Digitized pictures

* To reduce time to transmit & to store digitized image over n/w, we should apply compression algorithm for this data of two dimentional array of pixel value.

So algorithm developed by Joint photographic expert group (JPEG) an international standard

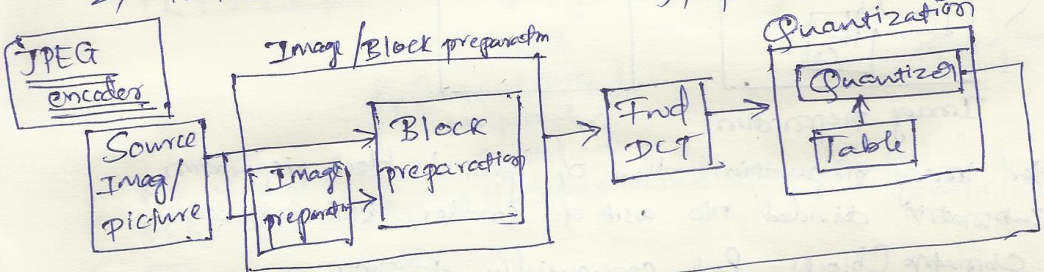
beased on IS-10918.

It also form most of video compression

* JPEG applicable for compression of monochrome & color image

It has 5 stages

1> Image / block preparation
2> Fwd DCT
3> quantization
4> Entropy encoding
5> frame building



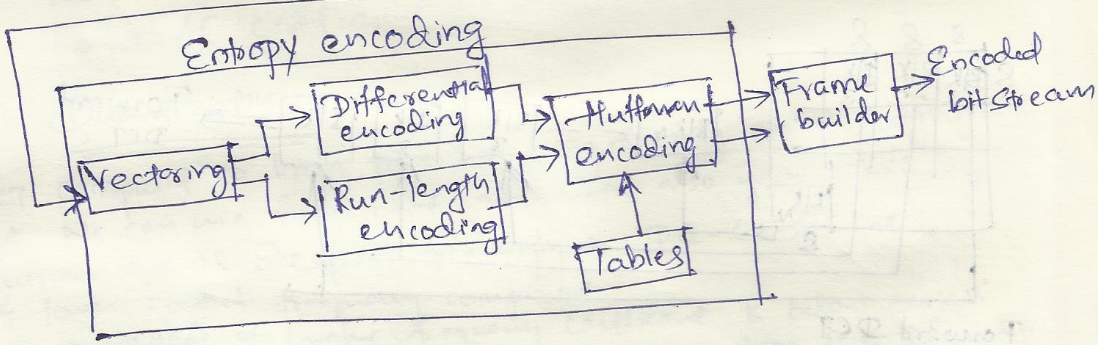


Image / Block preparation

* Src image is made up of one or more 2 Dimentional matrix of pixel value

In case of monochrome just single 2D matrix to store set of gray level (8 bit)

* In case of color image if CLUT (colour lookup table) is used than only one matrix is enough

