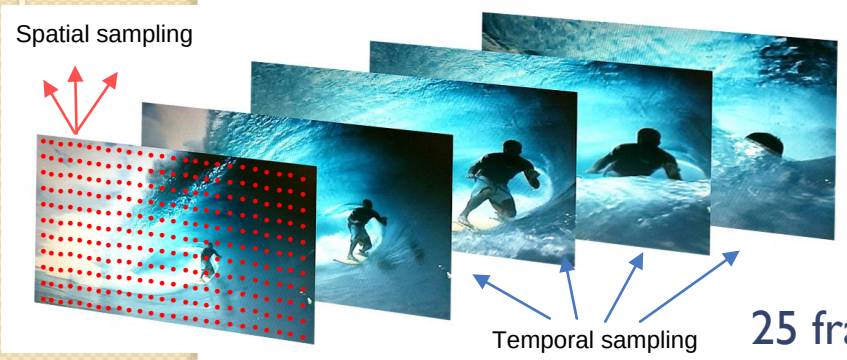
Faculty of Electrical Engineering
University of Montenegro, Podgorica

MULTIMEDIA SYSTEMS AND SIGNALS

Digital video



- **Digital audio signals** - sampled
- **Digital images** - sampled in the spatial domain
- **Digital video signal** - sampled in both space and time, sample is called **frame**

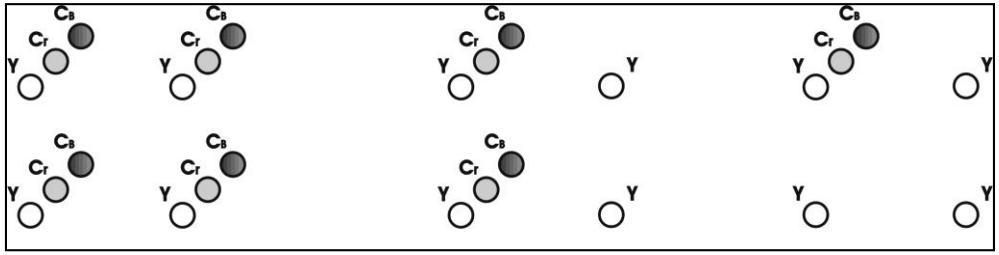
25 frames/s or 30 frames/s – sampling frequency

- Two fields are used instead of frame - one containing **even** and the other with **odd** lines ⇒ sampling rate is 50 fields/s or 60 fields/s
- YCrCb - color model used in digital video ⇒

$$Y = 0.299R + 0.587G + 0.114B$$
$$Cb = 0.564(B - Y)$$
$$Cr = 0.713(R - Y)$$

- required number of bits per sampling schemes:

-4:4:4 - 24 b/pixel
-4:2:2 - 16 b/pixel
-4:2:0 - 12 b/pixel



4:4:4 4:2:2 4:2:0

Digital video standards

- **ITU-R BT.601-5** (International Telecommunication Union, Radiocommunications Sector - ITU-R) – standard for digital video broadcasting . Specifies:
- 60 fields/s for NTSC
- 50 fields/s for PAL system
- NTSC requires 525 lines per frame
- PAL system requires 625 lines,
- 8 b/sample in both systems

- The bit rate of video data is 216 Mb/s in both cases

- Frequently used video formats:

- 4CIF - 704x576 pixels
- CIF - 352x288 pixels
- QCIF - 176x144 pixels
- SubQCIF - 128x96 pixels

- video signal bit rate depends on the video frame format
- bit rate of 216 Mb/s corresponds to the quality used in standard television
- it is obvious that the signal must be compressed in order to be transmitted over the network

- MPEG algorithms belong to the ISO standard
- ITU standards cover VCEG algorithms

MPEG - MPEG-1, MPEG-2, MPEG-4, MPEG-7
VCEG - H.261, H.263, H.264

Motion parameters estimation in video sequences

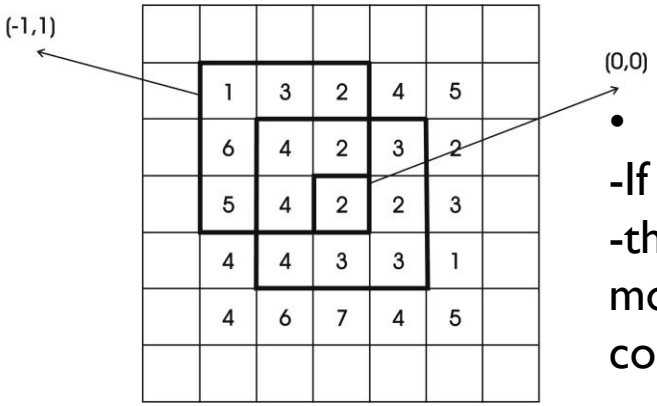
- **block matching technique**
- **logarithmic search**

$$MSE = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}(C_{i,j} - R_{i,j})^2$$

$$SAE = \sum_{i=1}^{N}\sum_{j=1}^{N}\left|C_{i,j} - R_{i,j}\right|$$

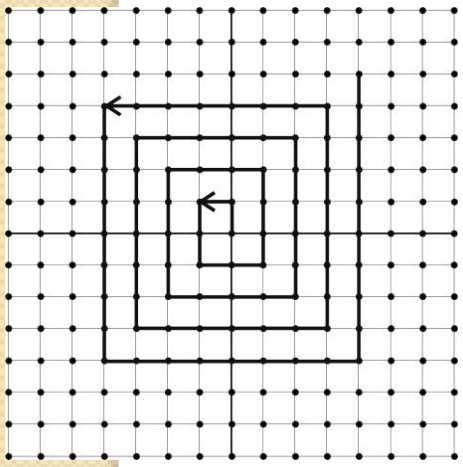
R_{ij} and C_{ij} - pixels in the reference and current frames

1	3	2
6	4	3
5	4	3



- minimal error is compared with a threshold

-If the minimal error is below the threshold, - - -the corresponding position represents the motion vector, which indicates the motion of considered block within the two frames



- procedure for motion vectors estimation in the case of larger blocks is -**full search algorithm**
- blocks of size 16x16
- search area of 31x31 pixels - the search is done over 15 pixels on each side from the central position (0,0)
- method is computationally demanding, since we need to calculate 31x31 MSEs for 16x16 blocks

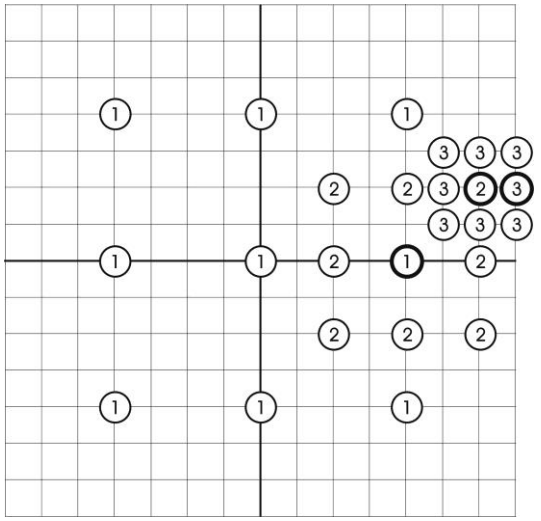
Motion parameters estimation in video sequences

- The fast search algorithms have been defined to reduce the number of calculations, still providing sufficient estimation accuracy
- Steps:

1. We observe the eight positions at the distance of *p* pixels (e.g., *p* = 4) from the central point (0,0). The MSEs are calculated for all nine points

- The position that provides the lowest MSE becomes the central position for the next step

