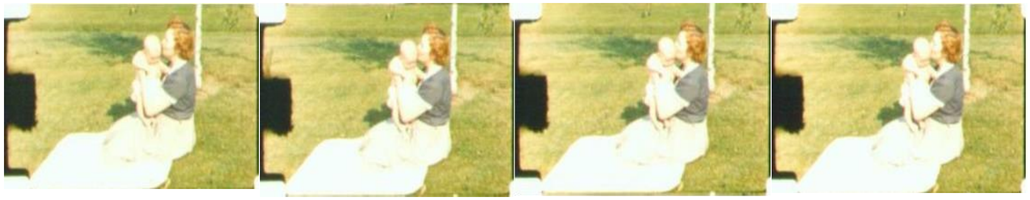
VIDEO COMPRESSION BASICS

1. OVERVIEW

- A video comprises of a sequence of frames.

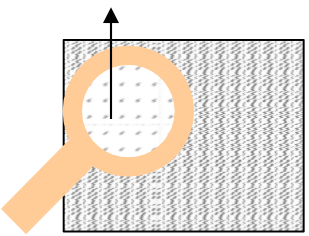
Frame 1 Frame 2 Frame 3 Frame 4



- A video, *of the duration of 1 second*, generated by a TV camera usually contains 24 frames or 30 frames.

- Each pixel in a frame is represented by three attributes *(each 8 bits long)* – **One luminance** attribute and **two chrominance** attributes. (*i.e.* ***YCbCr***)

{ , Cb, Cr}



Frame

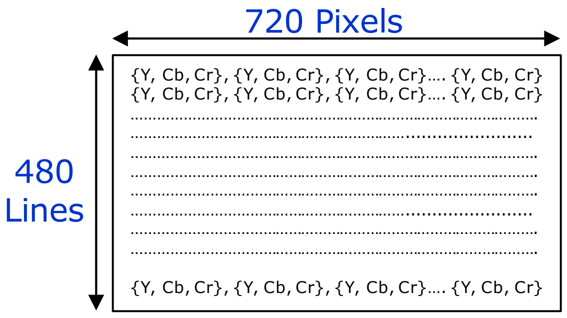
Luminance (Y) : Describes the brightness of the pixel.

Chrominance (CbCr) : Describes the color of the pixel.

- An *uncompressed* video data is big in size.

For example-

A single frame having the resolution of 720 X 480 *(no. of pixels in each horizontal line is 720 and total no. of horizontal lines per frame is 480)* will be described by



Frame (720 X 480)

720 X 480 X 8 + 720 X 480 X 8 + 720 X 480 X 8 bits

= *8294400 bits.* ~ **8.29 Mb.**

Or

A complete video of 1 second will be described by

(720 X 480 X 8 + 720 X 480 X 8 + 720 X 480 X 8) X 24 bits

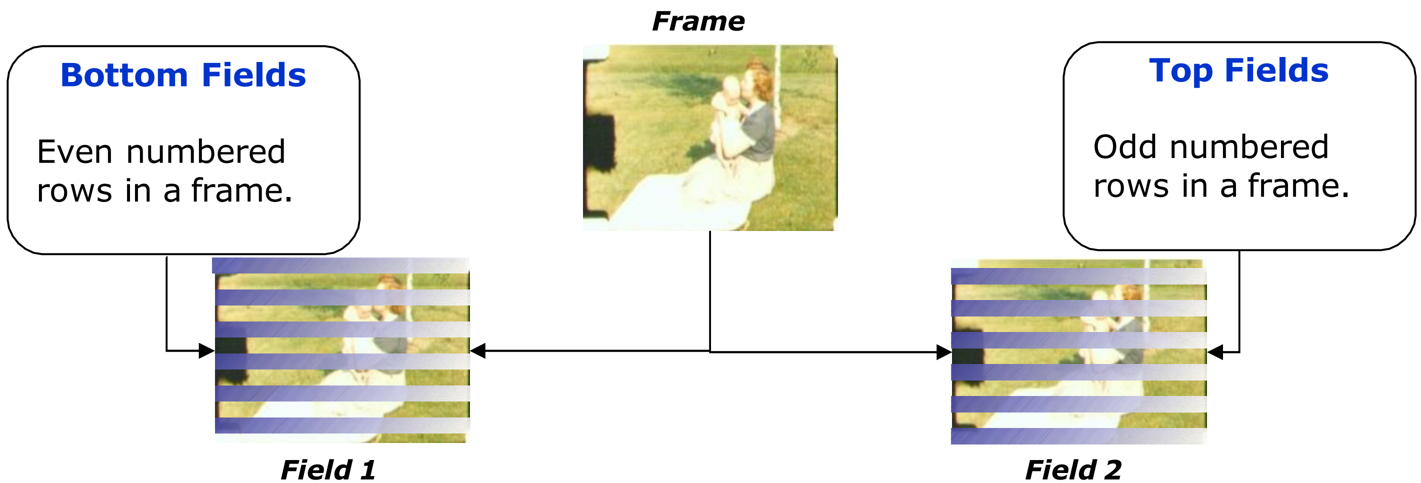
= *199065600 bits* ~ **199 Mb.**

Thus, for the entire movie, the data would be too big to fit on DVDs or to transmit it using the bandwidth of available TV channels.

2. VIDEO SCHEME

Types of video schemes used for transmission -

- **Interlaced Video Scanning**
  - In this, a frame is divided into two separate fields – Top Fields and Bottom fields.



  - The two successive fields *(field 1 & field 2)* are called a frame.
  - Both the fields are sent one after another and display puts them back together before displaying the full frame.
  - Quality degraded as sometimes the frames come out of sync.
  - It conserve the bandwidth.
  - Maximum frame rate is 60 frames/ second.

Progressive Video Scanning

- In this, complete frame is send to display.
- Bandwidth requirement is twice as compared to Interlaced video scanning.
- Quality is good as frames come in sync and image is much sharper.
- Maximum frame rate is 30 frames/ second.

Frame

3. VIDEO COMPRESSION

The concept of video compression lies on two main factors-

- The data in frames is often redundant in space and time.

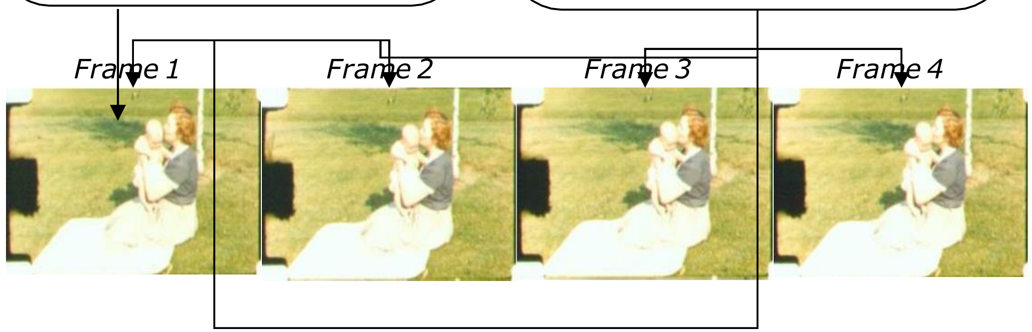
  For example-

Spatial redundancy

In a frame, adjacent pixels are usually correlated. *e.g. - The grass is green in the background of a frame.*

Time based redundancy

In a video, adjacent frames are usually correlated. *e.g. - The green background is persisting frame after frame.*



- The human eye better resolve the brightness details than color details. So the way human eye works, it is also possible to delete some data from the frame with almost no noticeable degradation in image quality.

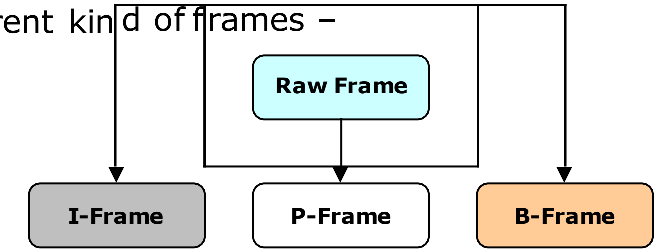
4. MPEG

MPEG stands for *Motion Picture Experts Group* established in 1988 as a working group within ISO/IEC that has defined standards for digital compression of audio & video signals. *Such as-*

- **MPEG-1 :** It was the very first project of this group and published in 1993 as ISO/IEC 11172 standard.
  - MPEG-1 defines coding methods to compress the progressively scanned video.
  - Commonly used in CD-i and Video CD systems.
  - It supports coding bit rate of 1.5 Mbit/s.
- **MPEG-2 :** is an extension of MPEG-1, published in 1995 as ISO/IEC 13818 standard.
  - MPEG-2 defines coding methods to compress progressively scanned video as well as interlaced scanned video.
  - Commonly used in broadcast format, such as – Standard Definition TV (SDT) and High Definition TV (HDT).
  - It supports coding bit rate of 3 - 15 Mbit/s for SDT and 15 – 20 Mbit/s for HDT.
- **MPEG-4 :** introduced in 1998 and still in development as ISO/IEC 14496 standard.
  - MPEG-2 defines object based coding methods for mixed media data and provides new features, such as – 3D rendering, animation graphics, DRM, various types of interactivity etc.
  - Commonly used in web based streaming media, CD, videophone, DVB, etc .
  - It supports coding bit rate of few Kbit/s – tens of Mbit/s.

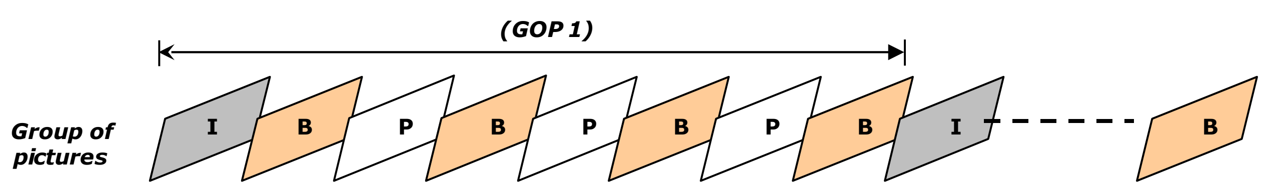
5. MPEG-2 VIDEO COMPRESSION

- MPEG-2 compresses a raw frame into three different kind of frames –
  - Intra coded frames *(I-frames)*,
  - Predictive coded frames *(P-frames)*, and
  - Bi-directionally predictive coded frames *(B- frames)*.



- Compression is based on -
  - Spatial redundancy
  - Time based redundancy

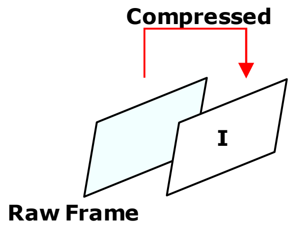
- Compressed frames (I, P & B frames) are organized in a sequence to form *Group of Pictures* (GOP).



6. MPEG-2 FRAMES

❑ I - Frame

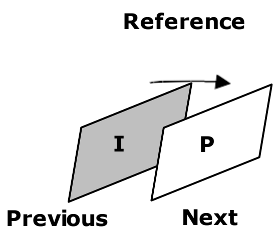
- Compressed directly from of a raw (uncompressed) frame.
- Compression is based on spatial redundancy in the current raw frame only and inability of human eye to detect certain changes in the image.

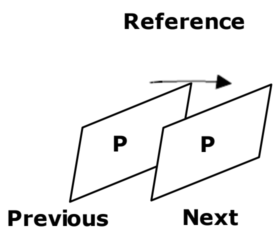


- *I-frame is a reference frame* and can be used to predict the P-frame immediately following it.

❑ P - Frame

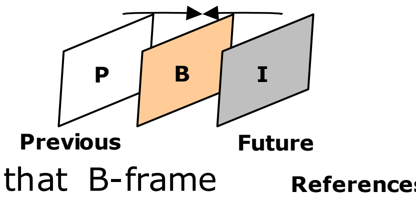
- Compression is based on spatial redundancy as well as on time based redundancy.
- P-frame can be predicted by referring I-frame or P-frame immediately preceding it. *(P-frame is also a reference frame)*.

