Lab 11 report Shuchen Zhang

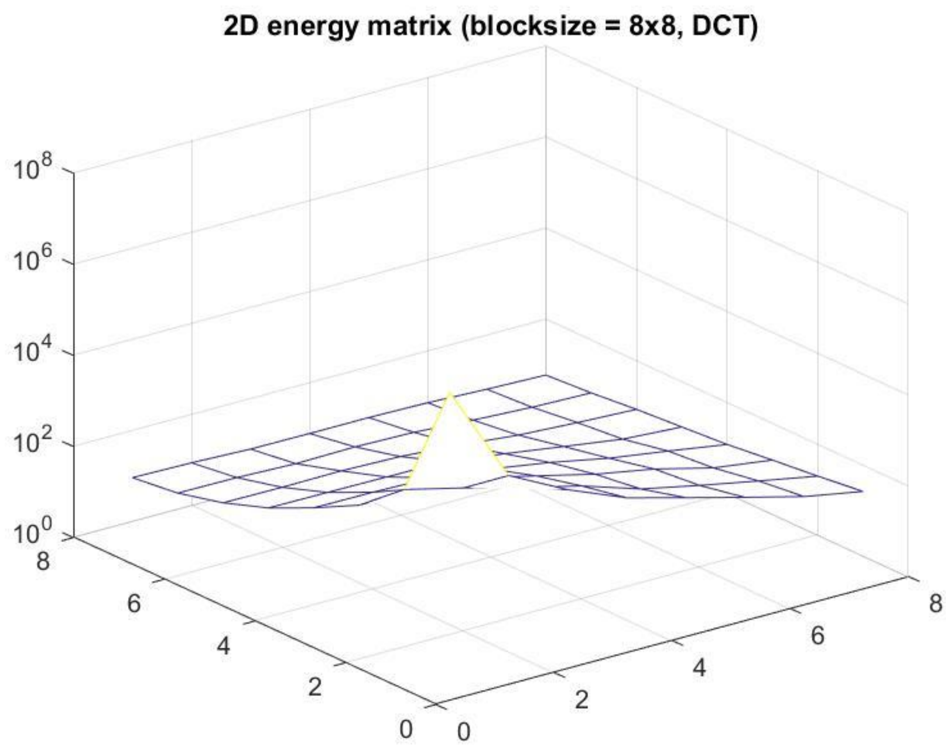
Introduction:

In this lab, we explore a few lossy image compression methods which are belong to transform coding. Transform coding take the advantage of energy compaction properties of Transform and find the optimal number of coefficients to keep so that the compressed image perceptually perfect. Two methods are implemented: DCT (discrete cosine transform) and KLT (Karhunen-Loeve Transform). Also, we examine the basic properties of JPEG image compression.

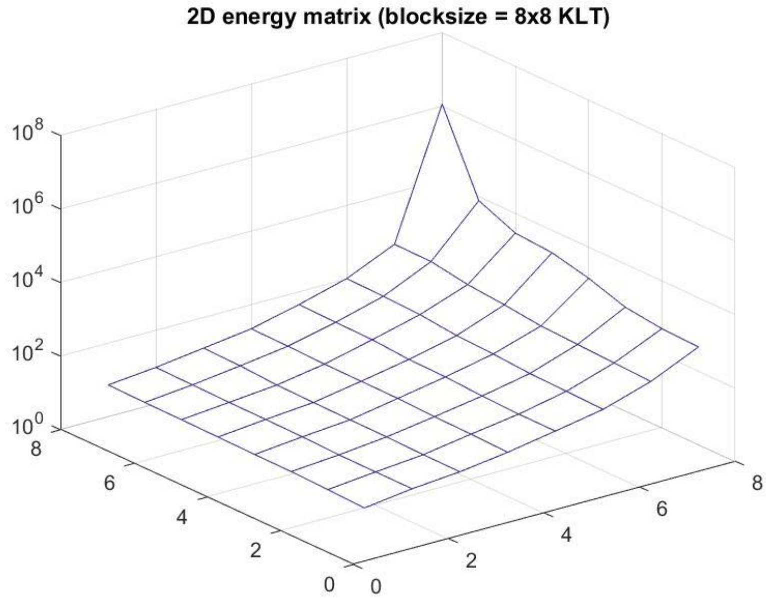
Results:

Q1.