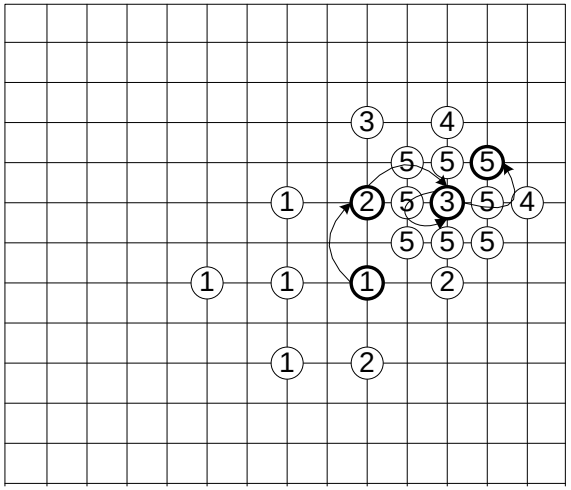
- The search procedure based on the three steps algorithm



Motion parameters estimation in video sequences

2. We consider locations on a distance $p/2$ from the new central position
   - the MSEs are calculated for eight surrounding locations (denoted by 2), and position related to the lowest MSE is a new central position
3. we consider another 8 points around the central position, with the step $p/4$
   -the position with minimal MSE in the third step determines the motion vector

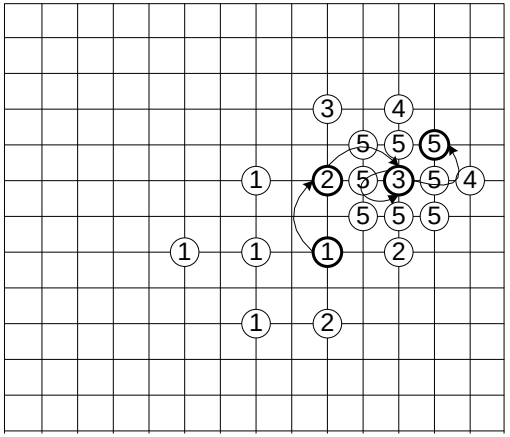
- Another interesting search algorithm is called the logarithmic search



Motion parameters estimation in video sequences

1. In the first iteration, it considers the position that form a "+" shape (denoted by 1)
   - The position with the smallest MSE is chosen for the central point
   2. Then, in the second iteration, the same formation is done around the central point and MSE is calculated
   3. The procedure is repeated until the same position is chosen twice in two consecutive iterations. After that the search continuous by using the closest eight points (denoted by 5)
   4. Finally, the position with the lowest MSE is declared as the motion vector.

The motion vectors search procedures can be combined with other motion parameters estimation algorithms to speed up the algorithm



Digital video compression

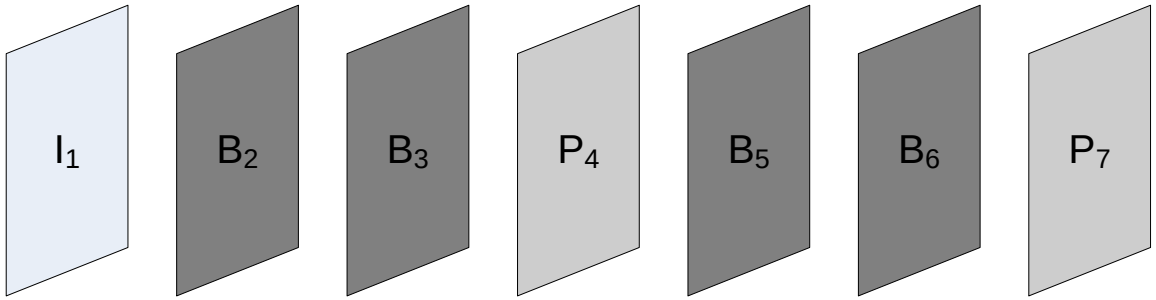
- Algorithms for compression - great importance for multimedia applications and digital video transmission
- The uncompressed video contains large amount of data, which requires significant transmission and storage capacities. Hence, the powerful MPEG algorithms are developed and used.

MPEG-1 video compression algorithm:

- The primary purpose of MPEG-1 algorithm was to store 74 minutes of digital video recording on CD, with a bit rate 1,4 Mb/s
- This bit rate is achieved by using the MPEG-1 algorithm with a VHS video quality.
- A low video quality obtained - one of the main drawbacks that of MPEG-1 algorithm
- MPEG-1 algorithm served as a basis for the development of MPEG-2 and was used in some Internet applications

MPEG-1 video compression algorithm

- The main characteristics of the MPEG-1 algorithm:
  - CIF format (352x288)
  - YCrCb 4:2:0 sampling scheme
- The basic coding units are 16x16 macroblocks
- 16x16 macroblocks are used for the Y component, while given the 4:2:0 scheme, 8x8 macroblocks are used for the Cr and Cb components
- The MPEG-1 algorithm consists of I, B and P frames
- I - firstly displayed, frame
- B and P frames – after I frame
- The scheme continuously repeats:



MPEG-1 video compression algorithm

- I frames are not coded by using the motion estimation
- I frames use only ***intracoding*** - the blocks are compared within the same frame
- P is ***intercoded*** frame and it is based on the forward prediction
  - P frame is coded by using motion prediction from the reference I frame.
- B frame is ***intercoded*** frame as well, but unlike the P frame, the forward and backward motion prediction is used
  - Namely, the motion prediction can be done with respect to I frame or P frame, depending on which gives more optimal results

- Hence, the motion vectors are of particular importance in MPEG algorithms
- They are calculated for the blocks of DCT coefficients

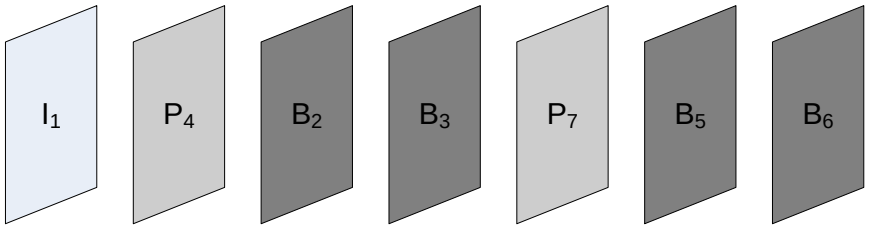
MPEG-1 video compression algorithm

- If we have a video scene in which there is a sudden change in the background happened, at the position of the second B frame
- It is much more efficient to code the first B frame with respect to I frame, while the second B frame is coded with respect to the P frame
- The sequence of frames used for transmission:

- I frame is transmitted first, followed by P and then B frames
- The frame transfer order is:

  I1 P4 B2 B3 P7 B5 ...
- To reconstruct the video sequence, we use the following order:

  I1 B2 B3 P4 B5 ... .



The data structure in MPEG-1 algorithm

The data in MPEG-1 are structured in several levels.

1. **Sequence layer.** The level of sequence contains information about the image resolution and a number of frames per second.
2. **Group of pictures layer.** This level contains information about I, P and B frames. For example, we transmit 12 frames: 1 I frame, 3 P frames, and 8 B frames.
3. **Picture layer.** It carries information on the type of images (e.g., I, P or B), and defines when the picture should be displayed in relation to other pictures.
4. **Slice layer.** Pictures consist of slices, which are further composed of macroblocks. The slice layer provides information about slice position within the picture.
5. **Macroblock layer.** The macroblock level consists of six 8x8 blocks (four 8x8 blocks represent the information about luminance and two 8x8 blocks are used to represent colors).
6. **Block layer.** This level contains the quantized transform coefficients from 8x8 blocks.