If Image is represented in RGB formate than 3 matrix

* In alternative formate Y, Cb, Cr
Y → luminance; Cb, Cr → chrominance

for Y one matrix
& Cb, Cr required 2 matrix
these arrangement are shown below

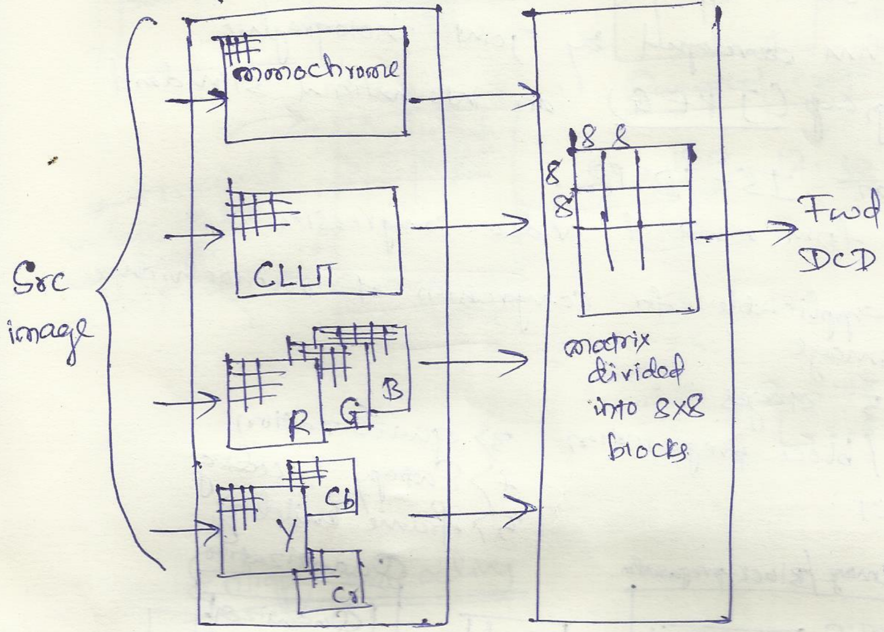
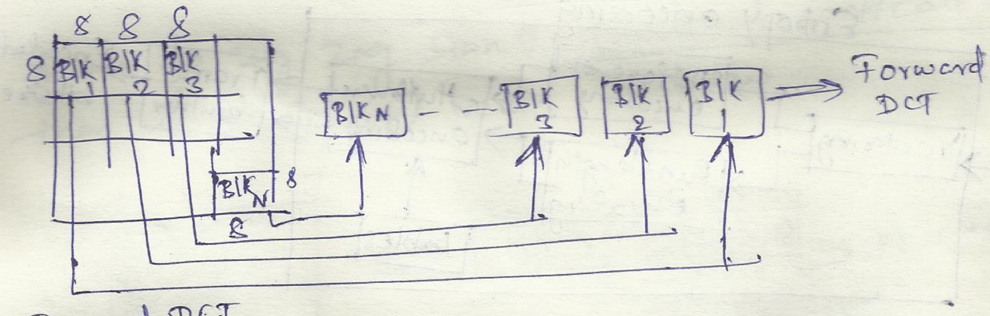


Image preparation

* After image preparation this o/p given to block preparation each submatrix divided into a set of smaller 8x8 submatrix
* each submatrix (block) fed sequentially to DCT



<u>Forward DCT</u>

* I/p 2D matrix is represented by $P(x,y)$ after DCT transfered to $F(i,j)$
DCT of 8x8 block of value computed using formula

$$F(i,j) = \frac{1}{4} C(i)\, C(j) \sum_{x=0}^{7} \sum_{y=0}^{7} P(x,y) \cos\frac{(2x+1)i\pi}{16} \cos\frac{(2y+1)j\pi}{16}$$

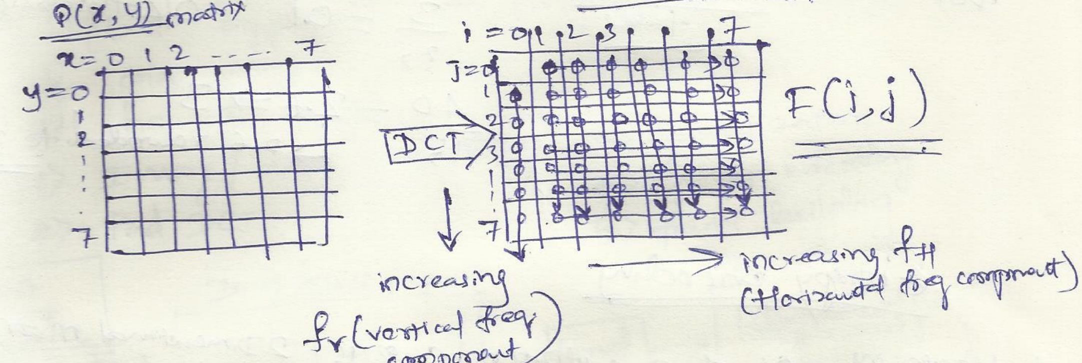
where $C(i)$ & $C(j) = 1/\sqrt{2}$ for $i, j = 0$

$= 1$ for all other value of i & j

& x, y, i & j all vary from 0 through 7.

