SYDE 575: Introduction to Image Processing

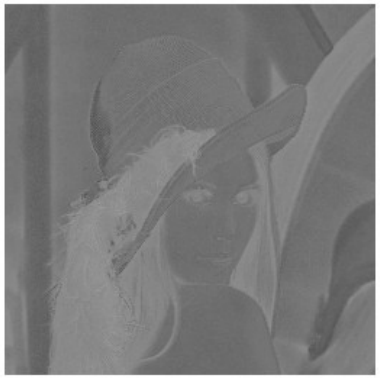
Image Compression Part 2:
Variable-rate compression
(Example of JPEG Compression)

Example of JPEG Compression

- Supposed we are given a color image represented in R8G8B8 format.

Color space transformation

- Convert image from RGB color space to YcbCr color space.



Chroma Subsampling

- Reduce the resolution of Cb and Cr channels by a factor of two in each dimension.

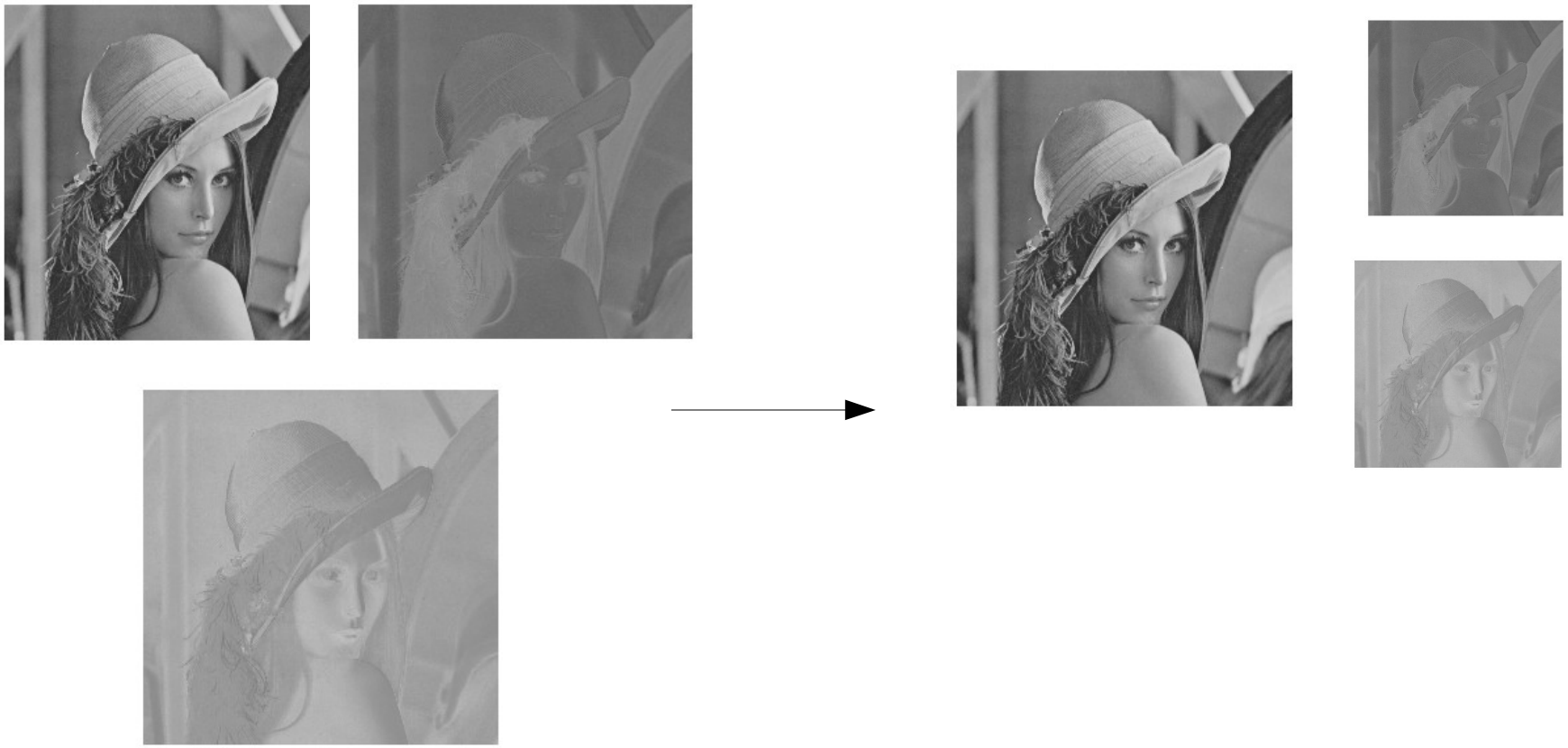
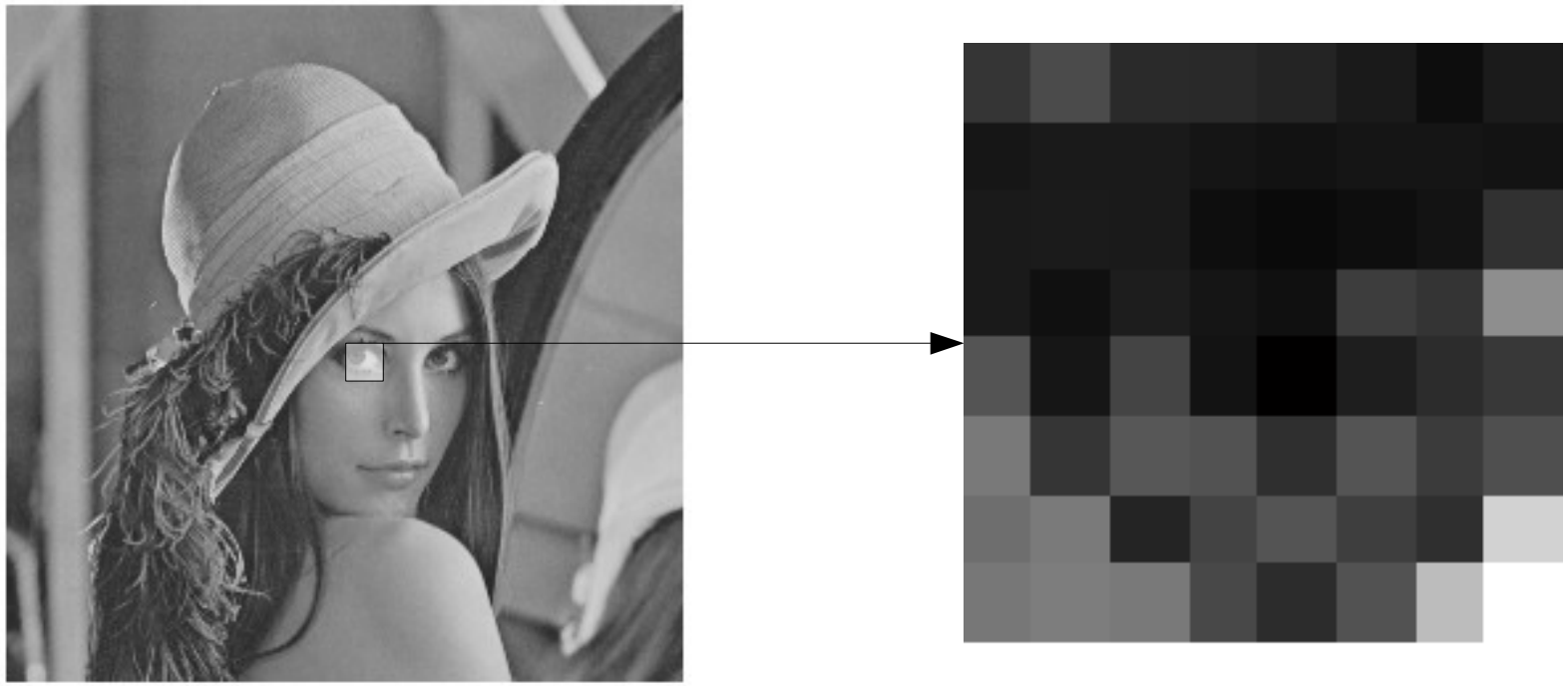


Image sub-division

- For each channel, divide the image into 8x8 blocks.