MPEG-11 video compression algorithm

- MPEG-2 is a part of the ITU-R 601 standard and it is still present in digital TV broadcasting
- The standard consists of the MPEG-1 audio algorithm, MPEG-2 video algorithm and a system for multiplexing and transmission of digital audio/video signals
- MPEG-2 is optimized for data transfer at a bit rate 3-5Mb/s
- Based on fields rather than he frames, i.e., the field pictures are coded separately
- One frame consists of two fields: odd and even numbered frame lines are placed within two fields
- If we observe a 16x16 block, the fields in DCT domain can be represented by using even and odd lines

- It is possible to split a DCT block into upper and lower blocks of size 16x8, which provides a separate motion estimation, and improves the performance of the algorithm, because a significant difference in motion may exist between fields with lower and higher frequencies

MPEG-1V video compression algorithm

- The MPEG-4 compression algorithm - designed for low bit rates
- **Object-based coding** and **content-based coding**
- The algorithm uses an ***object*** as the basic unit instead of a frame (the entire scene is split into the objects and background)
- Equivalently to the frame, in MPEG-4 we have a **video object plane**
- MPEG-4 provides higher compression ratio - interaction between objects is much higher than among frames

- MPEG-4 video algorithms with very low bit rate video (MPEG-4 VLBV) is basically identical to the H.263 protocol for video communications
- The sampling scheme is 4:2:0 Y:Cr:Cb and it supports formats 16CIF, 4CIF, CIF, QCIF, SubQCIF, with 30 frames/s
- The motion parameters estimation is performed for 16x16 or 8x8 blocks
- DCT is used together with the entropy coding

MPEG-1V video compression algorithm

The data structure of MPEG-4 VLBV algorithm is:

- **Picture layer**
  - provides the information about the resolution, the type of encoding (inter, intra) and relative temporal positions
- **Group of blocks layer**
  - This layer contains a group of macroblocks (with a fixed size defined by the standard) and has a similar function as slices in MPEG-1 and MPEG-2
- **Macroblock layer** - similar to MPEG-1
  - consists of 4 blocks carrying information about luminance and 2 blocks with chrominance components
  - its header contains information about the type of macroblock, about the motion vectors, etc
- **Block layer** consists of coded coefficients from the 8x8 blocks

MPEG-1V video compression algorithm

- Shape coding is used to provide the information about the shape of video object plane - to determine whether a pixel belongs to an object or not
- It defines the contours of the video object
- The shape information can be coded as a binary (pixel either belongs to the object or not) or gray scale information (coded by 8 bits to provide more description about possible overlapping, pixel transparency, etc)
- Objects are encoded by using 16x16 blocks
  - all pixels within the block can completely belong to an object, but can also be on the edge of the object. For blocks that are completely inside the object plane, the motion estimation is performed similarly to MPEG1 and MPEG2 algorithms
  - For the blocks outside the object (blocks with transparent pixels) no motion estimation is performed.

MPEG-1V video compression algorithm

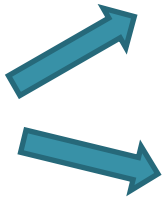
- Motion estimation for the blocks on the boundaries of the video object plane:
  - In the reference frame, the blocks (16x16 or 8x8) on the object boundary are padded by the pixels from the object edge, in order to fill the transparent pixels
  - The block in the current frame is compared with the blocks in the referent frame
  - The MSE (or SAE) is calculated only for pixels that are inside the video object plane

Motion estimation is done for video object plane as follows:

- For the I frame, the motion estimation is not performed
- For the P frame, the motion prediction is based on I frame or the previous P frame
- For the B frame, the video object plane is coded by using the motion prediction from I and P frames (backward and forward)

MPEG-1V video compression algorithm

The MPEG-4 structure



Spatial scalability

Temporal scalability

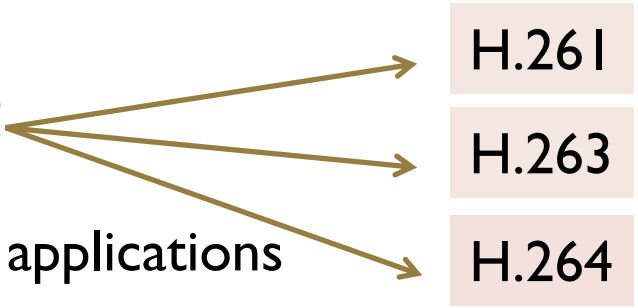
- Resolution could be changed with spatial scaling
- Time resolution for objects and background could be changed with time scaling
  - we can display objects with more frames/s, and the background with less frames/s

Larger backgrounds than the one that is actually displayed at the moment, could be transsmited at the beginning of the video sequence

when zooming or moving the camera, the background information already exists

VCEG algorithms

- VCEG algorithms – for video coding
- Belong to the ITU standards
- More related to the communication applications



H.261

- The main objective - to establish the standards for video conferencing via an ISDN network with a bit rate equal to px64 Kb/s
- A typical bit rates achieved with this standard are in the range 64-384 Kb/s
  - CIF and QCIF formats
  - 4:2:0 YCrCb scheme
  - **The coding unit** is a macroblock containing 4 luminance and 2 chrominance blocks (of size 8x8)
  - This compression approach requires relatively simple hardware and software, but has a poor quality of video signals at bit rates below 100 Mb/s.

VCEG algorithms

H.263

- An extension of H.261standard
- Supports video communication at bit rates below 20 Mb/s with a quite limited video quality
- The functionality of H.263 is identical to the MPEG-4 algorithm
  - 4:2:0 sampling scheme
- motion prediction - for each of the four 8x8 luminance blocks or for the entire 16x16 block
- Introduces an extra PB frame
- Macroblocks of the PB frame contain data from P and B frames, which increases the efficiency of compression (H263+ optional modes)

VCEG algorithms

- H264/MPEG4-AVC is one of the latest standards for video encoding
- Joint project of the ITU-T Video Coding Experts Group (VCEG) and ISO/IEC Moving Picture Experts Group (MPEG)
- Covers many current applications - for mobile phones (mobile TV), video conferencing, IPTV, HDTV, HD video applications, etc

Five types of frames

H.264/MPEG4-AVC supports 5 types of frames:

I frame
P frames
B frames,
SP frames
SI frames

VCEG algorithms

- SP and SI - provide transitions from one bit rate to another
- Intra-coding: 4x4 or 16x16 blocks
- Intra 4x4 coding - based on the prediction of 4x4 blocks
  - used to encode the parts of images that contain the details
- Intra 16x16 coding - based on the 16x16 blocks
  - used to encode uniform (smooth) parts of the frame

VCEG algorithms

SP and SI frames

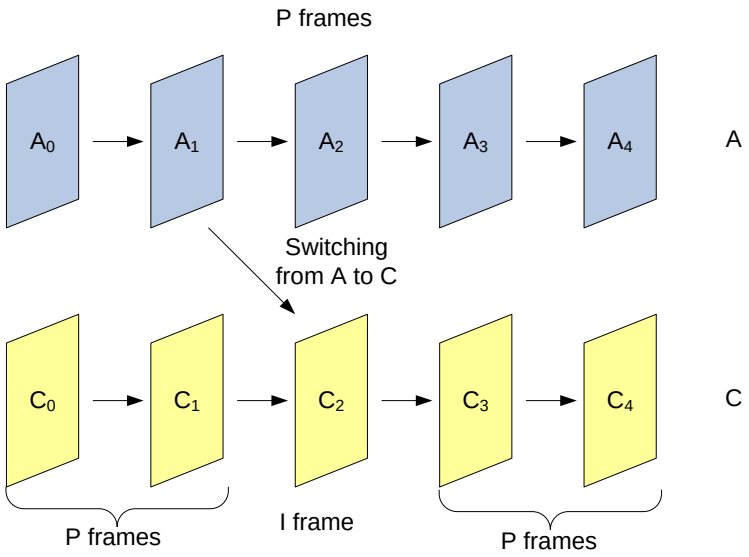
- SP and SI frames are specially encoded
- Provide a transition between different bit rates
- Provide operations such as frame skipping, fast forwarding, the transition between two different video sequences
- The SP and SI frames are added only if it is expected that some of these operations will be carried out

- During the transfer of signals over the Internet, the same video is encoded for different (multiple) bit rates. The decoder attempts to decode the video with the highest bit rate, but often there is a need to automatically switch to a lower bit rate, if the incoming data stream drops

VCEG algorithms

- During the decoding of sequence with bit rate A, we have to switch automatically to the bit rate C
- Also, assume that the P frames are predicted from one reference I frame. After decoding P frames denoted by A0 and A1 (sequence A), the decoder needs to switch to the bit rate C and decode frames C2, C3, etc. Now, since the frames in the sequence C are predicted from other I frames, the frame in A sequence is not appropriate reference for decoding the frame in C sequence.

