10/1 (2012), 43–55
tmcs@math.klte.hu
http://tmcs.math.klte.hu

Teaching
Mathematics and
Computer Science

Demonstrating the feature of energy saving of transforms in secondary schools

Sándor Király

Abstract. When we are teaching the digital image formats and we are explaining the JPEG format we may get into difficulties how to explain the point and the usefulness of the discrete cosine transform (DCT) to our students. Why do we need this transform before compressing? Students probably do not know that the feature of energy saving of this transform makes the chance of good compression. In this article we show how we can demonstrate that feature of these transforms which make the chance of good compression while saving the most energy of images.

Key words and phrases: discrete cosine transform, wavelet transform, matrix multiplication.

ZDM Subject Classification: B50, C70, D40, D50, F50, F90, U70.

1. Introduction

In this article we use MS Excel to demonstrate the feature of energy saving of discrete cosine transform (DCT) and wavelet transform (WT). This software is available in secondary schools, we do not need to download and install before teaching and students know this well. Using MS Excel spreadsheet students can do DCT on constant and near constant data and they can see the result of DCT immediately. Using different sample data students can understand why we use this transform. Before demonstrating the feature of energy saving of transforms we need to explain the matrix multiplication to our students if they do not know. Knowing the transpose of a matrix is also necessary. Although the goal of this article is demonstrating the feature of energy saving of DCT and WT, we can teach the whole JPEG coding based on this article, not only the transform (DCT). Although the image format of JPEG2000 is not so current, it is worth demonstrating the Wavelet transform (WT) used for JPEG2000 as well, based on DCT.

In case of the Wavelet transform we use the Haar wavelet instead of CDF97 and LeGall52 used for JPEG2000, as the Haar wavelet is simpler and easy to understand.

2. Discrete cosine transform

DCT is a Fourier transform but it works with real numbers instead of complex numbers, and it generates the signal as not the sum of cosine and sine functions but only the sum of cosine functions. And what is very useful for us, this transform can be implemented as a matrix multiplication using a 8×8 square matrix. Represent and explain this matrix for students.

We can define the DCT with the following formula:

$$B = UAU^T$$

In this expressing the matrix U is a special matrix and U^T is the transpose of the matrix U. (Write the rows of U as the columns of U^T.) [7] The matrix A includes the value to be transformed and matrix B is the result matrix.

The matrix U is the following [9]:

$$U=\frac{1}{2}\begin{bmatrix}
\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\
\cos\frac{\pi}{16} & \cos\frac{3\pi}{16} & \cos\frac{5\pi}{16} & \cos\frac{7\pi}{16} & \cos\frac{9\pi}{16} & \cos\frac{11\pi}{16} & \cos\frac{13\pi}{16} & \cos\frac{15\pi}{16} \\
\cos\frac{2\pi}{16} & \cos\frac{6\pi}{16} & \cos\frac{10\pi}{16} & \cos\frac{14\pi}{16} & \cos\frac{18\pi}{16} & \cos\frac{22\pi}{16} & \cos\frac{26\pi}{16} & \cos\frac{30\pi}{16} \\
\cos\frac{3\pi}{16} & \cos\frac{9\pi}{16} & \cos\frac{15\pi}{16} & \cos\frac{21\pi}{16} & \cos\frac{27\pi}{16} & \cos\frac{33\pi}{16} & \cos\frac{39\pi}{16} & \cos\frac{45\pi}{16} \\
\cos\frac{4\pi}{16} & \cos\frac{12\pi}{16} & \cos\frac{20\pi}{16} & \cos\frac{28\pi}{16} & \cos\frac{36\pi}{16} & \cos\frac{44\pi}{16} & \cos\frac{52\pi}{16} & \cos\frac{60\pi}{16} \\
\cos\frac{5\pi}{16} & \cos\frac{15\pi}{16} & \cos\frac{25\pi}{16} & \cos\frac{35\pi}{16} & \cos\frac{45\pi}{16} & \cos\frac{55\pi}{16} & \cos\frac{65\pi}{16} & \cos\frac{75\pi}{16} \\
\cos\frac{6\pi}{16} & \cos\frac{18\pi}{16} & \cos\frac{30\pi}{16} & \cos\frac{42\pi}{16} & \cos\frac{54\pi}{16} & \cos\frac{66\pi}{16} & \cos\frac{78\pi}{16} & \cos\frac{90\pi}{16} \\
\cos\frac{7\pi}{16} & \cos\frac{21\pi}{16} & \cos\frac{35\pi}{16} & \cos\frac{49\pi}{16} & \cos\frac{63\pi}{16} & \cos\frac{77\pi}{16} & \cos\frac{91\pi}{16} & \cos\frac{105\pi}{16}
\end{bmatrix}$$

Figure 1. DCT matrix

Row 0 contains the constant value of $\frac{\sqrt{2}}{2}$.

In the next seven rows there are eight equally spaced points starting at $k\pi/16$ with the distance between each point equal to $2k\pi/16$ where k is the row number (as the first row is numbered as 0). We can think of creating eight equally spaced points on the interval $[k\pi/16, k\pi - k\pi/16]$, evaluating cosine at each point, and dividing the result by two.

