8.2.8 Block transform coding: JPEG (page 601)

CODING

(1) Subdivide the image into 8×8 subimages and process them in a from-left-to-right and from-top-to-bottom fashion **Why?**

(2) Level shifting
Subtract 2^{n-1} from each subimage, where 2^n is the number of gray scales used **Why?**

(3) Calculate the two-dimensional DCT of each subimage **Why?**

(4) Quantize each DCT...

(4.1) Divide with a multiple of a normalization matrix
This multiple is directly proportional to the degree of compression `(figure 8.30 (b), page 600)`

(4.2) Round off the result

$$T(u,v) = \textbf{DCT}\{\, f(x,y)\,\}, \quad \hat{T}(u,v) = \textbf{round}\left\{\frac{T(u,v)}{Z(u,v)}\right\}$$

(5) Construct a one-dimensional array from these quantized coefficients, using a zig-zag pattern `(figure 8.29 (d), page 598)`

(6) Code this array, using run-length and variable-length coding

(6.1) DC coefficient: Code the difference between the DC coefficient of the current subimage and the DC coefficient of the previous subimage `(tables A.3 and A.4, page 934 & 935)`

(6.2) AC coefficients: Take the value of each non-zero AC coefficient and the number of zero-valued coefficients that preceedes the non-zero AC coefficient into account `(tab A.3, p 934, tab A.5, p 935 & 936)`

- **EOB and 16 successive zeros have special codes**

(subimage $\equiv$ "block")

DECODING

(1) Decode the array and "zig-zag" back $\rightarrow \hat{T}(u,v)$

(2) Multiply with normalization matrix: $\dot{T}(u,v) = \hat{T}(u,v)\, Z(u,v)$

(3) Calculate two-dimensional DICT: $\hat{f}(x,y) =$ **DICT**$\{\, \dot{T}(u,v) \,\}$

(4) Level shift back: Add 2^{n-1} **to each subimage**

(5) Place the subimages in their correct positions

A typical normalization matrix

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

Threshold coefficient ordering sequence (zig-zag)

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

JPEG coefficient coding categories

Range	DC Difference Category	AC Category
0	0	N/A
−1, 1	1	1
−3, −2, 2, 3	2	2
−7, …, −4, 4, …, 7	3	3
−15, …, −8, 8, …, 15	4	4
−31, …, −16, 16, …, 31	5	5
−63, …, −32, 32, …, 63	6	6
−127, …, −64, 64, …, 127	7	7
−255, …, −128, 128, …, 255	8	8
−511, …, −256, 256, …, 511	9	9
−1023, …, −512, 512, …, 1023	A	A
−2047, …, −1024, 1024, …, 2047	B	B
−4095, …, −2048, 2048, …, 4095	C	C
−8191, …, −4096, 4096, …, 8191	D	D
−16383, …, −8192, 8192, …, 16383	E	E
−32767, …, −16384, 16384, …, 32767	F	N/A

JPEG default DC code

Category	Base Code	Length	Category	Base Code	Length
0	010	3	6	1110	10
1	011	4	7	11110	12
2	100	5	8	111110	14
3	00	5	9	1111110	16
4	101	7	A	11111110	18
5	110	8	B	111111110	20