One solution is to determine a priori the transition points (e.g., the C2 frame within the C sequence) and to insert an I frame



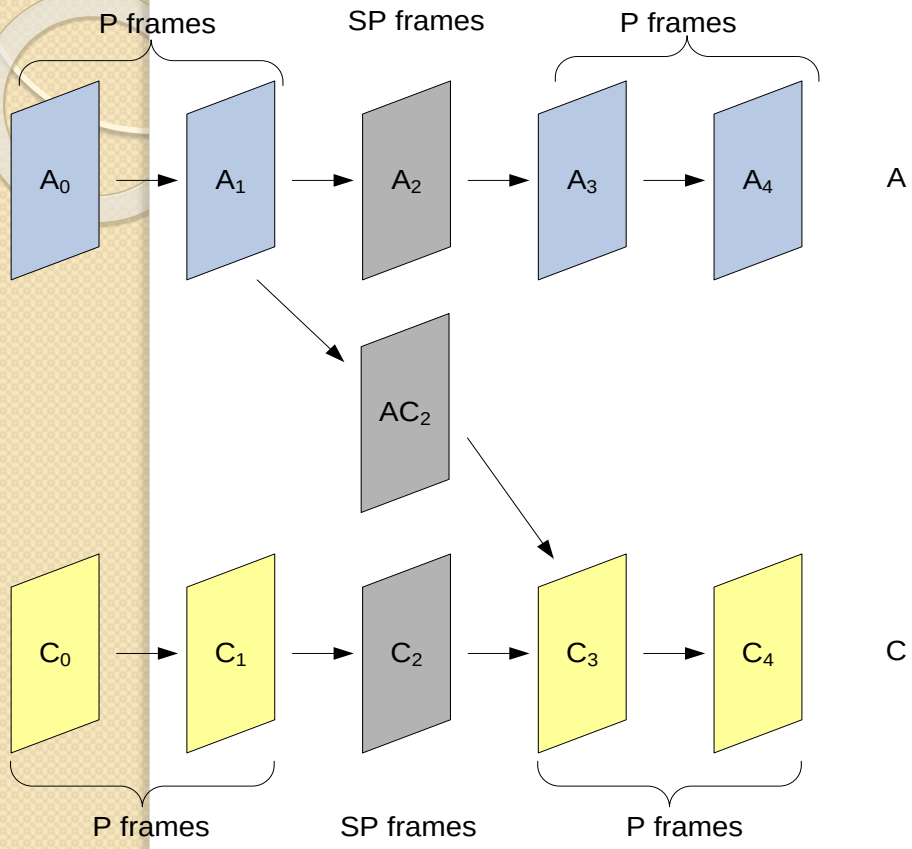
VCEG algorithms

As a result of inserting I frames, the transitions between two video sequences would produce peaks in the bit rate

SP-frames are designed to support the transition from one bit rate to another, without increasing the number of I frames.

- Transition points are defined by SP frames (A2, C2 and AC2)
- We can distinguish two types of SP frames: primary (A2 and C2, which are the parts of the video sequences A and C) and switching SP frame
- If there is no transition, SP frame A2 is decoded by using the frame A1, while the SP frame C2 is decoded using C1

VCEG algorithms



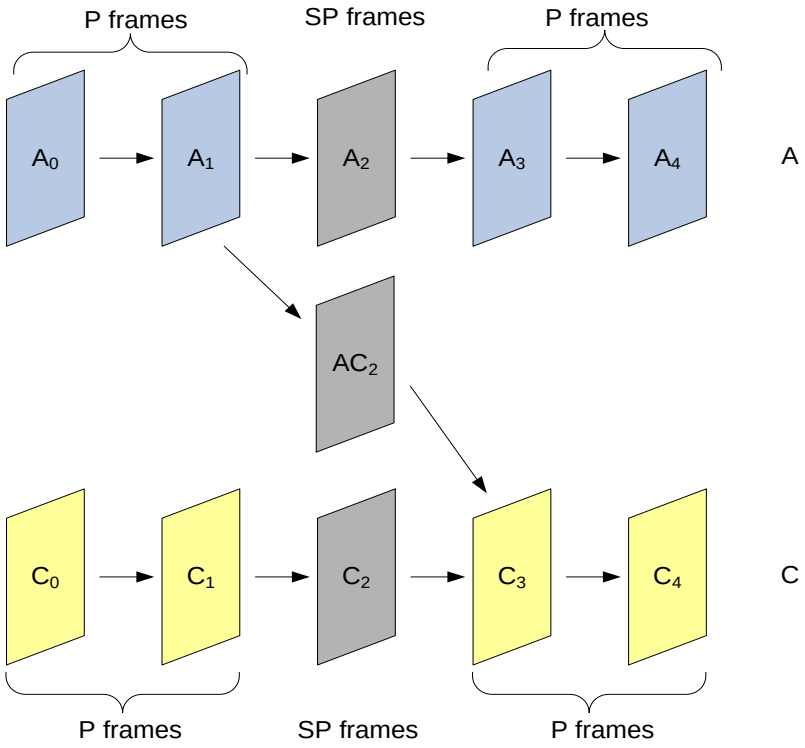
- When switching from A to C sequence, the switching secondary frame (AC2) is used
  - AC2 should provide the same reconstruction as the primary SP frame C2 and to be reference frame for C3
  - The switching frame needs to have characteristics that provide the smooth transition between the sequences

Unlike coding of the P frames, the SP frames coding requires an additional re-quantization procedure with a quantization step that corresponds to the step used in the switching SP frame

VCEG algorithms

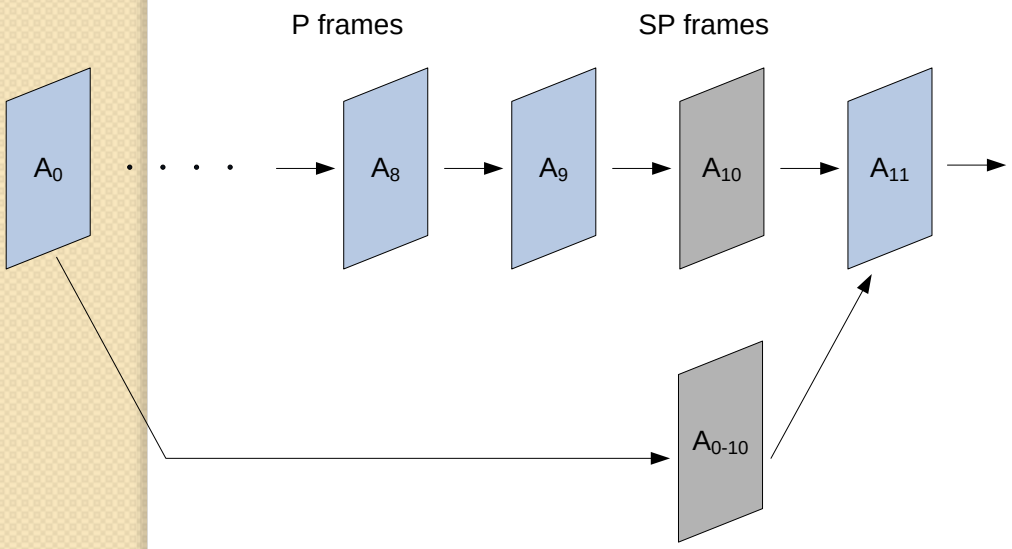
- Obviously, the switching frame should contain also the information about the motion vector correction in order to provide an identical reconstruction in both cases: with and without the switching between the sequences

