Visualizing 2D DCT

1D DCT basis set for R^8 can be visulized as

$$D = \begin{bmatrix} \vert & \vert & \vert & \vert & \vert & \vert & \vert & \vert \\ b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & b_7 & b_8 \\ \vert & \vert & \vert & \vert & \vert & \vert & \vert & \vert \end{bmatrix}.$$

For $R^{8\times8}$ 2D space , we need 64 $R^{8\times8}$ bases. These bases can be created using outerproduct of 1D bases.

The bases can be thought of arranged in the form

$$\begin{bmatrix} b_1b_1^T & . & . & b_1b_4^T & . & . & . & b_1b_8^T \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ b_4b_1^T & . & . & b_4b_4^T & . & . & . & b_4b_8^T \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ b_8b_1^T & . & . & b_8b_4^T & . & . & . & b_8b_8^T \end{bmatrix}$$

for convenience

Note that each entry is 8x8 matrix . In total, there are 64 base matrices

For 8x8 2D data we find 2D DCT coefficients by dotproducting the the data with 64 bases. The result of dotproduct is

another 8x8 matrix of DCT coefficients.

Let $x = \begin{bmatrix} x_{11} & . & . & x_{14} & . & . & . & x_{18} \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ x_{41} & . & . & x_{44} & . & . & . & x_{48} \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ x_{81} & . & . & x_{84} & . & . & . & x_{88} \end{bmatrix}$ Then DCT of x is

$$X = \begin{bmatrix} \langle b_1 b_1^T, x \rangle & . & . & \langle b_1 b_4^T, x \rangle & . & . & . & \langle b_1 b_8^T, x \rangle \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ \langle b_4 b_1^T, x \rangle & . & . & \langle b_4 b_4^T, x \rangle & . & . & . & \langle b_4 b_8^T, x \rangle \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ \langle b_8 b_1^T, x \rangle & . & . & \langle b_8 b_4^T, x \rangle & . & . & . & \langle b_8 b_8^T, x \rangle \end{bmatrix}$$

Note that both x and X are 8x8 matrices

16. For $R^{8\times 8}$, Create DCT basis set vector (here it is 2D) , DCT_3_4 from 1D basis set for R^8

(basis set matrix created by outer producting 3rd and 4th 1D basis set vectors in order)

```
B=idct(eye(8)); b3=B(:,3); b4=B(:,4);
DCT_3_4=b3*b4';
disp(num2str(DCT_3_4,'%.3f '))
```

```
0.192 -0.045 -0.227 -0.128  0.128  0.227  0.045 -0.192
0.080 -0.019 -0.094 -0.053  0.053  0.094  0.019 -0.080
```

```
-0.080  0.019  0.094  0.053 -0.053 -0.094 -0.019  0.080
-0.192  0.045  0.227  0.128 -0.128 -0.227 -0.045  0.192
-0.192  0.045  0.227  0.128 -0.128 -0.227 -0.045  0.192
-0.080  0.019  0.094  0.053 -0.053 -0.094 -0.019  0.080
 0.080 -0.019 -0.094 -0.053  0.053  0.094  0.019 -0.080
 0.192 -0.045 -0.227 -0.128  0.128  0.227  0.045 -0.192
```

17. Take M=randi([0 255],8,8) matrix find coefficient corresponding to DCT_3_4

```
rng("default")
M=randi([0 255],8,8);
B=idct(eye(8)); b3=B(:,3); b4=B(:,4);
DCT_3_4=b3*b4';
Coef_3_4=dot( M(:), DCT_3_4(:) ); %Vectorise and
take dot product
disp(num2str(Coef_3_4,'%2.3f '))
```

```
-57.858
```