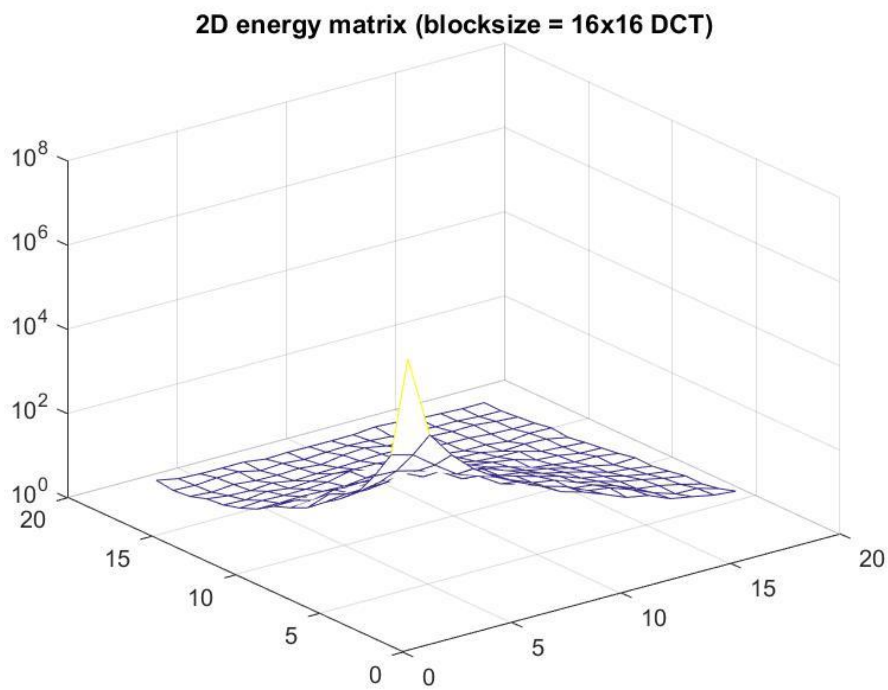
Plot of ‘energy’ matrix (block size = 8x8, DCT):



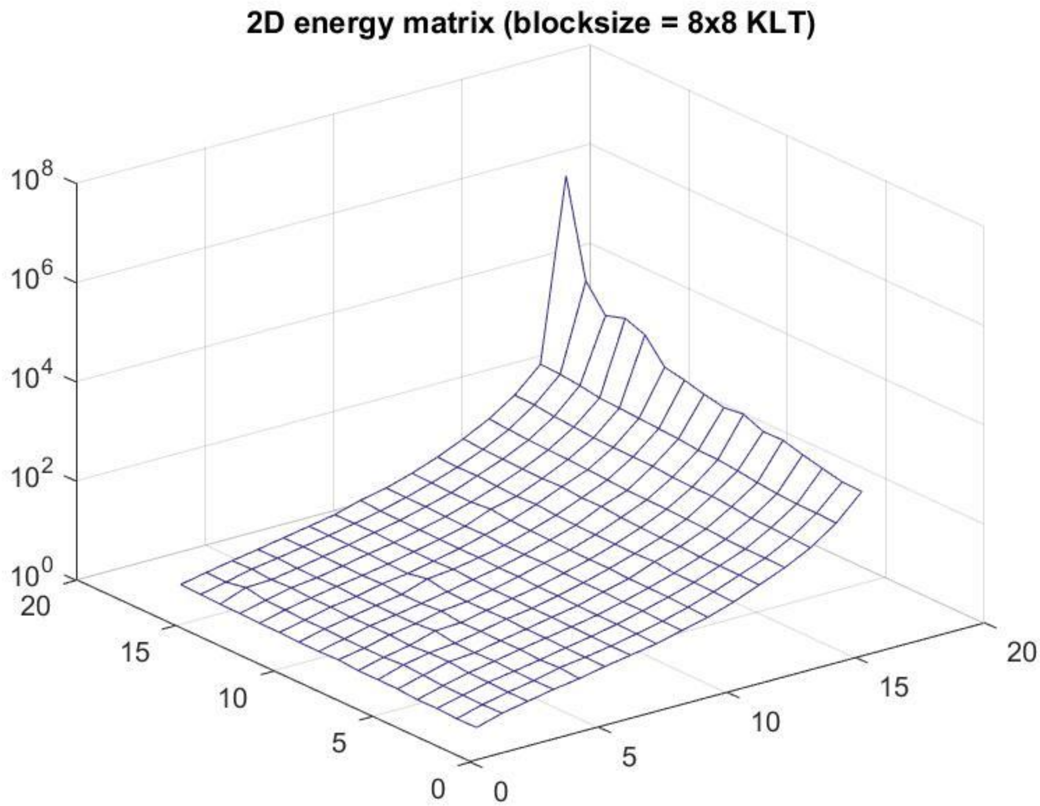
Plot of ‘energy’ matrix (block size = 8x8, KLT):