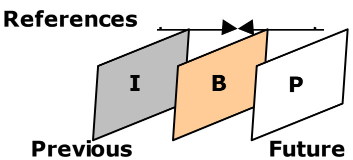




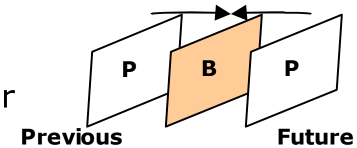
- P-frame provides better compression than I-frame as it uses the data of previous I-frame or P-frame.

❑ B - Frame



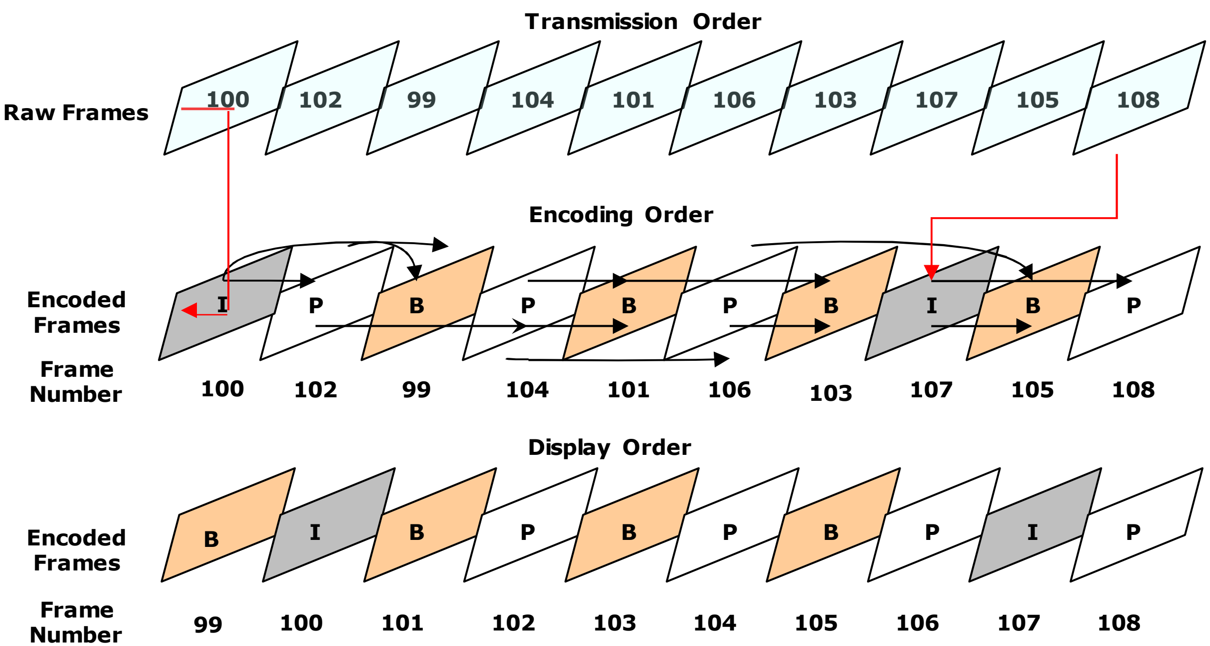


- Compression is similar to P-frame except that B-frame compression is done by referring previous as well as following I-frame and/ or P frame.



- B-frame required frame sequence must be transmitted or stored (I or P frames) out of order, so that future frame is available for reference.

  It causes some delay through out the system.



- B-frame provides better compression than P-frame & I-frame, as it uses the data of previous as well as succeeding I- frame and/ or P-frame.

  It requires a memory buffer of double in size to store data of two reference/ anchor frames

- B-frame is not a *reference* frame.

- *There is no defined limit to the number of consecutive B-frames within a group of*

7. MPEG-2 VIDEO ENCODING

MPEG-2 video encoding can be broadly categorized into – *Intra Frame Encoding and Non-Intra Frame Encoding.*

Intra Frame Encoding

It takes the advantage of spatial redundancy.

Compression techniques are applied using the data of the current frame only.

It uses combination of various lossless and lossy compression techniques. Such as –

- **Video filter**

  Compress spatial redundancy at chrominance plane.

- **Discrete cosine transform (DCT)**

  Convert spatial variation into frequency variation.

- **DCT coefficient quantization**

  Reduces higher frequency DCT coefficients to zero .

- **Run-Length amplitude/ Variable length encoding**

  Compression using entropy encoding, run-length encoding & Huffman encoding .

- **Bit rate control**

Non-Intra Frame Encoding

It takes the advantage of time-based redundancy as well as spatial redundancy.

Compression techniques are applied using the data of the current frame as well as preceding and/ or succeeding frames.

It mainly uses lossy compression techniques. Such as –

- **Forward interpolated prediction**

  Uses data of previously coded frame.

- **Forward & backward interpolated prediction**

  Uses data of previously coded frame & future frame.

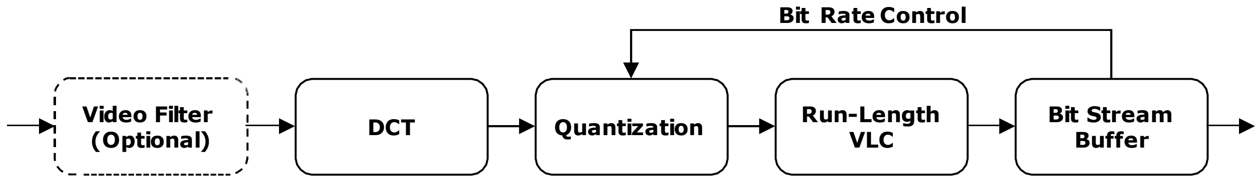
- **Temporal prediction**

  Uses motion estimation & vectors to generate predicted frame.

- **Residual error frame & its coding**

  It is generated by subtracting predicted frame from its reference frame, which is further spatially coded & transmitted.

7.1 INTRA FRAME ENCODING



Intra Frame Encoder

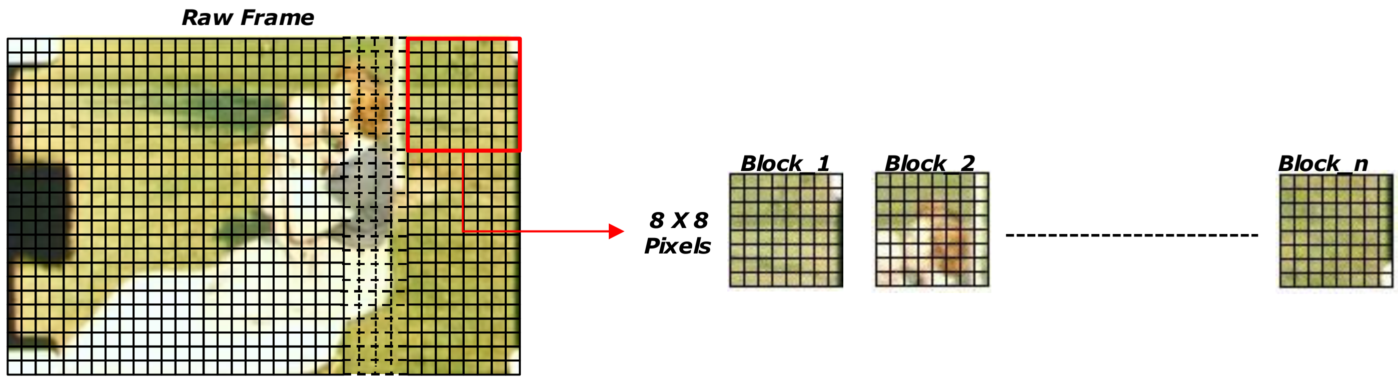
❑ Video Filtering

Video filtering is a lossy compression technique and is used to compress the spatial redundancies on *macro-block* basis within the current frame.

It operates on color space *(i.e. YCbCr encoding & CbCr sub-sampling)*, as Human Visual System is less sensitive to variations in color as compared to variations in brightness.

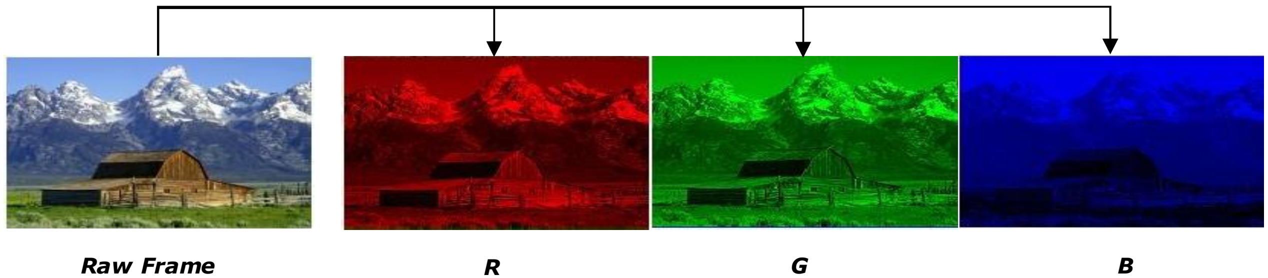
Video filtering includes-

- **Macro-block :** Macro-blocks are created by dividing raw frame into 8 x 8 pixels blocks. *For example-*



- **YCbCr Encoding:** converts block's RGB data into YCbCr color space. *For example-*

A raw frame contains an image in RGB color space.



RGB color space contains mutual redundancies, so it requires large space for storage and high bandwidth for transmission.

Encoding RGB into YCbCr color space reduces the mutual redundancies

Conversion: Y = + 0.299 * R + 0.587 * G + 0.114 * B

Cb = 128 - 0.168736 * R - 0.331264 * G + 0.5 * B

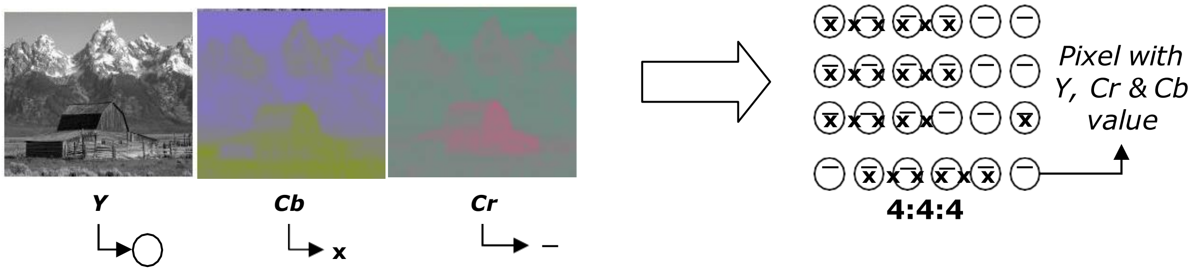
Cr = 128 + 0.5 * R - 0.418688 * G - 0.081312 * B

Where-

R, G & B values are 8 bits long and lies in {0, 1, 2, ..., 255} range.

- **Chrominance (CbCr) Sub-Sampling:** provides further compression at chrominance plane to reduced number of bits to represent an image. *For example-*

A YCbCr encoded frame can be represented as- *4:4:4 Sampling Format*



4:4:4 sampling format states that, for every four Y samples – there is four Cb & four Cr samples.

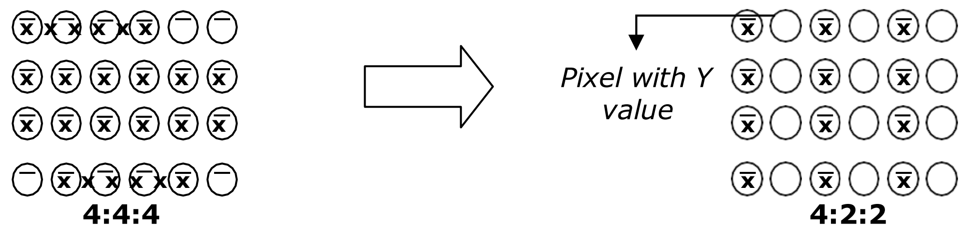
If an image resolution is 640 x 480 pixels then, the number. of

Y samples = 640 x 480, Cr samples = 640 x 480 & Cb samples = 640 X 480

Number of bits required = <u>640 x 480 x 8</u> + <u>640 x 480 x 8</u> + <u>640 x 480 x 8</u>

= 7372800 bits ~ 7.3728 Mb

4:4:4 sampling format can be sub–sampled to 4:2:2 format, where Cr & Cb are sub-sampled to half the horizontal resolution of Y.



