

MULTIMEDIA COMMUNICATION

Laboratory Session: Discrete Cosine Transform

Fernando Pereira

The objective of this lab session about the Discrete Cosine Transform (DCT) is to get the students familiar with the way the DCT works with the target to compress images. For that purpose, the application “Audio and Video Communication Simulation System”¹ will be used since it includes among others a module demonstrating the various operations involved in compression of an 8×8 image block using the DCT.

1. THE DCT TRANSFORM

With no doubt, the DCT is the most popular transform in the field of image and video compression since it is used by the JPEG, H.261, H.263, MPEG-1 Video, MPEG-2 Video and MPEG-4 Visual and H.264/AVC coding standards, due to its great compression versus complexity trade-off.

The DCT is computed as the product of two matrices, notably the image block (luminance or chrominances) and the DCT basis functions (see Figure 1), according to the following expressions:

$$F(u,v) = 1/4 C(u) C(v) \sum_{x=0}^7 \sum_{y=0}^7 f(x,y) \cos[\pi u(2x+1)/16] \cos[\pi v(2y+1)/16]$$

$$\text{with } u = 0, 1, 2, \dots, 7 \\ v = 0, 1, 2, \dots, 7$$

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

Moreover, the inverse DCT is computed as:

$$f(x,y) = 1/4 \sum_{u=0}^7 \sum_{v=0}^7 C(u)C(v)F(u,v)\cos[\pi u(2x+1)/16]\cos[\pi v(2y+1)/16]$$

$$\text{with } x = 0, 1, 2, \dots, 7 \\ y = 0, 1, 2, \dots, 7$$

¹ The application “Audio and Video Communication Simulation System” has been developed by Pedro Fernandes in the context of his M.Sc. Thesis. I would like to express here my appreciation for his work considering the possibilities this application has opened for the lab sessions of the Multimedia Communication course.

