Multimedia Communication (SW-416)

JPEG COMPRESSION

Dr. Areej Fatemah

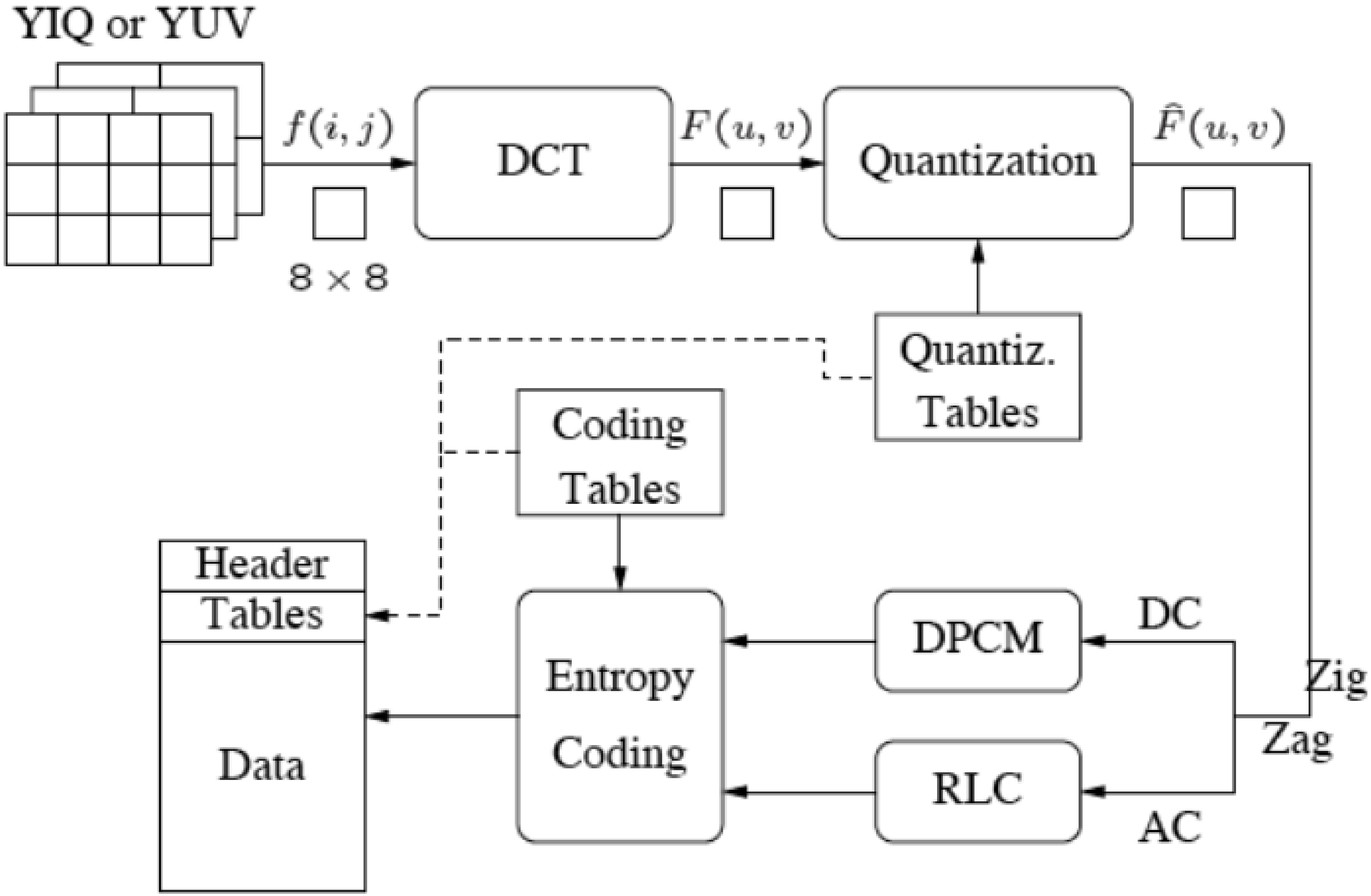
email: areej.fatemah@faculty.muet.edu.pk