Zero-shift

- For each block, shift values by -128 to center around zero
- Reduces internal precision requirements during DCT

70	84	63	62	59	52	44	53
50	52	52	49	47	49	49	48
52	53	52	45	42	45	48	67
52	46	54	49	46	75	69	128
90	50	80	48	35	55	64	72
114	70	92	89	66	90	74	87
107	115	59	79	90	76	66	173
113	117	114	82	64	89	158	202

-58	-44	-65	-66	-69	-76	-84	-75
-78	-76	-76	-79	-81	-79	-79	-80
-76	-75	-76	-83	-86	-83	-80	-61
-76	-82	-74	-79	-82	-53	-59	0
-38	-78	-48	-80	-93	-73	-64	-56
-14	-58	-36	-39	-62	-38	-54	-41
-21	-13	-69	-49	-38	-52	-62	45
-15	-11	-14	-46	-64	-39	30	74

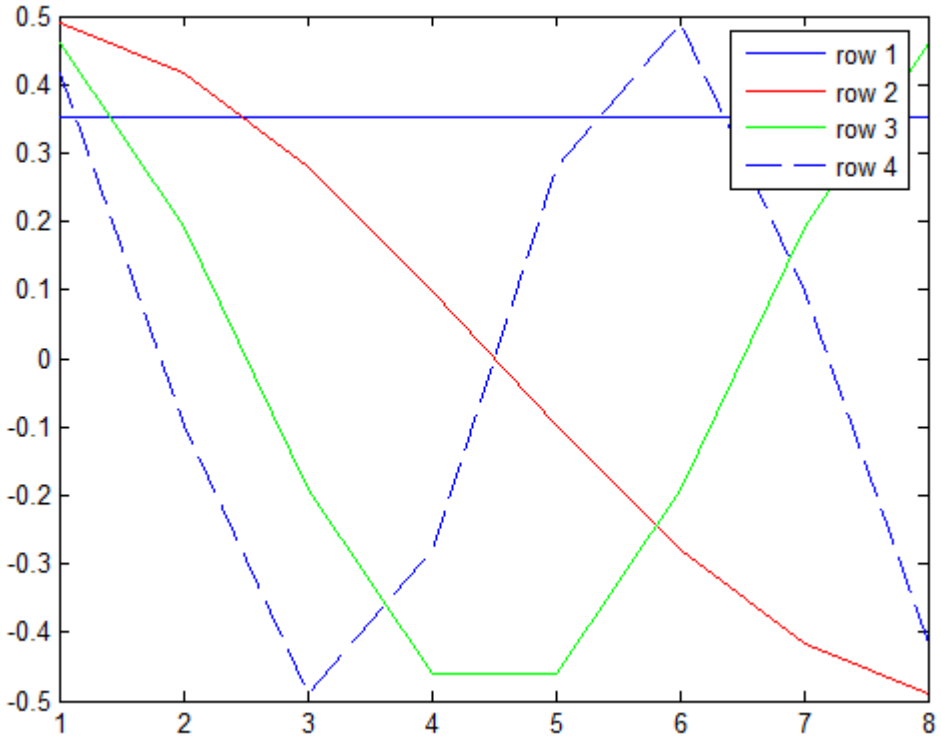
DCT Matrix

- DCT Matrix is composed of 8 1-D cosines of varying frequencies (arranged in rows)

0.3536	0.3536	0.3536	0.3536	0.3536	0.3536	0.3536	0.3536
0.4904	0.4157	0.2778	0.0975	-0.0975	-0.2778	-0.4157	-0.4904
0.4619	0.1913	-0.1913	-0.4619	-0.4619	-0.1913	0.1913	0.4619
0.4157	-0.0975	-0.4904	-0.2778	0.2778	0.4904	0.0975	-0.4157
0.3536	-0.3536	-0.3536	0.3536	0.3536	-0.3536	-0.3536	0.3536
0.2778	-0.4904	0.0975	0.4157	-0.4157	-0.0975	0.4904	-0.2778
0.1913	-0.4619	0.4619	-0.1913	-0.1913	0.4619	-0.4619	0.1913
0.0975	-0.2778	0.4157	-0.4904	0.4904	-0.4157	0.2778	-0.0975

