CMPT 365 Multimedia Systems

Final Review – 2

Spring 2017

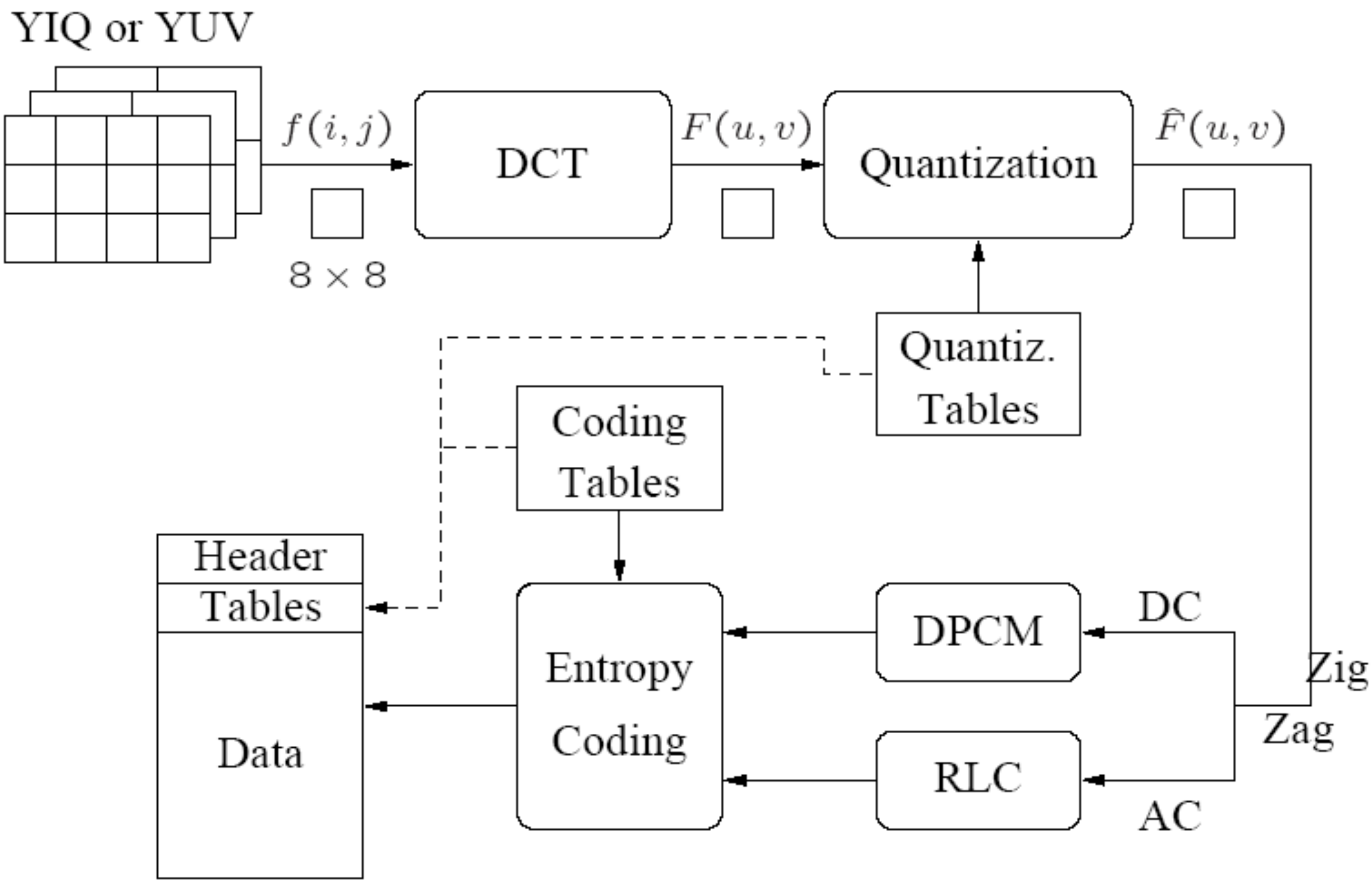
Administrative

- Final Exam:
  - C9002, April 18th, 15:30-18:30
  - Calculator Allowed, No cheat sheet
- Project:
  - Due at 11:59pm, April 18th
  - Demo day: April 20th
  - Slot register: https://docs.google.com/spreadsheets/d/11sDObECkxmkEKhBMz6lLz9LNa45Mxm_rJ0a26aOKR4c/edit?usp=sharing
  - Need to bring printed report
- Wed, Friday 2:30pm~ 3:20pm
  - Office hour in TASC1-8002

Outline

- Jpeg
- H.261
- Audio

JPEG Diagram



JPEG Steps

1 Block Preparation
  - RGB to YUV (YIQ) planes
2 Transform
  - 2D Discrete Cosine Transform (DCT) on 8x8 blocks.
3 Quantization
  - Quantized DCT Coefficients (lossy).
4 Encoding of Quantized Coefficients
  - Zigzag Scan
  - Differential Pulse Code Modulation (DPCM) on DC component
  - Run Length Encoding (RLE) on AC Components
  - Entropy Coding: Huffman or Arithmetic

Block Effect

- Using blocks, however, has the effect of isolating each block from its neighboring context.
  - choppy ("blocky") with high *compression ratio*

Compression Ratio: 7.7

Compression Ratio: 33.9

Compression Ratio: 60.1

More about Quantization

- **Quantization is the main source for loss**
  - $Q(u, v)$ of larger values towards lower right corner
    - More loss at the higher spatial frequencies
    - Supported by Observations 1 and 2.
  - Q(u,v) obtained from psychophysical studies
    - maximizing the compression ratio while minimizing perceptual losses

JPEG: Zigzag Scan

Maps an 8x8 block into a 1 x 64 vector
Zigzag pattern group low frequency coefficients in top of vector.

150	80	20	4	1	0	0	0
92	75	18	3	1	0	0	0
26	19	13	2	1	0	0	0
3	2	2	1	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

JPEG: Encoding of Quantized DCT Coefficients

- DC Components (zero frequency)
  - DC component of a block is large and varied, but often close to the DC value of the previous block.
  - Encode the difference from previous
    - Differential Pulse Code Modulation (DPCM).
- AC components:
  - Lots of zeros (or close to zero)
  - Run Length Encoding (RLE, or RLC)
    - encode as (skip, value) pairs
    - Skip: number of zeros, value: next non-zero component
  - (0,0) as end-of-block value.

DPCM on DC coefficients

- The DC coefficients are coded separately from the AC ones. *Differential Pulse Code modulation* (*DPCM*) is the coding method.

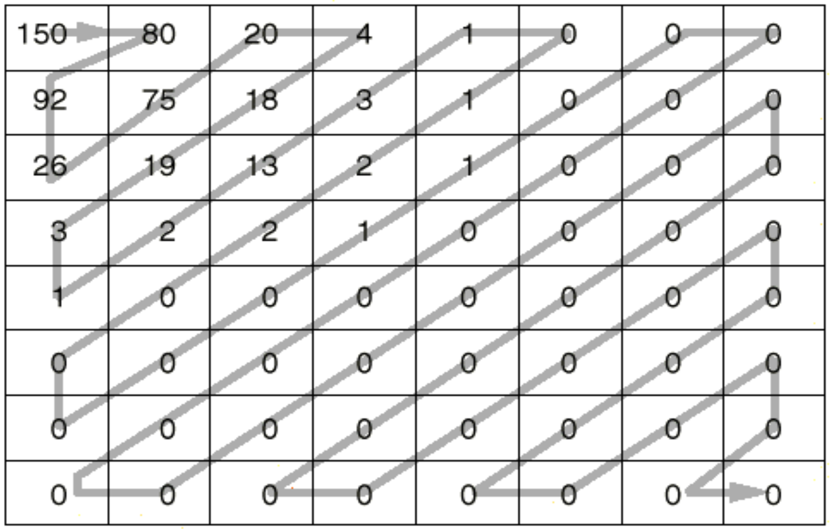
- If the DC coefficients for the first 5 image blocks are 150, 155, 149, 152, 144, then the DPCM would produce 150, 5, -6, 3, -8, assuming $d_i = DC_{i+1} - DC_i$, and $d_0 = DC_0$.