JPEG

- Developed by the “Joint Photographic Experts Group”
  - a group of image processing experts that devised a standard for compressing an image
- JPEG (or JPG) is not really a file format but rather an image compression standard.
- Accepted as an international standard in 1992.
- The JPEG algorithm is designed specifically for the human eye.
- It exploits the following biological properties of human sight:
  - We are more sensitive to the luminosity of color, rather than the chromatic value of an image, and
  - We are not particularly sensitive to high-frequency content in images.

JPEG

- The algorithm can be neatly broken up into several stages.
- JPEG combines several lossless and lossy compression techniques:
  - Color model transformation
  - 2d Discrete Cosine Transform on 8x8 blocks
  - Quantization (filtering)
  - Differential Pulse Code Modulation (DPCM), Run-length coding
  - Entropy coding (Huffman)
- The result of these steps is a compressed image in the .jpg file format. This format contains the compressed image as well as information that is needed for it to be uncompressed, with other information to allow for re-expanding the image.


YIQ or YUV
$f(i,j)$
8×8
DCT
$F(u,v)$
Quantization
$\hat{F}(u,v)$
Quantiz.
Tables
Coding
Tables
Header
Tables
Data
Entropy
Coding
DPCM
RLC
DC
AC
Zig
Zag

JPEG Compression
Transform RGB to YUV or YCbCr and subsample color

- **Step-1:**
- Human eye is more sensitive to (luminosity than other color (100 million rods, 6 million cones)
- **YUV Color Model**
  - *Y: luminance information (gray)*
  - U and V: chrominance information (color)
- **YCbCr Color Model**
  - *Y: luminance information (gray)*
  - Cb and Cr: chrominance information (color)
- We can use the full resolution for the luminance and use half resolution for the chrominance components U and V

JPEG Compression
Transform RGB to YUV or YCbCr and subsample color

- **RGB v/s YCbCr Color Model**

JPEG Compression
Transform RGB to YUV or YCbCr and subsample color

- YCbCr uses the RGB color model to effectively compress color.
  - YCbCr divides data into three separate channels; luminance (Y), chroma blue (Cb) and chroma red (Cr).
  - The luminance (Y) data from each red, green and blue channel is extracted and separated from the chroma data.
- Cb and Cr is the blue component and red component related to the chroma component. That means: **Cb is the blue component relative to the green component. Cr is the red component relative to the green component.** These components are less sensitive to the human eyes.
- The combined luminance data of RGB is encoded separately to create one luminance channel (Y). This data alone makes up a complete and sufficient black and white image.
- Next, all we need to do is add the color.

