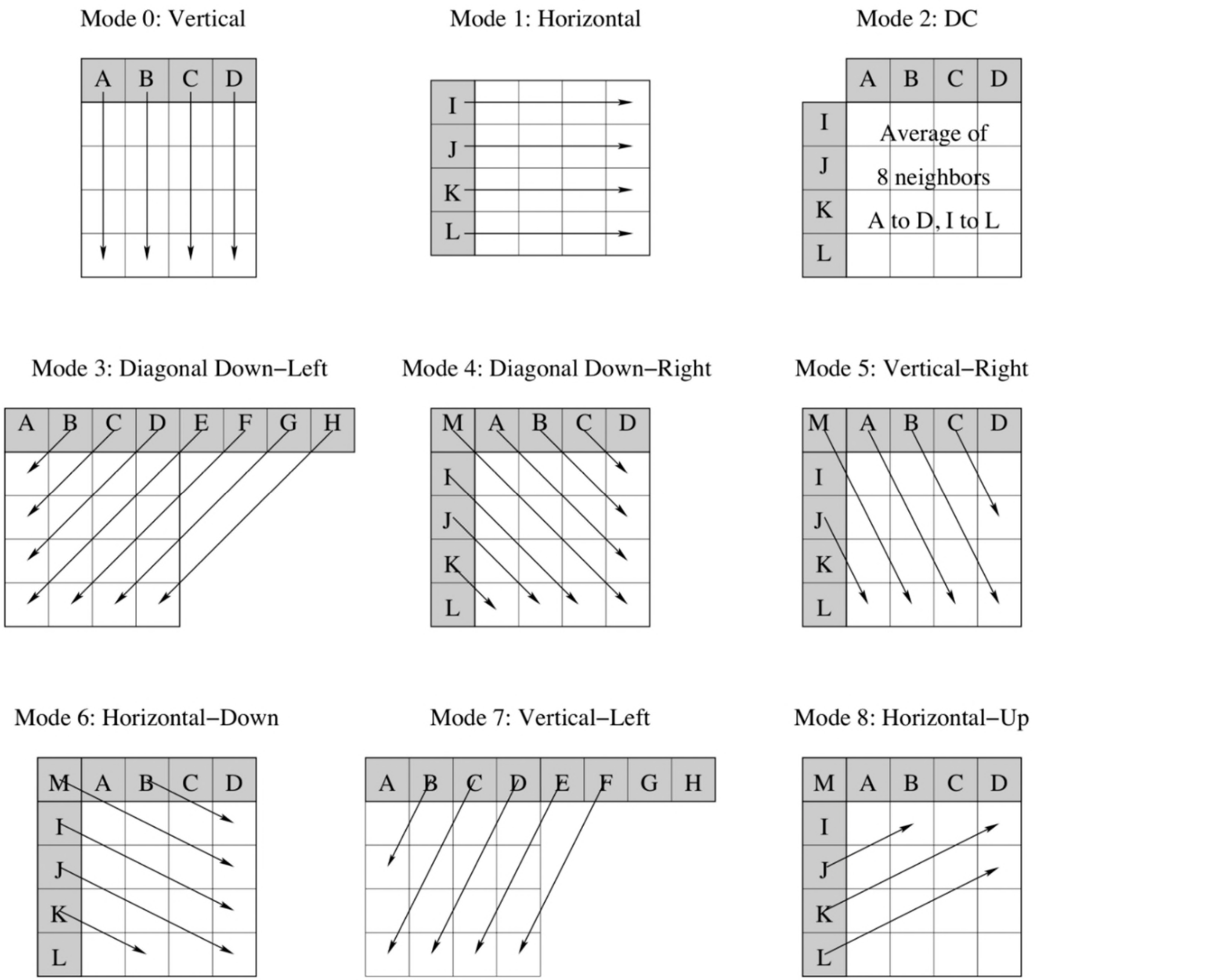
H.264: Intra Coding

- For each prediction mode, the predicted value and the actual value will be compared to produce the prediction error / residuals.
- The mode that produces the least residual will be chosen as the prediction mode for the block.
- The residuals are then sent to transform coding where the 4×4 integer transform is employed.