In the case of switching from C to A bit rate, the switching frame CA2 is needed



VCEG algorithms

- Another application of SP frames is to provide arbitrary access to the frames of a video sequence, as shown in figure
- SP frame (A10) and the switching SP frame (A0-10) are on the position of the 10th frame. The decoder perform a fast forward from the A-0 frame to the A-11 frame, by first decoding A0, then the switching SP frame A0-10, which will used the motion prediction from A0 to decode the frame A11



The second type of transition frames are SI frames. They are used in a similar way as SP frames. These frames can be used to switch between completely different video sequences

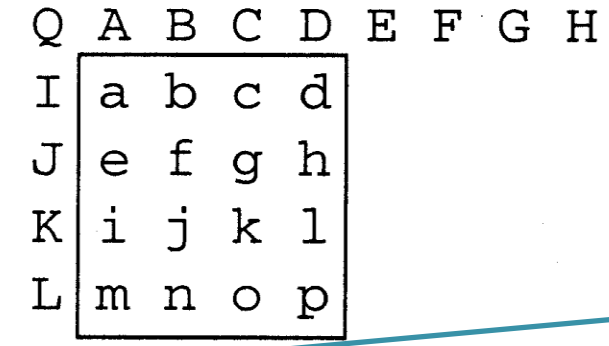
Intra coding in the spatial domain

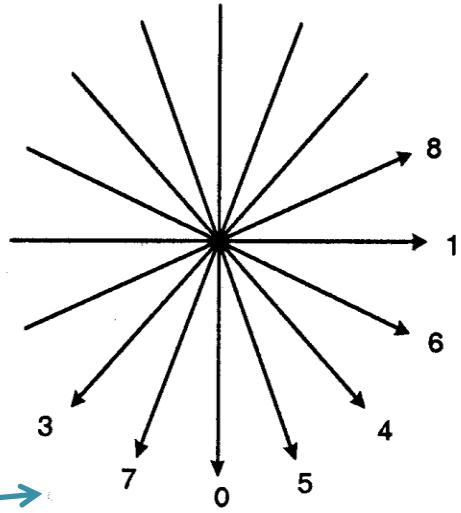
- H264/MPEG4 intra coding is performed in the spatial domain – pixel domain
- For the intra coding, the prediction of each 4x4 block is based on the neighboring pixels

Sixteen pixels in the 4x4 block are denoted by a, b ,...., p and they are coded by using the pixels: A, B, C, D, E, F, G, H, I, J, K, L, Q, belonging to the neighboring blocks

Intra 4x4 prediciton of block a-p based o the pixels A-Q

Eight prediction directions for Intra coding

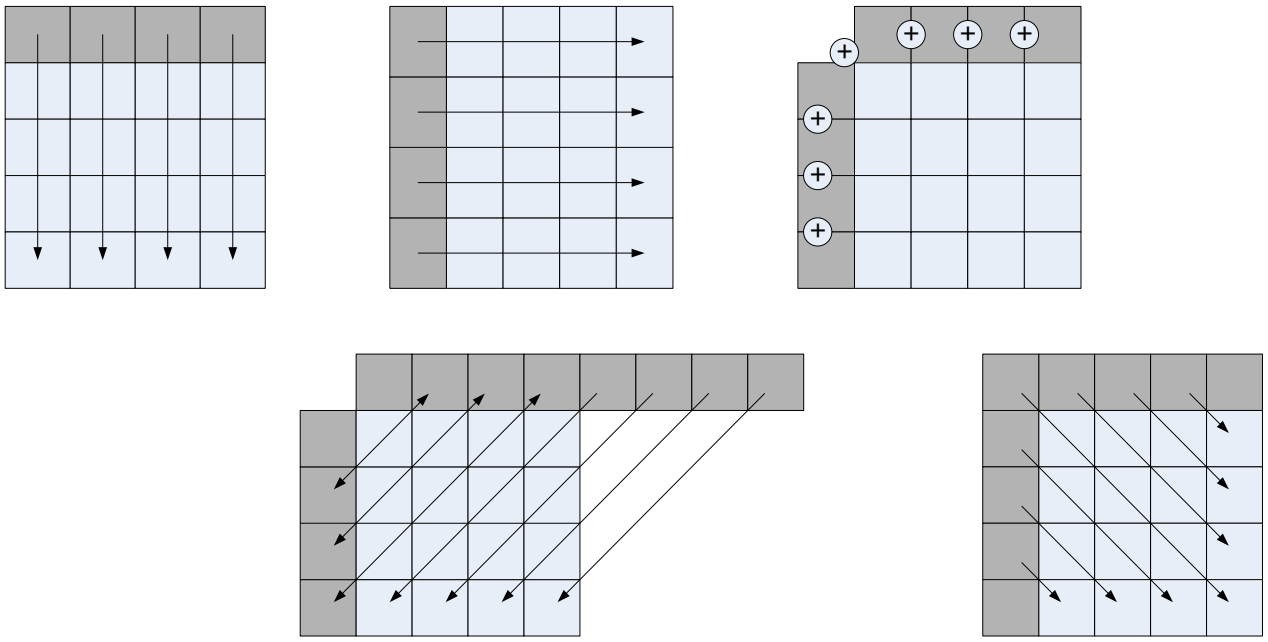




Intra coding in the spatial domain

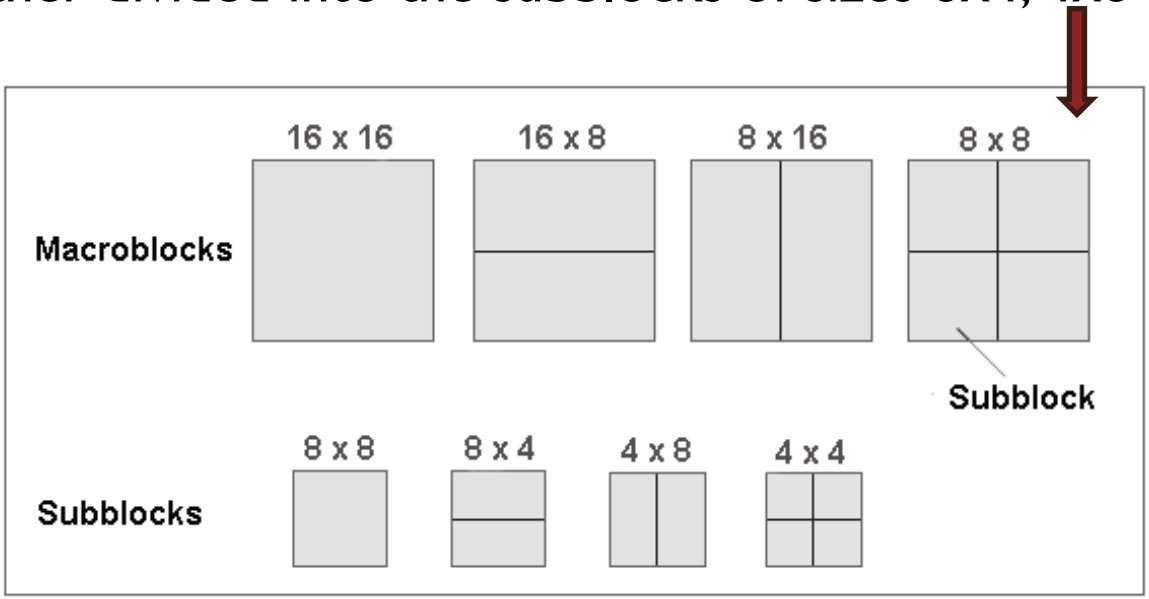
The vertical prediction indicates that the pixels above the current 4x4 block are copied to the appropriate positions according to the illustrated direction