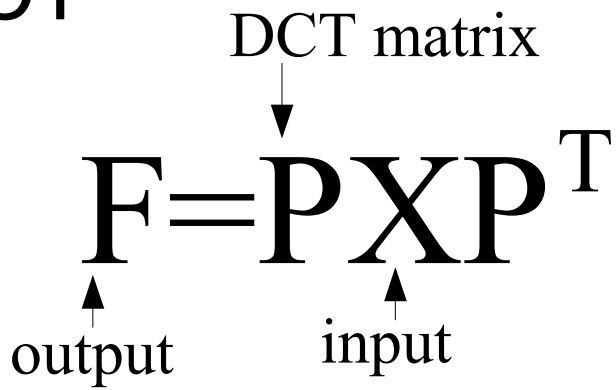
DCT Matrix

- DCT Matrix is composed of 8 1-D cosines of varying frequencies (arranged in rows)



DCT

- Compute DCT



-58	-44	-65	-66	-69	-76	-84	-75
-78	-76	-76	-79	-81	-79	-79	-80
-76	-75	-76	-83	-86	-83	-80	-61
-76	-82	-74	-79	-82	-53	-59	0
-38	-78	-48	-80	-93	-73	-64	-56
-14	-58	-36	-39	-62	-38	-54	-41
-21	-13	-69	-49	-38	-52	-62	45
-15	-11	-14	-46	-64	-39	30	74

X

-438	-24	90	-39	22	-11	25	-12
-159	37	-67	26	-19	2	-8	3
73	3	18	-15	-1	-20	-31	-3
-8	58	-33	29	8	8	9	-3
24	-22	26	-18	-19	13	4	9
14	-16	6	-3	31	-44	-5	-22
16	15	-10	2	-17	26	4	6
-25	26	6	-1	5	-10	-9	5

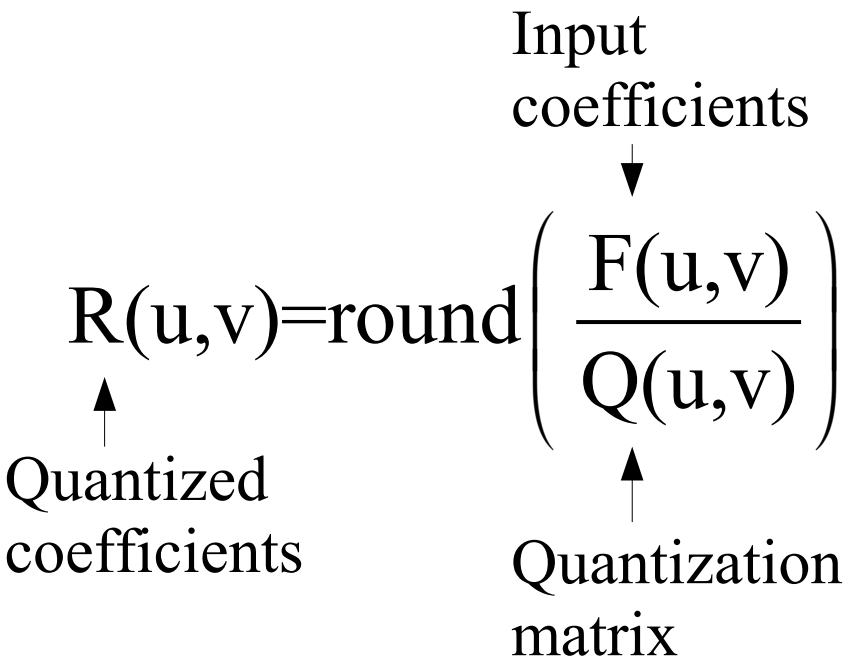
F

DCT Memory Requirements

- Notice that the DC coefficient (i.e., F(0,0)) requires more than 8 bits to represent
  - Since much of the energy is stored in the DC coefficient
- Therefore, internally, more bits are required for storage during DCT computation
- This does not increase memory requirements that much as we are only operating on small 8x8 blocks at a time.