JPEG Compression
Transform RGB to YUV or YCbCr and subsample color



JPEG Compression
Transform RGB to YUV or YCbCr and subsample color

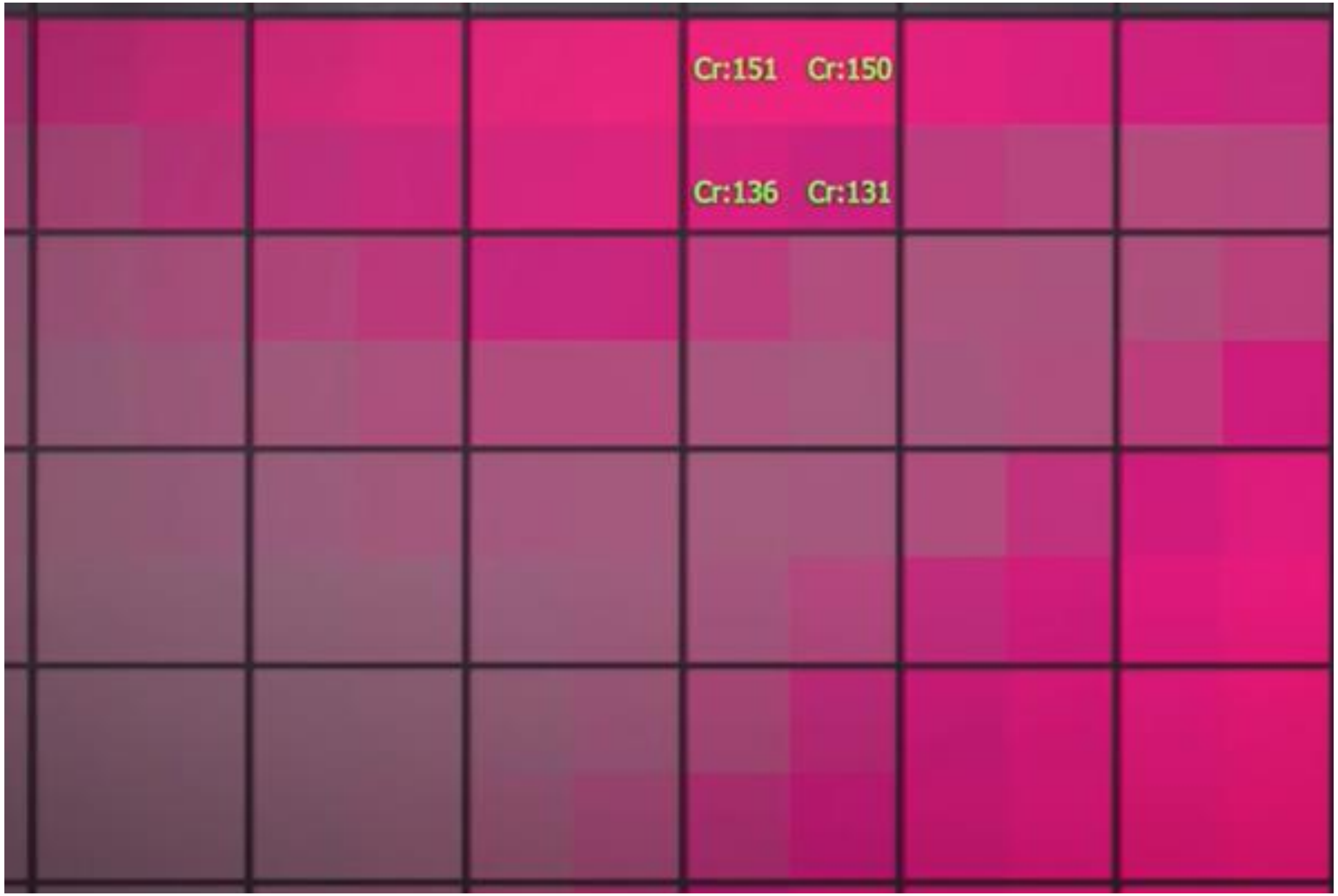
$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

JPEG Compression
Transform RGB to YUV or YCbCr and subsample color

- **Step-1:**
- This step does not reduce the amount of data as it just changes the representation of the same information.
- But as the human observer is much more sensitive to the intensity information than to the color information, the color information can be sub-sampled without a significant loss of visible image information.
- But of course the amount of data is reduced significantly.