Why ZigZag Scan

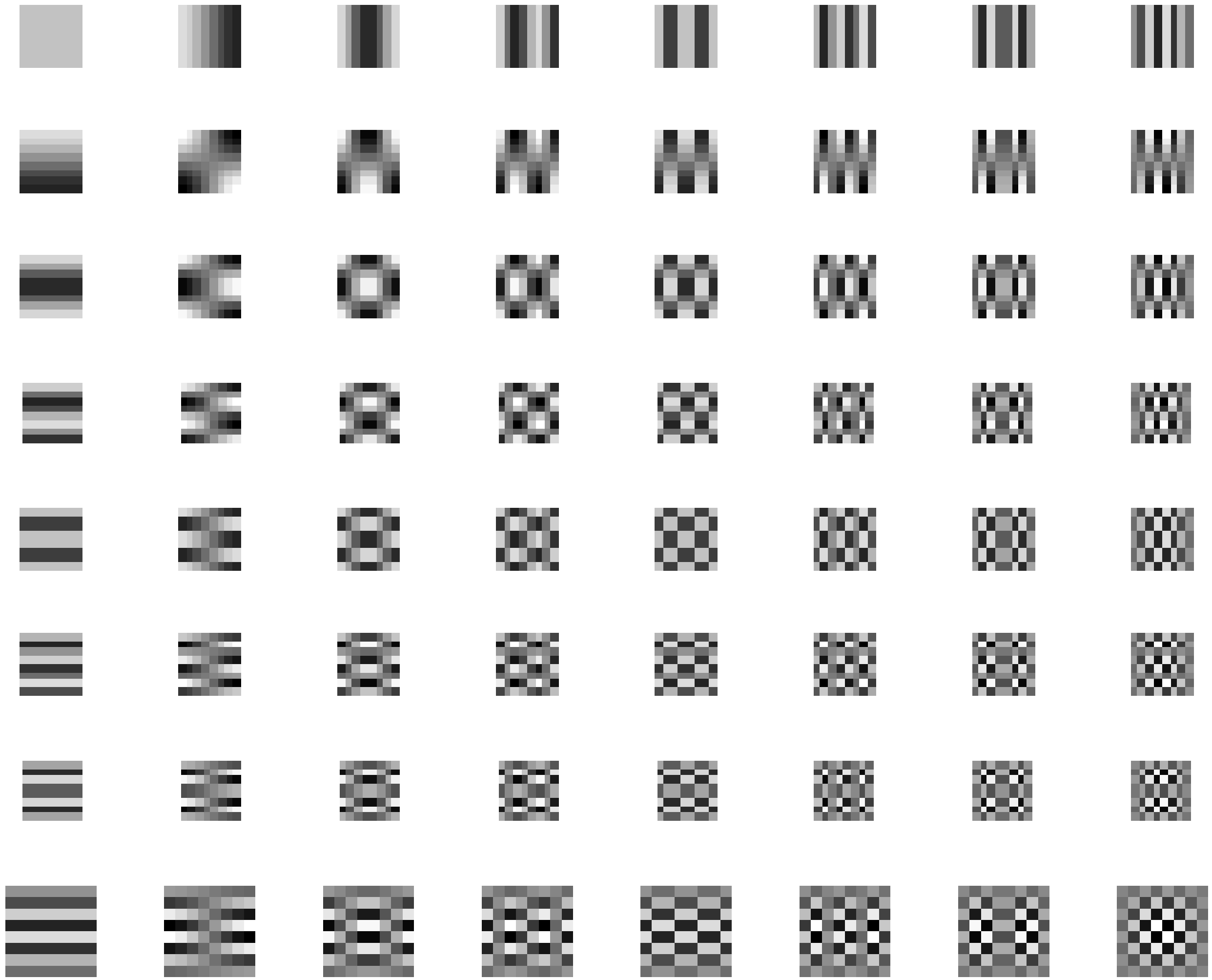
- RLC aims to turn the block values into sets

  <#-zeros-to-skip , next non-zero value>.

- ZigZag scan is more effective



Recall: 2-D DCT Basis Matrices: 8-point DCT



Runlength Encoding (RLE)

A typical 8x8 block of quantized DCT coefficients.
Most of the higher order coefficients have been quantized to 0.

12	34	0	54	0	0	0	0
87	0	0	12	3	0	0	0
16	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Zig-zag scan: the sequence of DCT coefficients to be transmitted:
12 34 87 16 0 0 54 0 0 0 0 0 0 12 0 0 3 0 0 0 .....
DC coefficient (12) is sent via a separate Huffman table.
Runlength coding remaining coefficients:
34 | 87 | 16 | 0 0 54 | 0 0 0 0 0 0 12 | 0 0 3 | 0 0 0 .....
(0,34),(0,87),(0,16),(2,54),(6,12),(2,3)...

- Further compression: statistical (entropy) coding

JPEG Modes

- Sequential Mode
  - default JPEG mode, implicitly assumed in the discussions so far. Each graylevel image or color image component is encoded in a single left-to-right, top-to-bottom scan.
- Progressive Mode.
- Hierarchical Mode.
- Lossless Mode

Progressive Mode

- Progressive
  - Delivers low quality versions of the image quickly, followed by higher quality passes.

- **Method 1. Spectral selection**

  - higher AC components provide detail texture information

  - Scan 1: Encode DC and first few AC components, e.g., AC1, AC2.
  - Scan 2: Encode a few more AC components, e.g., AC3, AC4, AC5.
  - ...
  - Scan k: Encode the last few ACs, e.g., AC61, AC62, AC63.

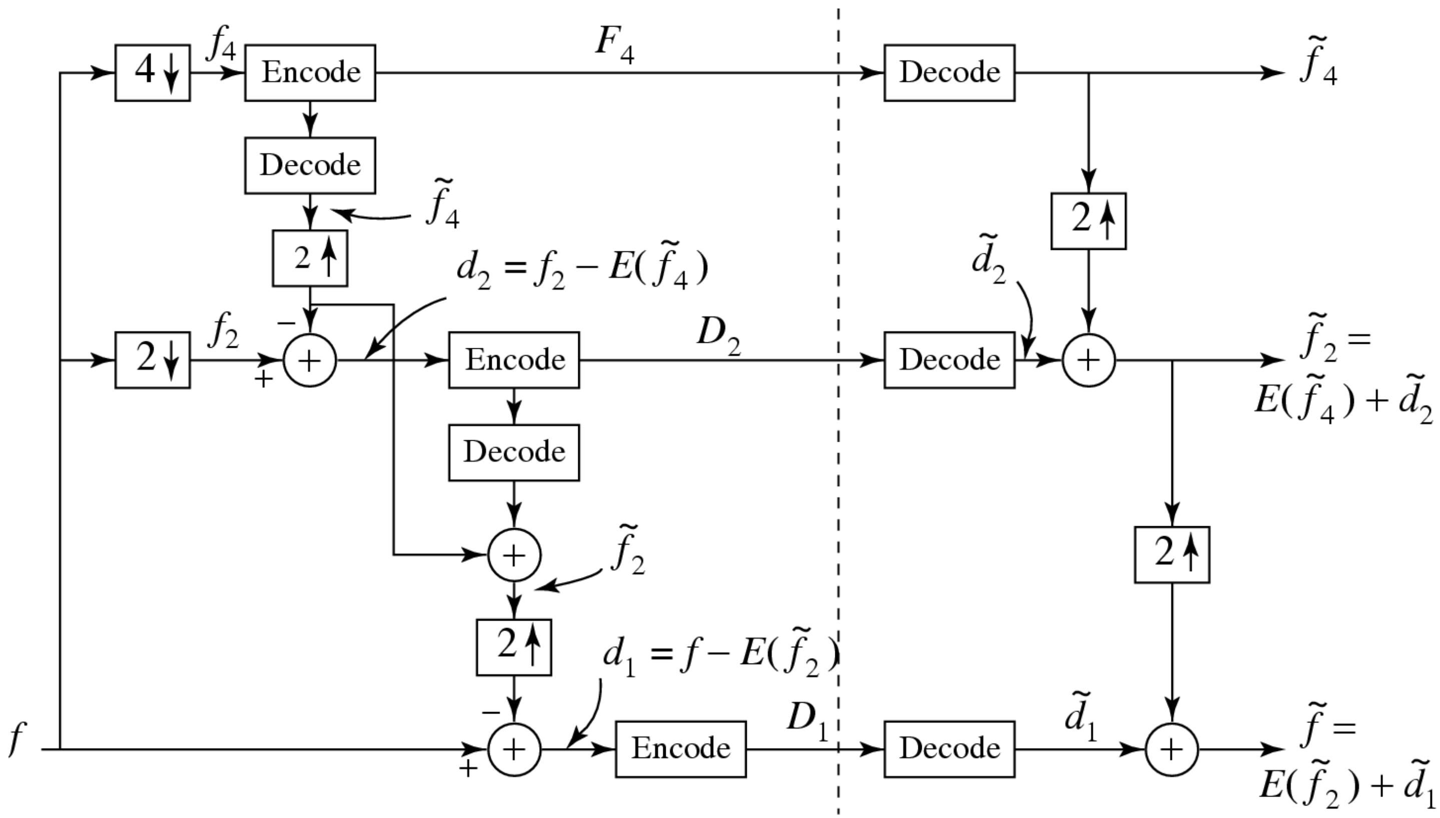
Progressive Mode cont'd

- **Method 2: Successive approximation**:
  - Instead of gradually encoding spectral bands, all DCT coefficients are encoded simultaneously but with their most significant bits (MSBs) first

  - Scan 1: Encode the first few MSBs, e.g., Bits 7, 6, 5, 4.
  - Scan 2: Encode a few more less significant bits, e.g., Bit 3.
  - ...
  - Scan m: Encode the least significant bit (LSB), Bit 0.