That is, in 4:2:0 sampling format - for every four 4 Y samples in horizontal direction, there would be 2 Cb & 2 Cr samples

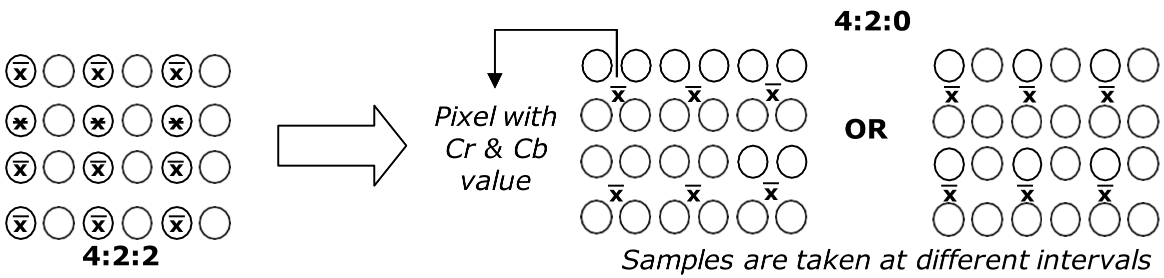
If an image resolution is 640 x 480 pixels then, the number. of

Y samples = 640 x 480, *Cr samples* = 320 x 480 & *Cb samples* = 320 X 480

Number of bits required = $\underline{640 \times 480 \times 8} + \underline{320 \times 480 \times 8} + \underline{320 \times 480 \times 8}$

= 4915200 bits ~ 4.9152 Mb

It can be further sub–sampled to 4:2:0 format, where Cb & Cr are sub-sampled to half the horizontal and vertical resolution of Y.



If an image resolution is 640 x 480 pixels then, the number. of

Y samples = 640 x 480, *Cr samples* = 320 x 240 & *Cb samples* = 320 X 240

Number of bits required = 640 x 480 x 8 + 320 x 240 x 8 + 320 x 240 x 8

= 3686400 bits ~ **3.6864 Mb**
(which is far less than 4:4:4 and 4:2:2 format)

Discrete Cosine Transformation

DCT converts the spatial variations within the macro-block into frequency variations without changing the data.

A two-dimension DCT function can be represented as:

$$F(u,v) = \frac{2}{N} C(u) C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \cos\frac{(2x+1)u\pi}{2N} \cos\frac{(2y+1)v\pi}{2N}$$

Where-

u *Horizontal spatial frequency, for the integers* $0 \le u < N$

v *Vertical spatial frequency, for the integers* $0 \le u < N$

$$C(u),\quad C(v) = \begin{cases} \frac{1}{\sqrt{2}} \text{ for } u,v = 0 \\ 1 \text{ otherwise} \end{cases}$$ *a normalizing function*

$f(x,y)$ *Pixel value at coordinates* (x, y)

$F(u,v)$ *DCT coefficient at coordinates* (u, v)

The output of a DCT function is a DCT coefficient matrix containing the data in frequency domain.

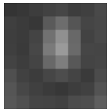
Data in frequency domain can be efficiently processed and compressed.

In DCT, 8x8 data block of Y, Cb and Cr components is converted into frequency domain. *For example-*

- Lets assume, 8x8 pixel block of Y component and its corresponding 8x8 data block

Block_1

8 X 8 Pixels



Y

8 X 8 Data block

52	55	61	66	70	61	64	73
63	59	55	90	109	85	69	72
62	59	68	113	144	104	66	73
63	58	71	122	154	106	70	69
67	61	68	104	126	88	68	70
79	65	60	70	77	68	58	75
85	71	64	59	55	61	65	83
87	79	69	68	65	76	78	94

- Subtract 128 from each pixel value.

8 X 8 Data block

-76	-73	-67	-62	-58	-67	-64	-55
-65	-69	-73	-38	-19	-43	-59	-56
-66	-69	-60	-15	16	-24	-62	-55
-65	-70	-57	-6	26	-22	-58	-59
-61	-67	-60	-24	-2	-40	-60	-58
-49	-63	-68	-58	-51	-60	-70	-53
-43	-57	-64	-69	-73	-67	-63	-45
-41	-49	-59	-60	-63	-52	-50	-34

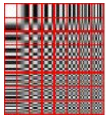
DCT

8 X 8 DCT Coefficient Matrix

-415	-30	-61	27	56	-20	-2	-0
4	-22	-61	10	13	-7	-9	5
-47	7	77	-25	-29	10	5	-6
-49	12	34	-15	-10	6	2	2
12	-7	-13	-4	-2	2	-3	3
-8	3	2	-6	-2	1	4	2
-1	0	0	-2	-1	-3	4	-1
0	0	-1	-4	-1	0	1	2

DCT

8 X 8 Pixels



Y ~

- Apply two-dimension DCT to get DCT coefficient matrix.

Quantization

Quantization reduces the amount of information in higher frequency DCT coefficient components using a default *quantization matrix* defined by MPEG-2 standard.

16	11	19	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Default Quantization Matrix

Default quantization matrix contains constant values.

It is a lossy operation that causes minor degradation in the image quality due to some subtle loss in brightness and colors.

Each component in DCT coefficient matrix is divided by its corresponding constant value in default quantization matrix and a quantized DCT coefficient matrix is computed.

A quantization function can be represented as:

$$B_{j,k} = \text{round}\left(\frac{G_{j,k}}{Q_{j,k}}\right) \quad \textit{Where-} \quad \begin{array}{l} j = 0, 1, 2, \cdots, N_1 - 1 \\ k = 0, 1, 2, \cdots, N_2 - 1 \end{array}$$

For example-

-415	-30	-61	27	56	-20	-2	-0
4	-22	-61	10	13	-7	-9	5
-47	7	77	-25	-29	10	5	-6
-49	12	34	-15	-10	6	2	2
12	-7	-13	-4	-2	2	-3	3
-8	3	2	-6	-2	1	4	2
-1	0	0	-2	-1	-3	4	-1
0	0	-1	-4	-1	0	1	2

8 X 8 DCT Coefficient Matrix

16	11	19	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Default Quantization Matrix

Quantization

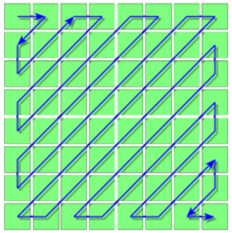
-26	-3	-6	2	2	-1	0	0
0	-2	-4	1	1	0	0	0
-3	1	5	-1	-1	0	0	0
-4	1	2	-1	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Quantized 8 X 8 DCT Coefficient Matrix

❑ Run Length Amplitude/ Variable Length Encoding

Run length amplitude/ Variable length encoding is a lossless compression techniques that includes *entropy encoding, run-length encoding and Huffman encoding*.

Entropy Encoding



- **Entropy Encoding :** Components of quantized DCT coefficient matrix is read in zigzag order.

  It helps in representing the frequency coefficients (both higher & lower) of quantized DCT coefficient matrix in an efficient manner. *For example-*

-26	-3	-6	2	2	-1	0	0
0	-2	-4	1	1	0	0	0
-3	1	5	-1	-1	0	0	0
-4	1	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

Quantized 8 X 8 DCT Coefficient Matrix

Entropy Encoding

-26							
-3	0						
-3	-2	-6					
2	-4	1	-4				
1	1	5	1	2			
-1	1	-1	2	0	0		
0	0	0	-1	-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	0				
0	0	0					
0	0						
0							

~

Zigzag ordering of quantized 8 X 8 DCT Coefficient Matrix

0 **15**

-26	-3	0	-3	-2	-6	2	-4	1	-4	1	1	5	1	2	-1
1	-1	2	0	0	0	0	0	-1	-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	0	0	0	0	0	0	0

48 **63**

- **Run Length Encoding :** is a lossless data compression technique of a sequence in which same data value occurs in many consecutive data elements.

WWWWWWWWWWWWBWWWWWWWWWWWWBBB

(Sequence)

Here, the runs of data are stored as single data value and count. That is, the above sequence can be represented as *12W1B12W3B.*

For example-

Zigzag ordering of quantized 8 X 8 DCT Coefficient Matrix

0 … **15** (top row); **48** … **63** (bottom row)

-26	-3	0	-3	-2	-6	2	-4	1	-4	1	1	5	1	2	-1
1	-1	2	0	0	0	0	0	-1	-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	0	0	0	0	0	0	0

Run Length Encoding

-26, -3, 0, -3, -2, -6, 2, -4, 1, -4, (two 1s), 5, 1, 2, -1, 1, -1, 2, (5 0s), (two -1s), (thirty eights zeroes)

Run Length Encoded DCT coefficients

- **Huffman Encoding :** is a lossless data compression technique that uses *variable length code table* for encoding a source symbol.

Where, Variable length code table is derived based on the estimated probability of occurrence of each possible value of the source symbol.

Variable length code table

char	Freq	Code
space	7	111
a	4	010
e	4	000
f	3	1101
h	2	1010

For example- MPEG-2 has special Huffman code word (i.e. EOB) to end the sequence prematurely when the remaining coefficients are zero and then it performs variable length encoding for further compression.

-26, -3, 0, -3, -2, -6, 2, -4, 1, -4, (two 1s), 5, 1, 2, -1, 1, -1, 2, (5 0s), (two -1s), (thirty eights zeroes)

Run Length Encoded DCT coefficients

↓

-26, -3, 0, -3, -2, -6, 2, -4, 1, -4, (two 1s), 5, 1, 2, -1, 1, -1, 2, (5 0s), (two -1s), *EOB*

DCT coefficients sequence with EOB

↓

Variable Length Encoding

↓

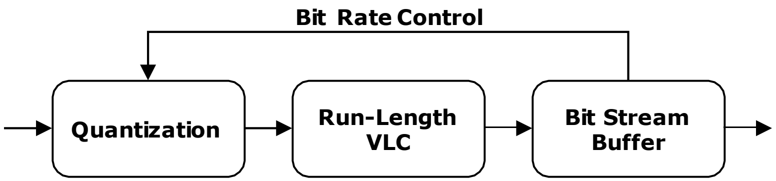
0011101010011011010110001010

Compressed Bit Stream Sequence

❑ Bit Rate Control

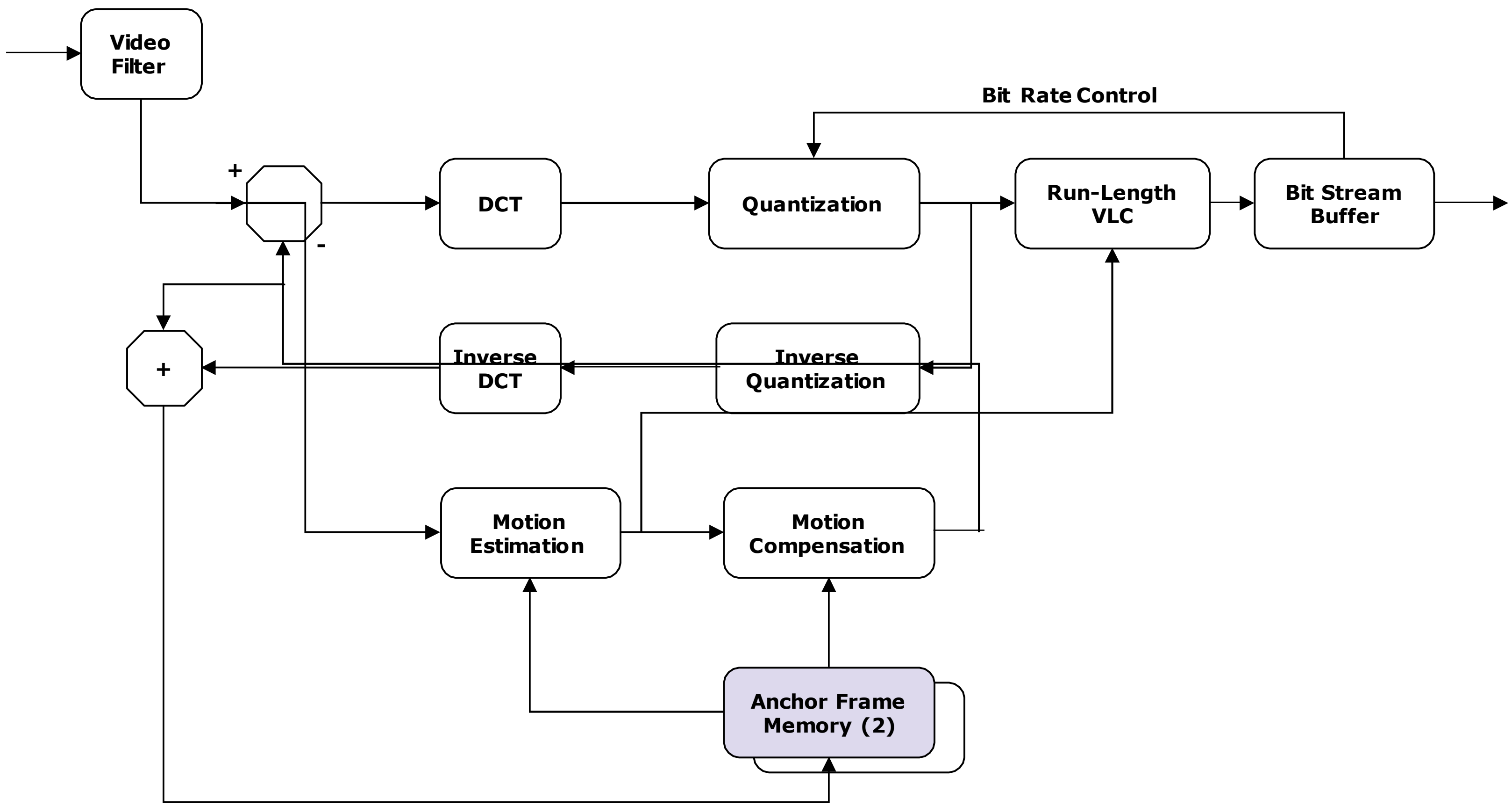
Bit rate control is a mechanism to prevent the underflow or overflow of buffer used for temporarily storage of encoded bit stream within the encoder.