Figure 2 shows the represtentations of the row one and two.

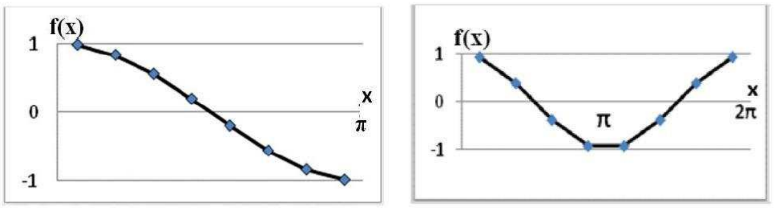


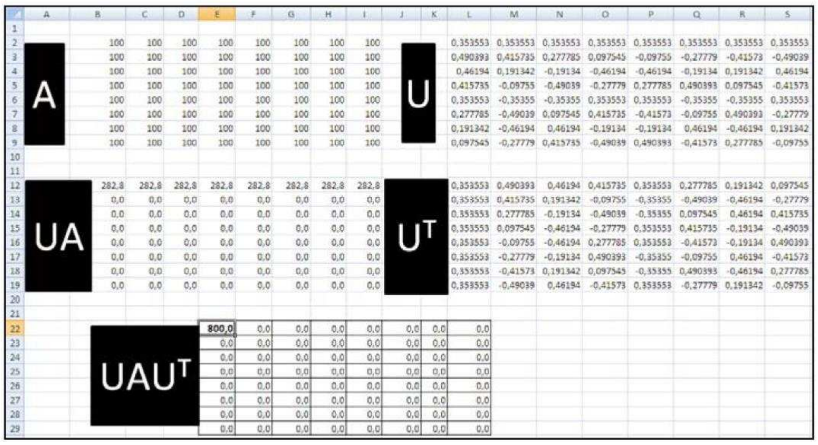
Figure 2. The first and the second row of the DCT matrix

Why do we use just this matrix? As at each row the heights of the marked points sum to zero. This means if we take any constant vector and multiply it by any of rows one through seven, the result will be zero. Moreover, if we multiply it by any of rows from one to seven with a near constant vector, the result will be either zero or values close to zero.

Doing the DCT first we multiply the block 8×8 (A) by $U.C = UA$. If the values of the matrix A are near-constant values elements in $C = UA$ will be (near) zero. Then we multiply the result C by U^T ($B = CU^T$). So if the elements in C are near constant values the elements in B will be near zero values. The DCT tends to store information about all 64 input values in a few values and takes them to the upper left hand corner of the result matrix. The remaining values are either zero or approximately zero. How can we demonstrate these facts to our students? Show them the following examples and the feature of energy saving of the DCT will be clear for them.

Consider the matrix A in Fig. 3. Do DCT on the matrix $A (B = UAU^T)$. Figure 3 shows the result.

As the old saying goes: “What I am hearing I'll forget, what I see I'll memorize and what I am doing I'll understand.” Make students do DCT on different matrixes by using Microsoft Excel. Create the matrix U and U^T and do DCT on different data (Fig. 3).



100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100

0,353553	0,353553	0,353553	0,353553	0,353553	0,353553	0,353553	0,353553
0,490393	0,415735	0,277785	0,097545	-0,09755	-0,27779	-0,41573	-0,49039
0,46194	0,191342	-0,19134	-0,46194	-0,46194	-0,19134	0,191342	0,46194
0,415735	-0,09755	-0,49039	-0,27779	0,277785	0,490393	0,097545	-0,41573
0,353553	-0,35355	-0,35355	0,353553	0,353553	-0,35355	-0,35355	0,353553
0,277785	-0,49039	0,097545	0,415735	-0,41573	-0,09755	0,490393	-0,27779
0,191342	-0,46194	0,46194	-0,19134	-0,19134	0,46194	-0,46194	0,191342
0,097545	-0,27779	0,415735	-0,49039	0,490393	-0,41573	0,277785	-0,09755

282,8	282,8	282,8	282,8	282,8	282,8	282,8	282,8
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	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	0,0	0,0
0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0

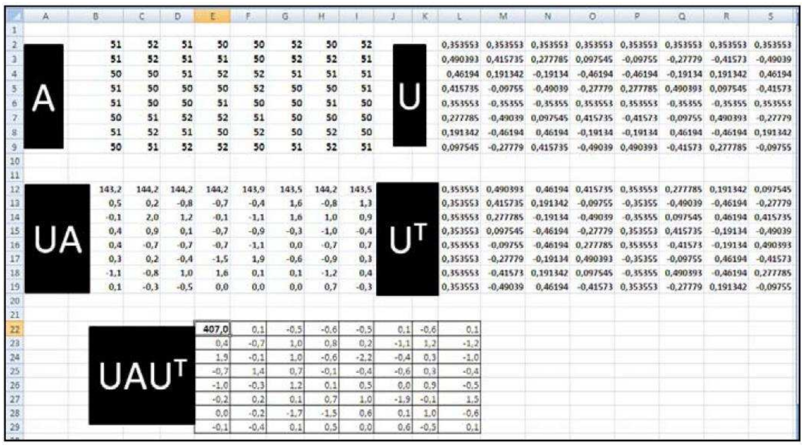
0,353553	0,490393	0,46194	0,415735	0,353553	0,277785	0,191342	0,097545
0,353553	0,415735	0,191342	-0,09755	-0,35355	-0,49039	-0,46194	-0,27779
0,353553	0,277785	-0,19134	-0,49039	-0,35355	0,097545	0,46194	0,415735
0,353553	0,097545	-0,46194	-0,27779	0,353553	0,415735	-0,19134	-0,49039
0,353553	-0,09755	-0,46194	0,277785	0,353553	-0,41573	-0,19134	0,490393
0,353553	-0,27779	-0,19134	0,490393	-0,35355	-0,09755	0,46194	-0,41573
0,353553	-0,41573	0,191342	0,097545	-0,35355	0,490393	-0,46194	0,277785
0,353553	-0,49039	0,46194	-0,41573	0,353553	-0,27779	0,191342	-0,09755