* Tranfored matrix $F(i, j)$ has $8 \times 8 = 64$ coefficient (DCT) by taking avg of these DCT coefficient we get **DC coefficient**

* value in all location of transformation matrix have frequency component in vertical & horizontal direction (like x axis & y axis variation) called as **AC coefficient**



$P(x, y) = 8 \times 8$ matrix of pixel value

$F(i, j) = 8 \times 8$ matrix of transformed value / spatial freq coefficient

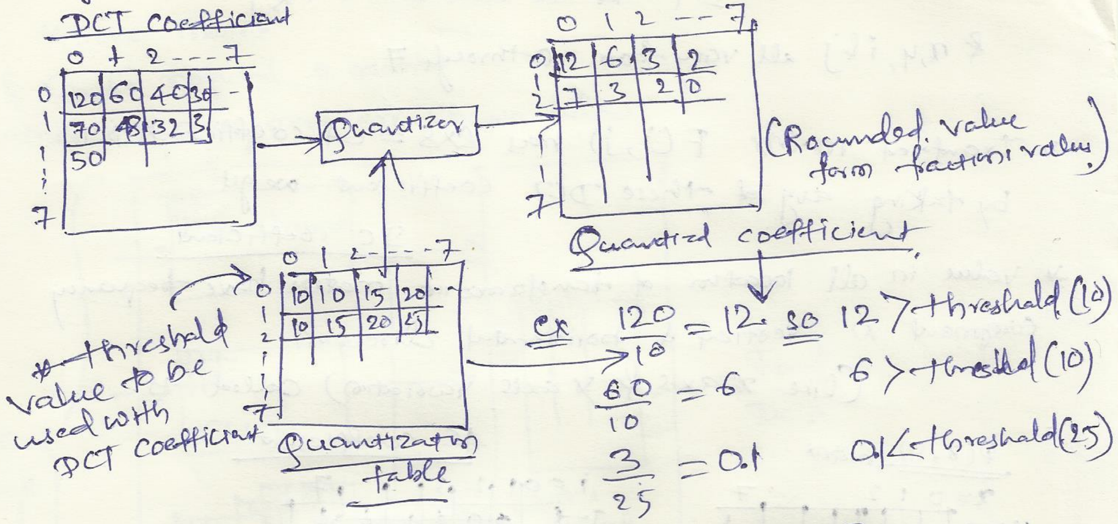
□ = DC coefficient

[o] = AC coefficient

Quantization

* DCT computed to high precision using floating point arithmatic but we can use fixed point arithmatic also with small loss
* human eye is respond primarily to DC-coefficient & lower spatial frequency component
* If magnitude of higher frequency coefficient is below certain threshold eye will not detect it, so we can drop this information
* So we should compare coefficient with corresponding threshold but insted of this division operation is performed using threshold.
* If result (quotient) < 0 means coefficient < threshold
  If " > 0 " coefficient > threshold.
* So threshold value ~~held~~ held by 64 DCT coefficient

arranged in matrix known as quantization table



ex $\frac{120}{10} = 12$ so $12 >$ threshold (10)

$\frac{60}{10} = 6$ $\quad 6 >$ threshold (10)

$\frac{3}{25} = 0.1$ $\quad 0.1 <$ threshold (25)

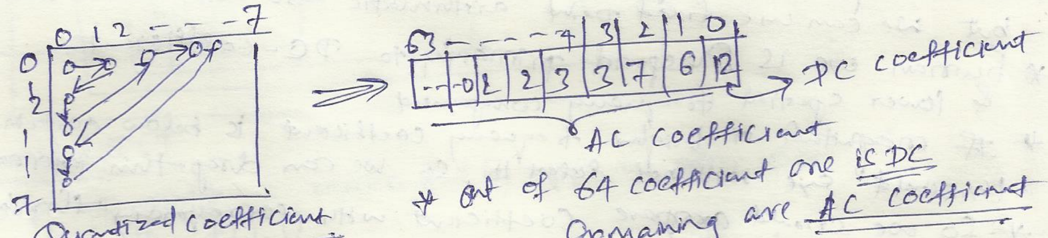
$\frac{40}{15} = 2.6 \Rightarrow 3$

↳ 2.6 rounded to 3

Entropy encoding

1) vectoring O/p of quantization stage is 2 Dimentional matrix of value to apply encoding. we should represent value in the form of single dimentional vector this operation is known as vectoring