Bit rate control is necessary in applications that requires fixed bit rate transmission of encoded bit-stream. *For example-*



- o Quantization process may affects relative buffer fullness which in turn affects the output bit rate, as quantization depends on default quantization matrix (picture basis) and quantization scale (macro-block basis).
- o Encoder has to pass these two parameters to bit rate control mechanism in order to control relative buffer fullness and constant bit rate.
- o Buffer under flow/ over flow can be prevented by repeating or dropping of entire video frames.

7.2 NON-INTRA FRAME ENCODING

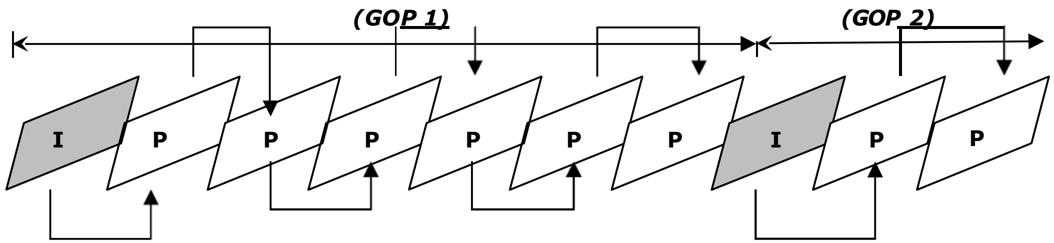


Non – Intra Frame Encoder

❑ Forward Interpolated Prediction

Using *forward interpolated prediction*, encoder can forward predict a future frame called P-frame.

The very first P-frame in a *group of picture*, is predicted from the I-frame immediately preceding it.

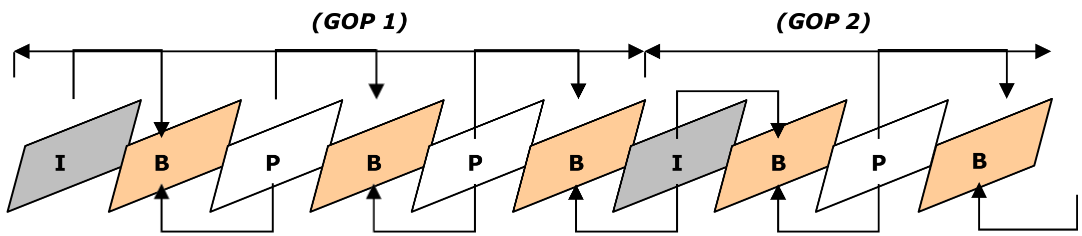


Other P-frame in a *group of picture* can be predicted from previous I-frame or P-frame immediately preceding it.

❑ Forward & Backward Interpolated Prediction

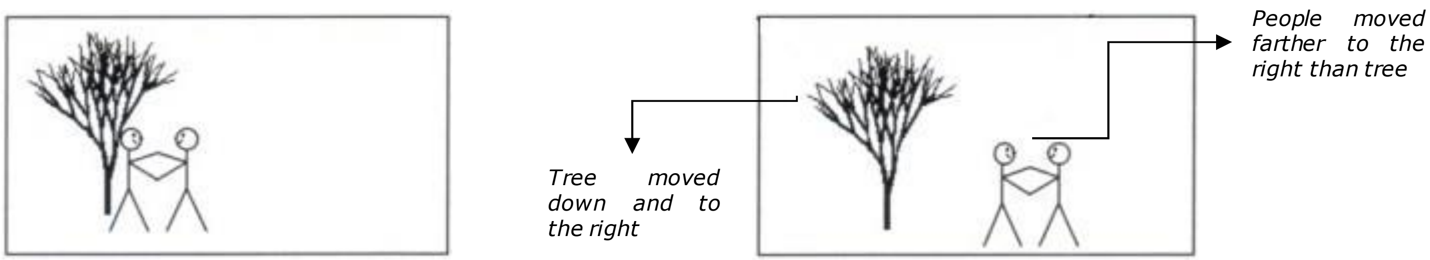
Using *forward & backward interpolated prediction*, encoder can forward predict a future frame called B-frame.

The B-frame in a *group of picture*, is predicted based on a forward prediction from a previous I or P as well as backward prediction from a succeeding I or P frame.



❑ Temporal Prediction

Mostly, consecutive video frames are similar except for the differences induced by the objects moving within the frames. *For example-*



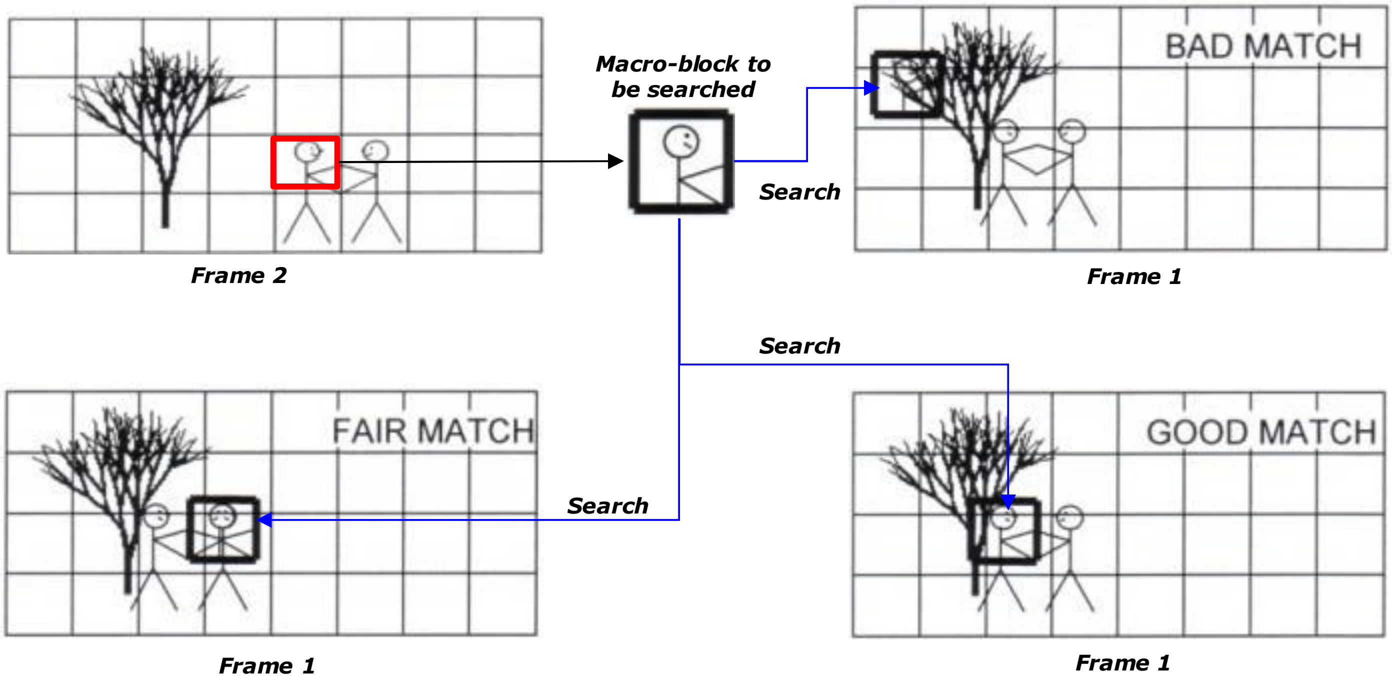
Frame 1 (Current)

Frame 2 (Next)

Temporal prediction uses ***motion estimation & motion vector techniques*** to predict these changes in the future frames.

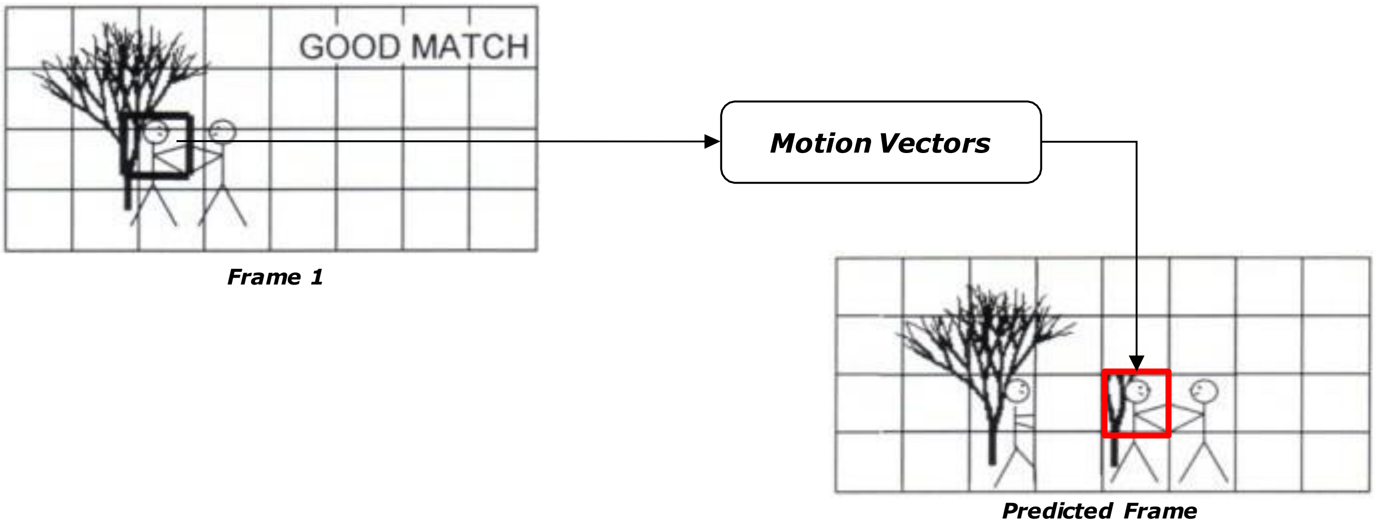
Motion estimation is applied at luminance plane only. *(It is not applied at chrominance plane as it is assumed that the color motion can be adequately presented with the same motion information as the luminance.*)

- **Motion Estimation :** It performs a 2-Dimesional spatial search for each luminance macro-blocks within the frame to get the best match. *For example-*



If there is no acceptable match, then encoder shall code that particular macro-block as an intra macro-block, even though it may be in a P or B frame.