800,0	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	0,0
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	0,0	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

Figure 3. The illustration of working DCT on constant values

First we multiply U by A, then we multiply the result by U^T. For the matrix multiplication we should use the function MMULT (as an array function). If we do DCT on a matrix with constant elements we get a matrix with zero elements except one non-zero value in its upper left hand corner.

Change the elements in A for values of 50, 51 and 52 and explain the result of the DCT to students (Fig. 4).



51	52	51	50	50	52	50	52
51	52	51	51	50	52	52	51
50	50	51	52	52	51	51	51
51	50	50	50	52	50	50	51
51	50	50	51	50	50	51	50
50	51	52	52	51	50	50	50
51	52	51	50	52	50	52	50
50	51	52	52	50	51	52	51

0,353553	0,353553	0,353553	0,353553	0,353553	0,353553	0,353553	0,353553
0,490393	0,415735	0,277785	0,097545	-0,09755	-0,27779	-0,41573	-0,49039
0,46194	0,191342	-0,19134	-0,46194	-0,46194	-0,19134	0,191342	0,46194
0,415735	-0,09755	-0,49039	-0,27779	0,277785	0,490393	0,097545	-0,41573
0,353553	-0,35355	-0,35355	0,353553	0,353553	-0,35355	-0,35355	0,353553
0,277785	-0,49039	0,097545	0,415735	-0,41573	-0,09755	0,490393	-0,27779
0,191342	-0,46194	0,46194	-0,19134	-0,19134	0,46194	-0,46194	0,191342
0,097545	-0,27779	0,415735	-0,49039	0,490393	-0,41573	0,277785	-0,09755

143,2	144,2	144,2	144,2	143,9	143,5	144,2	143,5
0,5	0,2	-0,8	-0,7	-0,4	1,6	-0,8	1,3
-0,1	2,0	1,2	-0,1	-1,1	1,6	1,0	0,9
0,4	0,9	0,1	-0,7	-0,9	-0,3	-1,0	-0,4
0,4	-0,7	-0,7	-0,7	-1,1	0,0	-0,7	0,7
0,3	0,2	-0,4	-1,5	1,9	-0,6	-0,9	0,3
-1,1	-0,8	1,0	1,6	0,1	0,1	-1,2	0,4
0,1	-0,3	-0,5	0,0	0,0	0,0	0,7	-0,3

0,353553	0,490393	0,46194	0,415735	0,353553	0,277785	0,191342	0,097545
0,353553	0,415735	0,191342	-0,09755	-0,35355	-0,49039	-0,46194	-0,27779
0,353553	0,277785	-0,19134	-0,49039	-0,35355	0,097545	0,46194	0,415735
0,353553	0,097545	-0,46194	-0,27779	0,353553	0,415735	-0,19134	-0,49039
0,353553	-0,09755	-0,46194	0,277785	0,353553	-0,41573	-0,19134	0,490393
0,353553	-0,27779	-0,19134	0,490393	-0,35355	-0,09755	0,46194	-0,41573
0,353553	-0,41573	0,191342	0,097545	-0,35355	0,490393	-0,46194	0,277785
0,353553	-0,49039	0,46194	-0,41573	0,353553	-0,27779	0,191342	-0,09755

407,0	0,1	-0,5	-0,6	-0,5	0,1	-0,6	0,1
0,4	-0,7	1,0	0,8	0,2	-1,1	1,2	-1,2
1,9	-0,1	1,0	-0,6	-2,2	-0,4	0,3	-1,0
-0,7	1,4	0,7	-0,1	-0,4	-0,6	0,3	-0,4
-1,0	-0,3	1,2	0,1	0,5	0,0	0,9	-0,5
-0,2	0,2	0,1	0,7	1,0	-1,9	-0,1	1,5
0,0	-0,2	-1,7	-1,5	0,6	0,1	1,0	-0,6
-0,1	-0,4	0,1	0,5	0,0	0,6	-0,5	0,1

Figure 4. The illustration of working DCT on near constant values

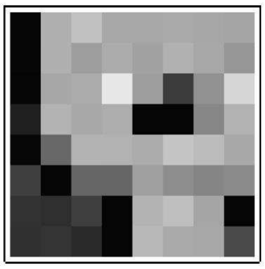
In the upper left hand corner there is the worth value, the others are close to zero. If we remove these values, we need to store significantly less amount of data.