Quantization

- Divide DCT coefficients by quantization matrix



$$Q=\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

Quantization

- Divide DCT coefficients by quantization matrix
- Quantized coefficients composed of mostly zeros and small values

-438	-24	90	-39	22	-11	25	-12
-159	37	-67	26	-19	2	-8	3
73	3	18	-15	-1	-20	-31	-3
-8	58	-33	29	8	8	9	-3
24	-22	26	-18	-19	13	4	9
14	-16	6	-3	31	-44	-5	-22
16	15	-10	2	-17	26	4	6
-25	26	6	-1	5	-10	-9	5

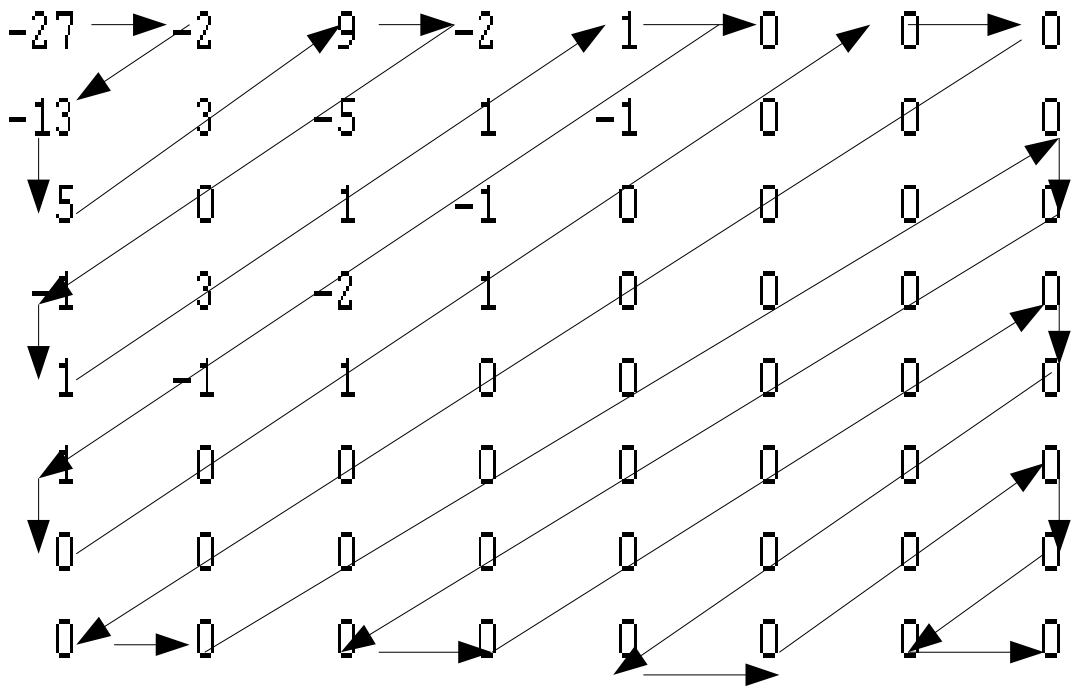
F

-27	-2	9	-2	1	0	0	0
-13	3	-5	1	-1	0	0	0
5	0	1	-1	0	0	0	0
-1	3	-2	1	0	0	0	0
1	-1	1	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