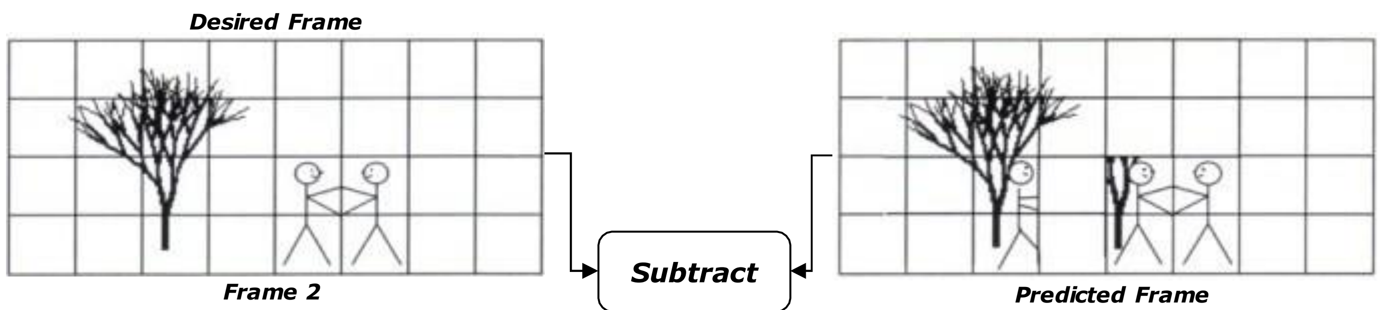
- **Motion Vector :** Assign motion vectors to the resultant macro-blocks to indicate how far the horizontally and vertically the macro-block must be moved so that a predicted frame can be generated *For example-*

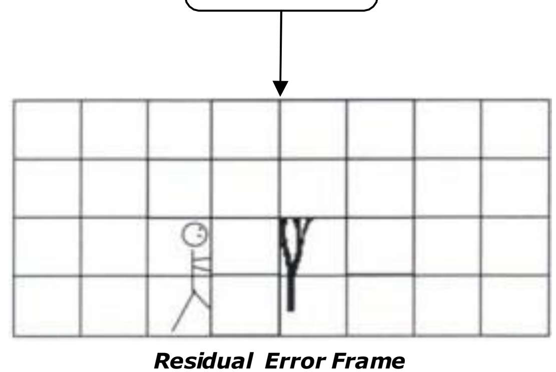


Since, each forward and backward macro-block contains 2 motion vectors, so a bi-directionally predicted macro-block will contain 4 motion vectors.

❑ Residual Error Frame

Residual error Frame is generated by subtracting the predicted frame from desired frame. *For example-*





- Residual error Frame is less complicated and can be encoded efficiently.
- More accurate the motion is estimated & matched, more likely the residual error will approach zero and the coding efficiency shall be highest.
- Since, motion vector tends to be highly correlated between macro-blocks – Horizontal & vertical component is compared to the previously valid horizontal & vertical motion vector respectively and difference is calculated.
- These differences *(i.e. Residual Error Frame)* are then coded and variable length code is applied on it for maximum compression efficiency.

- **Residual Error Frame Coding :** Coding of residual error frame is similar to I-frame with some differences. *Such as-*

    - Default quantization matrix for non-intra frame is flat matrix with constant value of 16 for each of the 64 locations.

    - Non-intra frame quantization contains a dead-zone around zero which helps in eliminating any lone DCT coefficient quantization values that might reduce the run-length amplitude efficiency.

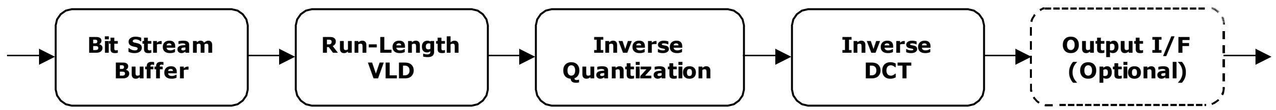
    - Motion vectors for the residual block information are calculated as differential values and coded with a variable length code according to their statistical likelihood of occurrence.

8. MPEG-2 VIDEO DECODING

MPEG-2 video decoding can be broadly categorized into – *Intra Frame Decoding and Non-Intra Frame Decoding.*

❑ Intra Frame Decoding

Intra frame decoding reverse the order of intra frame encoding process . *For example-*



Intra Frame Decoder

- **Buffer :** Contains input bit-stream. For fixed rate applications, constant bit-stream is buffered in the memory and read out at variable rate based on the coding efficiency of the macro-blocks and frames to be decoded.

- **VLD :** Reverse the order of run length amplitude/ variable length encoding done in encoding process and recover the quantized DCT coefficient matrix.
  - It is a most complex and computationally expensive portion in decoding.
  - Perform bitwise decoding of input bit-stream using table look-ups to generate quantized DCT coefficients matrix.

- **Inverse Quantization :** Reverse the order of quantization done in encoding process and recover the DCT coefficient matrix.
  - Components of decoded quantized DCT coefficient matrix is multiplied by the corresponding value of the default quantization matrix and the quantization scale factor.
  - Resulting coefficient is clipped to the region -2048 to +2047.
  - Perform IDCT mismatch control to prevent long term error propagation within the sequence.
- **Inverse DCT :** Reverse the order of DCT done in encoding process and recover the original frame.
  - A two-dimension DCT function can be represented as:

$$f(x,y) = \frac{2}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} C(u)C(v)F(u,v)\cos\frac{(2x+1)u\pi}{2N}\cos\frac{(2y+1)v\pi}{2N}$$

Where-

$$C(u),\quad C(v) = \begin{cases} \frac{1}{\sqrt{2}} \text{ for } u,v = 0 \\ 1 \text{ otherwise} \end{cases}$$

a normalizing function

$f(x,y)$ *Pixel value at coordinates* (x, y)

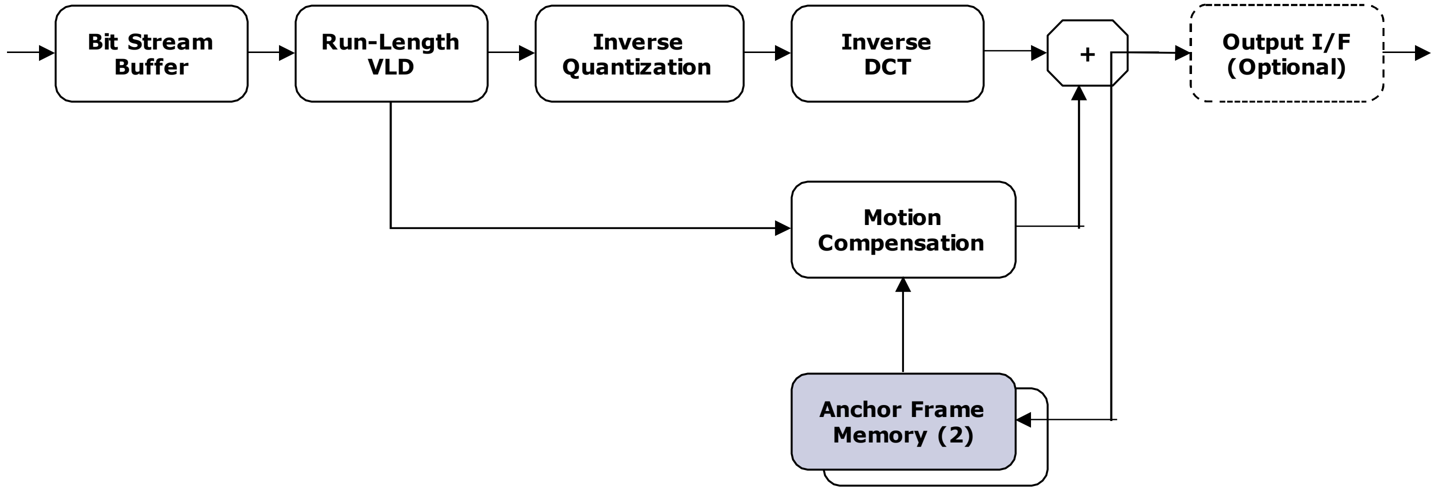
$F(u,v)$ *DCT coefficient at coordinates* (u, v)

u *Horizontal spatial frequency, for the integers* $0 \leq u < N$

v *Vertical spatial frequency, for the integers* $0 \leq u < N$

❑ Non-Intra Frame Decoding

Non-Intra frame decoding is similar to intra frame decoding with the addition of motion compensation support. *For example-*



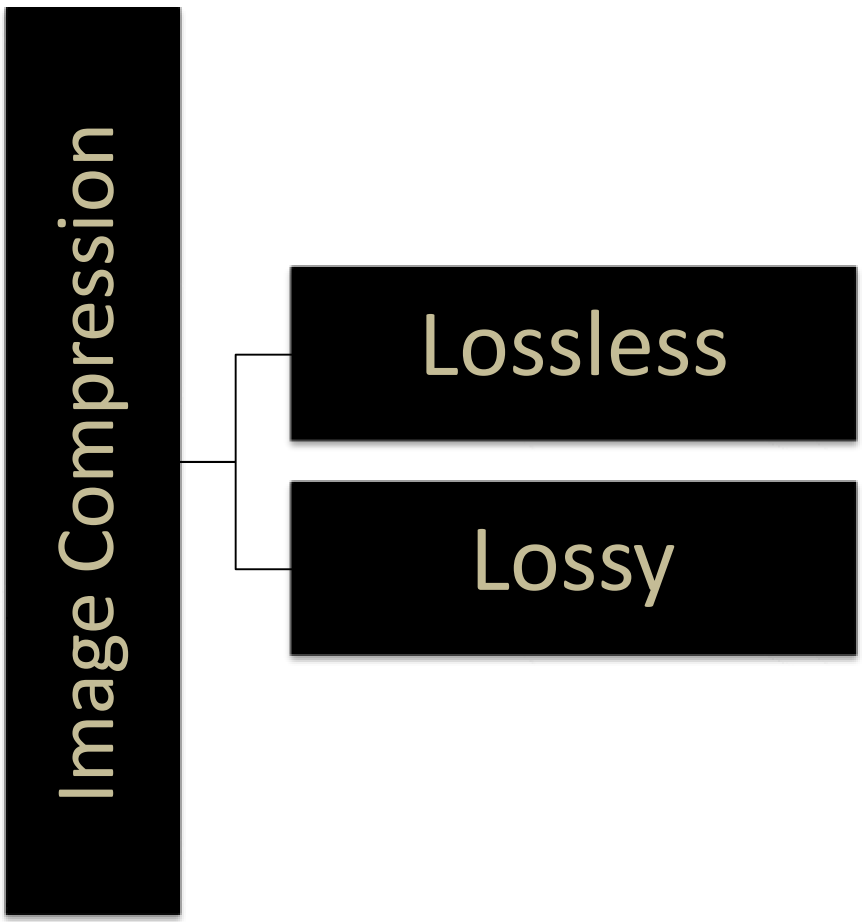
Non-Intra Frame Decoder

JPEG Image Compression

What is Image Compression?

The objective of image compression is to reduce irrelevant and redundant image data in order to be able to store or transmit data in an efficient form.

Types



- **Lossless** image compression is a compression algorithm that allows the original image to be perfectly reconstructed from the original data.
- **Lossy** image compression is a type of compression where a certain amount of information is discarded which means that some data are lost and hence the image cannot be decompressed with 100% originality.

Overview

- JPEG, which stands for Joint Photographic Experts Group (the name of the committee that created the JPEG standard) is a lossy compression algorithm for images.
- A lossy compression scheme is a way to inexactly represent the data in the image, such that less memory is used yet the data appears to be very similar. This is why JPEG images look almost the same as the original images they were derived from most of the time, unless the quality is reduced significantly, in which case there will be visible differences.
- The JPEG algorithm takes advantage of the fact that humans can't see colours at high frequencies. These high frequencies are the data points in the image that are eliminated during the compression. JPEG compression also works best on images with smooth colour transitions.