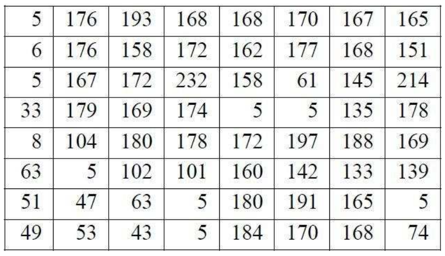
How can we use this feature of DCT in digital image processing? Why can we leave near zero values? Consider matrix A as an image or part of an image. The elements in A are the intesity values of the pixels of the image. Using DCT we will not get less data, but what we get, it is the flavour of the spectrum of the image. The human eye is less sensitive for the high frequency components than low ones. Leaving the 50% of high frequeny components after the inverse DCT only the 5% original data will be lost. (DCT is an inverse transform so the inverse DCT can transform the origin values from the frequency values. [2]

How do we use DCT in JPEG process? The pixel instensity values in the original RGB bitmap file will be converted to the YCbCr color space, then we partition the image into block of size 8×8 pixels. The next part of the processing is to subtract 128 from each pixel intensity in each block. This step centers the intensities about the value 0. [10]

Then we do DCT on each 8×8 block, and the following step is the quantization before Huffman coding. After quantization the elements near zero will be converted to zero and other elements will be shrunk so that their values are closer to zero. Then all quantized values will be rounded to integers. Quantization makes the JPEG algorithm an example of lossy compression. The DCT step is completely invertible, as $U^T U = 1$ (unit matrix) that is $U^T = U^{-1}$. (This is a special feature of this matrix.) It is that we applied the DCT to each block D by computing $E = UDU^T$. As we can recover D by the computation $D = U^T EU$. When we "shrink" values, it is possible to recover them. However, converting small values to 0 and rounding all quantized values are not reversable steps. We lose the ability to recover the original image forever. We perform quantization in order to obtain integer values and to convert a large number of the values to 0. The Huffman coding algorithm will be much more effective with quantized data and the hope is that when we view the compressed image, do not give up too much resolution. For applications such as web browsing, the resolution lost in order to gain storage space/transfer speed is acceptable. To practise we can do the DCT on a small image or a part of an image then do quantization on them.

Let's see the following image part and intensity values (Fig. 5). [1]





5	176	193	168	168	170	167	165
6	176	158	172	162	177	168	151
5	167	172	232	158	61	145	214
33	179	169	174	5	5	135	178
8	104	180	178	172	197	188	169
63	5	102	101	160	142	133	139
51	47	63	5	180	191	165	5
49	53	43	5	184	170	168	74

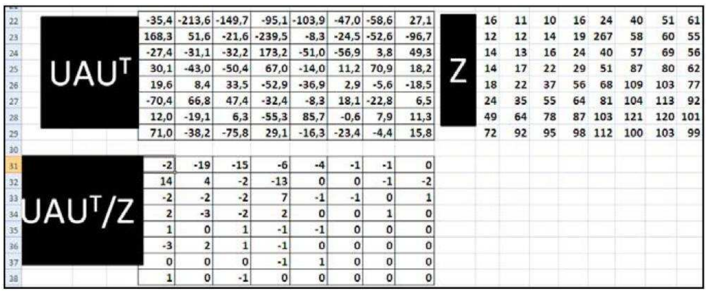
Figure 5. Part of an image and the intensity values

Subtract 128 from each value (Fig. 6):

-123	48	65	40	40	42	39	37
-122	48	30	44	34	49	40	23
-123	39	44	104	30	-67	17	86
-95	51	41	46	-123	-122	7	50
-120	-24	52	50	44	69	60	41
-65	-123	-26	-27	32	14	5	11
-77	-81	-65	-123	52	63	37	-123
-79	-75	-85	-123	56	42	40	-54

Figure 6. Intensity values after subtraction 128

Type these values into our Excel table in matrix A. The result of the DCT is the following matrix (Fig. 7):



22	UAU^T	-35,4	-213,6	-149,7	-95,1	-103,9	-47,0	-58,6	27,1	Z	16	11	10	16	24	40	51	61
23		168,3	51,6	-21,6	-239,5	-8,3	-24,5	-52,6	-96,7		12	12	14	19	267	58	60	55
24		-27,4	-31,1	-32,2	173,2	-51,0	-56,9	3,8	49,3		14	13	16	24	40	57	69	56
25		30,1	-43,0	-50,4	67,0	-14,0	11,2	70,9	18,2		14	17	22	29	51	87	80	62
26		19,6	8,4	33,5	-52,9	-36,9	2,9	-5,6	-18,5		18	22	37	56	68	109	103	77
27		-70,4	66,8	47,4	-32,4	-8,3	18,1	-22,8	6,5		24	35	55	64	81	104	113	92
28		12,0	-19,1	6,3	-55,3	85,7	-0,6	7,9	11,3		49	64	78	87	103	121	120	101
29		71,0	-38,2	-75,8	29,1	-16,3	-23,4	-4,4	15,8		72	92	95	98	112	100	103	99
30																		
31	UAU^T/Z	-2	-19	-15	-6	-4	-1	-1	0									
32		14	4	-2	-13	0	0	-1	-2									
33		-2	-2	-2	7	-1	-1	0	1									
34		2	-3	-2	2	0	0	1	0									
35		1	0	1	-1	-1	0	0	0									
36		-3	2	1	-1	0	0	0	0									
37		0	0	0	-1	1	0	0	0									
38		1	0	-1	0	0	0	0	0									