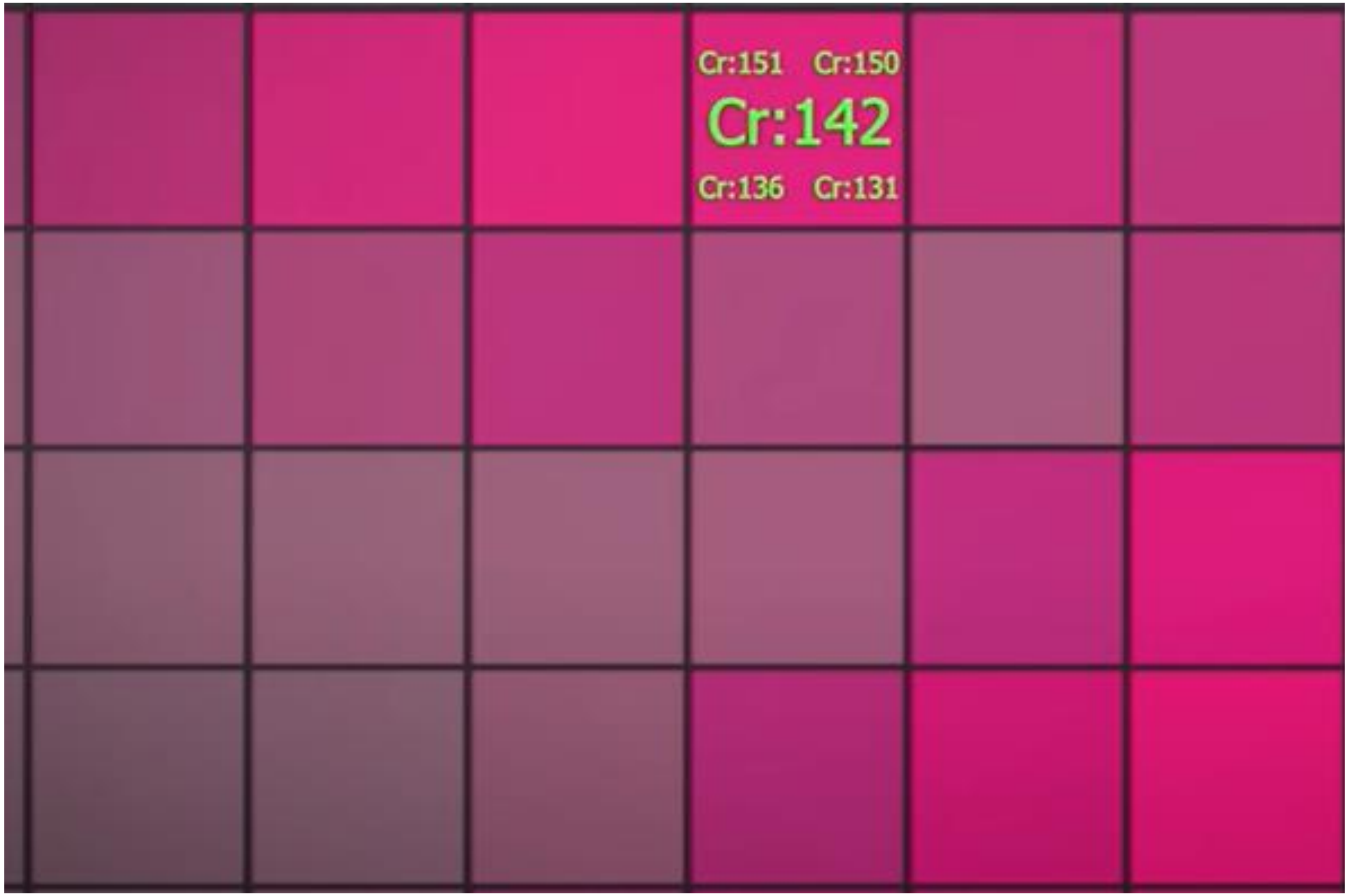
JPEG Compression

Transform RGB to YUV or YCbCr and subsample color



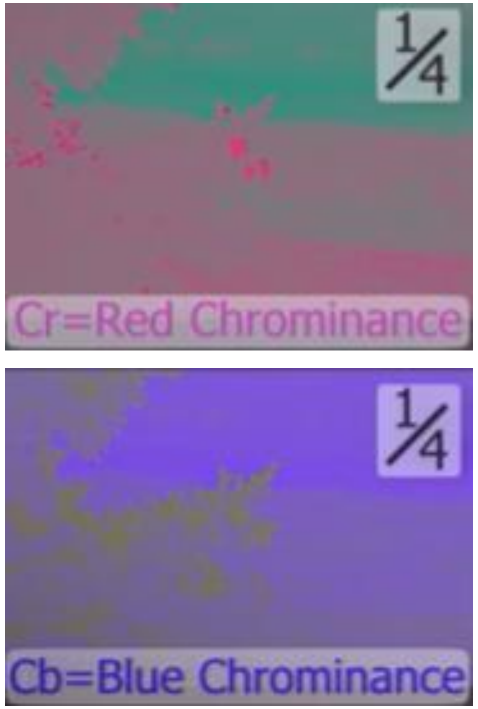
JPEG Compression

Transform RGB to YUV or YCbCr and subsample color



JPEG Compression
Transform RGB to YUV or YCbCr and subsample color

Before $1 + 1 + 1 = 3.0$
After $\frac{1}{4} + \frac{1}{4} + 1 = 1.5$



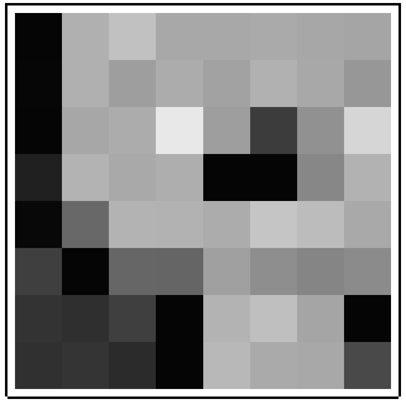
JPEG Compression
DCT on 8x8 image blocks

- **Step-2:**
- The YCbCr image data with the subsampled color components is then divided into 8x8pixel blocks.
- The complete algorithm is performed on these pixel blocks.
- Each block is transformed using the discrete Cosines Transformation (DCT).
- What does this mean?
  - An image needs to be separated into 8*8 pixel blocks.
  - That means each block has 8*8 pixels and it is 64 pixels in one block.