H.264: Intra Coding

Fig. 12.6 H.264 Intra-frame prediction. **a** Intra_4 × 4 prediction using neighboring samples A to M. **b** Eight directions for Intra_4 × 4 predictions

H.264: Intra Coding



The nine intra_4 × 4 prediction modes in H.264 intra frame prediction.

Exercise: H.264 Intra Coding

Consider a simplified H.264 4×4 intra coding that uses only Mode 0 (vertical), Mode 1 (horizontal), and Mode 2 (DC).

(i) Assume the 8×8 image below, find the least prediction error d(i, j) for the indicated 4×4 pixel block.

(ii) The prediction error d(i, j) is inputted to the following integer transform. Find D(u, v), the output of the integer transform.

20	20	40	40	60	60	80	80
20	20	40	40	60	60	80	80
30	30	50	50	70	70	90	90
30	30	50	50	70	70	90	90
40	40	60	60	80	80	100	100
40	40	60	60	80	80	100	100
50	50	70	80	90	90	110	110
50	50	70	80	90	90	110	110

$$D(u,v) = H \cdot d(i,j) \cdot H^T, \textbf{ where } H = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}$$

Solution

Exercise: H.264 Integer Transform in P-frame Encoding

P-frame coding in H.264 uses *Integer Transform*. For this exercise, assume:

$$F(u,v) = H \cdot f(i,j) \cdot H^T, \text{ where } H = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}.$$

(a) What are the two advantages of using Integer Transform?

(b) Assume the Target Frame below is a P-frame. For simplicity, assume the size of macroblock is 4×4. For the macroblock shown in the Target Frame:

(i) What should be the Motion Vector?

(ii) What are the values of $f(i,j)$ in this case? (iii) Show all values of $F(u,v)$.

20	40	60	80	100	120	140	155
30	50	70	90	110	130	150	165
40	60	80	100	120	140	160	175
50	70	90	110	130	150	170	185
60	80	100	120	140	160	180	195
70	90	110	130	150	170	190	205
80	100	120	140	160	180	200	215
85	105	125	145	165	185	205	220

Reference Frame

110	132	154	176	–	–	–	–
120	142	164	186	–	–	–	–
130	152	174	196	–	–	–	–
140	162	184	206	–	–	–	–
–	–	–	–	–	–	–	–
–	–	–	–	–	–	–	–
–	–	–	–	–	–	–	–
–	–	–	–	–	–	–	–