* we do zig-zag scan of quantized coefficient matrix than we get (1X64) vector as below



* out of 64 coefficient one is DC remaining are AC coefficient

(2) Differential encoding (for DC coefficient)

* " encoding is for DC-coefficient
* Diffⁿ in magnitude of DC-coefficient's quantized block so nub of bits to code DC-coefficient required is less

ex 12, 13, 11, 10, 10

diffⁿ encoded value 12, 1, -2, 0, -1
difference value.

* First element in vector table has DC-coefficient which has information related with Y, Cb, Cr/color in 8×8 block of pixel.
* block has small physical area, so DC coefficient vary slowly from one block to next block
* encoding is done in the form of (SSS, value)
  - SSS → Nub of bits for coding the value
  - value → Nub is binary form

ex

Value	SSS	Value
12	4	1100
1	1	1
-2	2	01
0	0	
-1	1	0

ex 12 is coded as (4, 1100)

* Rules to encode the value as below

Diffⁿ value	SSS	Encoded value
0	0	
-1, 1	1	1=1, -1=0
-3, -2, 2, 3	2	2=10, -2=01 3=11, -3=00
⋮		

* Negative number is compliment of +ve number.

Runlength encoding (for AC coefficient)

* Out of 64 in vector table remaining 63 are AC-coefficient we get 63 by zig-zag scan can be coded by (skip, value)
  - skip → Nub of zero in string
  - value → Next non zero coefficient

ex for vector table

63						3	2	1	0
---	0	0	2	2	3	3	7	6	12

⇒ coded as below
(0, 6) (0, 7) (0, 3) (0, 3) (0, 2) (0, 2) - - -
- - - (0, 0)

(0, 6): 0 → skip, 6 → value.
(0, 0) → end of string in block

Huffman coding

* We can replace long string of binary digit by string of much shorted codeword. So compression can be obtained.
* We have huffman codeward table & we use o/p of Differential & runlength encoder o/p to do Huffman coding.

SSS	Huffman codeward.
0	010
1	011
2	100
3	00
4	101
11	11111110

ex DC Differential o/p is as below

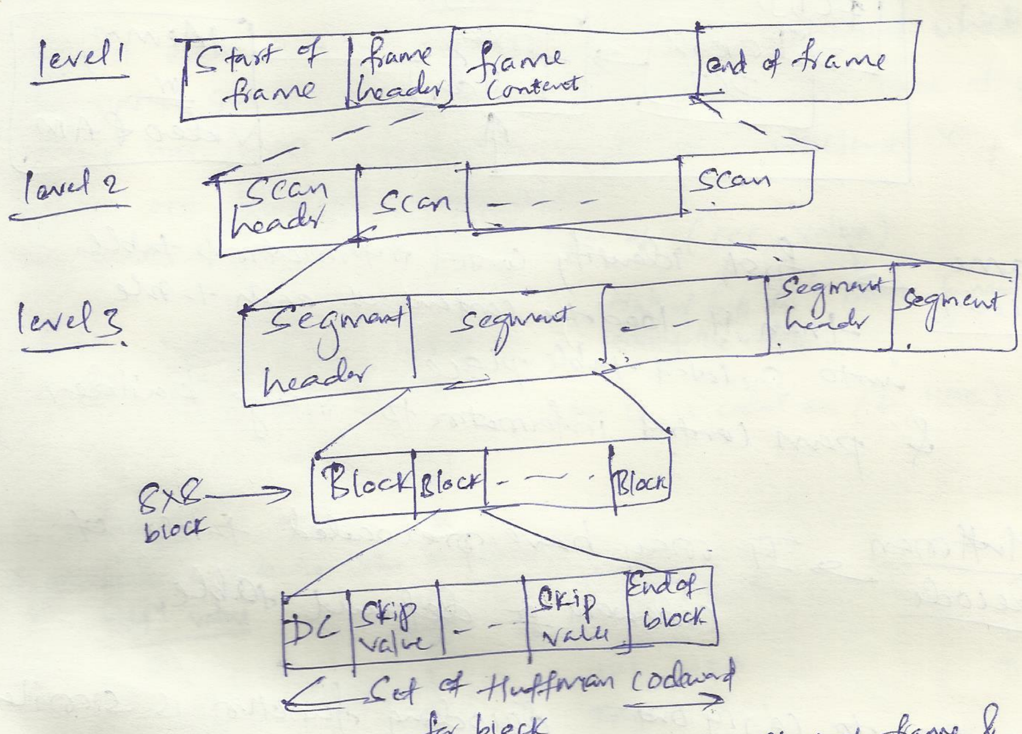
Value	SSS	Huffman encoded	Encoded value	Encoded bit stream
12	4	101	1100	101 1100
1	1	011	1	011 1
−2	2	100	01	100 01
0	0	010		010
−1	1	011	0	011 0