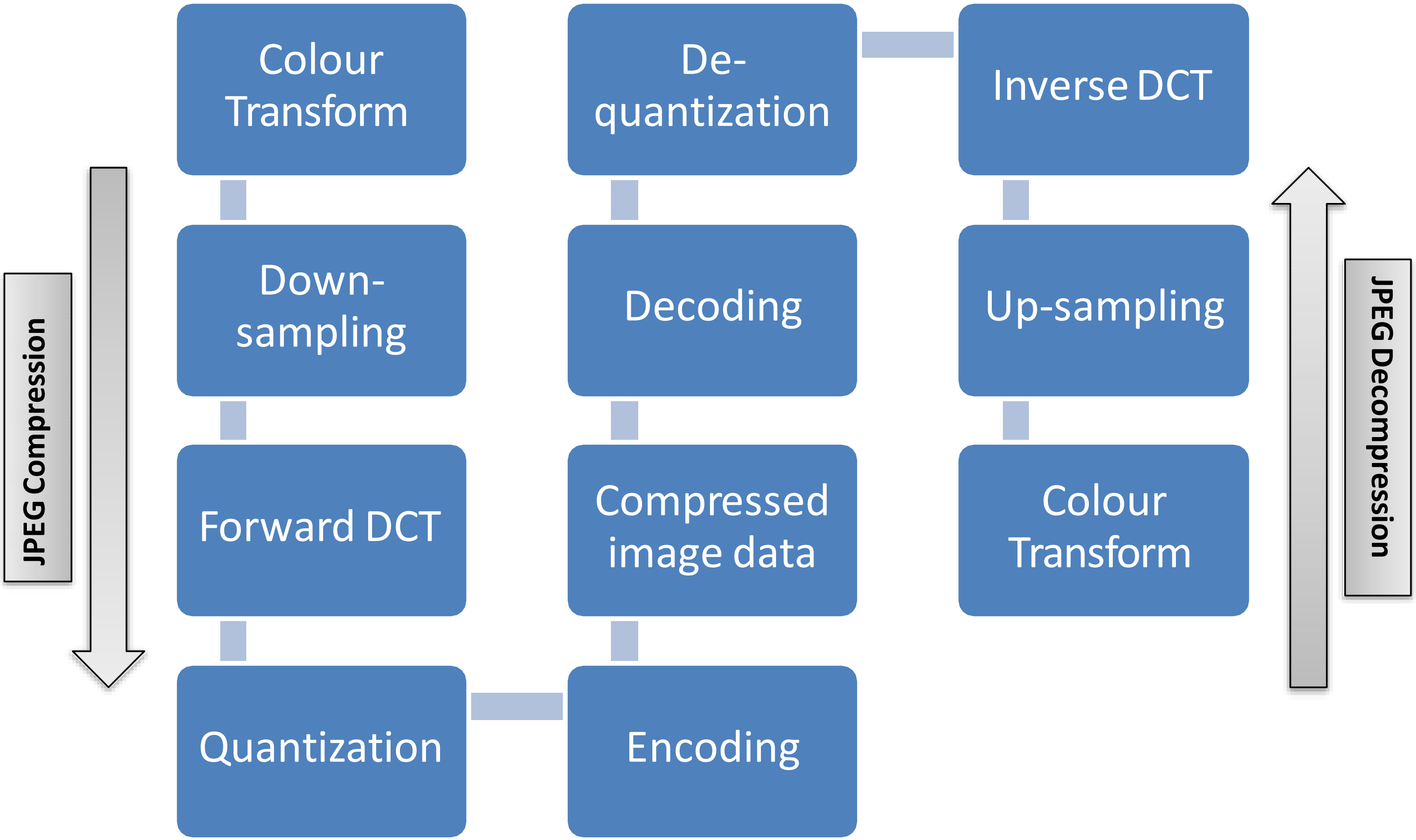
What is JPEG compression?

JPEG is a commonly used method of lossy compression for digital images. The degree of compression can be adjusted, allowing a tradeoff between storage size and image quality with a compression ratio 10:1; but with little perceptible loss in image quality.

Why JPEG?

- JPEG uses ***transform coding***, it is largely based on the following observations:
  - A large majority of useful image contents change relatively slowly across images, i.e., it is unusual for intensity values to alter up and down several times in a small area, for example, within an 8 x 8 image block. A translation of this fact into the spatial frequency domain, implies, generally, lower spatial frequency components contain **more information** than the high frequency components which often correspond to less useful details and noises.
  - Experiments suggest that humans are more immune to loss of higher spatial frequency components than loss of low frequency components. Human vision is insensitive to high frequency components.

JPEG Schematic



Algorithm

- **Splitting**: Split the image into 8 x 8 non-overlapping pixel blocks. If the image cannot be divided into 8-by-8 blocks, then you can add in empty pixels around the edges, essentially zero-padding the image.
- **Colour Space Transform** from [R,G,B] to [Y,Cb,Cr] & Subsampling.
- **DCT**: Take the Discrete Cosine Transform (DCT) of each 8-by-8 block.
- **Quantization**: quantize the DCT coefficients according to psycho-visually tuned quantization tables.
- **Serialization**: by zigzag scanning pattern to exploit redundancy.
- **Vectoring**: Differential Pulse Code Modulation (**DPCM**) on DC components
- Run Length Encoding (**RLE**) on AC components
- **Entropy Coding**:
  - Run Length Coding
  - Huffman Coding or Arithmetic Coding

Step I - Splitting

The input image is divided into smaller blocks having 8 x 8 dimensions, summing up to 64 units in total. Each of these units is called a ***pixel***, which is the smallest unit of any image.

Original image

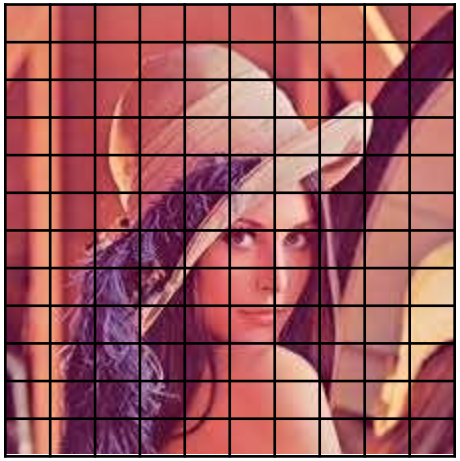


Image split into multiple 8x8 pixel blocks

Step II - RGB to YCbCr conversion

- JPEG makes use of [Y,Cb,Cr] model instead of [R,G,B] model.
- The precision of colors suffer less (for a human eye) than the precision of contours (based on luminance)

Simple color space model: [R,G,B] per pixel

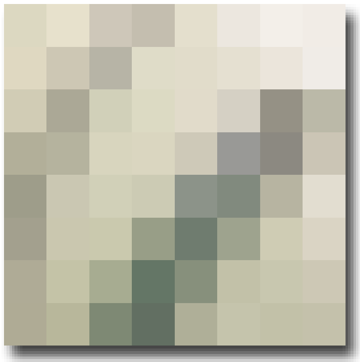
JPEG uses [Y, Cb, Cr] Model

Y = Brightness

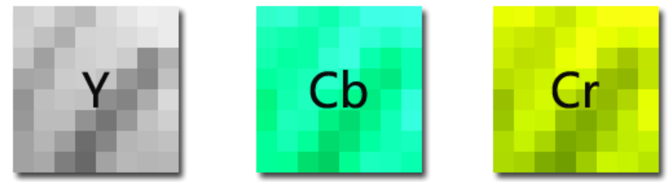
Cb = Color blueness

Cr = Color redness

Colour conversion



8x8 pixel
1 pixel = 3 components



MCU with sampling factor (1, 1, 1)

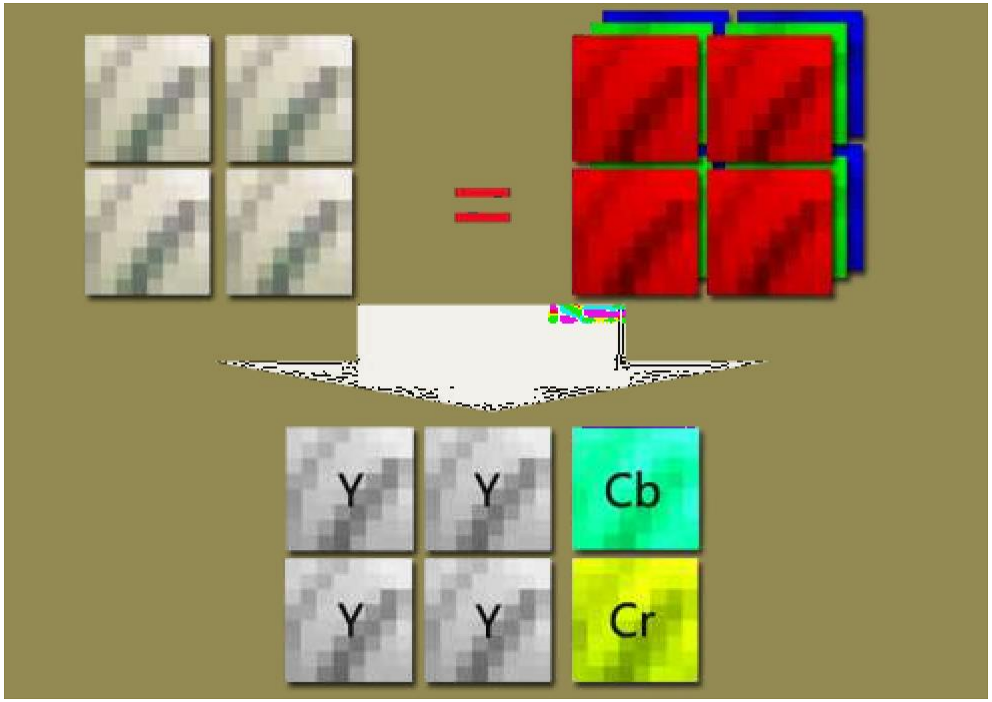
Y = 0.299R + 0.587G + 0.114B
Cb = -0.1687R – 0.3313G + 0.5B + 128
Cr = 0.5R – 0.4187G – 0.0813B + 128

Down-sampling

Y is taken for every pixel, and Cb, Cr are taken for a block of 2x2 pixels.

4 blocks
16 x16 pixel

MCU with sampling factor (2, 1, 1)



MCU = Minimum Coded Unit (smallest unit that can be coded)

Data size reduces to half

Step III - Forward DCT

- The DCT uses the cosine function, therefore not interacting with complex numbers at all.
- DCT converts the information contained in a block(8x8) of pixels from ***spatial*** domain to the ***frequency*** domain.

Why DCT?

- Human vision is insensitive to high frequency components, due to which it is possible to treat the data corresponding to high frequencies as redundant. To segregate the raw image data on the basis of frequency, it needs to be converted into frequency domain, which is the primary function of DCT.

DCT Formula

- 1-D DCT –

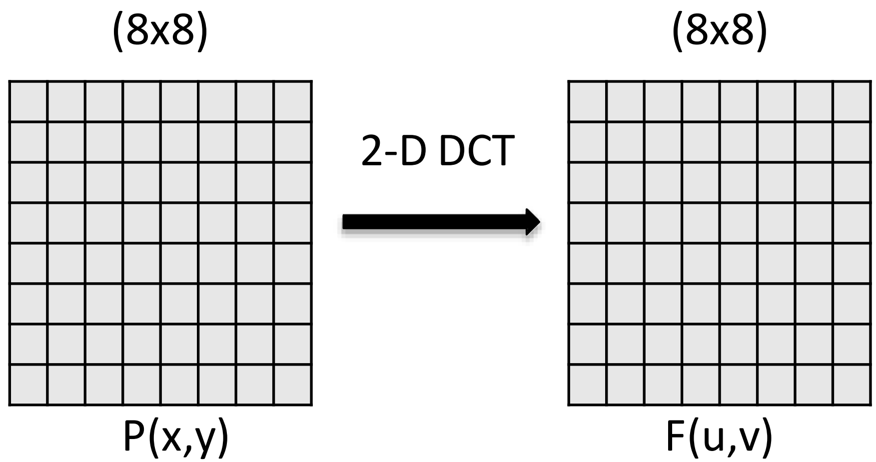
$$F(w) = \frac{a(u)}{2} \sum_{n=0}^{N-1} f(n) cos \frac{(2n+1)w\pi}{16}$$

- But the image matrix is a 2-D matrix. So we can either apply 1-D DCT to the image matrix twice. Once row-wise, then column wise, to get the DCT coefficients. Or we can apply the standard 2-D DCT formula for JPEG compression. If the *input matrix* is *P(x,y)* and the *transformed matrix* is *F(u,v) or G(u,v)* then the DCT for the 8 x 8 block is computed using the expression:-

- 2-D DCT –

$$G_{u,v} = \frac{1}{4}\alpha(u)\alpha(v) \sum_{x=0}^{7} \sum_{y=0}^{7} g_{x,y} cos \left[\frac{(2x+1)u\pi}{16}\right] cos \left[\frac{(2y+1)v\pi}{16}\right]$$

DC and AC components



F(0,0) is called the ***DC component*** and the rest of P(x,y) are called ***AC components***.