Target Frame

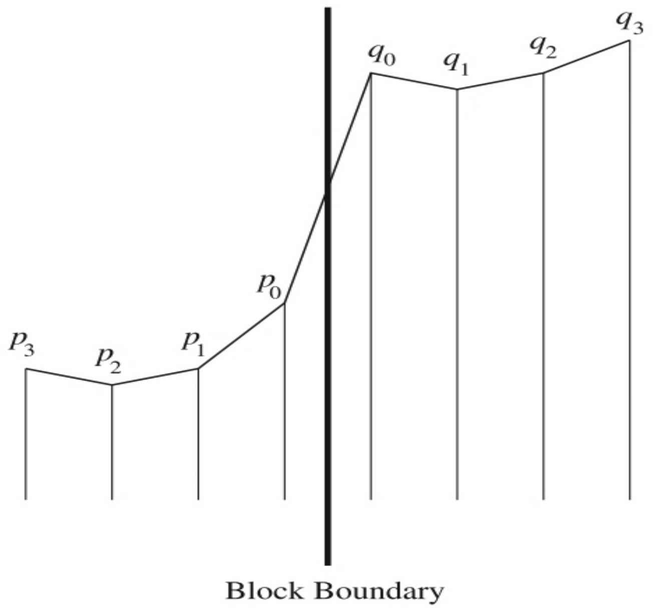
Solution

H.264: In-Loop Deblocking Filtering

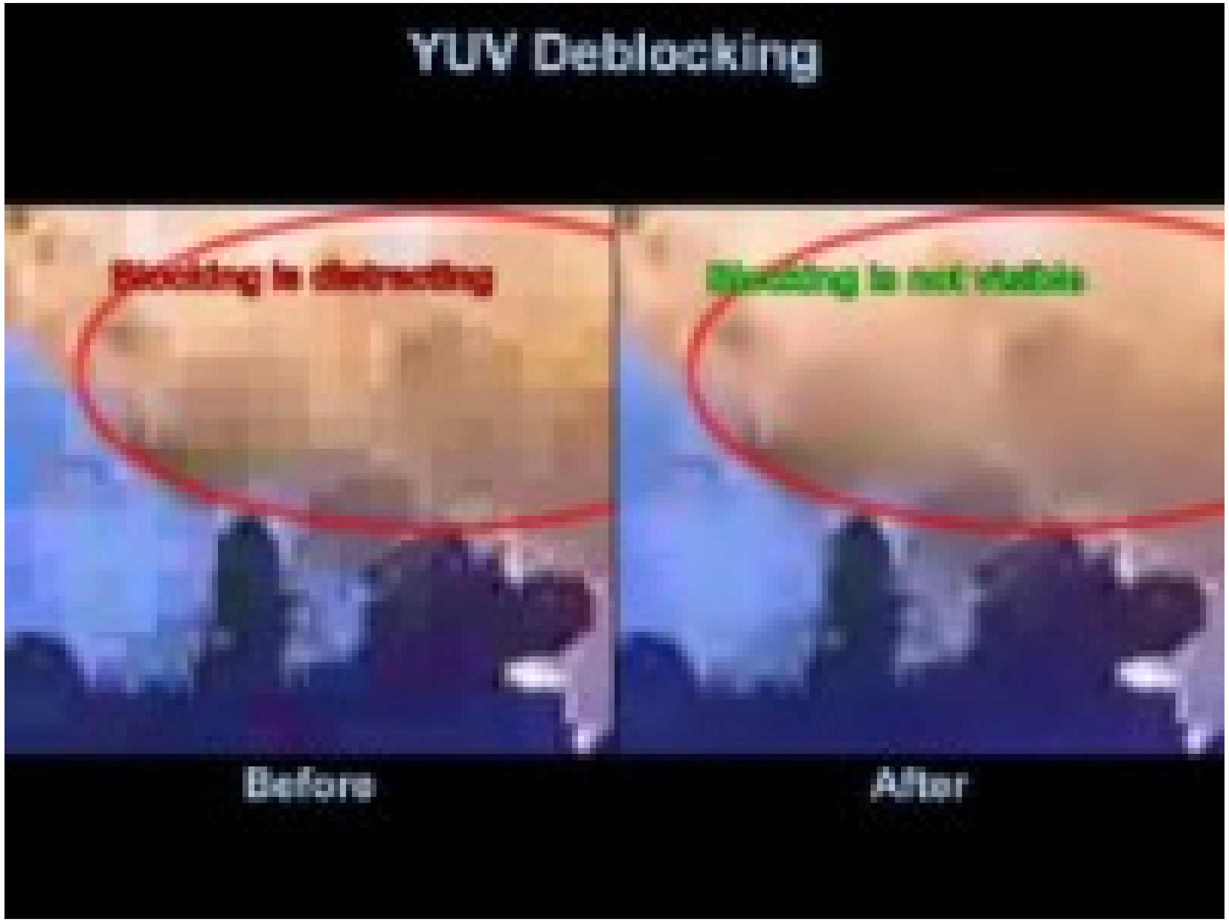
- Block-based coding causes blocking artifacts.
- In-loop deblocking filter aims to eliminate blocking artifacts.
- Increase subjective quality of the videos.
- A signal-adaptive deblocking filter is applied on the 4×4 block edges.

H.264: In-Loop Deblocking Filtering

- A signal-adaptive deblocking filter is applied on the 4 × 4 block edges.
- For example, a "4-tap filtering" will take some weighted average of the values of *p*1, *p*0, *q*0, and *q*1 to generate new *p*0 or *q*0.
- Filter length, strength, and type (deblocking / smoothing) will depend on macroblock coding parameters (intra- or inter-coded, reference-frame differences, coefficients coded, etc.) and spatial activity (edge detection), so that blocking artifacts are eliminated without distorting visual features.