Figure 7. DCT and quantization

Then do quantization on the result matrix by matrix Z. [4]. It means that we divide the elements in $B = UAU^T$ by the elements in Z and round them to integer. (We create a new matrix Q whose elements are $q_{jk} = R(b_{jk}/z_{jk})$ for $j, k = 0, \ldots, 7$. Here, $R(.)$ is the rounding function, b_{jk} and z_{jk} are the elements of matrices B and Z.) Note that the values are the largest in the lower right corner of Z. These divides should produce values close to zero so that the rounding function will convert the quotient to zero. [5] [12] [3]

The energy of the image centered in the upper left hand side corner of the transformed and quantized image. Before Huffman coding we could transform the image so that most energy of the image has been saved and we do not need most data, we can remove them or we can use it for compressing. In the regular JPEG standard process the quantized values will be coded by an advanced version of Huffman coding.

Make students draw an arbitrary image and ask them to save the image as an BMP file then ask them to save it as an JPEG file. After that they have to compare the size of these files. The algorithm above resulted the different sizes.

3. Wavelet transform

The main difference between DCT and WT is their base functions. As opposed to DCT the base functions of WT are not cosine waves but wavelets (small waves). These are the compressed and stretched versions of the base wavelet (named mother wavelet). Wavelets are not predefined. The key to the success of the transform is the selection of the adequate mother wavelet. Our goal is only to demonstrate the feature of energy saving of wavelet transform so the mother wavelet will be the Haar wavelet as it is easy to understand.

Before defining the Haar wavelet let's represent the necessity of the transform with the following example. We have eight numbers 100, 200, 44, 50, 20, 20, 3, 1 (those could be greyscale intensities of an image) and we want to send an approximation of the list to a network station but we are allowed to send only four values from the eight ones. What values would we send that represent an approximation of the eight given values the most? One solution can be that we take the eight numbers, two at a time, and average them. So we send 150 $((100 + 200)/2)$, $47((44 + 50)/2)$, $20((20 + 20)/2)$ and $2((1 + 3)/2)$. These four numbers represent the given eight numbers well. At the host they can not produce the original eight values from the four values (150, 47, 20, 2). Supposing that we are allowed to send an additional four values what values would we send? Send

50, 3, 0 and −1. They are the directed distances from the pair-wise averages to the second number in each pair: the half of 200 − 100 is 50, the half of 50 − 44 is 3, the half of 20 − 20 is 0 and the half of 1 − 3 is −1. So with the lists $(150, 47, 20, 2)$ and $(50, 3, 0, -1)$ we can completely reconstruct the original list $(100, 200, 44, 50, 20, 20, 3, 1)$.

For a given a and b we perform the following transform:

$$(a, b) \Rightarrow \left(\frac{b+a}{2}, \frac{b-a}{2}\right)$$

The beginning of output of the transform is the average of the input, the second part of the output of the transform is the half of the difference of the input. The question is: why would we send a list $(150, 47, 20, 2, 50, 3, 0, -1)$ instead of the original list $(100, 200, 44, 50, 20, 20, 3, 1)$? First of all, the differences in the transformed list tell us about the trends in the data: big differences indicate large jumps between values while small values tell us that there is relatively little change in that portion of the input. Besides in case of lossy compression the small differences can be converted to zero and in this way we can improve the efficiency of the coder. In this case instead of eight numbers, sending the list $(150, 47, 20, 2, 50, 3, 0, 0)$ would be enough. The receiver would get the list $(150 - 50, 150 + 50, 47 - 2, 47 + 2, 20 - 0, 20 + 0, 2 - 0, 2 + 0) = (100, 200, 45, 49, 20, 20, 2, 2)$ which is very similar to the original list $(100, 200, 44, 50, 20, 20, 3, 1)$.

If we consider the eight numbers as a vector of eight elements we can perform the previous transform by a matrix multiplication. (If we have even piece of numbers.) Based on this we can define the Haar wavelet transform (HWT) as the following $N \times N$ matrix W_N (Fig. 8) [6].

$$W_N = \begin{bmatrix} \sqrt{2}/2 & \sqrt{2}/2 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \sqrt{2}/2 & \sqrt{2}/2 & & 0 & 0 \\ \vdots & & & & \ddots & & \vdots \\ 0 & 0 & 0 & 0 & \dots & \sqrt{2}/2 & \sqrt{2}/2 \\ -\sqrt{2}/2 & \sqrt{2}/2 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & -\sqrt{2}/2 & \sqrt{2}/2 & & 0 & 0 \\ \vdots & & & & \ddots & & \vdots \\ 0 & 0 & 0 & 0 & \dots & -\sqrt{2}/2 & \sqrt{2}/2 \end{bmatrix}$$

Figure 8. The HWT matrix

The first $N/2$ rows of the HWT are the weighted average of the input list taken two at a time. The weight factor is $\sqrt{2}$.