Hierarchical Mode

- Encoding
  - First, lowest resolution picture (using low-pass filter)
  - Then, successively higher resolutions
    - additional details (encoding differences)
- Transmission:
  - transmitted in multiple passes
  - progressively improving quality
  - Similar to Progressive JPEG



- **Fig. 9.5: Block diagram for Hierarchical JPEG.**

Outline

- Jpeg
- H.261
- Audio

Temporal Redundancy

- Characteristics of typical videos:
  - A lot of similarities between adjacent frames
  - Differences caused by object or camera motion

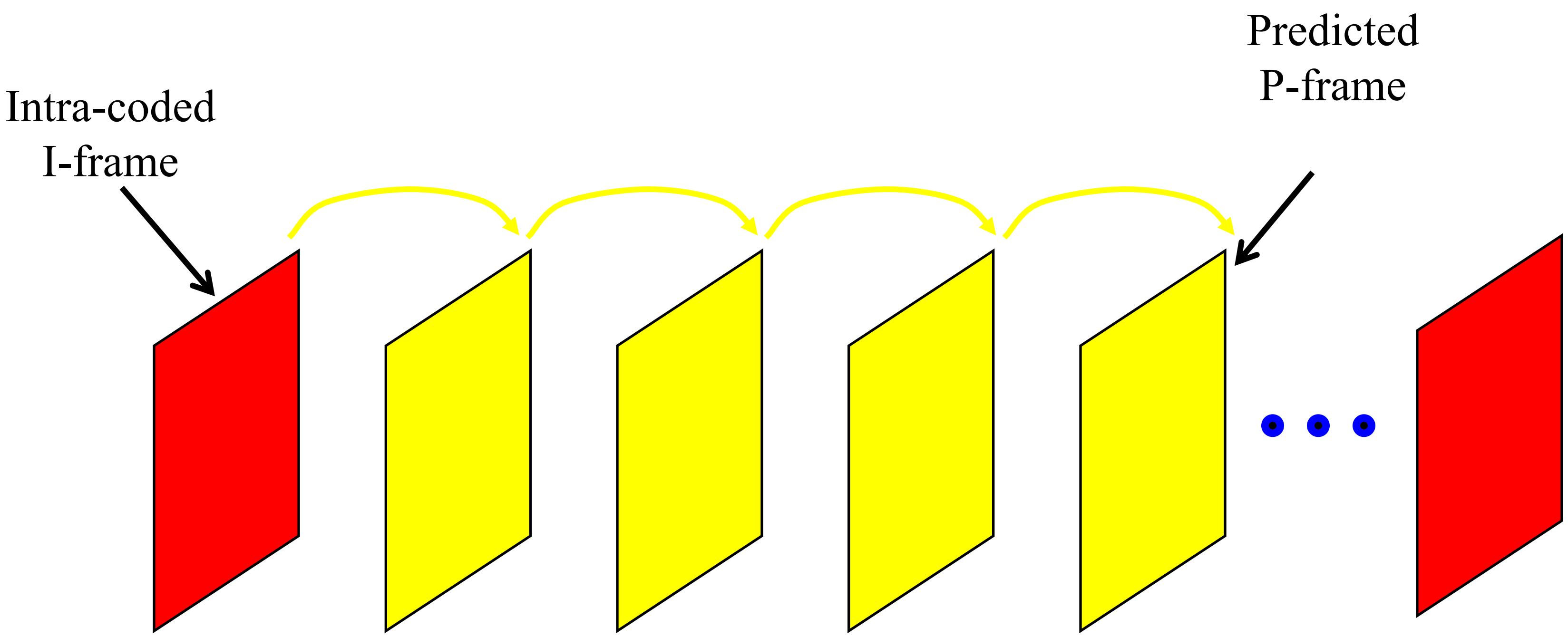
Frame 1

Frame 2

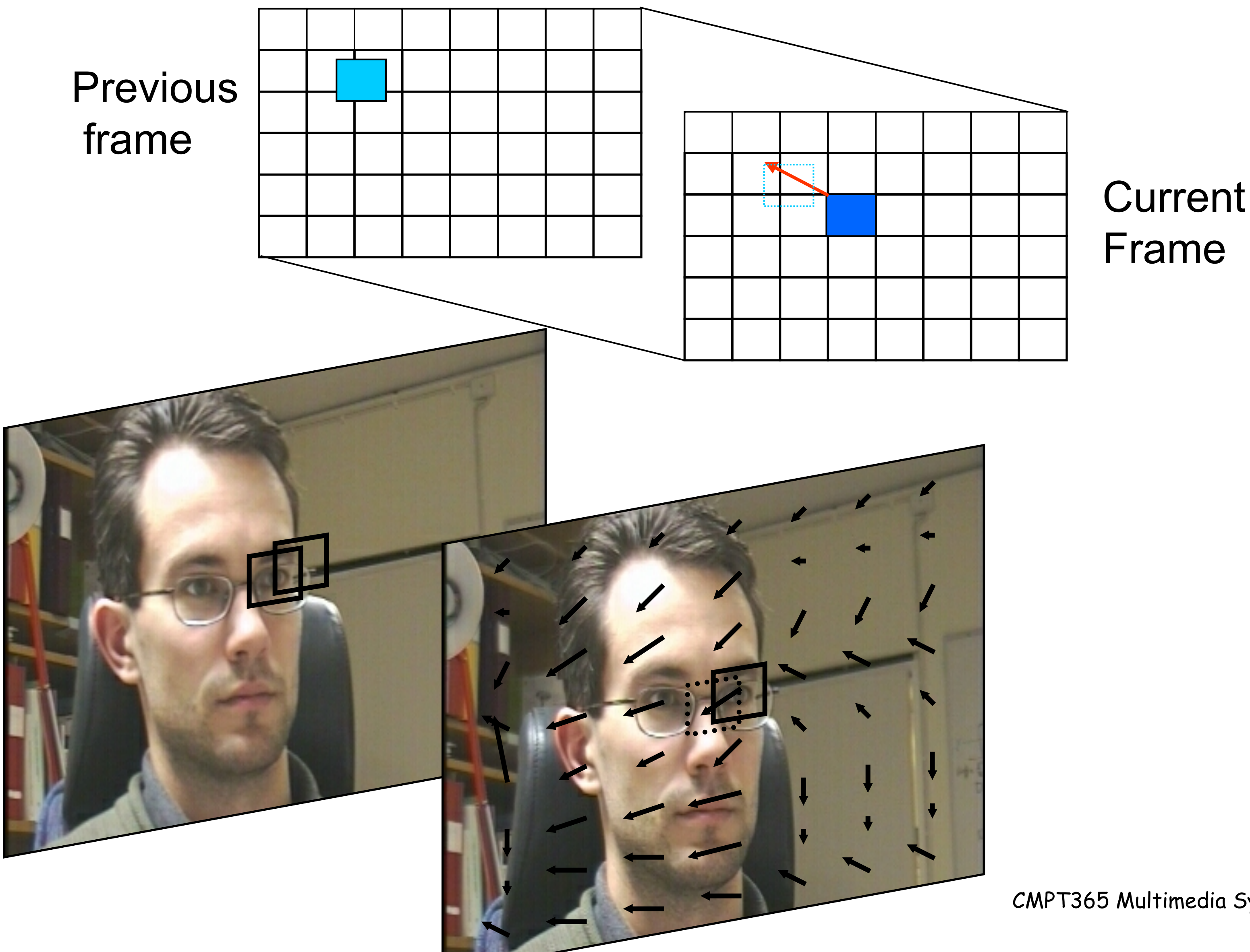
Direct Difference

Key Idea in Video Coding

- Predict each frame from the previous frame and only encode the prediction error:
  - Pred. error has smaller energy and is easier to compress

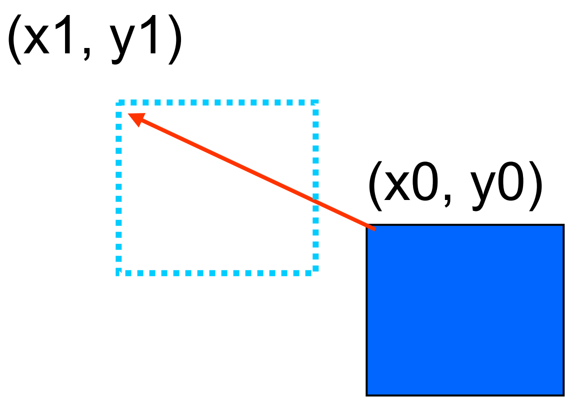


Motion ?



Motion Estimation (ME)