Deblocking Filtering Visualization



https://www.youtube.com/watch?v=PKXIwixuA_E&ab_channel=YUVsoft

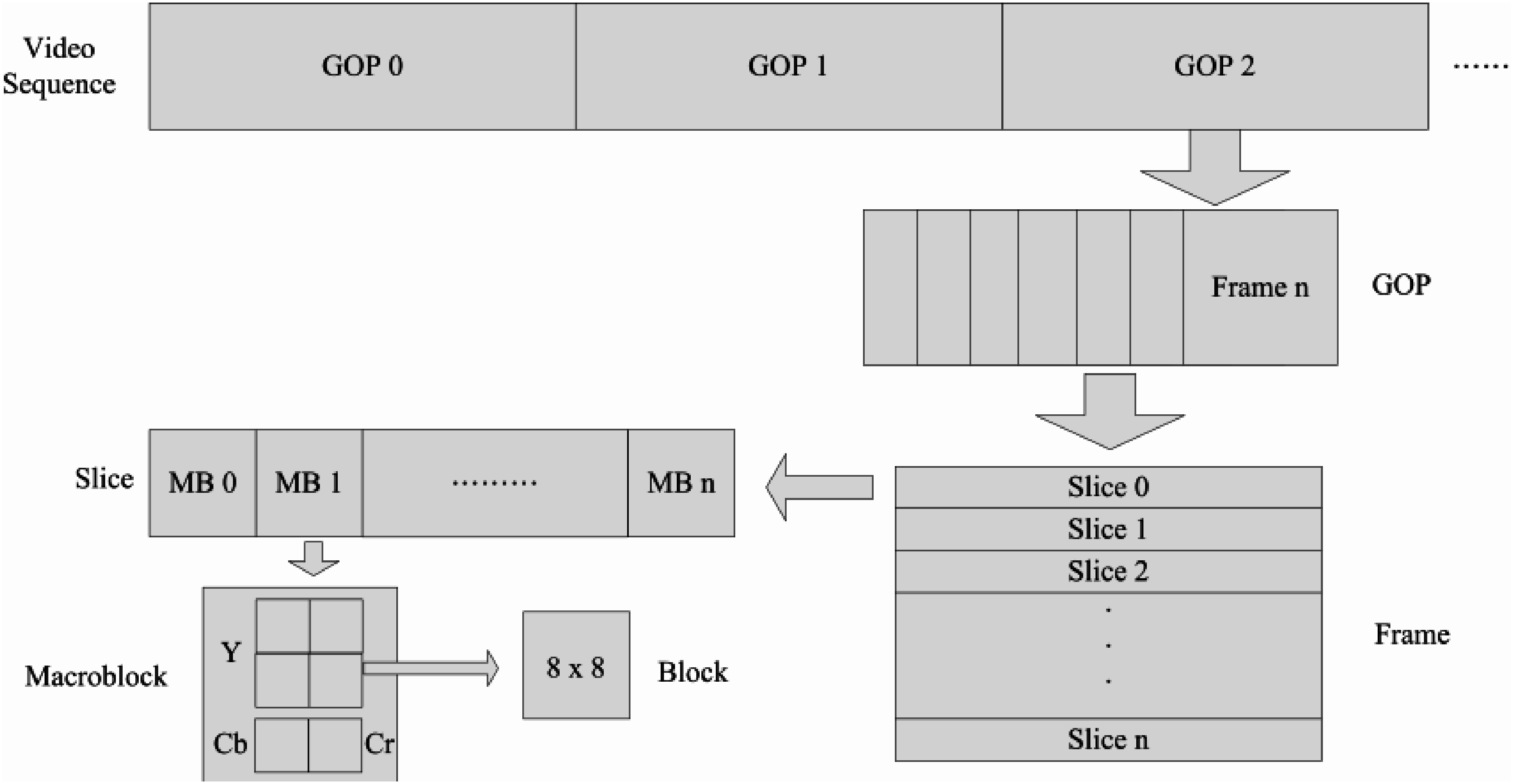
H.264: Context-Adaptive Variable Length Coding (CAVLC)

- Previous video coding standards such as MPEG-2 use fixed VLC.
- In CAVLC, multiple VLC tables are predefined for each data type.
- Predefined rules predict the optimal VLC table based on the context such as parameters from block data.

H.264: Context-Adaptive Binary Arithmetic Coding (CABAC)

- To achieve high coding efficiency in H.264 Main and High profiles.
- High complexity: the implementation has tremendous details.

H.264: Bitstream Structure



H.264: Profiles

- 4 different profiles: Baseline, Extended, Main, High profiles for applications ranging from mobile devices to broadcast HDTV.