JPEG default AC code

Run/ Category	Base Code	Length	Run/ Category	Base Code	Length
0/0	**1010 (= EOB)**	**4**			
0/1	00	3	8/1	11111010	9
0/2	01	4	8/2	111111111000000	17
0/3	100	6	8/3	1111111110110111	19
0/4	1011	8	8/4	1111111110111000	20
0/5	11010	10	8/5	1111111110111001	21
0/6	111000	12	8/6	1111111110111010	22
0/7	1111000	14	8/7	1111111110111011	23
0/8	1111110110	18	8/8	1111111110111100	24
0/9	1111111110000010	25	8/9	1111111110111101	25
0/A	1111111110000011	26	8/A	1111111110111110	26
1/1	1100	5	9/1	111111000	10
1/2	111001	8	9/2	1111111110111111	18
1/3	1111001	10	9/3	1111111111000000	19
1/4	111110110	13	9/4	1111111111000001	20
1/5	11111110110	16	9/5	1111111111000010	21
1/6	1111111110000100	22	9/6	1111111111000011	22
1/7	1111111110000101	23	9/7	1111111111000100	23
1/8	1111111110000110	24	9/8	1111111111000101	24
1/9	1111111110000111	25	9/9	1111111111000110	25
1/A	1111111110001000	26	9/A	1111111111000111	26
2/1	11011	6	A/1	111111001	10
2/2	11111000	10	A/2	1111111111001000	18
2/3	1111110111	13	A/3	1111111111001001	19
2/4	1111111110001001	20	A/4	1111111111001010	20
2/5	1111111110001010	21	A/5	1111111111001011	21
2/6	1111111110001011	22	A/6	1111111111001100	22
2/7	1111111110001100	23	A/7	1111111111001101	23

JPEG default AC code (continued...)

2/8	1111111110001101	24	A/8	1111111111001110	24
2/9	1111111110001110	25	A/9	1111111111001111	25
2/A	1111111110001111	26	A/A	1111111111010000	26
3/1	111010	7	B/1	111111010	10
3/2	111110111	11	B/2	1111111111010001	18
3/3	11111110111	14	B/3	1111111111010010	19
3/4	1111111110010000	20	B/4	1111111111010011	20
3/5	1111111110010001	21	B/5	1111111111010100	21
3/6	1111111110010010	22	B/6	1111111111010101	22
3/7	1111111110010011	23	B/7	1111111111010110	23
3/8	1111111110010100	24	B/8	1111111111010111	24
3/9	1111111110010101	25	B/9	1111111111011000	25
3/A	1111111110010110	26	B/A	1111111111011001	26
4/1	111011	7	C/1	1111111010	11
4/2	1111111000	12	C/2	1111111111011010	18
4/3	1111111110010111	19	C/3	1111111111011011	19
4/4	1111111110011000	20	C/4	1111111111011100	20
4/5	1111111110011001	21	C/5	1111111111011101	21
4/6	1111111110011010	22	C/6	1111111111011110	22
4/7	1111111110011011	23	C/7	1111111111011111	23
4/8	1111111110011100	24	C/8	1111111111100000	24
4/9	1111111110011101	25	C/9	1111111111100001	25
4/A	1111111110011110	26	C/A	1111111111100010	26

JPEG default AC code (continued...)

5/1	1111010	8	D/1	1111111010	12
5/2	111111001	12	D/2	1111111111000011	18
5/3	1111111100011111	19	D/3	1111111111000100	19
5/4	1111111101000000	20	D/4	1111111111000101	20
5/5	1111111101000001	21	D/5	1111111111000110	21
5/6	1111111101000010	22	D/6	1111111111000111	22
5/7	1111111101000011	23	D/7	1111111111001000	23
5/8	1111111101000100	24	D/8	1111111111001001	24
5/9	1111111101000101	25	D/9	1111111111001010	25
5/A	1111111101000110	26	D/A	1111111111001011	26
6/1	1111011	8	E/1	111111110110	13
6/2	1111111000	13	E/2	1111111111001100	18
6/3	1111111101000111	19	E/3	1111111111001101	19
6/4	1111111101001000	20	E/4	1111111111001110	20
6/5	1111111101001001	21	E/5	1111111111001111	21
6/6	1111111101001010	22	E/6	1111111111010000	22
6/7	1111111101001011	23	E/7	1111111111010001	23
6/8	1111111101001100	24	E/8	1111111111010010	24
6/9	1111111101001101	25	E/9	1111111111010011	25
6/A	1111111101001110	26	E/A	1111111111010100	26
7/1	1111001	9	F/0	11111110111	12
7/2	1111111001	13	F/1	1111111111010101	17
7/3	1111111101001111	19	F/2	1111111111010110	18
7/4	1111111101010000	20	F/3	1111111111010111	19
7/5	1111111101010001	21	F/4	1111111111011000	20
7/6	1111111101010010	22	F/5	1111111111011001	21
7/7	1111111101010011	23	F/6	1111111111011010	22