Plot of ‘energy’ matrix (block size = 16x16, DCT):



Plot of ‘energy’ matrix (block size = 16x16, KLT):



(i) 8x8 block, 10 coefficients kept:

a. Average block energy matrix:

1113526.26	7369.495	1600.408	472.6202	212.2493	95.93018	44.43204	25.64177
2845.291889	1366.788	619.1983	300.9496	114.7786	61.09009	34.09324	19.33793
447.4468234	458.6184	354.6095	173.9751	84.28035	48.7527	22.99782	15.20656
136.3994807	133.3624	123.5172	86.11809	53.05185	29.35361	18.75848	11.62838
47.24636078	46.06798	44.77216	43.36764	28.81882	18.46549	11.93056	9.827922
21.34602531	20.81486	21.23125	19.53493	15.60878	11.43845	9.589914	8.333294
12.14941378	11.84354	10.66156	11.08766	10.32013	8.49496	7.257209	6.675492
9.229664694	8.716275	7.755715	7.597281	7.562072	7.136904	6.278756	5.739915

b. Block coefficient mask:

	1	2	3	4	5	6	7	8
1	1	1	1	1	0	0	0	0
2	1	1	1	0	0	0	0	0
3	1	1	1	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0

c. Resulting MSE per pixel: 36.9188

(ii) 8x8 block, 5 coefficients kept:

a. Average block energy matrix: same with a. in (i)

b. Block coefficient mask:

	1	2	3	4	5	6	7	8
1	1	1	1	0	0	0	0	0
2	1	1	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0

c. Resulting MSE per pixel: 73.6765

(iii) 16x16 block, 40 coefficients kept:

a. Average block energy matrix:

4392481	49936.74	11955.11	4077.505	1892.138	1036.197	612.6555	319.2826	226.6707	144.3699	88.40918	71.27611	43.15714	34.82272	27.50977	24.65898
15099.27	10924.3	4536.448	1995.914	1310.457	615.3602	439.0047	281.5795	200.0729	116.8899	97.8373	64.2986	39.12767	26.70002	26.02746	18.55388
3104.02	2964.131	2190.862	1520.635	735.3763	584.9146	378.1431	204.5901	125.1485	86.04491	75.23078	52.03298	31.88623	27.42288	21.26908	15.75085
836.3508	1044.851	1279.093	744.256	733.6491	460.8319	251.3699	170.8372	106.7716	87.82005	56.6841	43.51826	31.00032	24.9745	19.48683	15.9719
388.7273	472.3987	480.1467	512.3895	418.6958	343.176	226.3727	147.0129	100.2263	70.92014	52.23671	34.00944	27.2799	19.936	15.98933	13.54172
206.8159	237.9029	220.7121	254.8958	282.0208	176.1841	143.0812	107.7534	68.17244	57.5767	39.84725	30.6647	22.79646	16.02337	14.00933	11.79908
101.6062	154.1746	115.423	141.1013	129.8546	136.1411	88.83169	66.32727	64.15353	41.71907	30.67936	24.68797	18.29884	14.8153	11.52714	12.18307
66.32012	71.76375	93.54336	76.19965	81.66236	67.66155	62.18324	63.00507	39.39582	33.44795	24.52823	16.90696	14.52143	12.10482	11.91042	10.13436
47.43298	47.66199	38.73416	46.33573	48.36397	47.53638	52.61275	41.13357	36.66602	24.77069	20.73646	17.30207	12.30504	11.26728	9.270624	8.946931
26.69206	26.58424	27.59643	26.67322	25.37542	29.12826	33.55545	27.65029	23.05822	18.21636	14.3248	13.12901	12.82623	9.344511	8.619672	7.932869
18.50789	20.2182	19.38841	19.69557	21.00017	27.23799	18.72978	19.86903	16.35357	13.32703	11.49889	10.92042	8.558529	9.150146	8.666635	6.856937
14.37901	13.69518	15.82048	13.48313	14.5229	15.57437	14.48063	14.37639	12.56665	10.56793	9.078033	7.904033	7.956674	6.899917	7.813554	7.583004
13.16958	11.3224	10.82248	11.87467	9.723232	10.30125	10.45296	10.89348	11.40377	11.00344	8.32724	7.708645	7.526664	6.806164	6.520124	6.438296
9.876284	9.005196	9.934769	8.862251	8.656709	9.053695	8.046934	9.744921	10.51647	8.501338	7.826323	7.494232	7.064021	7.261719	6.42101	5.916919
9.008136	8.561001	9.346854	9.46984	7.267325	7.796683	7.000182	7.542821	7.810988	7.42938	6.758444	6.561694	5.983596	5.975297	5.48827	5.406868
10.22306	8.112308	7.650061	7.544255	7.618819	6.477731	6.761943	6.650168	6.645195	6.690302	7.467806	7.007751	6.090077	5.844769	5.518064	5.562819