Feature	MPEG-2 (MP@ML)	MPEG-4 (ASP)	H.264 Baseline	H.264 Main	H.264 Extended
Block size	16 x 16	16 x 16	Variable: 16 x 16 to 4 x 4		
Frequency transform	8 x 8 DCT	8 x 8 DCT	Hierarchical bit-exact 4 x 4 integer DCT		
1/4-pixel motion vector resolution	No	Yes	Yes		
In-loop deblocking	No	No	Yes		
Reference frames for block prediction	Up to 2	Up to 2	Can be more than 2		
Motion vector prediction	No	No	Yes		
Intra-frame prediction	No	Yes	Yes, several directional spatial modes		
Weighted prediction	No	No	No	Yes	Yes
Entropy coding	Huffman	Huffman	CAVLC	CAVLC/CABAC	CAVLC
Interlaced video support	Yes	Yes	No	Yes	No
Frame/slice types	I, P, B	I, P, B	I, P	I, P, B	I, P, B
Slice groups and Arbitrary Slice Order	No	No	Yes	No	Yes
Redundant slices	No	No	Yes	No	Yes
Switching slices (SP & SI)	No	No	No	No	Yes
Implementation complexity	Low	Low	Medium	High	Medium
Computational requirements	Low	Low	Medium	Very high	High

Table 1 Comparison of some of the key features of H.264 to MPEG-2 and MPEG-4. Refer to 'Squeeze play: how video compression works' in the March 2004 edition of **Inside DSP** for useful background and definitions of terms used in this table.

H.264: Scalable Video Coding (SVC)

- Provide bitstream scalability for networks with different bandwidths.
- Provide temporal scalability, spatial scalability, quality scalability, and their possible combinations.
- Compared to previous standards, the coding efficiency is greatly improved.

H.264: Multiview Video Coding (MVC)

- Multi-view Video Coding (MVC) is an emerging issue.
- It has potential applications in some new areas such as Free Viewpoint Video (FVV) where users can specify their preferred views.

H.264 Summary

- Intra coding:
  - Directional spatial prediction.
- Inter coding:
  - Integer Transform.
  - Variable block-size motion compensation.
  - Subpixel motion compensation.
- New features:
  - In-loop deblocking filtering.
  - Multiple reference frames.
  - Effective scalable video coding.
  - Higher resolution & frame-rate.
  - Context-Adaptive Variable Length Coding (CAVLC) & Context-Adaptive Binary Arithmetic Coding (CABAC).

H.265, H.266

H.264 vs H.265 Comparison

https://www.youtube.com/watch?v=Fawcboio6g4&ab_channel=HandyAndyTechTips

H.265: Introduction

- H.265/HEVC (High Efficiency Video Coding) was developed by the Joint Collaborative Team on Video Coding (JCT-VC) from ITU-T VCEG and ISO/IEC MPEG.
- HEVC / ITU-T H.265 / MPEG-H Part 2 is standardized in 2013.

H.265: Introduction

- Improve coding efficiency due to increasing resolution (e.g., UHD).
- Speed up more complex methods by exploiting parallel processing.
- H.264 and H.265 are current choices for many video applications.