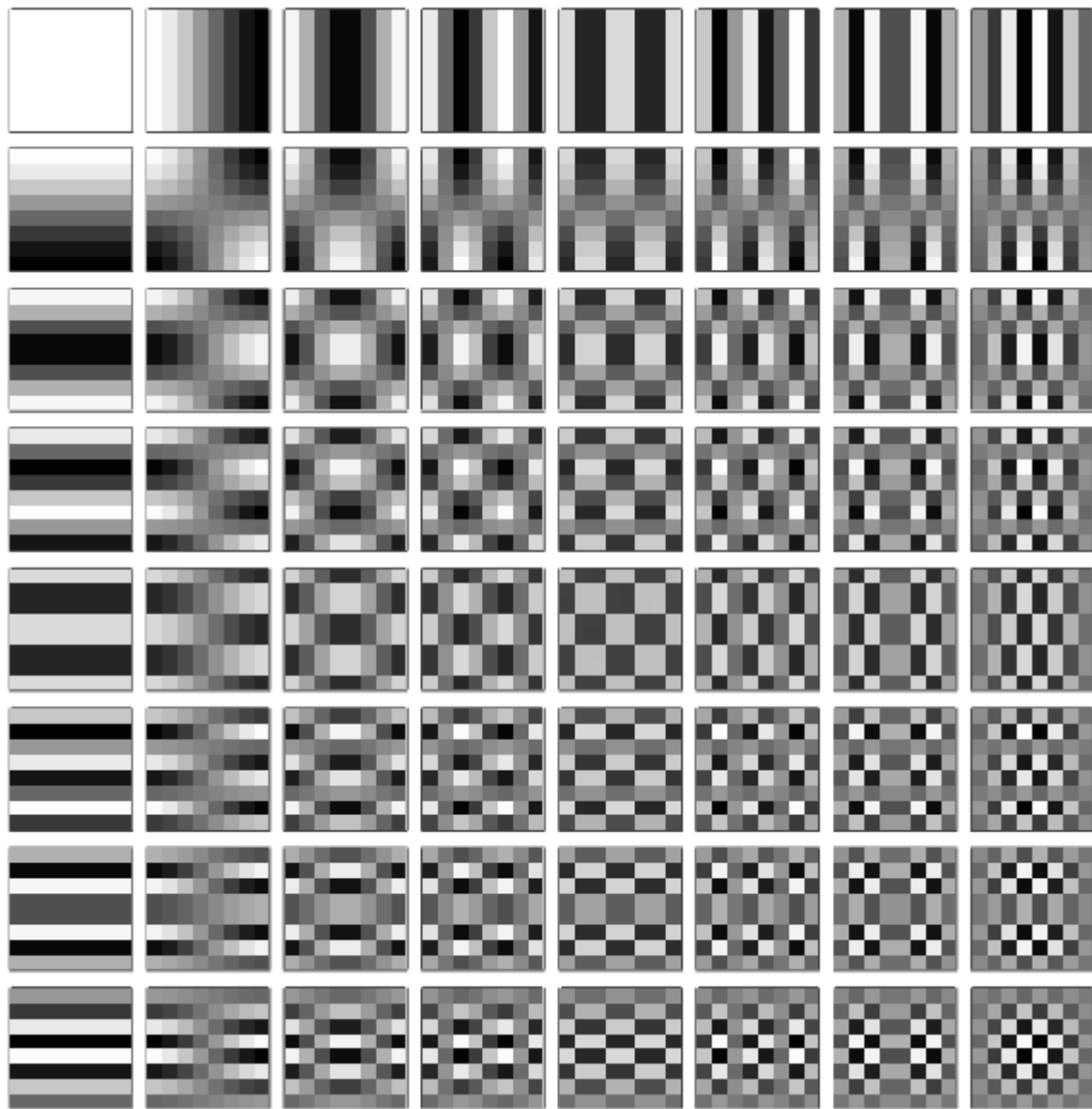


Figure 1 – Visual representation of the DCT basis functions.

2. THE DCT MODULE

The DCT module uses several matrices which are described in the following:

2.1 Data Matrices

There are several data matrices, notably corresponding to the data in pixel and DCT transform domains, before and after quantization:

- **Original Image** – Matrix with the 8×8 samples image block over which the DCT is applied, this means a spatial block with 64 PCM samples. Since each sample is represented with 8 bits, each luminance sample has a value between 16 and 235 (reduced range), if the ITU-R 601 recommendation is adopted, and between 0 and 255 (full range), otherwise; the ‘H.261 mode’ button allows switching between the reduced and full ranges. It is possible to see the luminance values on the matrices by using the ‘Show image values’ button. The application makes

available a set of example image blocks which may be selected using the ‘Prev’ and ‘Next’ buttons. The ‘Invert’ button allows changing an original image block to its complementary luminance image, this means ‘darks’ become ‘lights’ and vice-versa or $Y_{invert}=255-Y$.

- **DCT Coefficients** – Matrix with the 8×8 DCT coefficients; although the DCT coefficients are theoretically real values, they are in practice represented with a certain limited arithmetic, this means a certain number of bits (in this case with the -2048 to 2048 range). It is possible to see the DCT coefficient values on the matrices by using the ‘Show DCT values’ button.
- **Selected DCT Coefficients** – Matrix with the DCT coefficients which have fulfilled the selection criteria defined by the user defined ‘Selection Filter’ matrix (see below); the selection criteria are not normative and, thus, each encoder developer may adopt different solutions. This matrix basically defines which coefficients will be submitted to quantization; the others are ‘killed’ at this stage.
- **Quantized DCT Coefficients** – Matrix with the quantized DCT coefficients using the quantization steps user defined in the ‘Quantization Step’ matrix (see below). Clipping may happen when the maximum quantization level is lower than the quantization level determined by the simple division of the DCT coefficient by the corresponding quantization step; in the H.261 standard, this clipping may happen since the maximum quantization level is lower than the maximum quantization level that may be needed when the quantization step is very low. An example is the quantization of a DCT coefficient with value 1660 with a quantization step of 2: if the maximum quantization level available is 512, as in the H.261 standard, there will be clipping since the ‘correct’ value should be 830. When clipping happens for a DCT coefficient, the corresponding cell will appear in red in screen.