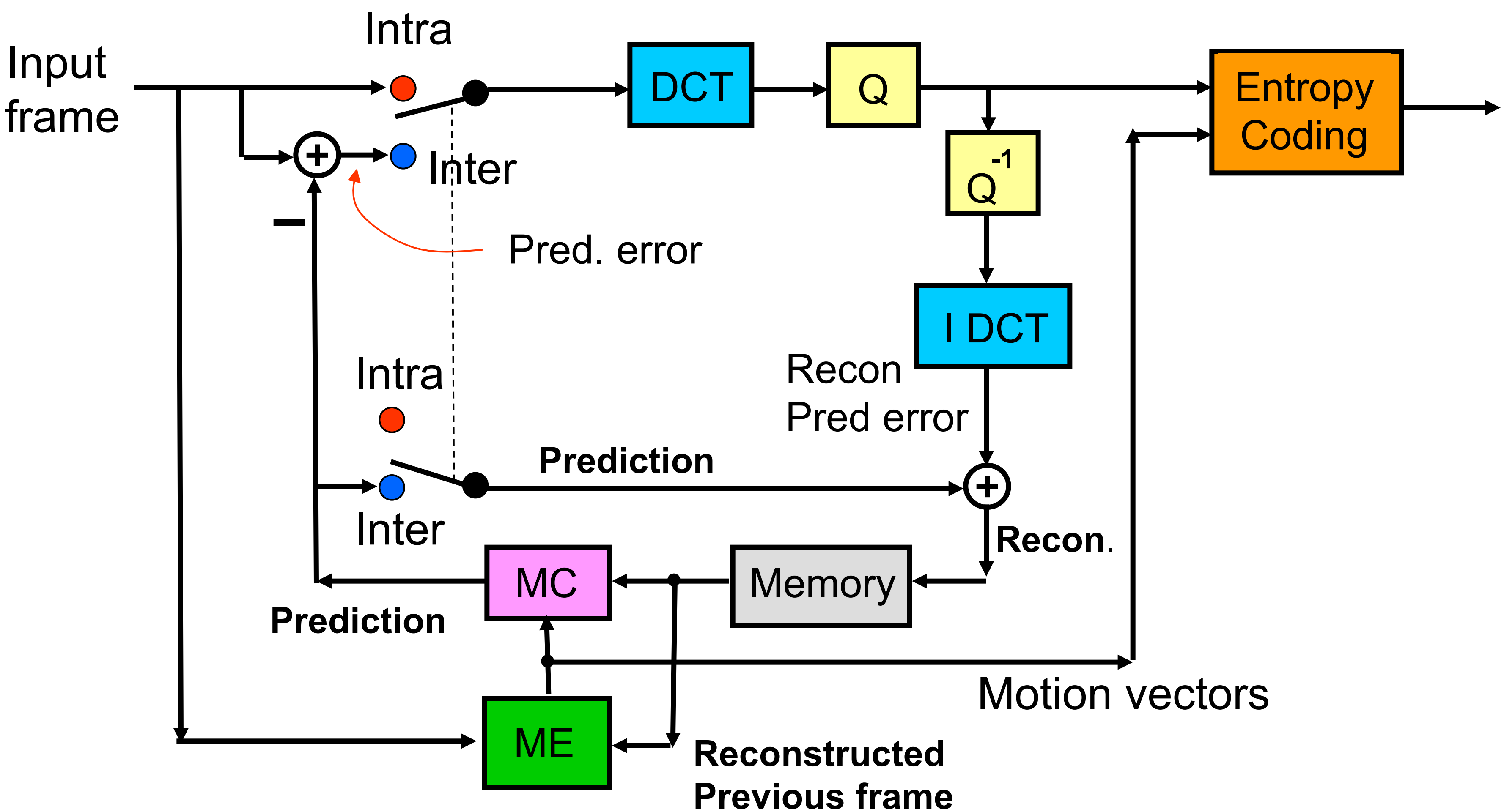
- For each block, find the best match in the previous frame (reference frame)
  - Upper-left corner of the block being encoded: (x0, y0)
  - Upper-left corner of the matched block in the reference frame: (x1, y1)
  - Motion vector (dx, dy): the offset of the two blocks:
    - (dx, dy) = (x1 – x0, y1 – y0)
    - (x0, y0) + (dx, dy) = (x1, y1)
  - Motion vector need to be sent to the decoder.



Motion Compensation (MC)

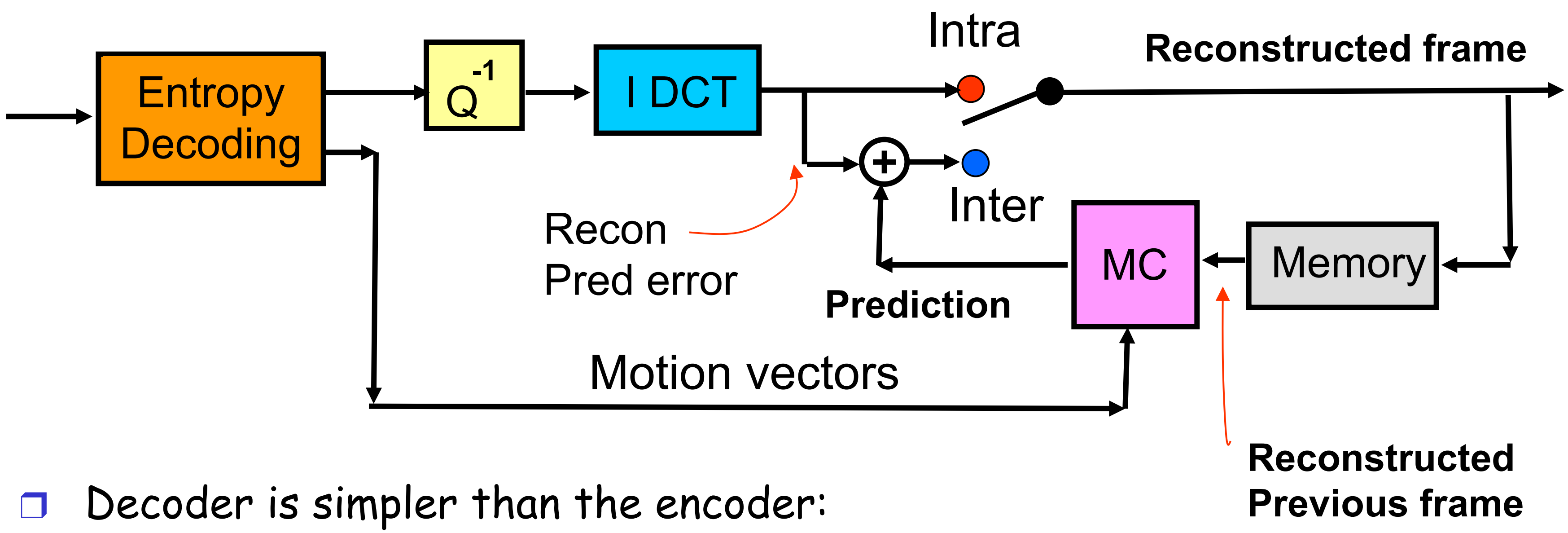
- Given reference frame and the motion vector, can obtain a prediction of the current frame
- Prediction error: Difference between the current frame and the prediction.
- The prediction error will be coded by DCT, quantization, and entropy coding.

Basic Encoder Block Diagram



Use reconstructed error in the loop to prevent drifting.
Original input is not available to the decoder.
Need a buffer to keep the reference frame.

Basic Decoder Block Diagram



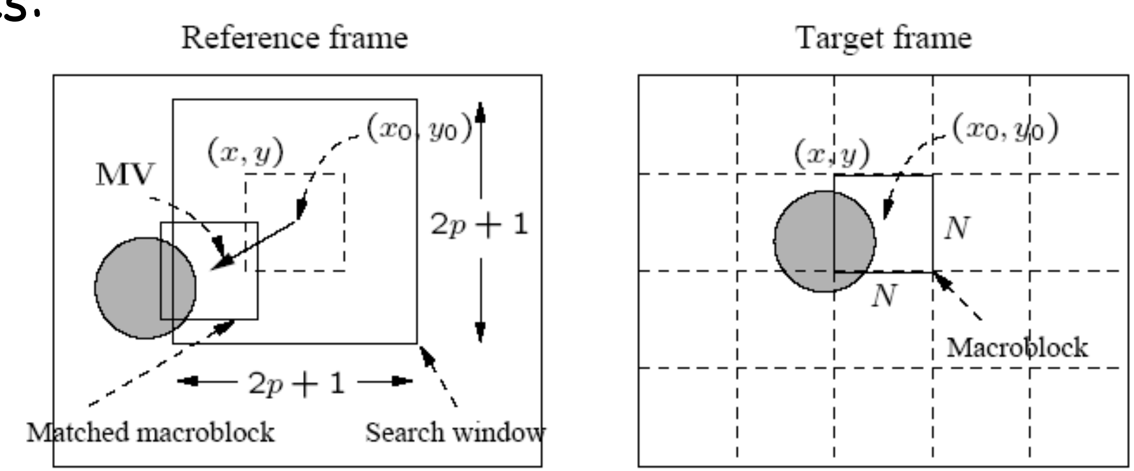
- Decoder is simpler than the encoder:
  - No need to do motion estimation.

Motion Estimation - Revisit

- Formulation:
- Find (i, j) in a search window (-p, p) that minimizes

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

- Mean square error (MSE)
  - If z=2
- Mean absolute distance (MAD):
  - If z = 1.
- # of search candidates:
  (2p+1) x (2p + 1)



MAD-based Motion Estimation

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

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 in Target frame,

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

- Objective
  - Find vector (*i, j*) as the motion vector **MV** = (**u**,**v**), such that *MAD*(*i*,*j*) is minimum

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

Naive Method

- **Sequential search (Full search):**
  - sequentially search the whole $(2p+1)$ $(2p+1)$ window in the Reference frame
    - a macroblock centered at each of the positions within the window is compared to the macroblock in the Target frame, pixel by pixel
    - respective *MAD* is derived
    - vector (i, j) that offers the least *MAD* is designated as the **MV** (u, v) for the macroblock in the target frame