Horizontal prediction indicates that the pixels are copied to the marked positions on the left side



Inter frame prediction with increased accuracy of motion parameters estimation

- Uses blocks of sizes 16x16, 16x8, 8x16, and 8x8. The 8x8 can be also further divided into the subblocks of sizes 8x4, 4x8 or 4x4

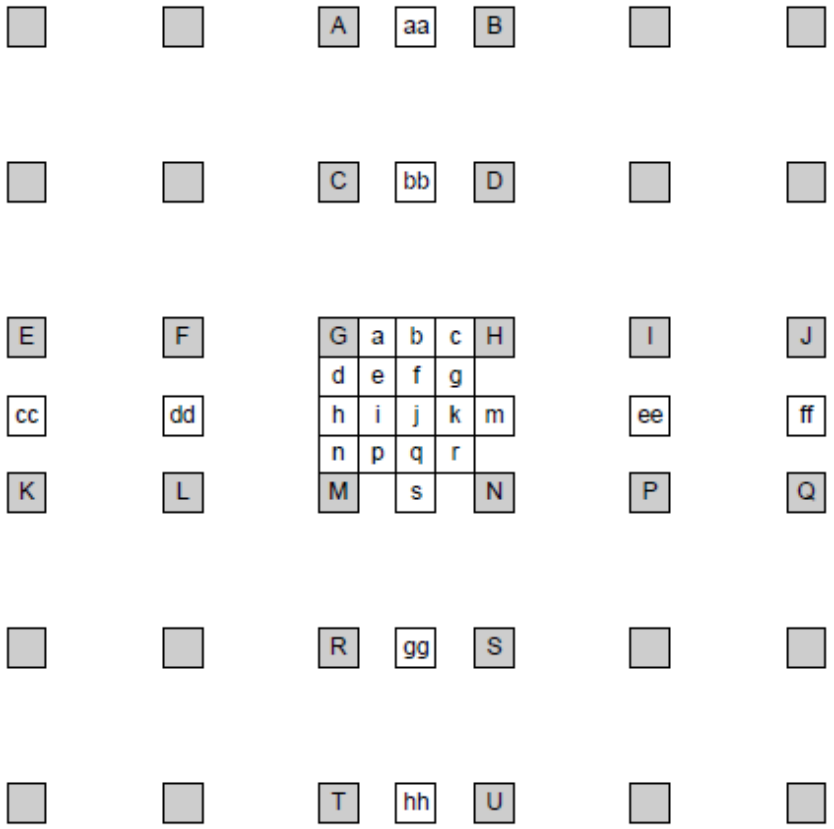


- H264/MPEG4 standard provides higher precision for the motion vectors estimation in comparison to other algorithms
- Its accuracy is equal to 1/4 of pixels distance in the luminance component
- For other algorithms, the precision is usually 1/2 of the distance

Inter frame prediction with increased accuracy of motion parameters estimation

If the motion vector does not indicates an integer position (within the existing pixels grid), the corresponding pixel can be obtained by using the interpolation

- Firstly, a 6-tap FIR filter is used to obtain the interpolation accuracy equal to ½
- Filter coefficients are (1, -5, 20, 20, -5, -1- it is **low-pass filter**
- The bilinear filter is applied to obtain the precision equal to 1/4 pixel



Inter frame prediction with increased accuracy of motion parameters estimation

Pixels *b, h, j, m* and *s* are obtained following the relations:

$$b = ((E - 5F + 20G + 20H - 5I + J) + 16)/32$$
$$h = ((A - 5C + 20G + 20M - 5R + T) + 16)/32$$
$$m = ((B - 5D + 20H + 20N - 5S + U) + 16)/32$$
$$s = ((K - 5L + 20M + 20N - 5P + Q) + 16)/32$$
$$j = ((cc - 5dd + 20h + 20m - 5ee + ff) + 512)/1024 \quad or$$
$$j = ((aa - 5bb + 20b + 20s - 5gg + hh) + 512)/1024$$

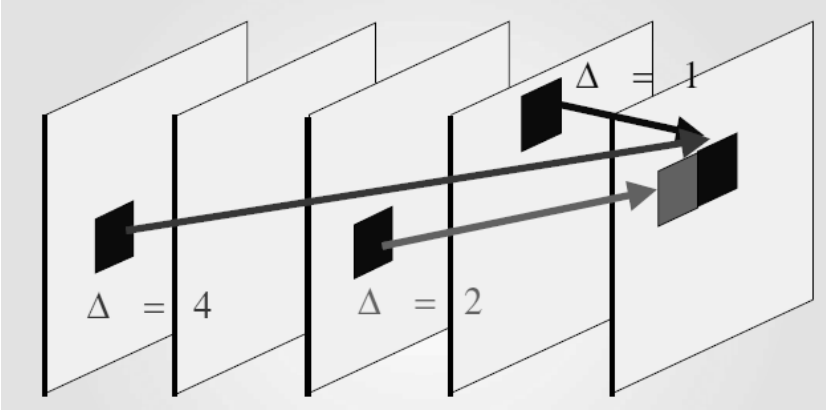
- To obtain a pixel *j*, it is necessary to calculate the values of pixels *cc*, *dd*, *ee*, and *ff*, or *aa*, *bb*, *gg*, and *hh*
- Pixels placed at the quarter of the distance between the pixels *a, c, d, e, f, g, i, k, n, p, q* are obtained as:

$$a = \frac{(G+b+1)}{2} \qquad c = \frac{(H+b+1)}{2}$$
$$d = \frac{(G+h+1)}{2} \qquad n = \frac{(M+h+1)}{2}$$
$$f = \frac{(b+j+1)}{2} \qquad i = \frac{(h+j+1)}{2}$$
$$k = \frac{(j+m+1)}{2} \qquad q = \frac{(j+s+1)}{2}$$
$$e = \frac{(b+h+1)}{2} \qquad g = \frac{(b+m+1)}{2}$$
$$p = \frac{(h+s+1)}{2} \qquad r = \frac{(m+s+1)}{2}$$

Multiple reference frames

- H264 introduces the concept of **multiple reference frames**
- Decoded reference frames are stored in the buffer (up to 16 frames)

- Finding the best possible references from the two sets of buffered frames - **List 0** is a set of past frames, **List 1** is a set of future frames



- The prediction for the block is calculated as a weighted sum of blocks from different multiple reference frames
- It is used in the scenes where there is a change in perspective, zoom, or the scene where new objects appear

  - Generalization of the B frames concept
  - B frames (bi-directional frames) can be encoded so that some macroblocks are obtained as the mean prediction based on different images from the list 0 and list 1

Hence, H264/MPEG4 allows 3 types of inter prediction: list 0, list 1, and bi-directional prediction

Coding in the transformation domain using the integer transform (please use the book for more details)