The quantization characteristic used in this application is:

$$Quantization_level = Int(DCT\ value / Quantization_step)$$

Although the quantization characteristic is not normative, uniform quantization as above is often used. The quantization process implies introducing an error that cannot be recovered, making the overall coding process lossy.

- **Restored DCT Coefficients** – Matrix with the DCT coefficient values recovered at the decoding based on the quantization levels and quantization steps used. This process is well known as ‘inverse quantization’ since it uses the received quantization levels and the corresponding quantization steps to recover the decoded DCT coefficient value, e.g. as in recommendation H.261:

$$Dequantized_DCT_value = Quantization_step \times (Quantization_level + 1/2)$$

Naturally, the *Dequantized_DCT_value* is typically different from the corresponding *DCT_value* due to the quantization process, since the quantization error cannot be ‘removed’

The inverse quantization process is typically normative which means the dequantized values for each quantization level and quantization step are fixed and well known in advance by the encoder; this fact may be used by the encoder to simulate any type of quantization characteristic.

- **Inverse DCT Transform** – Matrix with the 8×8 samples resulting from the inverse DCT applied to the dequantized DCT coefficients which corresponds to the decoded image block. These decoded sample values must be close to the ‘Original image’ values; naturally, the lower the quantization steps, the lower the error between the original and the decoded image.

- **Differences** – Matrix with the pixel level luminance differences between the original image block and the decoded image block resulting from the inverse DCT. These differences should be smaller for lower quantization steps. The application shows the values for some objective quality metrics, notably the Root Mean Squared Error (RMS), which is the square root of the Mean Squared Error (MSE), the Signal to Noise Ratio (SNR) and the Peak SNR (PSNR):

$$MSE = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (x_{ij} - y_{ij})^2 \quad (dB)$$

$$SNR = 10 \log_{10} \frac{\sum_{i=1}^M \sum_{j=1}^N x_{ij}^2}{\sum_{i=1}^M \sum_{j=1}^N (x_{ij} - y_{ij})^2} \quad (dB)$$

where x is the original sample value at position (i,j) and y is the corresponding decoded value. To obtain the PSNR (and not the SNR), the SNR must be computed using the peak value for the signal, this means 255 for samples with 8 bits:

$$PSNR = 10 \log_{10} 255^2 / \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (x_{ij} - y_{ij})^2 \quad (dB)$$

The PSNR is typically more used than the SNR since it shows a higher correlation with what would be the subjective assessment of a human user. A good objective quality metric should be as correlated as possible with the human subjective assessments.

2.2 Control Matrices

There are also some control data matrices which serve for to define some of the parameters used in the compression process:

A. Selection Filter – This filter uses two types of information, and thus two matrices, to decide how to filter the DCT coefficients:

- **Selection** – Matrix establishing the basic selection criteria for the DCT coefficients filtering according to:





The corresponding DCT coefficient is always selected for the next processing phase, this means the quantization, independently of its value (green cells).



The corresponding DCT coefficient never goes to the quantization phase, independently of its value (red cells).



The corresponding DCT coefficient goes to the quantization phase if its amplitude is equal or higher than the corresponding threshold specific in the Filter Values matrix (white cells).

- **Filter Values** – This matrix defines the selection threshold for the DCT coefficients with  in the *Selection matrix* above; this means that from the DCT coefficients with a , only those which amplitude is equal or higher than this threshold will go to the quantization phase.