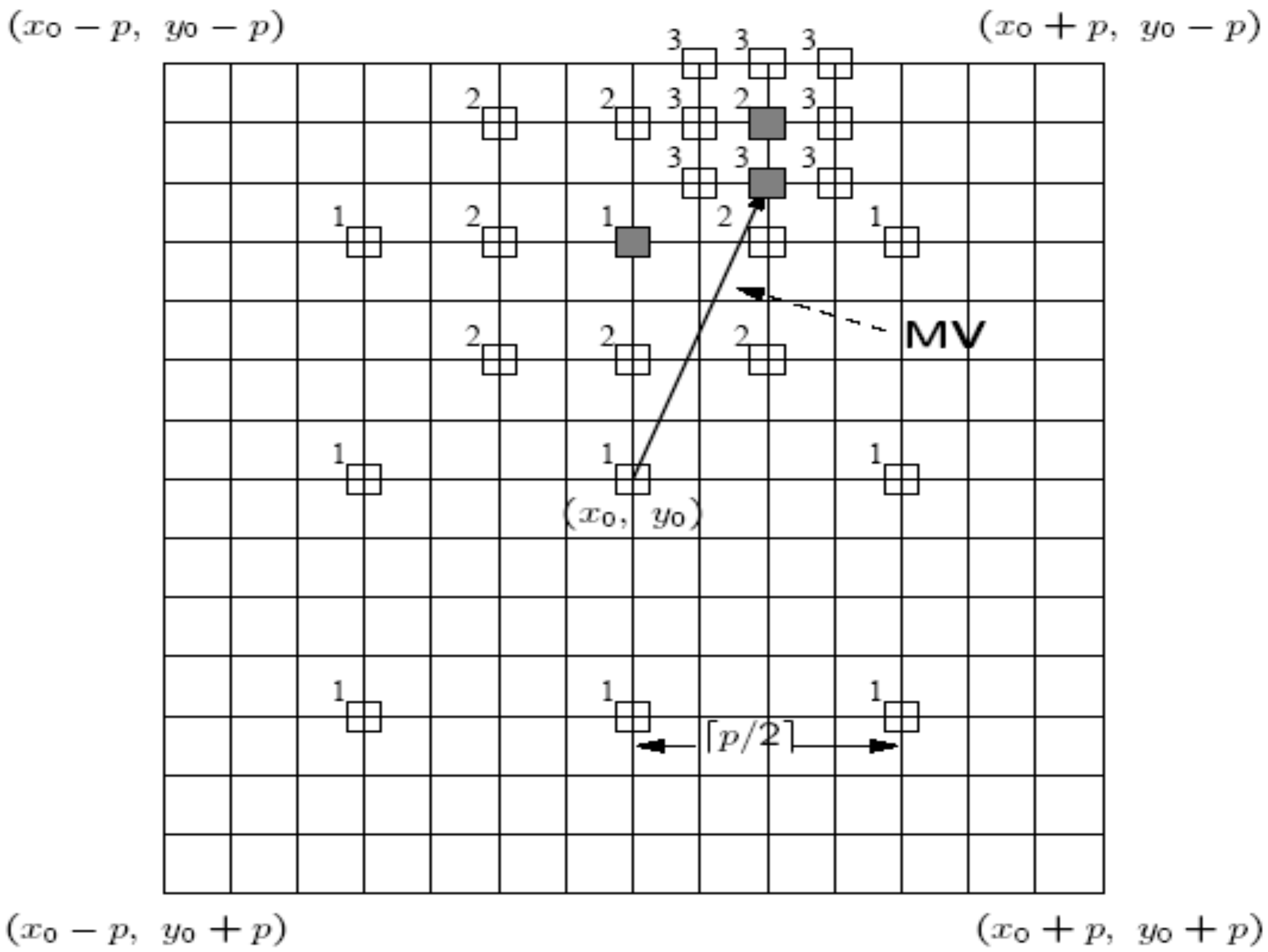
Fast Motion Estimation

- Full-search motion estimation is time consuming:
  - Each (i, j) candidate: N^2 summations
  - If search window size is W^2, need $W^2 \times N^2$ comparisions / MB
    - W=2p+1=31, N=16: ➔ 246016 comparisons / MB !
    - Each comparison three operations (subtraction, absolute value, addition)
- Fast motion estimation is desired:
  - Lower the number of search candidates
  - Many methods

2-D Log Search

- **Logarithmic search**:
  - a cheaper version
  - *suboptimal* but still usually effective.
- Procedure – similar to a binary search
  - Initially, only nine locations in the search window are used as seeds for a MAD-based search; marked as `1'.
  - After the one that yields the minimum *MAD* is located, the center of the new search region is moved to it and the step-size ("offset") is reduced to half.
  - In the next iteration, the nine new locations are marked as `2', and so on.

Log Search



$(x_0 - p,\ y_0 - p)$
$(x_0 + p,\ y_0 - p)$
3 3 3
2 2 3 2 3
3 3 3
1 2 1 2 1
MV
2 2 2
1 1 1
$(x_0,\ y_0)$
1 1 1
$\lceil p/2 \rceil$
$(x_0 - p,\ y_0 + p)$
$(x_0 + p,\ y_0 + p)$

Computations

- W=2p+1=31, N=16 (p=15)

$$\left(8 \cdot (\lceil \log_2 p \rceil + 1) + 1\right) \cdot N^2$$

- 10496 Comparison per Macroblock

Hierarchical Search

- **Hierarchical search**:
  - $W^2 \times N^2$: Comparison Per macroblock for sequential search
  - The search can benefit from a hierarchical (multiresolution) approach in which initial estimation of the motion vector can be obtained from images with a significantly reduced resolution.
  - Since the size of the macroblock is smaller and p can also be proportionally reduced, the number of operations required is greatly reduced.

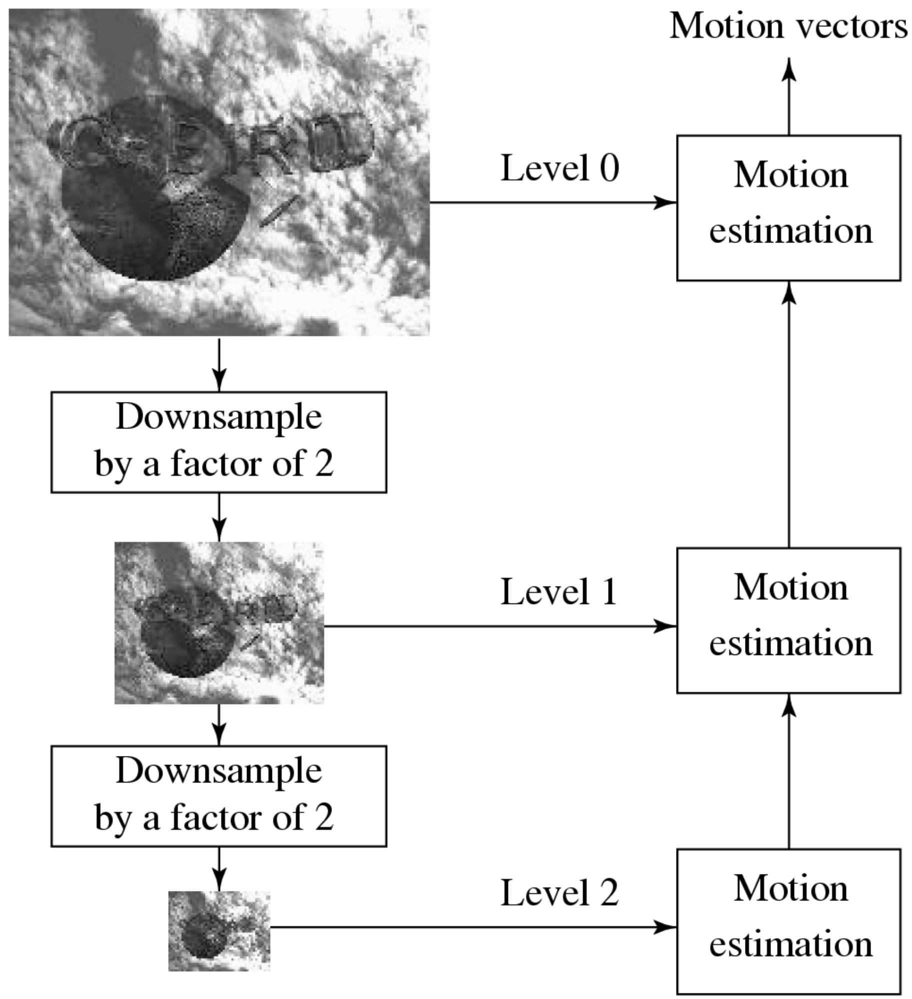


Fig. 10.3: A Three-level Hierarchical Search for Motion Vectors.