The last $N/2$ row of the HWT is the weighted difference of the input list taken two at a time. The weight factor is also $\sqrt{2}$.

We define the Haar filter as the pair of the non-zero values in the first row of the matrix. So the Haar filter is: $h = (h_0, h_1) = \big(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\big)$.

As this filter averages the pairs of numbers, it is a lowpass filter. (It lets the low frequencies through.) Emphasize to our students that the sum of the filter values is $\sqrt{2}$. We call the filter that is used to build the bottom half of the HWT is higpass filter. (It lets the high frequencies through.) In this case the filter is:

$$g = (g_0, g_1) = \Big(-\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\Big).$$

Figure 9 shows the Haar wavelet [11].

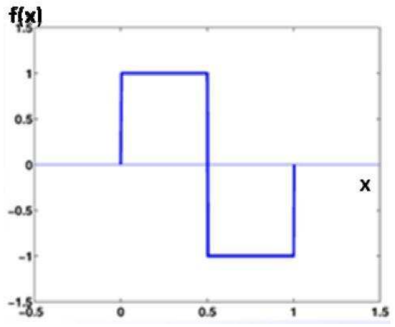
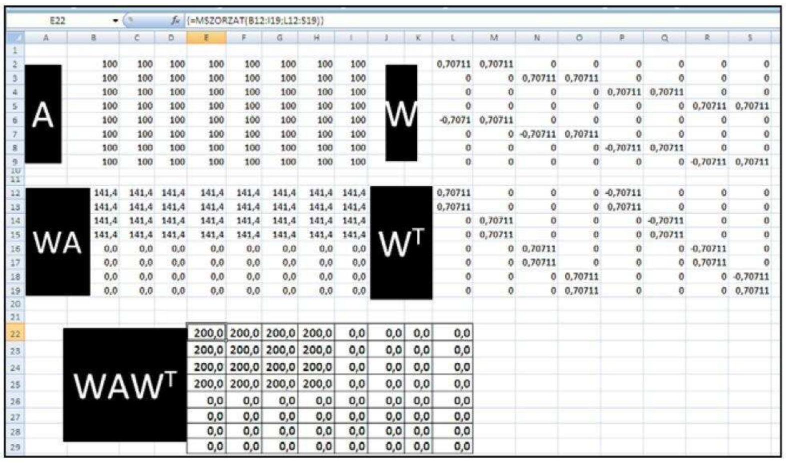


Figure 9. The Haar mother-wavelet

This filter processes data exactly opposite of lowpass filter. If two numbers are near in value, it will return a value near zero. If two numbers are (near) opposites of each other, the highpass filter will return a weighted version of one of the two numbers. Use our Excel spreadsheet again (Fig. 10).

Substitute the elements in W with elements in the Haar wavelet matrix. Demonstrate to our students that the transform collects the energy in the upper left hand side of the result matrix. The result of the first multiplication is a matrix which contains non-zero values in the first half of rows.

Do HWT on matrix A with 50, 51, 52 numbers. The result is the same (Fig. 11).