R

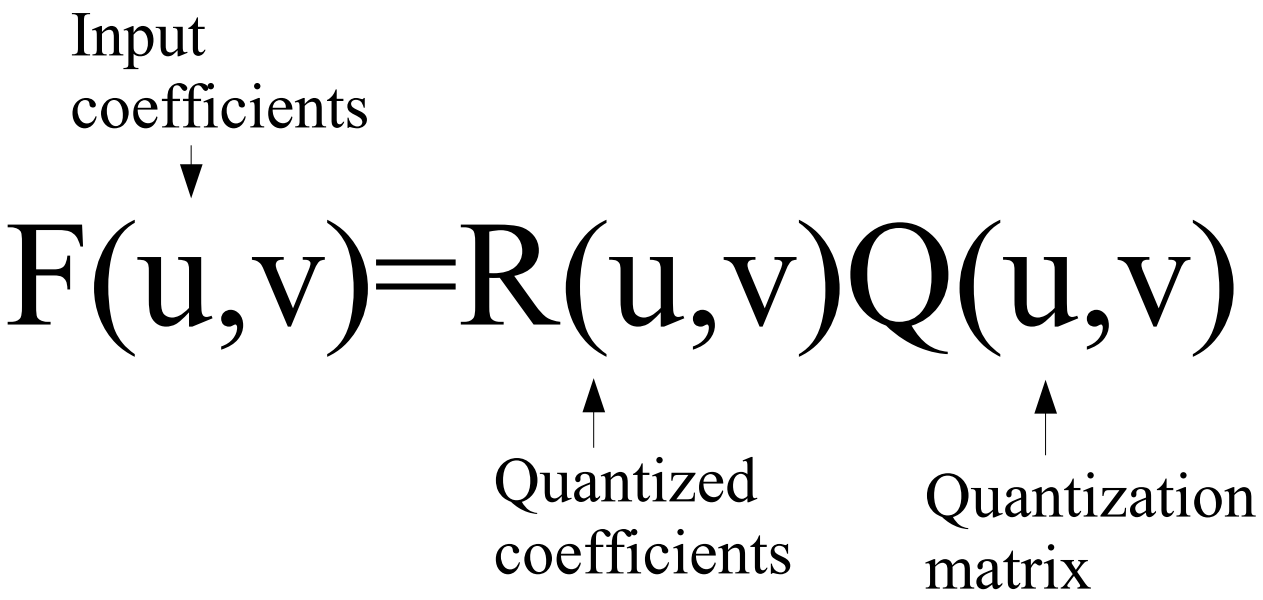
Variable-Length Coding

- Arrange coefficients in zig-zag order
- Apply a run-length encoding scheme
- Perform variable-length coding (e.g., Huffman)



Decoding

- Multiply quantized DCT coefficients by quantization matrix



Decoding

- Multiply quantized DCT coefficients by quantization matrix

-27	-2	9	-2	1	0	0	0
-13	3	-5	1	-1	0	0	0
5	0	1	-1	0	0	0	0
-1	3	-2	1	0	0	0	0
1	-1	1	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

R

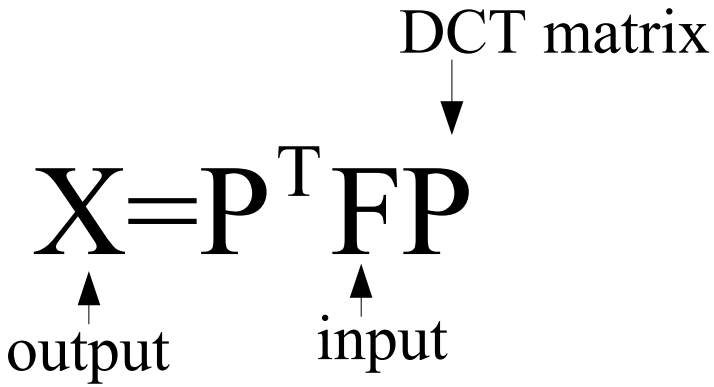
→

-432	-22	90	-32	24	0	0	0
-156	36	-70	19	-26	0	0	0
70	0	16	-24	0	0	0	0
-14	51	-44	29	0	0	0	0
18	-22	37	0	0	0	0	0
24	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

F

Decoding

- Compute Inverse DCT



-432	-22	90	-32	24	0	0	0
-156	36	-70	19	-26	0	0	0
70	0	16	-24	0	0	0	0
-14	51	-44	29	0	0	0	0
18	-22	37	0	0	0	0	0
24	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

F

-58	-56	-59	-66	-71	-72	-75	-80
-80	-73	-69	-74	-82	-85	-83	-81
-78	-75	-73	-78	-84	-80	-65	-51
-57	-68	-78	-82	-81	-70	-47	-26
-44	-63	-77	-76	-72	-68	-58	-44
-29	-46	-54	-49	-50	-60	-58	-45
-10	-25	-33	-35	-44	-48	-21	17
-2	-19	-35	-48	-61	-48	17	87

X

Decoding

- For each block, shift values by 128 to remove the effects of zero-shift during encoding

-58	-56	-59	-66	-71	-72	-75	-80
-80	-73	-69	-74	-82	-85	-83	-81
-78	-75	-73	-78	-84	-80	-65	-51
-57	-68	-78	-82	-81	-70	-47	-26
-44	-63	-77	-76	-72	-68	-58	-44
-29	-46	-54	-49	-50	-60	-58	-45
-10	-25	-33	-35	-44	-48	-21	17
-2	-19	-35	-48	-61	-48	17	87

70	72	69	62	57	56	53	48
48	55	59	54	46	43	45	47
50	53	55	50	44	48	63	77
71	60	50	46	47	58	81	102
84	65	51	52	56	60	70	84
99	82	74	79	78	68	70	83
118	103	95	93	84	80	107	145
126	109	93	80	67	80	145	215