  - Let's assume that the dimensions of this image are 240*320. That means this image has 76800 pixels.
  - If we divide this in 64 we can get the number of blocks. 76800 = 64 * 1200.
  - We have 1200 blocks in this image.

JPEG Compression
DCT on 8x8 image blocks

- **Step-2:**
- In the spatial domain (before we have transformed) the data is described via digital value for each pixel.
- So we represent the image content by a list of pixel number and pixel value.
- ***After the transformation***, the image content is described by the coefficient of the spatial frequencies for vertical and horizontal orientation.
- So in the spatial domain, we need to store 64 pixel values.
- In the frequency domain, we have to store 64 frequency coefficients.
- No data reduction so far.

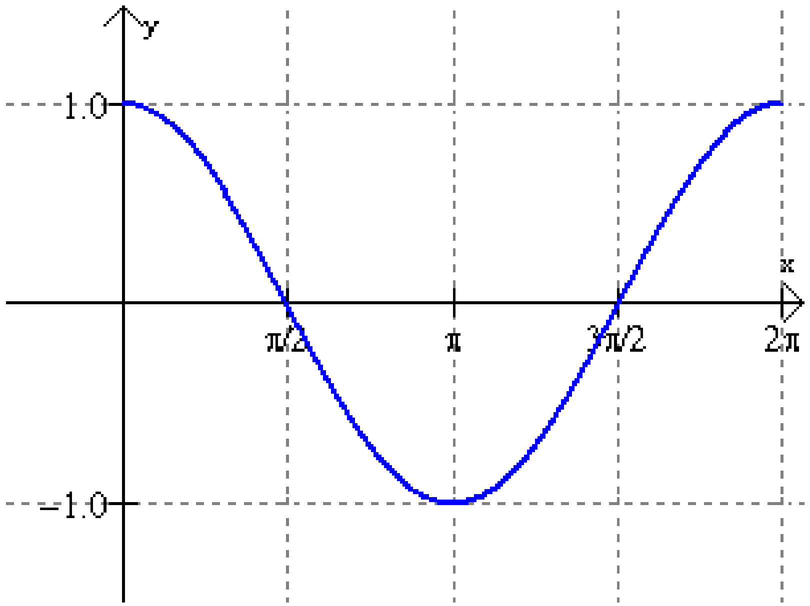
JPEG Compression
DCT on 8x8 image blocks