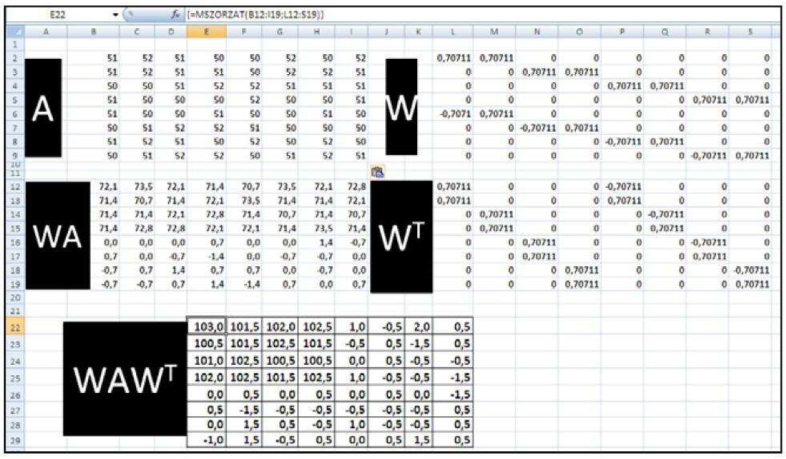


E22 {=MSZORZAT(B12:I19;L12:S19)}

	B	C	D	E	F	G	H	I		L	M	N	O	P	Q	R	S
2	100	100	100	100	100	100	100	100		0,70711	0,70711	0	0	0	0	0	0
3	100	100	100	100	100	100	100	100		0	0	0,70711	0,70711	0	0	0	0
4	100	100	100	100	100	100	100	100		0	0	0	0	0,70711	0,70711	0	0
5	100	100	100	100	100	100	100	100		0	0	0	0	0	0	0,70711	0,70711
6	100	100	100	100	100	100	100	100		-0,7071	0,70711	0	0	0	0	0	0
7	100	100	100	100	100	100	100	100		0	0	-0,70711	0,70711	0	0	0	0
8	100	100	100	100	100	100	100	100		0	0	0	0	-0,70711	0,70711	0	0
9	100	100	100	100	100	100	100	100		0	0	0	0	0	0	-0,70711	0,70711
12	141,4	141,4	141,4	141,4	141,4	141,4	141,4	141,4		0,70711	0	0	0	-0,70711	0	0	0
13	141,4	141,4	141,4	141,4	141,4	141,4	141,4	141,4		0,70711	0	0	0	0,70711	0	0	0
14	141,4	141,4	141,4	141,4	141,4	141,4	141,4	141,4		0	0,70711	0	0	0	-0,70711	0	0
15	141,4	141,4	141,4	141,4	141,4	141,4	141,4	141,4		0	0,70711	0	0	0	0,70711	0	0
16	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0		0	0	0,70711	0	0	0	-0,70711	0
17	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0		0	0	0,70711	0	0	0	0,70711	0
18	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0		0	0	0	0,70711	0	0	0	-0,70711
19	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0		0	0	0	0,70711	0	0	0	0,70711

	E	F	G	H	I	J	K	L
22	200,0	200,0	200,0	200,0	0,0	0,0	0,0	0,0
23	200,0	200,0	200,0	200,0	0,0	0,0	0,0	0,0
24	200,0	200,0	200,0	200,0	0,0	0,0	0,0	0,0
25	200,0	200,0	200,0	200,0	0,0	0,0	0,0	0,0
26	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
27	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
28	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
29	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0

Figure 10. The wavelet transform using the Haar wavelet



E22 {=MSZORZAT(B12:I19;L12:S19)}

	B	C	D	E	F	G	H	I		L	M	N	O	P	Q	R	S
2	51	52	51	50	50	52	50	52		0,70711	0,70711	0	0	0	0	0	0
3	51	52	51	51	50	52	52	51		0	0	0,70711	0,70711	0	0	0	0
4	50	50	51	52	52	51	51	51		0	0	0	0	0,70711	0,70711	0	0
5	51	50	50	50	52	50	50	51		0	0	0	0	0	0	0,70711	0,70711
6	51	50	50	51	50	50	51	50		-0,7071	0,70711	0	0	0	0	0	0
7	50	51	52	52	51	50	50	50		0	0	-0,70711	0,70711	0	0	0	0
8	51	52	51	50	52	50	52	50		0	0	0	0	-0,70711	0,70711	0	0
9	50	51	52	52	50	51	52	51		0	0	0	0	0	0	-0,70711	0,70711
12	72,1	73,5	72,1	71,4	70,7	73,5	72,1	72,8		0,70711	0	0	0	-0,70711	0	0	0
13	71,4	70,7	71,4	72,1	73,5	71,4	71,4	72,1		0,70711	0	0	0	0,70711	0	0	0
14	71,4	71,4	72,1	72,8	71,4	70,7	71,4	70,7		0	0,70711	0	0	0	-0,70711	0	0
15	71,4	72,8	72,8	72,1	72,1	71,4	73,5	71,4		0	0,70711	0	0	0	0,70711	0	0
16	0,0	0,0	0,0	0,7	0,0	0,0	1,4	-0,7		0	0	0,70711	0	0	0	-0,70711	0
17	0,7	0,0	-0,7	-1,4	0,0	-0,7	-0,7	0,0		0	0	0,70711	0	0	0	0,70711	0
18	-0,7	0,7	1,4	0,7	0,7	0,0	-0,7	0,0		0	0	0	0,70711	0	0	0	-0,70711
19	-0,7	-0,7	0,7	1,4	-1,4	0,7	0,0	0,7		0	0	0	0,70711	0	0	0	0,70711

	E	F	G	H	I	J	K	L
22	103,0	101,5	102,0	102,5	1,0	-0,5	2,0	0,5
23	100,5	101,5	102,5	101,5	-0,5	0,5	-1,5	0,5
24	101,0	102,5	100,5	100,5	0,0	0,5	-0,5	-0,5
25	102,0	102,5	101,5	102,5	1,0	-0,5	-0,5	-1,5
26	0,0	0,5	0,0	0,5	0,0	0,5	0,0	-1,5
27	0,5	-1,5	-0,5	-0,5	-0,5	-0,5	-0,5	0,5
28	0,0	1,5	0,5	-0,5	1,0	-0,5	-0,5	0,5
29	-1,0	1,5	-0,5	0,5	0,0	0,5	1,5	0,5

Figure 11. Do WT on a near constant matrix A

Show our students a real transformed image. Figure 12 illustrates the whole transform (WAW^T).

We can perform the transform many times with the image in the upper left hand corner (see Figure 13). The lower left-hand corner holds information about horizontal in the image: large values indicate a large horizontal change as we

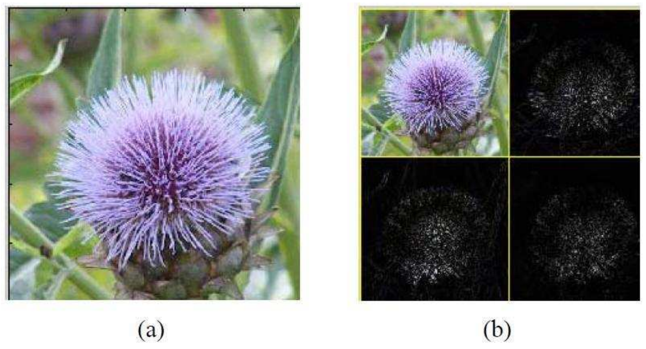


Figure 12. The original image in (a) and the result of the WT in (b)

move down the image and small values indicate little horizontal change. The lower right-hand block is different across both columns and rows and the result is a bit harder to see. It turns out that this product measures changes along ±45-degree lines. The upper right-hand corner holds information about vertical in the image: large values indicate a large vertical change as we move across the image and small values indicate little vertical change.