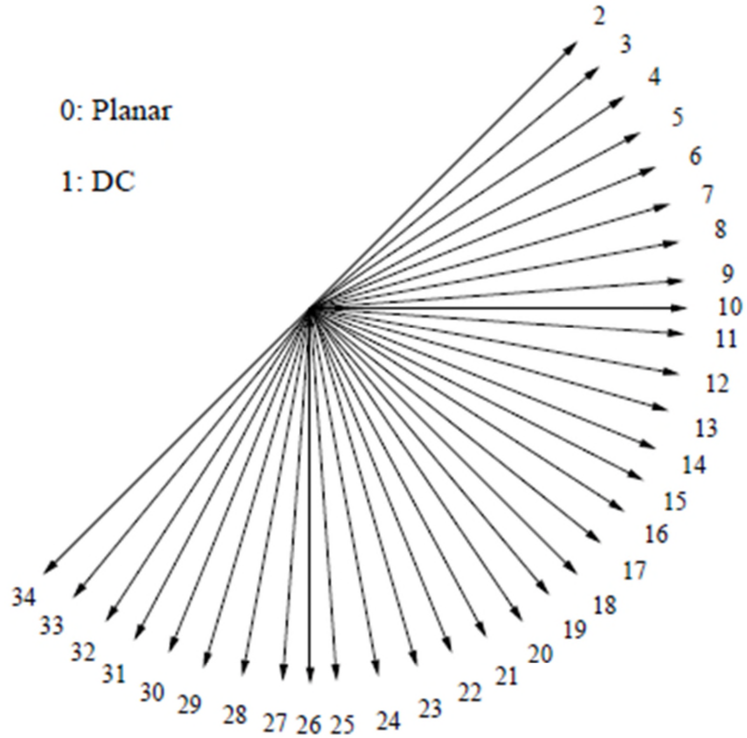
H.265: Improvements over H.264

- Support for UltraHD video up to 8K resolutions with frame rates up to 120 fps.
- Greater flexibility in prediction modes and transform block sizes.
- More sophisticated interpolation and deblocking filters.
- In-Loop Deblocking Filter: 8×8 in H.265 vs 4×4 in H.264
- Support parallel processing.
- More efficient compared to H.264 in terms of bitrate savings for the same picture quality.

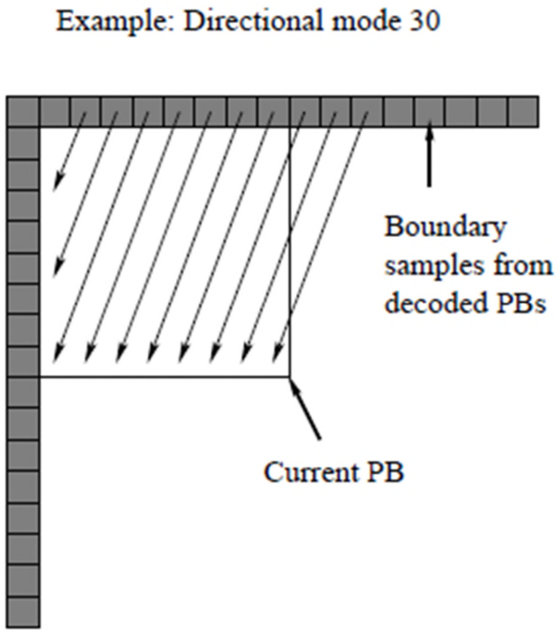
H.265: Main Features

- Macroblock structure is replaced by a quadtree structure of coding blocks at various levels and sizes.
- Improved interpolation methods for the quarter-pixel accuracy in motion vectors.
- Expanded directional spatial prediction (33 angular directions) for intra coding.

H.265: Intra-Coding



(a) Modes and Intra prediction directions



(b) Intra prediction for an 8×8 block

Fig. 12.16 H.265 Intra prediction

H.266: Overview

- H.266 / Versatile Video Coding (VVC) / MPEG-I Part 3.
- Suitable for SDR, HDR, 360◦ video, panoramic.
- Support up to 16K UHD videos, YCbCr 4:4:4, 4:2:2, 4:2:0 with 10 to 16 bits for each sample.
- Achieve about 50% more compression than H.265 for the same perceptual quality.