- H264/MPEG4 - encode the difference between the reference frame and the frame obtained by prediction
- Unlike previous standards (such as MPEG2 and H263) based on the DCT coefficients, H264/MPEG4 uses **integer transform**
- Integer transform is simpler to implement and allows more accurate inverse transform
- Commonly used 4x4 transform matrix is given by:

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

H264 uses advanced entropy coding, such as CAVLC (context adaptive variable length coding), and especially CABAC (context based adaptive arithmetic coding)

H264 provides significantly better compression ratio than the existing standards

Examples

2. Determine the bit rate of the PAL video sequence for CIF format and sampling schemes:

4:4:4
4:2:2
4:2:0

Solution:

The CIF format resolution is 352x288. Hence, we obtain the following bit rates:

a) 352·288·24b·25/s= 60825600b/s = 60.825Mb/s
b) 352·288·16b·25/s= 40550400b/s = 40.55Mb/s
c) 352·288·12b·25/s=30412800 b/s = 30.412Mb/s

Examples

3.How many minutes of uncompressed video data in CIF format with sampling scheme 4:2:2 can be stored on a DVD (capacity 4,7GB)? The PAL system is assumed.

Solution:

$$t=\frac{4.7\cdot 1024\cdot 1024\cdot 1024\cdot 8b}{25\cdot 352\cdot 288\cdot 16b/s}\approx 16.5\,\mathrm{min}$$

Examples

4. Consider a 3x3 block of pixels within a current frame and the 5x5 region centered at the same position in the reference frame. Determine the motion vector by using the block matching technique based on the MSE and assuming that the motion vector is within the given 5x5 block. The threshold value is 2.

1	4	7
9	11	8
4	5	6

2	5	7	17	19
9	11	8	8	5
4	6	6	4	1
0	9	15	7	4
4	8	7	3	1

Solution:

The observed 3x3 block is compared with the corresponding 3x3 block (within 5x5 block) in the reference frame, centered at (0,0). The MSE is calculated. Then, the procedure is repeated for eight positions around the central one.

Examples

$$\begin{matrix}1 & 4 & 7\\ 9 & 11 & 8\\ 4 & 5 & 6\end{matrix} \xleftrightarrow{MSE_{00}} \begin{matrix}11 & 8 & 8\\ 6 & 6 & 4\\ 9 & 15 & 7\end{matrix}$$

$MSE_{00}=((1-11)^2+(4-8)^2+(7-8)^2+(9-6)^2+(11-6)^2+(8-4)^2+(4-9)^2+(5-15)^2+(6-7)^2)/9=32.55$

$$\begin{matrix}1 & 4 & 7\\ 9 & 11 & 8\\ 4 & 5 & 6\end{matrix} \xleftrightarrow{MSE_{-10}} \begin{matrix}9 & 11 & 8\\ 4 & 6 & 6\\ 0 & 9 & 15\end{matrix}$$

$MSE_{-10} = ((1-9)^2+(4-11)^2+(7-8)^2+(9-4)^2+(11-6)^2+(8-6)^2+(4-0)^2+(5-9)^2+(6-15)^2)/9= 31.22$

$$\begin{matrix}1 & 4 & 7\\ 9 & 11 & 8\\ 4 & 5 & 6\end{matrix} \xleftrightarrow{MSE_{-11}} \begin{matrix}2 & 5 & 7\\ 9 & 11 & 8\\ 4 & 6 & 6\end{matrix}$$

$MSE_{-11} = ((1-2)^2+(4-5)^2+(7-7)^2+(9-9)^2+(11-11)^2+(8-8)^2+(4-4)^2+(5-6)^2+(6-6)^2)/9=0.33$

$$\begin{matrix}1 & 4 & 7\\ 9 & 11 & 8\\ 4 & 5 & 6\end{matrix} \xleftrightarrow{MSE_{11}} \begin{matrix}7 & 17 & 19\\ 8 & 8 & 5\\ 6 & 4 & 1\end{matrix}$$

$MSE_{11} = ((1-7)^2+(4-17)^2+(7-19)^2+(9-8)^2+(11-8)^2+(8-5)^2+(4-6)^2+(5-4)^2+(6-1)^2)/9=38.88$

Examples

$$\begin{matrix}1 & 4 & 7\\ 9 & 11 & 8\\ 4 & 5 & 6\end{matrix} \xleftrightarrow{MSE_{-1-1}} \begin{matrix}4 & 6 & 6\\ 0 & 9 & 15\\ 4 & 8 & 7\end{matrix}$$

$MSE_{-1-1}=((1-4)^2+(4-6)^2+(7-6)^2+(9-0)^2+(11-9)^2+(8-15)^2+(4-4)^2+(5-8)^2+(6-7)^2)/9=17.55$

$$\begin{matrix}1 & 4 & 7\\ 9 & 11 & 8\\ 4 & 5 & 6\end{matrix} \xleftrightarrow{MSE_{0-1}} \begin{matrix}6 & 6 & 4\\ 9 & 15 & 7\\ 8 & 7 & 3\end{matrix}$$

$MSE_{0-1}=((1-6)^2+(4-6)^2+(7-4)^2+(9-9)^2+(11-15)^2+(8-7)^2+(4-8)^2+(5-7)^2+(6-3)^2)/9=9.33$

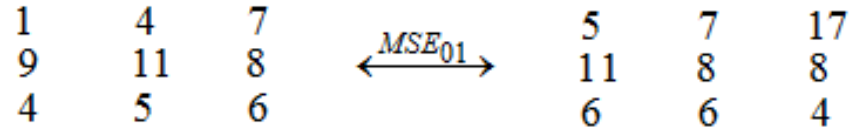
$$\begin{matrix}1 & 4 & 7\\ 9 & 11 & 8\\ 4 & 5 & 6\end{matrix} \xleftrightarrow{MSE_{1-1}} \begin{matrix}6 & 4 & 1\\ 15 & 7 & 4\\ 7 & 3 & 1\end{matrix}$$

$MSE_{1-1}=((1-6)^2+(4-4)^2+(7-1)^2+(9-15)^2+(11-7)^2+(8-4)^2+(4-7)^2+(5-3)^2+(6-1)^2)/9=18.55$

$$\begin{matrix}1 & 4 & 7\\ 9 & 11 & 8\\ 4 & 5 & 6\end{matrix} \xleftrightarrow{MSE_{10}} \begin{matrix}8 & 8 & 5\\ 6 & 4 & 1\\ 15 & 7 & 4\end{matrix}$$

$MSE_{10}=((1-8)^2+(4-8)^2+(7-5)^2+(9-6)^2+(11-4)^2+(8-1)^2+(4-15)^2+(5-7)^2+(6-4)^2)/9=33.88$

Examples



$MSE_{01}=((1-5)^2+(4-7)^2+(7-17)^2+(9-11)^2+(11-8)^2+(8-8)^2+(4-6)^2+(5-6)^2+(6-4)^2)/9=16.33$

Since $\min(MSE_{nm})=MSE_{-11}$, holds, and it is below the threshold ($MSE_{-11}< 2$), we may say that the position (-1,1) represents the motion vector.

5. At the output of the video coder, the average bit rate is R=5.07 Mb/s for CIF video format in PAL system. The quantization is done by the matrix Q_1. The bit rate control system sends the information back to the coder to reduce the bit rate by increasing the quantization step. The coder switches to quantization Q_2 and increases the compression degree. Determine the new average bit rate at the coder output.