18. Find all 2D DCT coefficients for M=randi([0 255],8,8)

```
rng("default")
M=randi([0 255],8,8);
D=dct2(M);
disp(num2str(D,'%2.3f ')); % verify DCT_3_4
coefficient
```

```
1116.250  102.814   45.745  -99.648  -96.250  -24.298   39.230  -26.612
 135.093   58.601   80.169  -21.028  -10.198  109.021  -72.408  -87.192
 -17.807  -30.900   99.582  -57.858    3.250   53.300   64.996  -31.462
 -21.003    9.791   43.952   22.132   90.329 -105.511 -189.789  -45.653
 142.500    2.407  136.573   34.842    1.000    5.638  -16.522   80.689
  60.065   -0.230 -112.897   43.329   30.101 -102.171    7.744    4.221
-211.538 -168.923   23.746   21.645    0.007   15.441   98.168  -80.516
  36.699  -63.852  -47.301 -135.451   95.246  -15.646   -0.175  -26.563
```

19. Find all 2D DCT coefficients for M=magic(8). what you infer

```
rng("default")
M=magic(8);
D=dct2(M);
disp(num2str(D,'%2.3f '));
```

```
260.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.000    0.000    0.000   13.990    0.000    0.000    0.000
  0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.000    0.000    0.000  103.888    0.000    0.000    0.000
  0.000    1.749    0.000   12.986    0.000   12.783    0.000    0.945
  0.000    0.000    0.000    0.000  102.266    0.000    0.000    0.000
  0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.000    0.000    0.000    7.561    0.000    0.000    0.000
```

```
Message='DCT allows data compression in transformed domain';
disp(Message)
```

```
DCT allows data compression in transformed domain
```

20. DCT for image compression (demo without Huffman coding).

Demonstrate that Transform in Transform domain, many coefficients are near zero.

and by cutting of those coefficients do not lead to percetible change in Reconstructed Image.

```
rng("default")
M=magic(8);
disp(num2str(M,'%2.0f '))
```

```
64  2  3 61 60  6  7 57
 9 55 54 12 13 51 50 16
17 47 46 20 21 43 42 24
40 26 27 37 36 30 31 33
32 34 35 29 28 38 39 25
41 23 22 44 45 19 18 48
49 15 14 52 53 11 10 56
 8 58 59  5  4 62 63  1
```

```
D=dct2(M);
threshold = 0.0001;
% Set values to zero if their absolute value is
less than the threshold
D(abs(D) < threshold) = 0;
% Display the modified matrix
disp(num2str(D,'%2.1f '));
```

```
260.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
  0.0   0.0   0.0   0.0  14.0   0.0   0.0   0.0
  0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
  0.0   0.0   0.0   0.0 103.9   0.0   0.0   0.0
  0.0   1.7   0.0  13.0   0.0  12.8   0.0   0.9
  0.0   0.0   0.0   0.0 102.3   0.0   0.0   0.0
  0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
  0.0   0.0   0.0   0.0   7.6   0.0   0.0   0.0
```

```
D_r=idct2(D);
disp(num2str(D_r,'%2.0f '));
```

```
64  2  3 61 60  6  7 57
 9 55 54 12 13 51 50 16
17 47 46 20 21 43 42 24
40 26 27 37 36 30 31 33
32 34 35 29 28 38 39 25
41 23 22 44 45 19 18 48
49 15 14 52 53 11 10 56
 8 58 59  5  4 62 63  1
```

Projects

Class room Project topics

1. Image Compression
2. Watermarking
3. Steganography
4. Audio watermarking

References for doing 2 hour projects.

1. Introduction to Steganography with MATLAB https://medium.com/@lhagenau/introduction-to-steganography-with-matlab-d8d2861a3686
2. https://github.com/AhmedAbdElghany97/LSB-Steganography Matlab code
3. Image Steganography using RSA and Hash LSB https://in.mathworks.com/matlabcentral/fileexchange/127893-image-steganography-using-rsa-and-hash-lsb/
4. https://github.com/SaiManojGubbala/Image-Steganography Good Matlab code
5. Audio_Steganography_in_MATLAB https://github.com/singhishita/Audio_Steganography_in_MATLAB
6. https://cft.vanderbilt.edu/guides-sub-pages/teaching-in-the-age-of-ai/