b. Block coefficient mask:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
2	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
3	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
4	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
5	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
6	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	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
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

c. Resulting MSE per pixel: 30.2193

(iv) 16x16 block, 10 coefficients kept:

a. Average block energy matrix: same with a in (iii.)

b. Block coefficient mask:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
2	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	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
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

c. Resulting MSE per pixel: 111.0360

KLT:

(i) 8x8 block, 10 coefficients kept:

a. Average block energy matrix:

5.351883	7.930937	10.22555	15.36457	25.93932	44.80842	114.2088	443.5506
5.885588	7.92351	10.35506	16.73876	26.01835	48.58515	127.9724	483.8919
6.326544	8.033769	11.01865	17.34049	28.21662	51.41986	141.3714	648.2377
6.427366	8.660772	11.16527	18.98026	32.498	60.86006	156.5932	1383.661
6.692372	8.957912	11.54252	19.3611	33.83434	67.06095	205.4076	2379.973
6.905103	9.064157	11.98473	19.74052	40.47456	86.88672	252.9087	2822.254
7.063862	9.479951	13.86675	21.11186	41.91792	95.21659	373.3221	7520.321
7.383065	9.829814	14.6635	21.85588	44.48344	100.6153	383.3447	1112790

b. Block coefficient mask:

1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1
0	0	0	0	0	0	1	1
0	0	0	0	0	0	1	1

c. Resulting MSE per pixel: 34.2895

(ii.) 8x8 block, 5 coefficients kept:

a. Average block energy matrix: same with a. in (i)

b. Block coefficient mask:

	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	1
5	0	0	0	0	0	0	0	1
6	0	0	0	0	0	0	0	1
7	0	0	0	0	0	0	0	1
8	0	0	0	0	0	0	0	1

c. Resulting MSE per pixel: 70.7324

(iii.) 16x16 block, 40 coefficients kept:

a. Average block energy matrix:

2.369959	3.330917	4.190329	5.272102	6.285074	7.970052	9.771789	12.01169	16.07049	21.39375	31.37256	50.22407	81.38747	148.0723	336.6625	1050.144
2.332576	3.258263	4.193031	5.193961	6.843746	8.019167	9.787918	12.47212	16.21146	22.43538	33.19362	52.18233	83.27848	154.0184	349.9859	1078.561
2.464427	3.311463	4.35079	5.277805	6.442939	8.125715	10.01929	12.77011	17.24256	23.4297	33.65978	52.75084	83.22232	161.4805	352.7122	1326.202
2.430561	3.361487	4.420327	5.511939	6.878993	8.411145	10.28121	12.78018	17.12574	23.68536	34.00095	52.04022	85.98721	164.1955	372.8812	1631.109
2.624678	3.595491	4.845384	5.354226	6.755071	8.358274	10.49844	12.94955	17.82722	23.76548	37.79341	55.64411	91.26381	183.9119	408.5203	2110.816
2.650481	3.810372	4.803954	5.451827	6.846963	8.753415	10.43553	13.46219	17.90354	25.01993	36.25061	54.90406	93.86685	197.4756	462.7796	2066.696
2.645917	3.547336	4.444754	6.333937	6.736382	8.612563	10.78504	13.4715	18.27636	25.02211	36.7428	57.52008	97.63734	197.1454	510.0237	3025.852
2.871909	3.562388	4.767493	5.806558	7.53895	8.486599	10.7168	14.16106	19.23741	25.30775	37.46025	60.05066	99.3309	204.5193	519.3437	2709.517
2.769809	3.592576	4.767447	5.739499	7.019576	8.778416	11.1189	14.01586	18.73537	26.32608	38.00238	65.56791	100.334	222.5534	575.9499	3295.721
2.799466	3.742066	4.674233	5.80817	7.0464	10.44122	11.82797	14.71859	18.82149	26.92142	39.66801	64.54438	107.6545	225.0982	604.4366	4154.311
2.884659	3.688501	4.887327	5.767117	7.528063	8.790721	11.31072	14.55557	20.0958	27.99197	43.06242	70.88006	112.9518	230.7002	642.0933	4988.742
2.905679	3.900327	4.900709	6.045012	7.279824	8.970615	11.44229	14.79919	19.63581	31.28725	41.36983	67.85254	119.4715	236.255	675.2414	14625.96
2.982081	3.979957	5.162129	5.945883	7.453368	9.414944	11.49229	15.0268	20.0348	28.58731	43.25427	69.49199	121.561	255.1312	708.16	20698.35
3.025331	4.703513	4.92744	6.028686	7.542048	9.383099	12.04748	15.28803	20.85966	28.83935	45.65044	72.25137	126.2609	272.6679	828.4125	15408.74
3.116713	3.980041	5.265613	6.316888	7.794121	9.494704	11.99778	15.54022	20.99277	29.90184	45.01557	73.85646	134.9505	306.3912	915.2131	50550.02
3.230931	4.074746	5.103386	6.219151	7.72297	9.546038	11.85235	16.75173	21.39964	30.64434	47.10282	79.55755	144.7806	316.7071	957.4898	4379323