JPEG Example: CODING

1 Original 8×8 subimage

52	55	61	66	70	61	64	73
63	59	66	90	109	85	69	72
62	59	68	113	144	104	66	73
63	58	71	122	154	106	70	69
67	61	68	104	126	88	68	70
79	65	60	70	77	68	58	75
85	71	64	59	55	61	65	83
87	79	69	68	65	76	78	94

2 Level shift

−76	−73	−67	−62	−58	−67	−64	−55
−65	−69	−62	−38	−19	−43	−59	−56
−66	−69	−60	−15	16	−24	−62	−55
−65	−70	−57	−6	26	−22	−58	−59
−61	−67	−60	−24	−2	−40	−60	−58
−49	−63	−68	−58	−51	−65	−70	−53
−43	−57	−64	−69	−73	−67	−63	−45
−41	−49	−59	−60	−63	−52	−50	−34

3 DCT

−415	−29	−62	25	55	−20	−1	3
7	−21	−62	9	11	−7	−6	6
−46	8	77	−25	−30	10	7	−5
−50	13	35	−15	−9	6	0	3
11	−8	−13	−2	−1	1	−4	1
−10	1	3	−3	−1	0	2	−1
−4	−1	2	−1	2	−3	1	−2
−1	−1	−1	−2	−1	−1	0	−1

4 Quantization

−26	−3	−6	2	2	0	0	0
1	−2	−4	0	0	0	0	0
−3	1	5	−1	−1	0	0	0
−4	1	2	−1	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
0	0	0	0	0	0	0	0

5 **Zig-zag $\rightarrow$ array**

6 **Coding**

$$[-26\ -3\ 1\ -3\ -2\ -6\ 2\ -4\ 1\ -4\ 1\ 1\ 5\ 0\ 2\ 0\ 0\ -1\ 2\ 0\ 0\ 0\ 0\ 0\ -1\ -1\ \textbf{EOB}]$$

1010110 0100 001 0100 0101 100001 0110 100011 001 100011
001 001 100101 11100110 110110 0110 11110100 000 1010

JPEG Example: DECODING

1 **Decode array and zig-zag back 8×8 subimage**

-26	-3	-6	2	2	0	0	0
1	-2	-4	0	0	0	0	0
-3	1	5	-1	-1	0	0	0
-4	1	2	-1	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
0	0	0	0	0	0	0	0

2 **Multiply with normalization matrix**

-416	-33	-60	32	48	0	0	0
12	-24	-56	0	0	0	0	0
-42	13	80	-24	-40	0	0	0
-56	17	44	-29	0	0	0	0
18	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

3 DICT

−70	−64	−61	−64	−69	−66	−58	−50
−72	−73	−61	−39	−30	−40	−54	−59
−68	−78	−58	−9	13	−12	−48	−64
−59	−77	−57	0	22	−13	−51	−60
−54	−75	−64	−23	−13	−44	−63	−56
−52	−71	−72	−54	−54	−71	−71	−54
−45	−59	−70	−68	−67	−67	−61	−50
−35	−47	−61	−66	−60	−48	−44	−44