The quantization matrices Q_1 and Q_2, as well as a sample 8x8 block of DCT coefficients (from video frames) are given below. In order to simplify the solution, one may assume that the ratio between the compression degrees achieved by Q_1 and Q_2 is proportional to the ratio between the number of non-zero DCT coefficients that remains within the 8x8 block after quantization.

$$Q_1 = \begin{bmatrix} 3 & 5 & 7 & 9 & 11 & 13 & 15 & 17 \\ 5 & 7 & 9 & 11 & 13 & 15 & 17 & 19 \\ 7 & 9 & 11 & 13 & 15 & 17 & 19 & 21 \\ 9 & 11 & 13 & 15 & 17 & 19 & 21 & 23 \\ 11 & 13 & 15 & 17 & 19 & 21 & 23 & 25 \\ 13 & 15 & 17 & 19 & 21 & 23 & 25 & 27 \\ 15 & 17 & 19 & 21 & 23 & 25 & 27 & 29 \\ 17 & 19 & 21 & 23 & 25 & 27 & 29 & 31 \end{bmatrix}$$

$$DCT = \begin{bmatrix} 96 & 35 & 82 & 41 & 11 & 0 & 0 & 0 \\ 70 & 70 & 40 & 21 & 5 & 0 & 0 & 0 \\ 45 & 40 & 20 & 29 & 13 & 19 & 0 & 0 \\ 27 & 44 & 42 & 15 & 0 & 20 & 0 & 0 \\ 34 & 23 & 0 & 35 & 11 & 0 & 10 & 0 \\ 68 & 34 & 32 & 34 & 10 & 10 & 0 & 0 \\ 38 & 25 & 0 & 10 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$Q_2 = \begin{bmatrix} 11 & 21 & 31 & 41 & 51 & 61 & 71 & 21 \\ 21 & 31 & 41 & 51 & 61 & 71 & 81 & 91 \\ 31 & 41 & 51 & 61 & 71 & 81 & 91 & 101 \\ 41 & 51 & 61 & 71 & 81 & 91 & 101 & 111 \\ 51 & 61 & 71 & 81 & 1 & 101 & 111 & 121 \\ 61 & 71 & 81 & 91 & 101 & 111 & 121 & 131 \\ 71 & 81 & 91 & 101 & 111 & 121 & 131 & 141 \\ 81 & 1 & 101 & 111 & 121 & 131 & 141 & 151 \end{bmatrix}$$

Examples

Solution:

Firstly, we calculate the average number of bits per pixel for the given bit rate R_v=5.07Mb/s.

R=25 frame/s · 352·288 pixel · x_1 b/pixel, $x_1 = \frac{10000}{25 \cdot 352 \cdot 288} = 2\text{b/pixel}$

$$DCT_{Q1}\begin{bmatrix} 32 & 7 & 12 & 6 & 2 & 0 & 0 & 0 \\ 14 & 10 & 4 & 3 & 1 & 0 & 0 & 0 \\ 6 & 4 & 4 & 3 & 2 & 5 & 0 & 0 \\ 3 & 4 & 4 & 5 & 0 & 4 & 0 & 0 \\ 3 & 2 & 0 & 2 & 1 & 0 & 1 & 0 \\ 5 & 5 & 4 & 2 & 1 & 0 & 0 & 0 \\ 5 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad DCT_{Q2} = \begin{bmatrix} 9 & 2 & 3 & 1 & 0 & 0 & 0 & 0 \\ 3 & 2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Examples

The number of non-zero coefficients after Q_1 and Q_2 are respectively:

$$No\{DCT_{Q1}\neq 0\}=30 \quad No\{DCT_{Q2}\neq 0\}=15$$

The average number of bits for the observed 8x8 block is:

$$Xb1=x_1\cdot 64 \quad Xb2=x_2\cdot 64$$

Since we assume that the ratio between compression degrees achieved by Q_1 and Q_2 is proportional to the ratio between the number of non-zero coefficients after Q_1 and Q_2, we may write:

$$\frac{Xb1}{Xb2}=k\frac{30}{15} \Rightarrow x_2=\frac{x_1}{2k}=\frac{1}{k}\ \text{b/pixel}$$

The new bit rate obtained at the coder output is: $R_{new}=25\text{frame/s}\cdot 352\cdot 288\text{pixels}\cdot x_2\ \text{b/pixel}=\frac{2.53}{k}$ Mb/s

Examples

6. Consider a video sequence with N=1200 frames. The frames are divided into 8x8 blocks, in order to analyze the stationarity of the coefficients. We assume that the stationary blocks do not vary significantly over the sequence duration. The coefficients from the stationary blocks are transmitted only once (within the first frame). The coefficients from the non-stationary blocks change significantly over time. In order to reduce the amount of data that will be sent, the non-stationary coefficients are represented by using K Hermite coefficients, where N/K=1.4. Determine how many bits are required for encoding the considered sequence and what is the compression factor? The original video frames can be coded by using on average 256 bits per block.

Blocks statistics	
Total number of frames	**1200**
Frame size	**300x450**
Stationary blocks	**40%**
Non-stationary blocks	**60%**

Examples

Solution:

The stationary blocks are transmitted only for the first frame. Thus, the total number of bits used to represent the coefficients from the stationary blocks is:

$$n_s = \frac{40}{100}\left(\frac{300\cdot 450}{64}\cdot 256b/block\right) = 216\cdot 10^3 b$$

In the case of non-stationary blocks, we observe the sequences of coefficients which are on the same position within different video frames. Hence, each sequence having N=1200 coefficients, is represented by using K Hermite coefficients, where N/K=1.4 holds. The total number of bits used to encode the coefficients from the non-stationary blocks is:

$$n_n = 1200\cdot\frac{K}{N}\cdot\left(\frac{60}{100}\cdot\left(\frac{300\cdot 450}{64}\cdot 256b/block\right)\right) = 2.77\cdot 10^8 b$$

$$p = 1200\cdot 300\cdot 450\cdot 4 = 6.4\cdot 10^8 b$$

- number of bits required for coding the original sequence

$$\frac{6.4\cdot 10^8}{216\cdot 10^3 + 2.77\cdot 10^8} = 2.33$$

compression factor

Examples

7. A part of the video sequence contains 126 frames in JPEG format (Motion JPEG - MJPEG format) and its total size is 1.38MB. The frame resolution is 384x288, while an average number of bits per 8x8 block is B=51.2. Starting from the original sequence, the DCT blocks are classified into stationary S and non-stationary NS blocks. The number of the blocks are No{S}=1142 and No{NS}=286. The coefficients from the S blocks are almost constant over time and can be reconstructed from the first frame. The coefficients from NS blocks are represented by using the Hermite coefficients. Namely, the each sequence of 126 coefficients is represented by 70 Hermite coefficients. Calculate the compression ratio between the algorithm based on the blocks classification and Hermite expansion, and the MJPEG algorithm.

Examples

A set of 126 frames in the JPEG format requires $No\{S\}\cdot B\cdot 126$ bits for stationary and $No\{NS\}\cdot B\cdot 126$ bits for non-stationary blocks

In other words, the total number of bits for the original sequence in the MJPEG format is:

$$No\{S\}\cdot B\cdot 126+No\{NS\}\cdot B\cdot 126=$$
$$(1142+286)\cdot 51.2\cdot 126=9.21\cdot 10^6 b$$

The algorithm based on the blocks classification will encode the stationary blocks from the first frame only:

$$No\{S\}\cdot B$$

For non-stationary blocks, instead of 126 coefficients over time, it uses 70 Hermite coefficients, with the required number of bits equal to:

$$No\{NS\}\cdot N\cdot B$$

total number of bits for stationary and non-stationary blocks:

$$No\{S\}\cdot B+No\{NS\}\cdot N\cdot B=1142\cdot 51.2+286\cdot 70\cdot 51.2=1.083\cdot 10^6 b$$