B. Quantization Step – Matrix with the quantization step to be used for each DCT coefficient. The DC coefficient (upper left corner) is usually quantized with a quantization characteristic without dead-zone, contrary to the other coefficients, since it corresponds to the average energy of the image block (see Figure). Generally, the quantization steps for all coefficients may go from 1 to 255. In the H.261 mode, the DC coefficient is mandatory quantized with step 8 due to its higher subjective impact, while the AC coefficients must use integer and even quantization step values, from 2 to 62. While the JPEG standard uses different quantization steps for the various DCT coefficients, depending on the human visual sensibility to each spatial frequency (e.g. based on the Lohscheller values), recommendation H.261 uses the same quantization step for all AC coefficients since prediction residues (noise) are coded. The DCT coefficients dequantization is made as recommended for the H.261 standard.

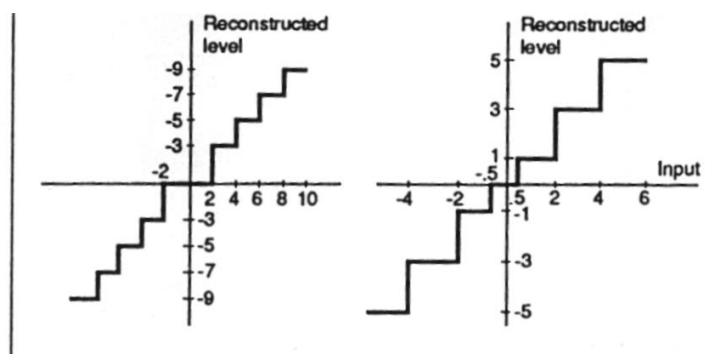


Figure 2 – Examples of quantization characteristics with dead zone.

3. INTERACTION WITH THE APPLICATION

3.1 Changing the Matrices

In order to understand what is going on, the users may change the Original Image, DCT Coefficients, Selection Filter and Quantization Step matrices. To do that, the user must press twice of the mouse left button which opens another working window with the values of the matrix to be changed. The changes may be made in the following way:

- **Change a value in the matrix** – Just change the value in the corresponding cell in the open working window.
- **Applying value from a certain DCT coefficient A to another coefficient B in zig-zag order** – The *value* is set for all positions in the zig-zag scanning order from position A to position B with the position counting starting with 0 (before closing the window, press *zig-zag fill*).
- **Random values** – The selected matrix is filled with random values having as maximum limit the value *value* (before closing the window, press *random*).
- **Lohscheller values** – The quantization step matrix is filled with half the Lohscheller values for the luminance; the Lohscheller values express the minimum human visual sensibility to the vertical and horizontal spatial frequency variations associated to each DCT coefficient.

3.2 Image Block Examples

The application includes 20 image block examples. The image blocks and associated data can be visualized using the 'Prev' and 'Next' buttons; the image block number is shown on the control box. The image examples include:

Example 1) Single black pixel at the upper left corner of a fully white block

Example 2) Line of black pixels on the top of a fully white block

Example 3) Column of black pixels at the left of a fully white block

Example 4) Line and column of black pixels on the top and at the left of a fully white block

Example 5) Black and white alternating lines

Example 6) Black and white alternating columns

Example 7) Alternate black with mixed black and white lines and columns

Example 8) Black central diagonal, downwards from left to right, in a fully white block

Example 9) Black central diagonal, upwards from left to right, in a fully white block

Example 10) Black cross made with central diagonals with a fully white block

Example 11) Black cross as example 10 but without the black central pixels

Example 12) Grey half image block with back diagonal in the other lighter grey half

Example 13) Four black central pixels in a fully white block

Example 14) Uniform grey block (128)

Example 15) Sinusoidal horizontal light-dark transition block

Example 16) Sinusoidal horizontal light-dark-light transition block

Example 17) Fast sinusoidal horizontal transitions

Example 18) Fast sinusoidal vertical transitions

Example 19) Vertical and horizontal light-dark-light transitions

Example 20) Vertical, horizontal and diagonal light-dark-light transitions