- **Step-2:**

JPEG Compression
DCT on 8x8 image blocks

- **Step-2:**
- Cosine signal lies in between the 1 and -1.
- What we do with this transform is that we represent the image pixels with different of cosine waves.
- By doing this we eliminate high frequencies in the signal as the human eye is not sensitive to the very high frequency changes of the image.
  - It is the graphical explanation of this transformation.
- If we talk about the numerical explanation, this transformation creates a new matrix that has the values in left upper corner of the matrix and other places are almost nearly zero.



JPEG Compression
Uniform Scalar Quantization

- **Step-3:**
- To reduce the needed amount of data to store the 64 coefficients, these are quantized.
- The human eyes are more reactive to low frequencies than to high ones.
  - Quantization is used to discard perceptibly insignificant information.
- Depending on the size of the quantization steps, more or less information is lost in this step.
- Most of the times, the user can define the strength of the JPEG compression.
- Quantization is the step where this user information has influence on the result (remaining image quality and file size).

JPEG Compression
Uniform Scalar Quantization

- **Step-3:**
- Each of the 64 coefficients are uniformly quantized
- Encode the DCT coefficients
  - Example (uniform quantization)
  - Quantize to 5 bits: $(10111)_2$ = 23 or Quantize to 4 bits: $(1011)_2$ = 11 or
  - Quantize to 3 bits: $(110)_2$ = 6 or Quantize to 2 bits: $(11)_2$ = 3
- Quantization error is the main source of lossy compression.
- It basically converts each real DCT coefficient to an integer by scaling it by a factor and then discarding the digits after the decimal point.