Hierarchical Search (Cont'd)

- Given the estimated motion vector (u^k, v^k) at Level k, a 3 x 3 neighborhood centered at $(2 \cdot u^k, 2 \cdot v^k)$ at Level $k - 1$ is searched for the refined motion vector.
- The refinement is such that at Level $k - 1$ the motion vector (u^{k-1}, v^{k-1}) satisfies:
  - $(2u^k - 1 \leq u^{k-1} \leq 2u^k + 1, 2v^k - 1 \leq v^{k-1} \leq 2v^k + 1)$
- Let $(x^k{}_0, y^k{}_0)$ denote the center of the macroblock at Level k in the target frame. The procedure for hierarchical motion vector search for the macroblock centered at $(x^0{}_0, y^0{}_0)$ in the Target frame can be outlined as follows:

PROCEDURE 10.3 Motion-vector:hierarchical-search

```
BEGIN
    // Get macroblock center position at the lowest resolution Level k
    x^k_0 = x^0_0 /2^k ;      y^k_0 = y^0_0 / 2^k;
    Use Sequential (or 2D Logarithmic) search method 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 at Level
        k − 1 centered at
        (2(x^k_0+u^k) − 1 ≤ x ≤  2(x^k_0+u^k) + 1; 2(y^k_0+v^k)−1 ≤ y ≤ 2(y^k_0 +v^k) + 1 );
        IF k = 1 THEN last = TRUE;
        k = k − 1;
        Assign (x^k_0; y^k_0 ) and (u^k, v^k) with the new center location and MV;
    }
END
```

Computations

- W=2p+1=31, N=16 (p=15)
- Reduced size

$$\left[\left(2\left\lceil\frac{p}{4}\right\rceil+1\right)^2\left(\frac{N}{4}\right)^2+9\left(\frac{N}{2}\right)^2+9N^2\right]$$

- 4176 Comparison per Macroblock

Outline

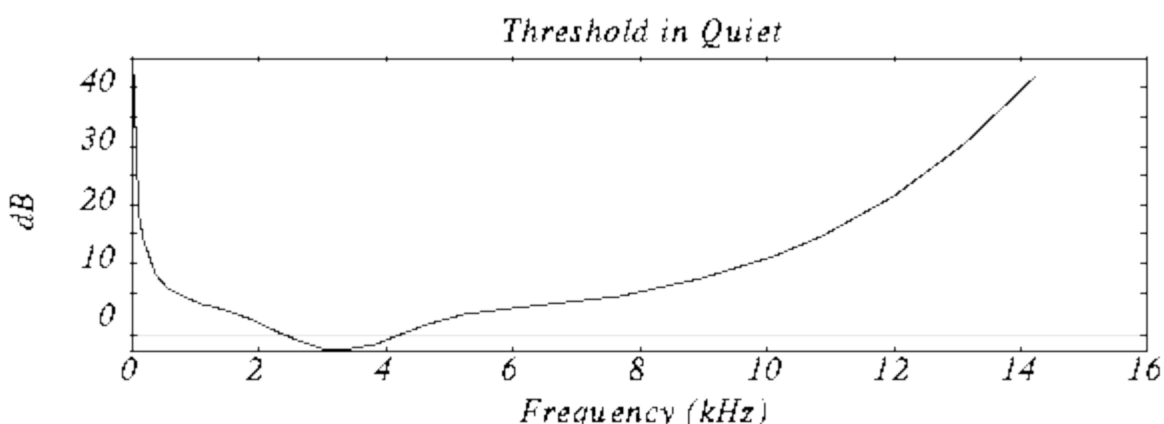
- Jpeg
- H.261
- Audio

Lossy coding: Perceptual Coding

- Hide errors where humans will not see or hear it
  - Study hearing and vision system to understand how we see/hear
  - Masking refers to one signal overwhelming/hiding another (e.g., loud siren or bright flash)
- Natural Bandlimitng
  - Audio perception is 20-20 kHz but most sounds in low frequencies (e.g., 2 kHz to 4 kHz)
  - Low frequencies may be encoded as single channel

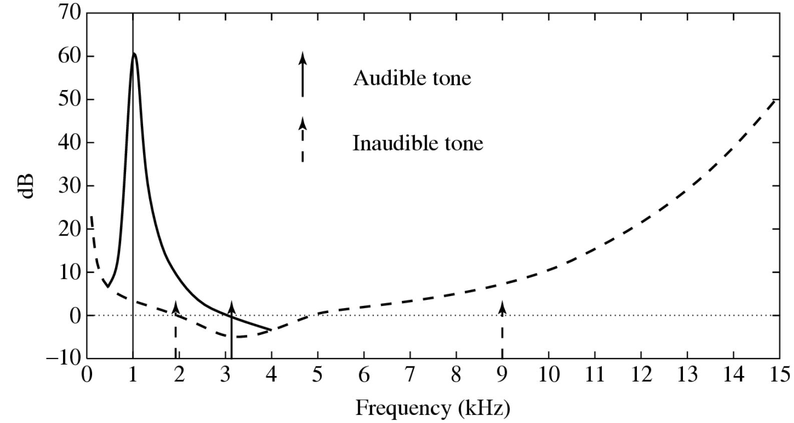
Psychoacoustic Model

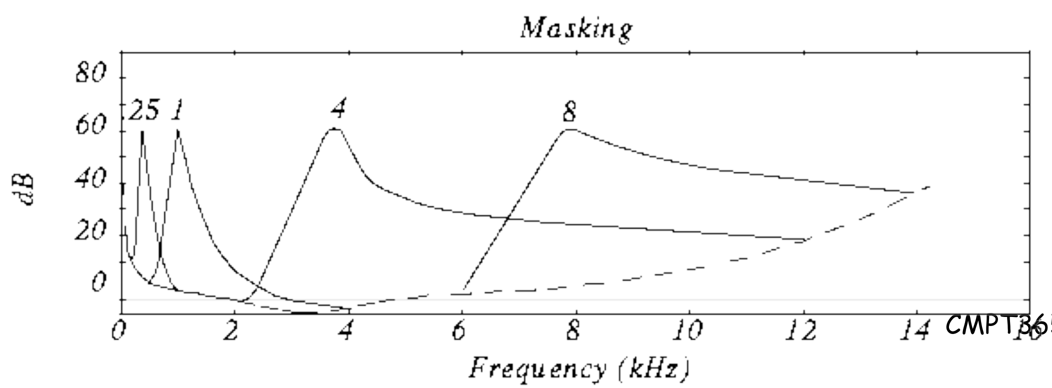
- Basically: If you can't hear the sound, don't encode it
  - Frequency range is about 20 Hz to 20 kHz, most sensitive at 2 to 4 KHz.
  - Dynamic range (quietest to loudest) is about 96 dB
  - Normal voice range is about 500 Hz to 2 kHz
    - Low frequencies are vowels and bass
    - High frequencies are consonants
- Threshold of Hearing
  - Experiment: Put a person in a quiet room. Raise level of 1 kHz tone until just barely audible. Vary the frequency and plot



Psychoacoustic Model con'td

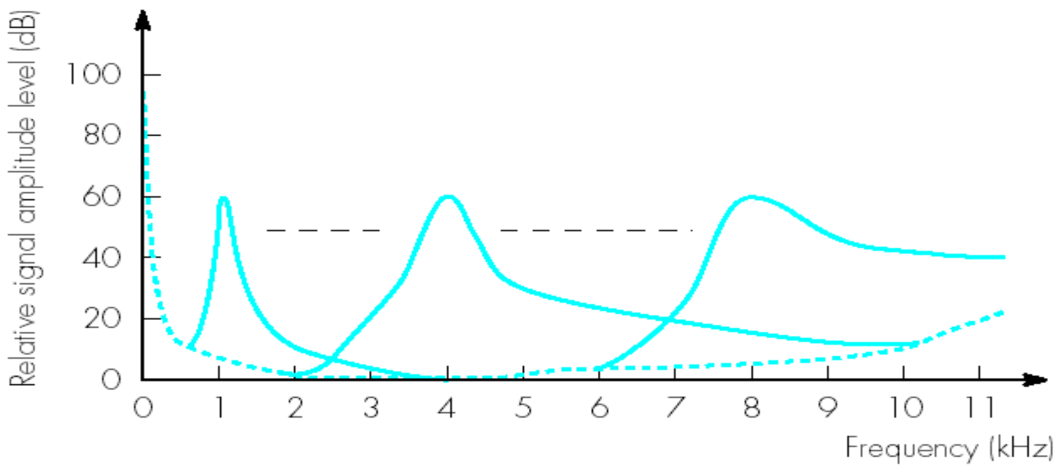
- Frequency masking: Do receptors interfere with each other?
- Experiment:
  - Play 1 kHz tone (*masking tone*) at fixed level (60 dB). Play *test tone* at a different level and raise level until just distinguishable.
  - Vary the frequency of the test tone and plot the threshold when it becomes audible:





Psychoacoustic Model con't d

- Frequency masking: If within a critical band a stronger sound and weaker sound compete, you can't hear the weaker sound. Don't encode it.

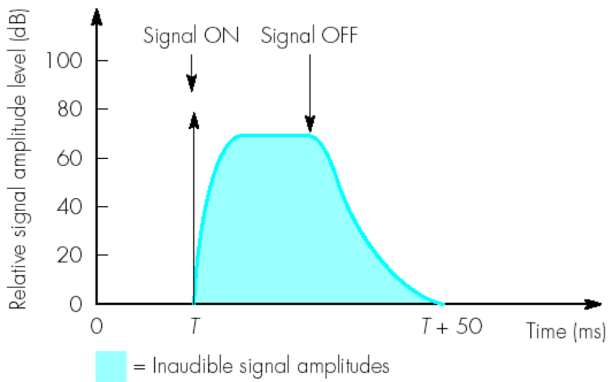


Our brains perceive the sounds through 25 distinct ***critical bands***. The bandwidth grows with frequency (above 500Hz).