b. Block coefficient mask:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
9	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
10	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
11	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
12	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
13	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
14	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
15	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
16	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

c. Resulting MSE per pixel: 24.8651

(iv.) 16x16 block, 10 coefficients kept:

a. Average block energy matrix: same with a. in (iii.)

b. Block coefficient mask:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

c. Resulting MSE per pixel: 105.1344

Q2.

For both DCT and KLT, as the number of coefficients kept increase and keep the block size the same, the visual quality of the compressed image will be better, the distortion will be lower and MSE will be lower. If keep the number of coefficients the same, for both DCT and KLT methods, as the block size increase, the visual quality of the compressed image will get lower, distortion will be higher and MSE will be higher. When keep the block size and the number of coefficients kept same for both DCT and KLT methods, KLT gives better visual quality and MSE will be smaller than DCT.

Q3. Zonal masking is effective if most of the energy is concentrated in the low frequency part. By doing the zonal masking, small high-frequency components will be discarded, so it might not be effective if there are a lot of high frequency components. Instead of zonal masking method, we could also do thresholding for each block, and those energy above the selected threshold will be preserved and the others will be discarded.

Q4. KLT has higher compression efficiency than DCT and results in better visual quality and lower MSE of compressed image if keeping the block size and number of coefficients the same. However, computing KLT requires a correlation matrix computation followed by an eigenvector decomposition which requires extra execution time and unlike DCT, there is no fast algorithm for computing the transformed coefficients with KLT method. The KLT works better than DCT in compression quality in the image that is less-correlated, while DCT is almost optimal in stationary image with high-correlation.

Q5. The percentage of total energy retained can be calculated as the sum of energy corresponding to the preserved coefficients divided by the total energy of the entire image.

DCT:

i)8x8 block, 10 coefficients kept : 0.9966
ii) 8x8 block, 5 coefficients kept : 0.9958
iii) 16x16 block, 40 coefficients kept : 0.9982
iv) 16x16 block, 10 coefficients kept : 0.9937

KLT:

i)8x8 block, 10 coefficients kept : 0.99806
ii) 8x8 block, 5 coefficients kept : 0.9959
iii) 16x16 block, 40 coefficients kept : 0.99859
iv) 16x16 block, 10 coefficients kept : 0.994053

Q6. Comparing DCT and KLT matrices, we could see that they are very different. For DCT, high energy concentrated on the top left corner of matrices (low frequency part) while for KLT, high energy concentrated on the bottom right corner of matrices (high frequency part). The assumption for KLT to perform best is that the image is highly uncorrelated, while DCT works best when the image is highly correlated.

Image obtained:

8X8 #coefficients = 5 DCT:

8X8 #coefficients = 10 DCT:

8X8 #coefficients = 5 KLT:

8X8 #coefficients = 10 KLT:

16X16 #coefficients = 10 DCT:

16X16 #coefficients = 40 DCT:

16X16 #coefficients = 10 KLT:

16X16 #coefficients = 40 KLT:

Conclusion:

In this lab, we compared the performance of two transform coding method DCT and KLT. DCT will perform more efficient and faster than KLT and works best especially for highly-correlated stationary image. However, KLT results in better visual quality and lower MSE compared to DCT for the lena image which is not very high-correlated. KLT will cost longer execution time due to its extra computation of correlation matrix and eigenvector decomposition but will results in better quality of compressed image when there is random correlation of the image.