JPEG Compression
Preparation for Entropy Coding

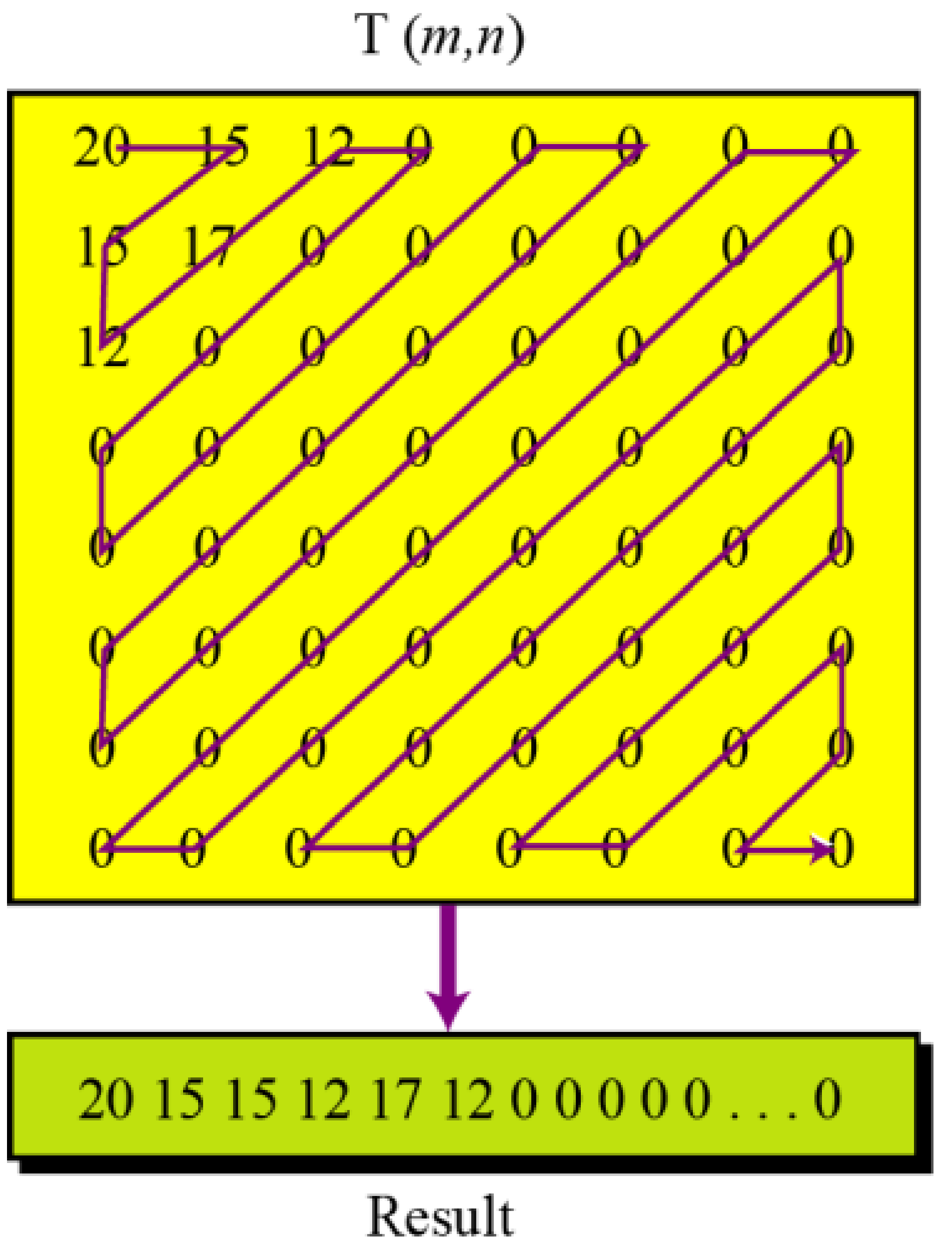
- **Step-4:**
- The remaining steps all lead up to entropy coding of the quantized DCT coefficients
- These additional data compression steps are lossless
- Most of the lossiness was in the quantization step

JPEG Compression
Preparation for Entropy Coding

- **Step-4:**
- Depending on the quantization, more and more coefficients are reduced to zero. And the probability is very high, that these coefficients with value 0 are found in the higher frequencies rather than in the lower frequencies.
- So the data is reordered that way, that the values are sorted by the spatial frequency, first the low frequencies, high frequencies come last. To take advantage of this, use zigzag scanning to create a 64-vector
- After reordering, it is very likely, that we have some values at the beginning and then a lot of 0s. Instead of storing “0 0 0 0 0 0 0 0 0 0”, we can easily store “10x0”.
  - That way, the amount of data is also reduced significantly.
- *Run Length Coding is a good idea.*