New / Emerging Coding Standards

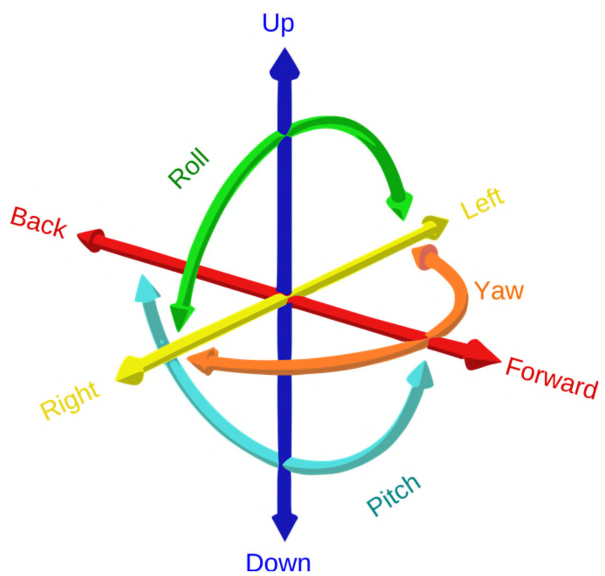
MPEG-I: Immersive Media

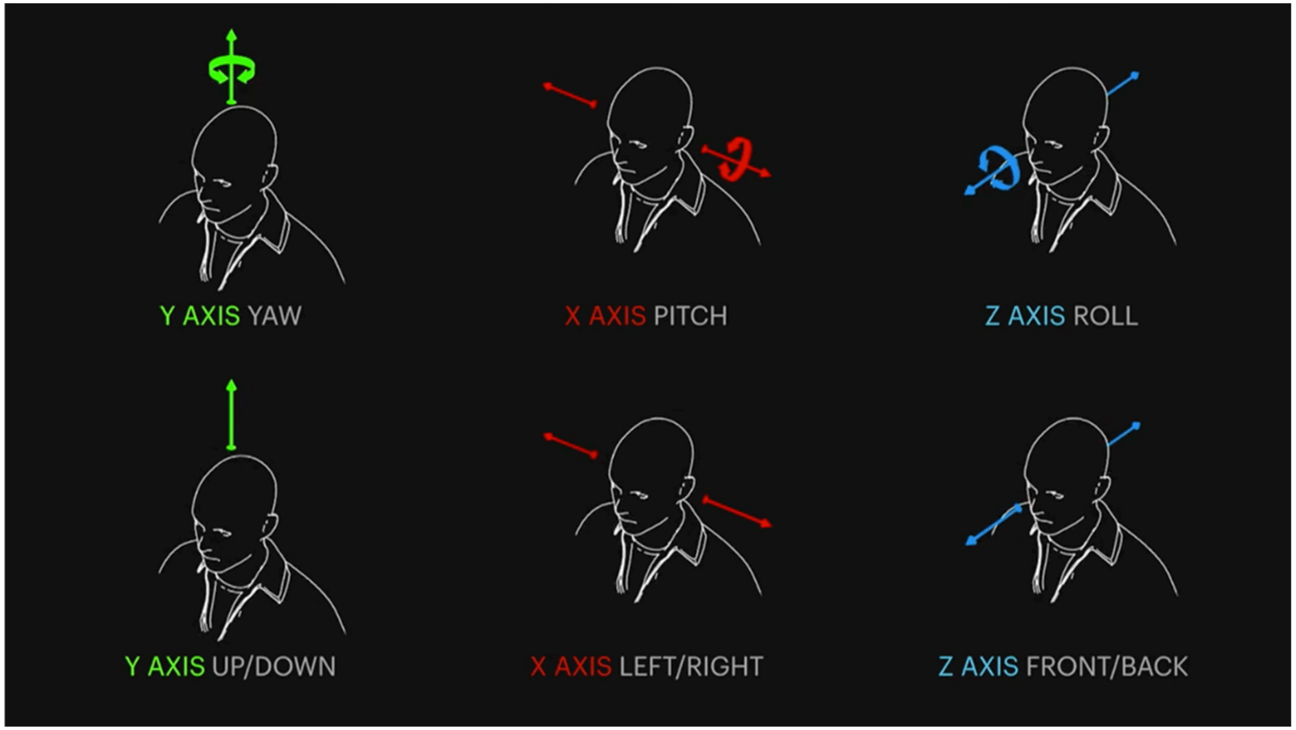
MPEG Immersive Video (MIV)

- Part of ISO/IEC 23090 MPEG-I (Part 12).
- MIV is undergoing standardization to digitally represent immersive media.
- Draft MIV standard provides support for six degrees of freedom (6DoF) video, also known as volumetric video.
- Enable to view immersive volumetric content captured by multiple cameras with 6DoF.

J. M. Boyce et al., "MPEG Immersive Video Coding Standard," in Proceedings of the IEEE, vol. 109, no. 9, pp. 1521-1536, Sept. 2021, doi: 10.1109/JPROC.2021.3062590.

6DoF Video





https://alex4d.com/notes/item/what-is-six-degrees-of-freedom-360-video

6DoF Video

https://www.youtube.com/watch?v=VuJ6o5r1n2c&ab_channel=Skarredghost

Volumetric Video in Film-making

https://www.youtube.com/watch?v=iwUkbi4_wWo&t=569s&ab_channel=TED

Summary

- This section covers the following:
  - Video Coding & Standards
  - MPEG Standards
  - H.26x Standards
  - New / Emerging Coding & Standards