- For *u = v = 0* the two cosine terms are 0 and hence the value in the location *F[0,0]* of the transformed matrix is simply a function of the summation of all the values in the input matrix.
- This is ***the mean*** of all 64 values in the matrix and is known as the **DC coefficient.**
- Since the values in all the other locations of the transformed matrix have a frequency coefficient associated with them they are known as **AC coefficients.**

Step IV - Quantization

- Humans are unable to see aspects of an image that are at really high frequencies. Since taking the DCT allows us to isolate where these high frequencies are, we can take advantage of this in choosing which values to preserve. By multiplying the DCT matrix by some mask, we can zero out elements of the matrix, thereby freeing the memory that had been representing those values.
- The resultant quantize matrix will only preserve values at the lowest frequencies up to a certain point.
- Why Quantization?
  - To reduce the number of bits per sample.
- Two types:
  - **Uniform quantization**
    - q(u,v) is a constant
  - **Non-uniform quantization**
    - Custom quantization tables can be put in image/scan header.
    - JPEG Standard defines two default quantization tables, one each for luminance and chrominance.

Quantization

Standard Formula: $\mathbf{F}(\mathbf{u},v) = round\left(\frac{F(u,v)}{Q(u,v)}\right)$

- F(u,v) represents a *DCT coefficient*, Q(u,v) is *quantization matrix*, and **F**(**u,v**) represents the *quantized DCT coefficients* to be applied for successive entropy coding.
- The quantization step is the major information losing step in JPEG compression.
- For non-uniform quantization, there are 2 psycho-visually tuned quantization tables each for luminance (Y) and chrominance (Cb,Cr) components defined by jpeg standards.

Quantization

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

The Luminance Quantization Table

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

The Chrominance Quantization Table

- The entries of Q(u,v) tend to have larger values towards the lower right corner. This aims to introduce more loss at the higher spatial frequencies.
- The tables above show the default Q(u,v) values obtained from psychophysical studies with the goal of maximizing the compression ratio while minimizing perceptual losses in JPEG images.

Quantization - Example

160	80	47	39	5	3	0	0
65	53	48	8	5	2	0	0
58	34	6	4	2	0	0	0
40	7	6	1	1	0	0	0
8	4	0	0	0	0	0	0
5	2	0	0	0	0	0	0
3	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0

DCT Coefficients F(u,v)

Quantizer

16	7	4	2	0	0	0	0
5	4	3	0	0	0	0	0
4	2	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

Quantized Coefficients **F**(u,v)

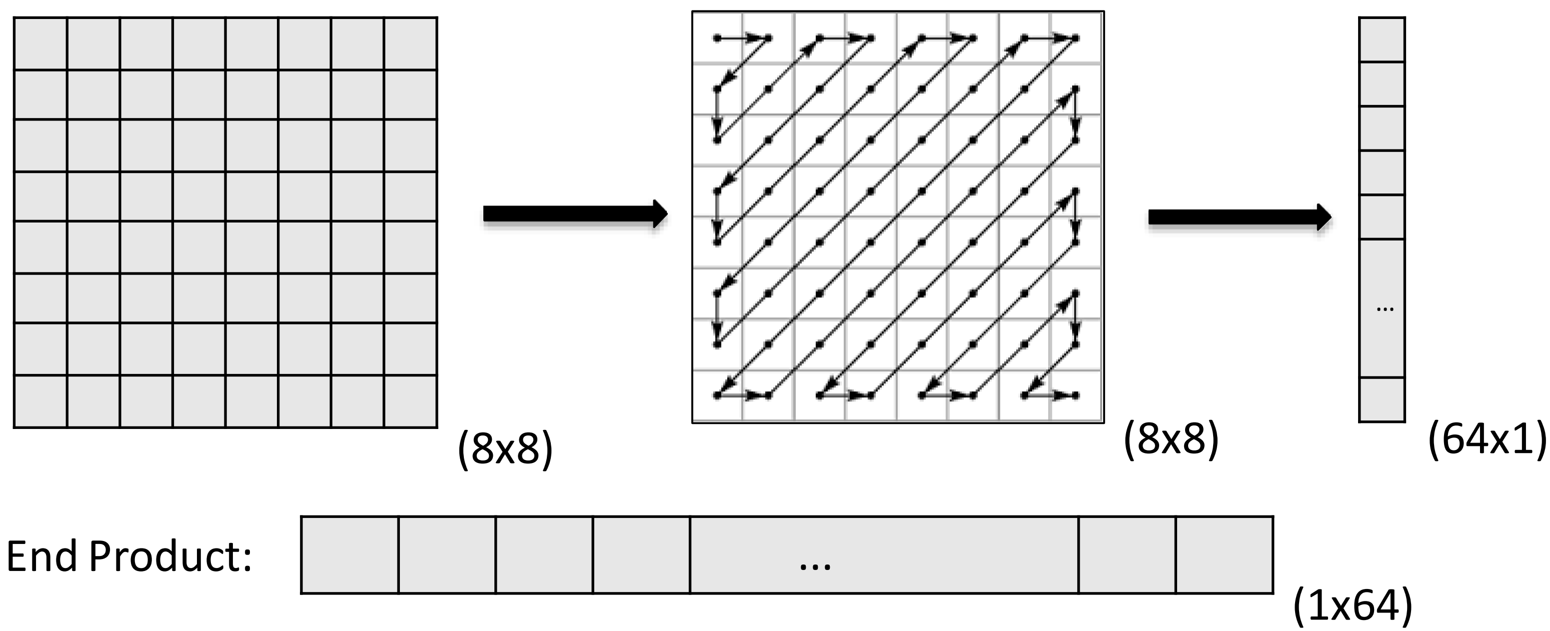
16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Quantization Table Q(u,v)

Each element of F(u,v) is divided by the corresponding element of Q(u,v) and then rounded off to the nearest integer to get the **F** (u,v) matrix.

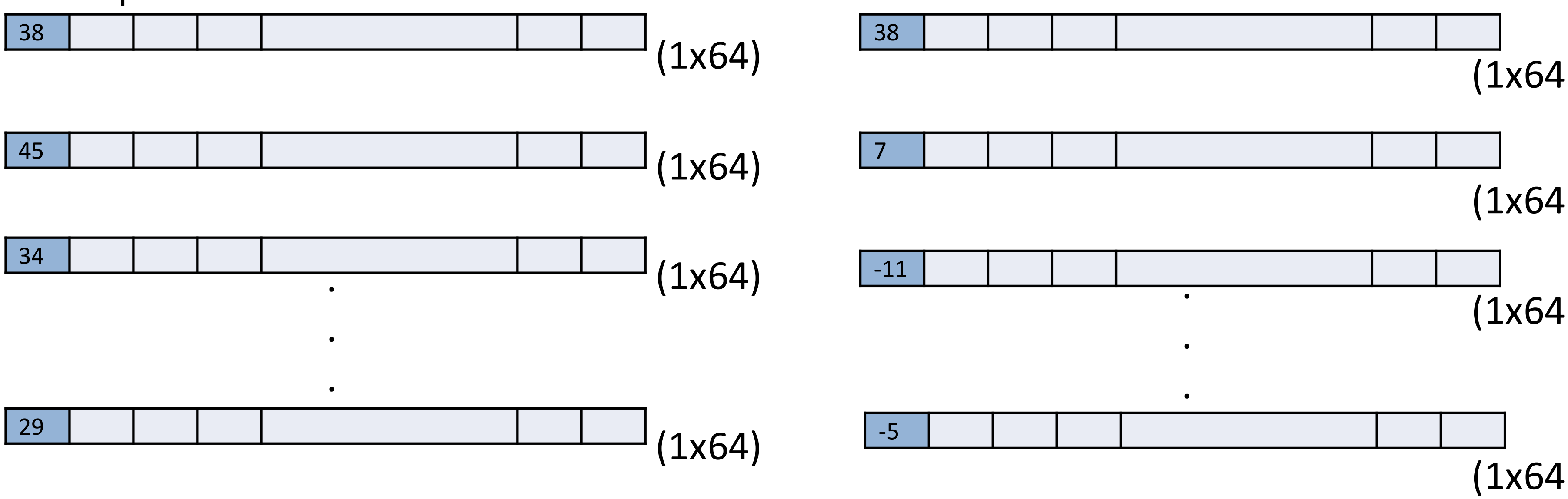
Step V – Zig-Zag Scan

- Maps 8 x 8 matrix to a 1 x 64 vector.
- Why zigzag scanning?
  - To group low frequency coefficients at the top of the vector and high frequency coefficients at the bottom.
- In order to exploit the presence of the large number of zeros in the quantized matrix, a zigzag of the matrix is used.



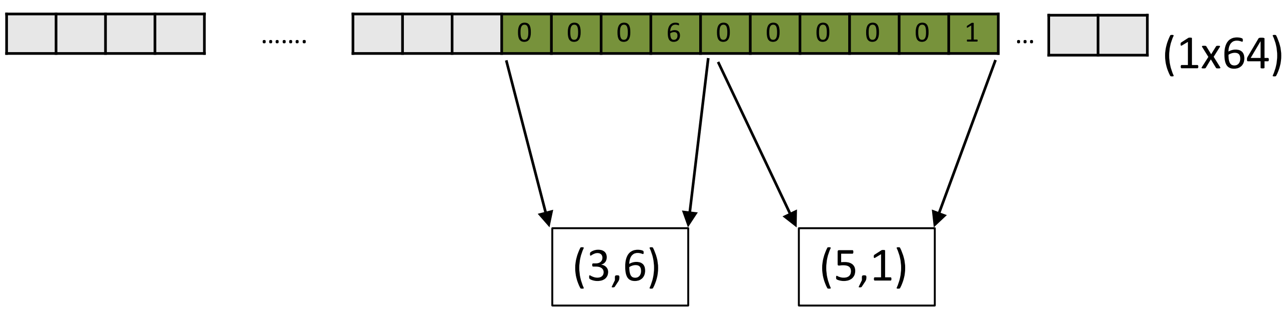
Step VI – Vectoring: DPCM on DC component

- Differential Pulse Code Modulation (DPCM) is applied to the DC component.
- DC component is large and varies, but often close to previous value.
- DPCM encodes the difference between the current and previous 8x8 block.
- The entropy coding operates on a 1-dimensional string of values (vector). However the output of the quantization matrix is a 2-D matrix and hence this has to be represented in 1-D form. This is known as ***Vectoring***.
- Example:-



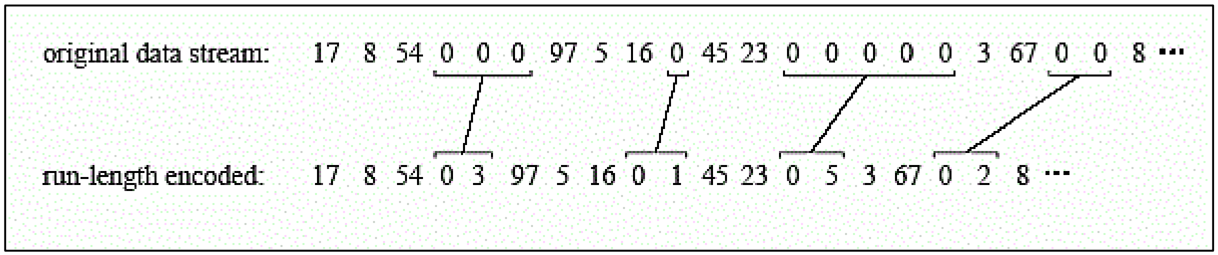
Step VII - RLE on AC components

- Run Length Encoding (RLE) is applied to the AC components.
- 1x63 vector (AC) has lots of zeros in it.
- Encoded as (***skip, value***) pairs, where *skip* is the number of zeros preceding a non-zero value in the quantized matrix, and *value* is the actual coded value of the non-zero component.
- (0,0) is sent as end-of-block (EOB) sentinel value and the zeros after that are discarded. Only the number of zeros is taken note of.