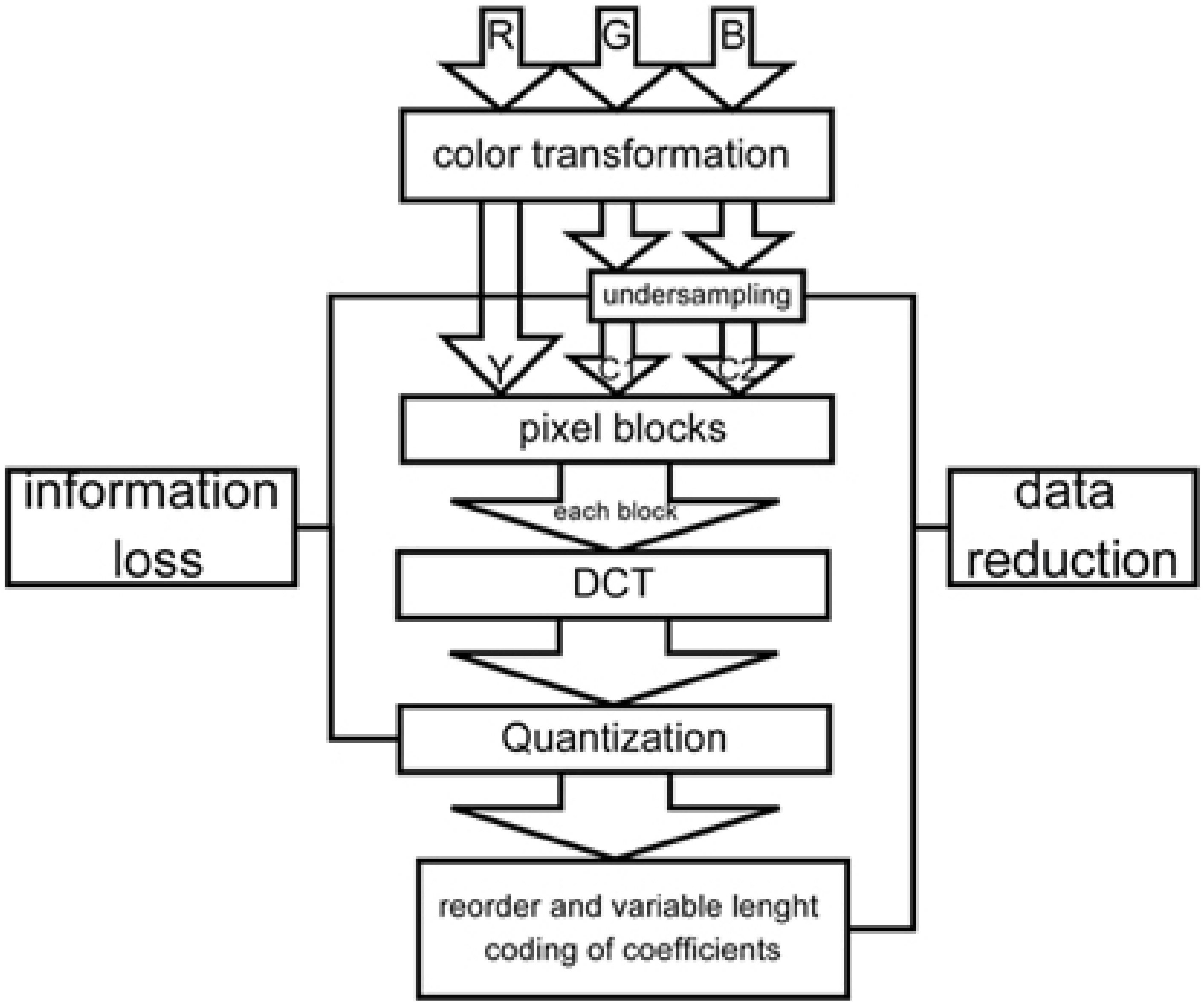
JPEG Compression
Zig-Zag Scanning

- **Step-4:**



JPEG Compression Summary

- The JPEG compression is a block based compression.
- The data reduction is done by the subsampling of the color information, the quantization of the DCT-coefficients and the Huffman-Coding (reorder and coding).
- The user can control the amount of image quality loss due to the data reduction by setting (or chose presets). For a high quality compression, the subsampling can be skipped and the quantization matrix can be selected in a way that the information loss is low.
- For high compression settings, the subsampling is turned on and the quantization matrix is selected to force most coefficients to 0.
- In that case, the image gets clearly visible artifacts after decompression.


R
G
B
color transformation
undersampling
Y
C1
C2
pixel blocks
each block
DCT
Quantization
reorder and variable lenght
coding of coefficients
information
loss
data
reduction