* This Huffman coding is also applied for o/p of runlength coding.
* Here skip & SSS treated as one & the same & encoded using Huffman {table 3.2 page 185}

{Refere example 3.8 page No− 186}

Frame building

* bit stream o/p of JPEG encoder is compressed. version of picture
* it is stored in memory of computer.
* JPEG standard also include information of total. Number of bit stream related to particular picture is known as frame
* It's out line structure is shown below



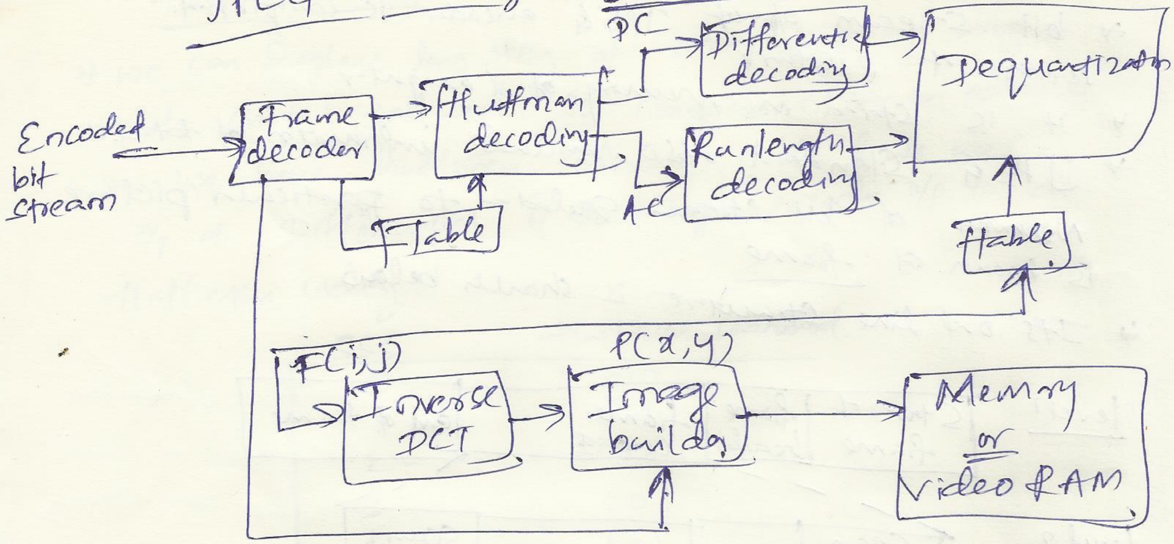
* We encapsulate all the information bet^n start of frame & end of frame

frame header has
* Overall width & height of image in pixel.
* Number & type of component that are used to represent the image (CLUT/, R, G, B, Y/Cb/Cr)
* digital formats used (4:2:2, 4:2:0, etc)
* frame content has scan

Scan has
* identity of component (R/G/B etc)
* No. of bits used to digitized each component
* quantization table of value that have been used to encode each component

JPEG decoding

decoder schematic



Frame decoder → first identify control information & table. than it load the content of each table into related table place & pass control information to image builder.

Huffman decode → It may have preloaded table of codeword or default table.

* so time to carry out decoding function is similar to that used to perform encoding.