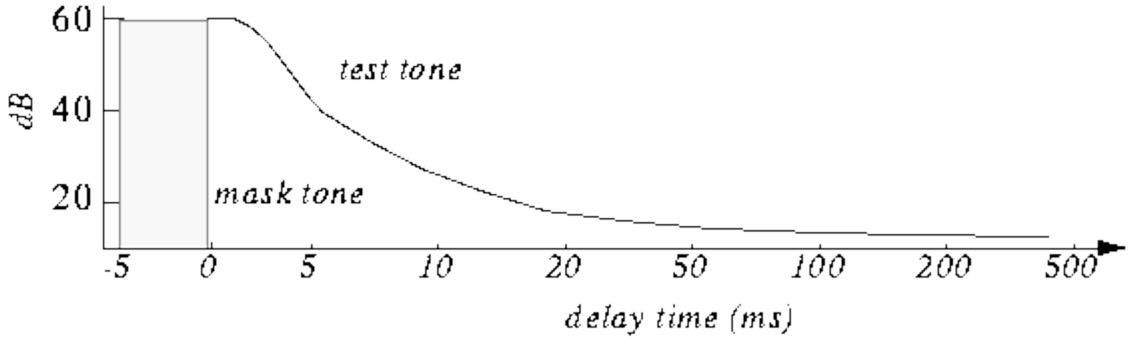
- At 100Hz, the bandwidth is about 160Hz;
- At 10kHz it is about 2.5kHz in width.

Psychoacoustic Model con'td

- Temporal masking: If we hear a loud sound, it takes a little while until we can hear a soft tone nearby.



- Experiment:
  - Play 1 kHz *masking tone* at 60 dB, plus a *test tone* at 1.1 kHz at 40 dB. Test tone can't be heard (it's masked). Stop masking tone, then stop test tone after a short delay.
  - Adjust delay to the shortest time when test tone can be heard.
  - Repeat with different level of the test tone and plot:



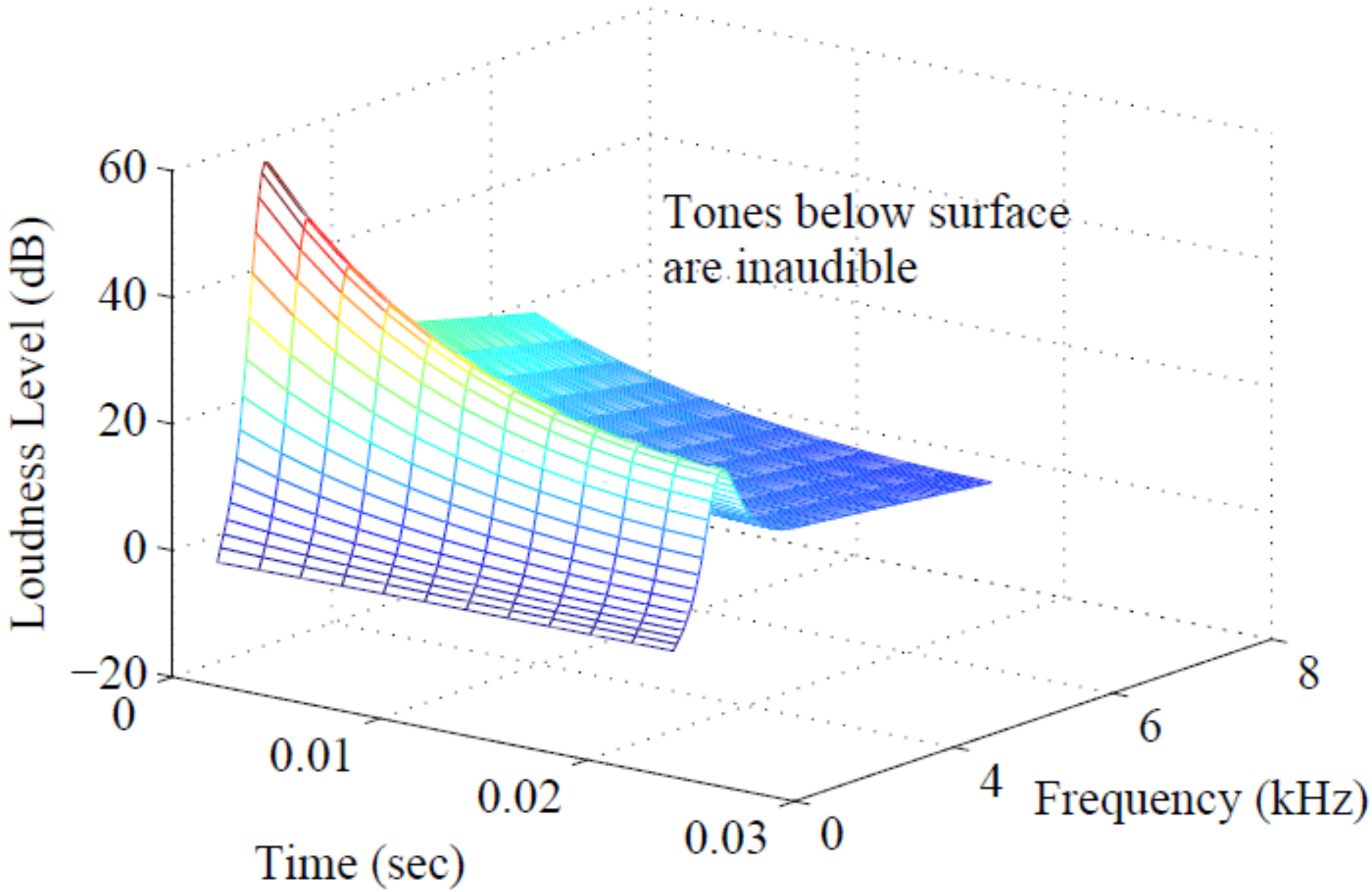
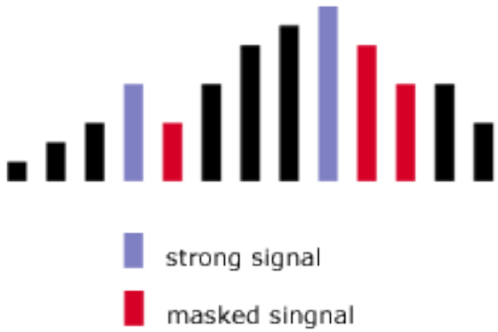


Fig. 14.7: Effect of temporal masking depends on both time and closeness in frequency.

Perceptual Coding

- Makes use of psychoacoustic knowledge to reduce the amount of information required to achieve the same perceived quality (lossy compression)



- Example:
  - Sony MiniDisc uses Adaptive TRAnsform Coding (ATRAC) to achieve a 5:1 compression ratio (about 141 kbps)
  - MPEG audio (MP3)

http://www.mpeg.org
http://www.minidisc.org/aes_atrac.html

MPEG Layers

- MPEG audio offers three compatible *layers*:
  - Each succeeding layer able to understand the lower layers
  - Each succeeding layer offering more complexity in the psychoacoustic model and better compression for a given level of audio quality
  - each succeeding layer, with increased compression effectiveness, accompanied by extra delay
- The objective of MPEG layers: a good tradeoff between quality and bit-rate

MPEG Audio Strategy

- • **MPEG approach to compression** relies on:
  - Quantization
  - Inaccuracy of human auditory system within the width of a critical band

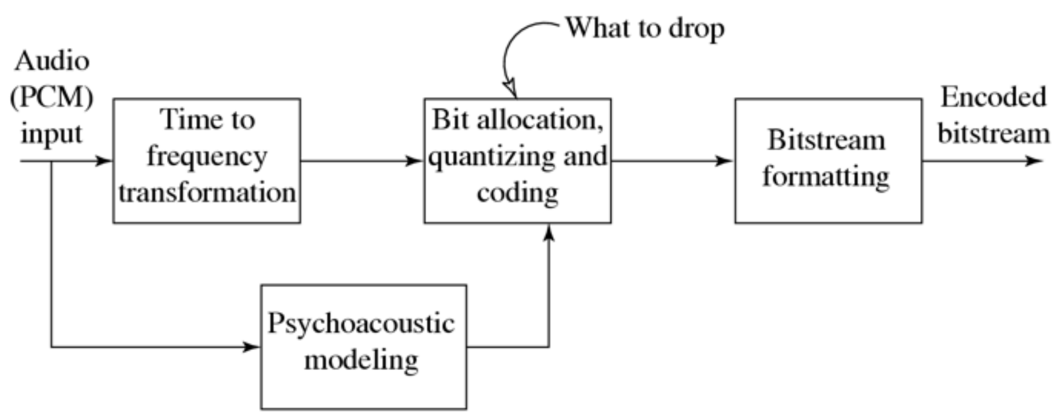
- • **MPEG encoder** employs a bank of filters to:
  - Analyze the frequency ("spectral") components of the audio signal by calculating a frequency transform of a window of signal values
  - Decompose the signal into subbands by using a bank of filters (Layer 1 & 2: "quadrature-mirror"; Layer 3: adds a DCT; psychoacoustic model: Fourier transform)

MPEG Audio Strategy (Cont'd)

- **Frequency masking**: by using a psychoacoustic model to estimate the just noticeable noise level:
  - Encoder balances the masking behavior and the available number of bits by discarding inaudible frequencies
  - Scaling quantization according to the sound level that is left over, above masking levels
- May take into account the actual width of the critical bands:
  - For practical purposes, audible frequencies are divided into 25 main critical bands (Table 14.1)
  - To keep simplicity, adopts a *uniform* width for all frequency analysis filters, using 32 overlapping subbands