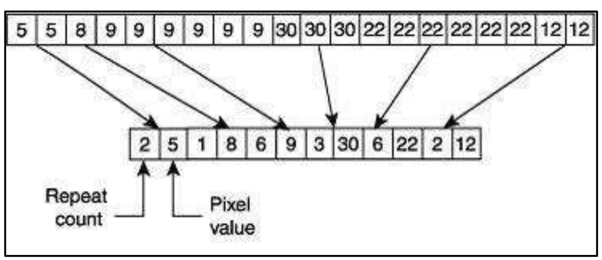


RLE Examples

Example 1-



Example 2-



The 1x64 vector is reduced in terms of bits by grouping the elements as groups of (repeat count,pixel value).

Step VIII – Huffman Coding

- DC and AC components finally need to be represented by a smaller number of bits
- Why Huffman Coding?
  - Significant levels of compression can be obtained by replacing long strings of binary digits by a string of much shorter code words.
- How?
  - The length of each code word is a function of its relative frequency of occurrence.
- Normally a table of code words is used with the set of code words pre-computed using the Huffman Coding Algorithm.
- In Huffman Coding, each DPCM-coded DC coefficient is represented by a pair of symbols : *(Size, Amplitude)*
- where *Size* indicates number of bits needed to represent the coefficient and *Amplitude* contains actual bits.

Huffman Coding: DC Components

- DC Components are coded as (*Size,Value*). The look-up table for generating code words for Value is as given below:-

SIZE	Value	Code
0	0	---
1	-1,1	0,1
2	-3, -2, 2,3	00,01,10,11
3	-7,..., -4, 4,..., 7	000,..., 011, 100,...111
4	-15,..., -8, 8,..., 15	0000,..., 0111, 1000,..., 1111
.		.
.		.
11	-2047,..., -1024, 1024,... 2047	...

Table 1: Huffman Table for DC components *Value* field

Huffman Coding – DC Components

- The look-up table for generating code words for *Size* is as given below:-

SIZE	Code Length	Code
0	2	00
1	3	010
2	3	011
3	3	100
4	3	101
5	3	110
6	4	1110
7	5	11110
8	6	111110
9	7	1111110
10	8	11111110
11	9	111111110

Table 2: Huffman Table for DC components *Size* field

Huffman DC Coding - Example

- If the DC component is 48, and the previous DC component is 52. Then the difference between the 2 components is (48-52) = -4.
- Therefore it is Huffman coded as: 100011
- 011: The codes for representing -4 (Table 1.)
- 100: The size of the value code word is 3. From Table 2, code corresponding to 3 is 100.

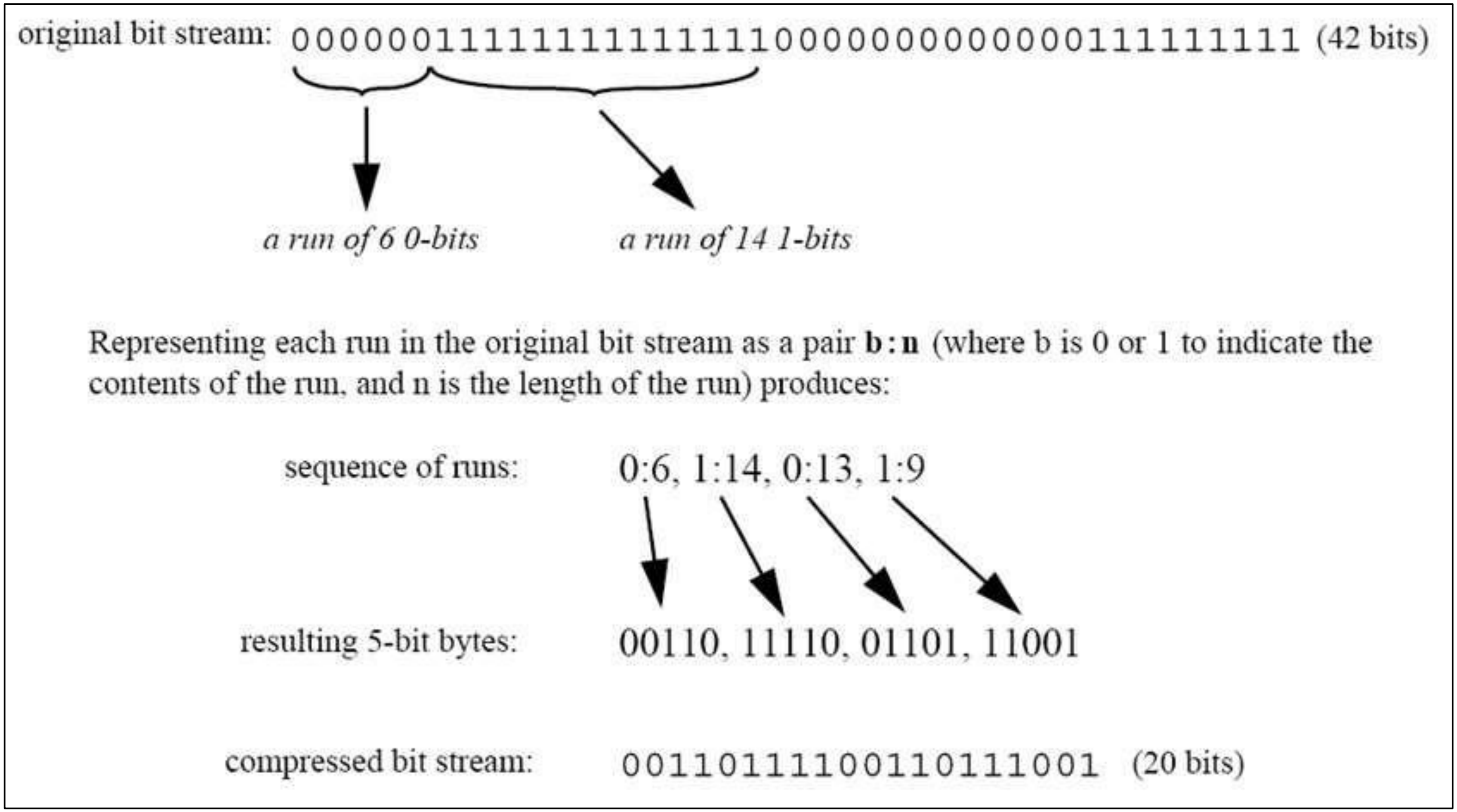
Huffman Coding: AC Components

- AC components are coded in two parts:
  - **Part1: (RunLength/SIZE)**
    - **RunLength:** The length of the consecutive zero values [0..15]
    - **SIZE:** The number of bits needed to code the *next* nonzero AC component's value. [0-A]
    - (0,0) is the End Of Block (EOB) for the 8x8 block.
    - **Part1** is Huffman coded (see Table 3)
  - **Part2: (Value)**
    - **Value:** Is the actual value of the AC component.(Table 1)

Run/ SIZE	Code Length	Code
0/0	4	1010
0/1	2	00
0/2	2	01
0/3	3	100
0/4	4	1011
0/5	5	11010
0/6	7	1111000
0/7	8	11111000
0/8	10	1111110110
0/9	16	1111111110000010
0/A	16	1111111110000011

Run/ SIZE	Code Length	Code
1/1	4	1100
1/2	5	11011
1/3	7	1111001
1/4	9	111110110
1/5	11	11111110110
1/6	16	1111111110000100
1/7	16	1111111110000101
1/8	16	1111111110000110
1/9	16	1111111110000111
1/A	16	1111111110001000
... 15/A	More	Such rows

Example of Run Length Encoding



Merits of JPEG Compression

- Works with colour and grayscale images, but not with binary images.
- Up to 24 bit colour images (Unlike GIF)
- Target photographic quality images (Unlike GIF)
- Suitable for many applications e.g., satellite, medical, general photography, etc.

MULTIMEDIA SYSTEMS AND APPLICATIONS

Data and File Format

- RTF
- TIFF
- RIFF
- MIDI
- JPEG
- AVI video file formats
- MPEG standards.

RTF

- The Rich Text Format (RTF) Specification provides a method of transferring encoding formatted text and graphics between different output devices, OS and applications.
- RTF uses the American National Standards Institute (ANSI), PC-8, Macintosh, or IBM PC character set to control the representation and formatting of a document, both on the screen and in print.

RTF(cont.)

- RTF files begins with control word "\rtfN"; where N is the major version number. An entire RTF file is consider a group and must be enclosed in braces. Current version is 1.7, hence the file will begin with
- {\rtf1\ansi\ansicpg1252\....}.....

TIFF

- **TIFF**: stands for **Tagged Image File Format**.
- The support for attachment of additional information (referred to as “tags”) provides a great deal of flexibility.

1.The most important tag is a format signifier: what type of compression etc. is in use in the stored image.

2.TIFF can store many different types of image: 1-bit, grayscale, 8-bit color, 24-bit RGB, etc.

3.TIFF was originally a lossless format but now a new JPEG tag allows one to opt for JPEG compression.

4.The TIFF format was developed by the Aldus Corporation in the 1980's and was later supported by Microsoft.

Resource Interchange File Format (RIFF)

- This is a file format for multimedia data on PCs. It can contain bit-mapped graphics, animation, digital audio and MIDI data.
- The WAV file format is the RIFF format for storing sound data.
- RIFF (similarly to TIFF - Tagged Image File Format) is a tagged file format. Tags allow applications capable of reading RIFF files to read RIFF files by another application, hence the word interchange in RIFF.
- Other Formats/Players - RealPlayer 7 (Windows NT) with RealAudio, MP3 (MPEG Audio Layer 3) audio, Midi players; MP3 players (MP3.com)

MP3

- Its full title is MPEG-1/2 Layer 3. It is a format for compressing sounds which uses a lossy technique that does not seriously degrade the quality of the sound because it filters out aspects of the original sound that the human ear cannot detect. After filtering it then applies further compression techniques. A form of coding called Huffman encoding is used to compress the data once it has been captured.
- One minute of music takes up around 1 MS of space. MP3 allows compression of CD•quality audio files by a factor of 12 with little loss in quality. This explains why it is such a widely used format.

MIDI

- The Musical Instrument Digital Interface (MIDI) is a standard interface used by musical instruments like keyboards, synthesisers and drum machines which enables notes played on an instrument to be saved on a computer system, edited and played back through a MIDI device.
- The information about the sound is stored in a MIDI file which the computer can then use to tell the instrument which notes to play.
- http://www.cool-midi.com/

MIDI (Cont.)

- When a MIDI sound is stored in a computer system the following attributes or properties of the sound are stored:

- Instrument, Pitch, Volume, Duration

- Defines the instrument being played. Each built-in sound on a MIDI keyboard has an instrument number assigned to it.

  When selected the instrument number is saved by the computer so that, on playback, the notes in the musical piece are played with the sound of that specific instrument.
- This sets the musical tone of a note which is determined by the frequency.

MIDI (Cont.)

- This controls the loudness or amplitude of the note.
- This determines the length of a note (the number of beats).
- Tempo Rate is the speed at which the piece of music is set.

Advantages of MIDI

- Allows musical pieces or messages to be exchanged and edited on different computers.
- It is an easily manipulated form of data. Changing the tempo is a straightforward matter of changing one of the attributes.
- A musician can store the messages generated by many instruments in one file. This enables a musician to put together and edit a piece of music generated on different midi instruments with complete control over each note of each instrument.

Advantages of MIDI(cont.)

- Produces much smaller file sizes than other sound formats.
- Because it is digital it is easy to interface instruments, such as keyboards, to computers. The musician can store music on the computer and the computer can then play the music back on the instrument.
- Disadvantage
- Browsers require separate plug-ins to play MIDI files.

EXIF

- **EXIF** (Exchange Image File) is an image format for digital cameras:
  1. Compressed EXIF files use the baseline JPEG format.

  2. A variety of tags (many more than in TIFF) are available to facilitate **higher quality printing,** since information about the camera and picture-taking conditions (flash, exposure, light source, white balance, type of scene, etc.) can be stored and used by printers for possible color correction algorithms.

  3. The EXIF standard also includes specification of file format for audio that accompanies digital images. As well, it also supports tags for information needed for conversion to FlashPix (initially developed by Kodak).