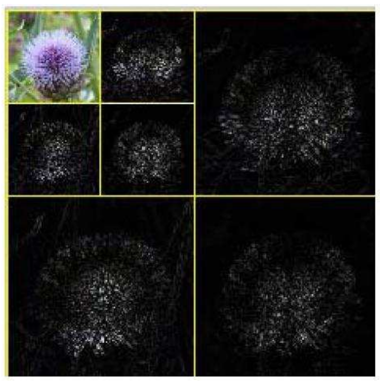


Figure 13. Two iterations of the WT

The iterated HWT conserves the energy of a digital image. Figure 14 shows the energy distribution for the original image (curve C), the curve B shows the one iteration of the HWT and the curve A shows the energy distribution of the three iterations of the HWT. [8] [13] The horizontal scale is the number of the pixels. For a given pixel value p, the height represents the percentage of energy stored in the largest p pixels of the image. Emphasize to our students that the HWT gets to 1 (100% of the energy) much faster than the original image and the iterated HWT is much better than either the HWT or the original image (Fig. 14).

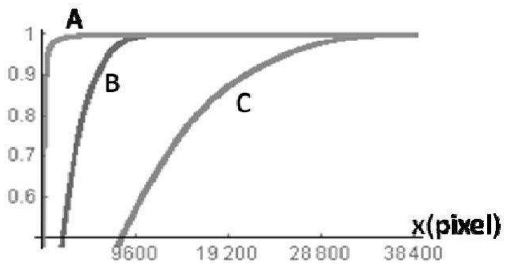


Figure 14. Illustrating the feature of energy saving of HWT

4. Summary

The main aim of this paper was to give ideas of demonstrating the feature of energy saving of DCT and WT to secondary students. We need this when we are teaching digital image formats. We presumed that students know the matrix multiplication. In case of DCT we review the transform by using DCT matrix. The Excel spreadsheets illustrate the feature of energy saving of DCT very spectacularly in cases of constant and almost constant values as well. After doing DCT on an image, the important parts for human eye are preserved but the image will be suitable for compressing so huge amount of data will be unnecessary.

The wavelet transform has the same feature and we use the same Excel spreadsheet to demonstrate this. We only have to replace the DCT matrix to the Haar wavelet matrix. Not only the spreadsheet but the sample images also illustrate this feature of the HWT very spectacularly. The spreadsheet used in this article can be changeable by students so they can realize how the transforms work in cases of different images. In case of HWT they can perform more and more iterations and they can follow the working of HWT and the feature of energy saving.

References

[1] Data compression, 2011, `http://www-cs-faculty.stanford.edu/ eroberts/courses/soco/projects/2000-01/data-compression/lossy/jpeg/dct.htm`.

[2] R. C. Gonzales and R. E. Woods, *Digital Image Processing*, Third Edition, Pearson Education Inc., 2008.

[3] M. Grgic’, S. Grgic’ and B. Zovko-Cihlar, DCTlab: educational software for still image compression and its application in a digital television course, 2011, `http://www.manchesteruniversitypress.co.uk/uploads/docs/380187.pdf`.

[4] JPEG, 2011, `http://computervision.wikia.com/wiki/JPEG`.

[5] JPEG Tutorial, 2011, `http://www.johnloomis.org/ece563/notes/compression/jpeg/tutorial/jpegtut1.html`.

[6] L. Lade and B. Westergen, *Mathematics Handbook for Science and Engineering*, 5th Edition, Springer, 2004.

[7] J. Gy. Obádovics, *Matematika I., II.*, Scolar kiadó, Budapest, 1994.

[8] R. Polikar, The Wavelet Tutorial, 2011, `http://users.rowan.edu/ polikar/WAVELETS/WTpart3.html`.

[9] K. R. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages and Applications*, Academic Press, San Diego, 1990.

[10] S. W. Smith, *The Scientist and Engineer’s Guide to Digital Signal Processing*, 2nd Edition, California Technical Publishing, 2003.

[11] G. Strang and T. Nguyen, *Wavelets and Filter Banks*, Wellesley–Cambridge Press, 1997.

[12] A. B. Watson, Image Compression Using the Discrete Cosine Transform, `http://www.mathematica-journal.com/issue/v4i1/article/81-88_Watson.mj.pdf`.

[13] Wavelet toolbox, 2011, `http://www.mathworks.com/products/wavelet/`.

SÁNDOR KIRÁLY
FAISKOLA U. 24/B
3300 EGER
HUNGARY

E-mail: `ksanyi@nejanet.hu`

(Received March, 2011)