Algorithm

- Divide the audio signal (e.g., 48 kHz sound) into 32 frequency subbands --> *subband filtering.*
  - Modified discrete cosine transform (MDCT) -
- Masking for each band caused by nearby band
  - *psychoacoustic model*
  - If the power in a band is below the masking threshold, don't encode it.
  - Otherwise, determine number of bits needed to represent the coefficient such that noise introduced by quantization is below the masking effect
    - One fewer bit introduces about 6 dB of noise).
- Format bitstream



(a) MPEG Audio Encoder

Example

- After analysis, the first levels of 16 of the 32 bands:

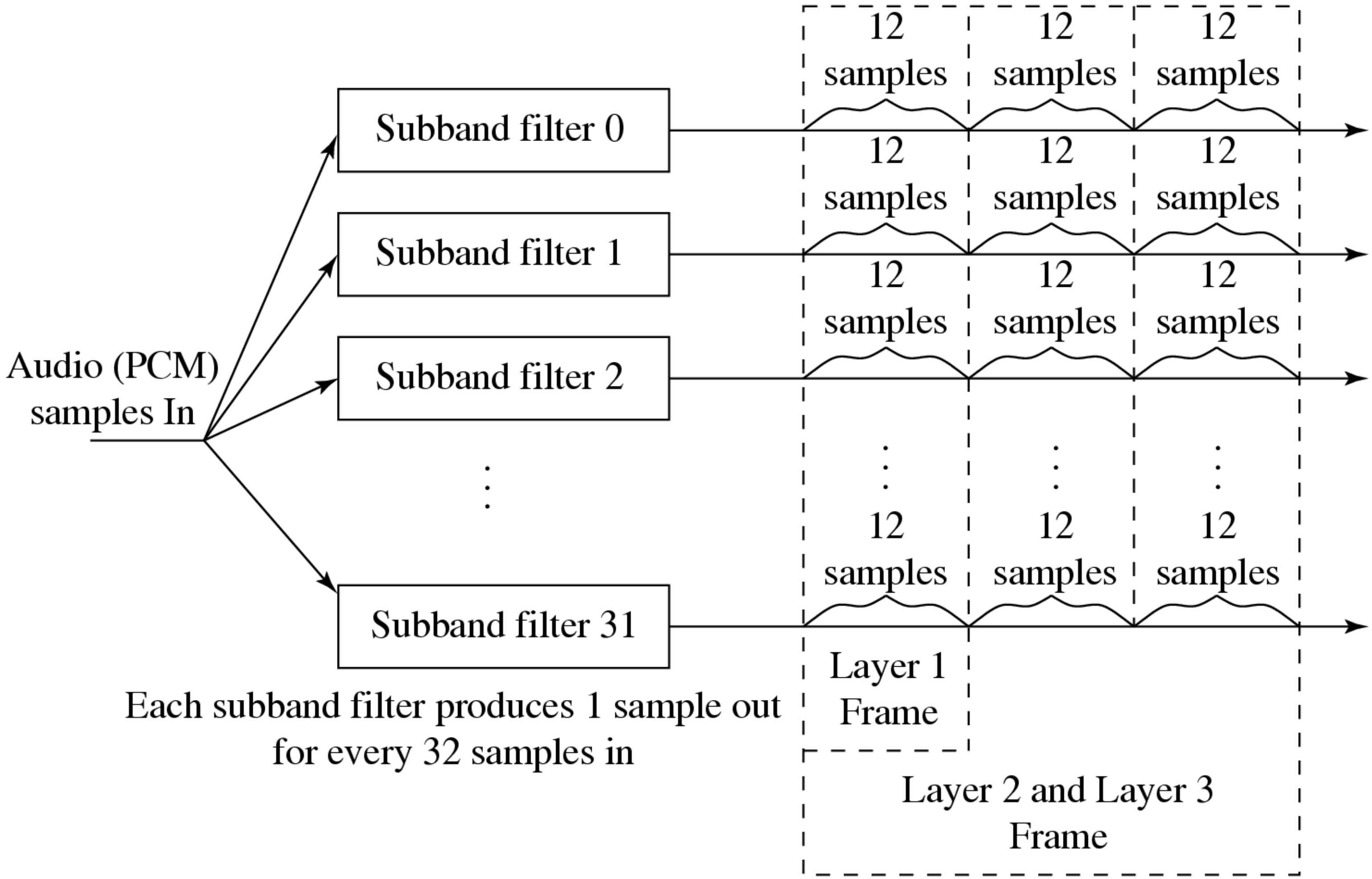
Band	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Level (db)	0	8	12	10	6	2	10	60	35	20	15	2	3	5	3	1

- If the level of the 8th band is 60dB, it gives a masking of 12 dB in the 7th band, 15dB in the 9th.
- Level in 7th band is 10 dB (< 12 dB), so ignore it.
- Level in 9th band is 35 dB (> 15 dB), so send it.

[Only the amount above the masking level needs to be sent, so instead of using 6 bits to encode it, we can use 4 bits -- a saving of 2 bits (12 dB).]

Basic Algorithm (Cont'd)

- The algorithm proceeds by dividing the input into 32 frequency subbands, via a filter bank
  - A linear operation taking 32 PCM samples, sampled in time; output is 32 frequency coefficients
- In the Layer 1 encoder, the sets of 32 PCM values are first assembled into a set of 12 groups of 32s
  - an inherent time lag in the coder, equal to the time to accumulate 384 (i.e., 12×32) samples
- Fig.14.11 shows how samples are organized
  - A Layer 2 or Layer 3, frame actually accumulates more than 12 samples for each subband: a frame includes 1,152 samples



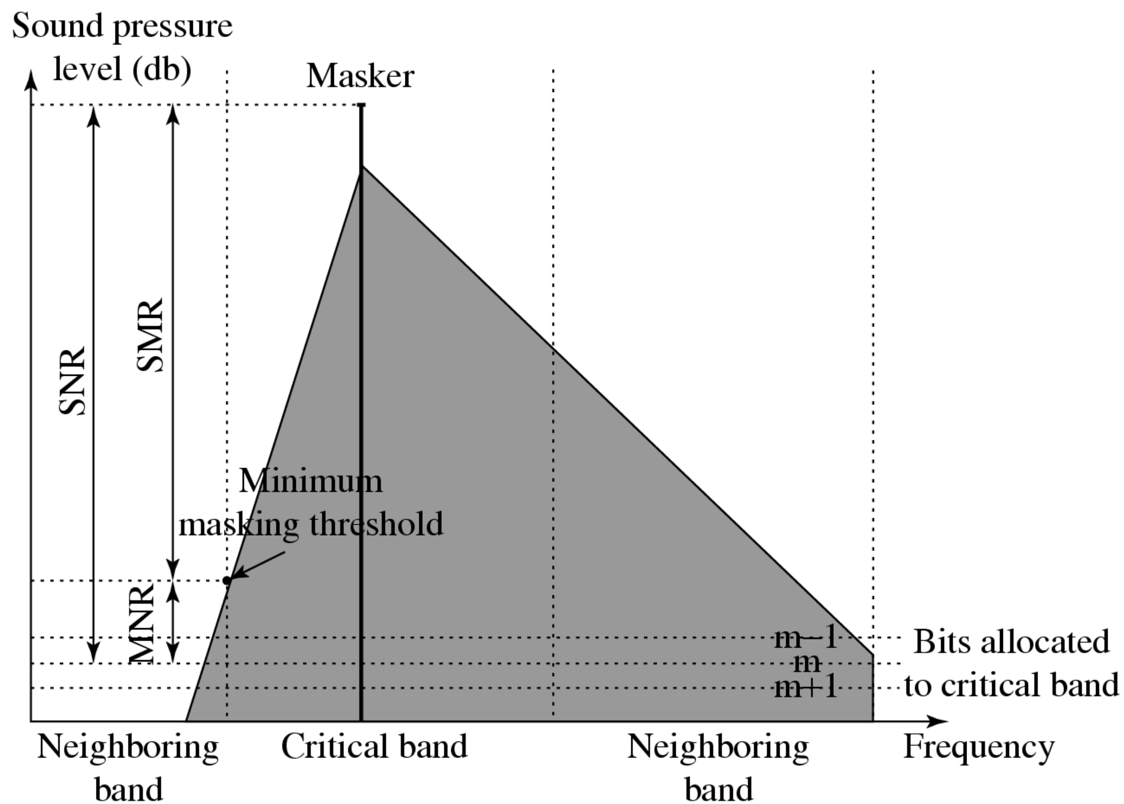
❒ **Fig. 14.11: MPEG Audio Frame Sizes**

Bit Allocation Algorithm

- **Aim**: ensure that all of the quantization noise is below the masking thresholds
- **One common scheme**:
  - For each subband, the psychoacoustic model calculates the *Signal-to-Mask Ratio* (SMR)in dB
  - Then the "Mask-to-Noise Ratio" (MNR) is defined as the difference (as shown in Fig.14.12):

$$\mathrm{MNR}_{\mathrm{dB}} \equiv \mathrm{SNR}_{\mathrm{dB}} - \mathrm{SMR}_{\mathrm{dB}} \qquad (14.6)$$

  - The lowest MNR is determined, and the number of code-bits allocated to this subband is incremented
  - Then a new estimate of the SNR is made, and the process iterates until there are no more bits to allocate



- **Fig. 14.12:** MNR and SMR. A qualitative view of SNR, SMR and MNR are shown, with one dominate masker and m bits allocated to a particular critical band.