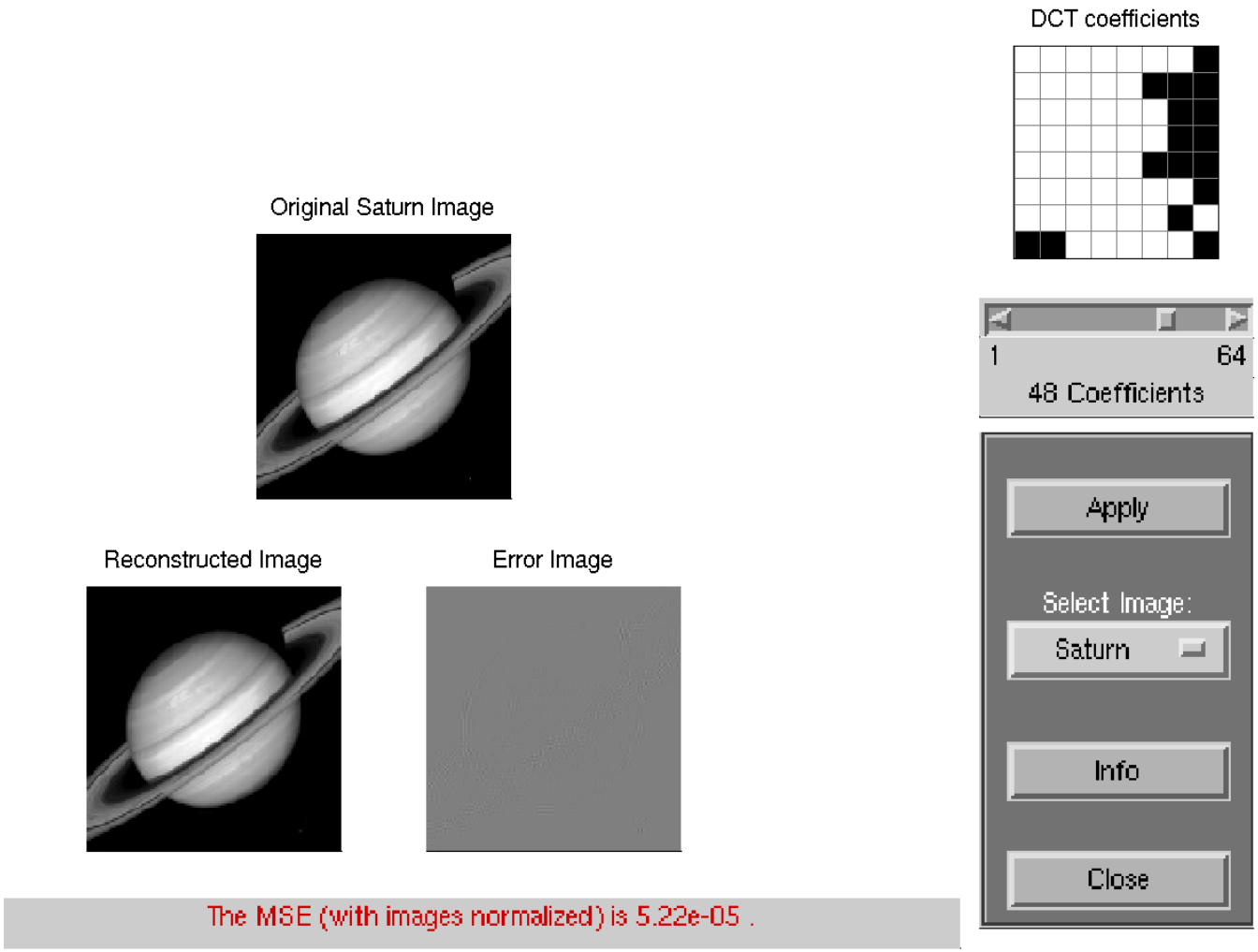
4 Level shift back

58	64	67	64	59	62	70	78
56	55	67	89	98	88	74	69
60	50	70	119	141	116	80	64
69	51	71	128	149	115	77	68
74	53	64	105	115	84	65	72
76	57	56	74	75	57	57	74
83	69	59	60	61	61	67	78
93	81	67	62	69	80	84	84

5 Place the subimages in their correct positions

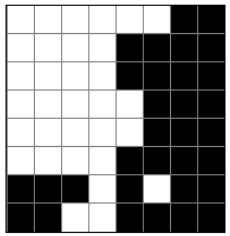
$$C_R = \frac{512}{92} \quad \Rightarrow R_D = 1 - \frac{1}{C_R} = 1 - \frac{92}{512} = 82\% \text{ redundancy}$$

JPEG Demo: 48 DCT coefficients retained

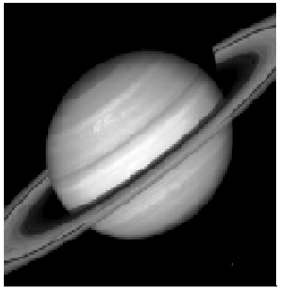


JPEG Demo: 32 DCT coefficients retained

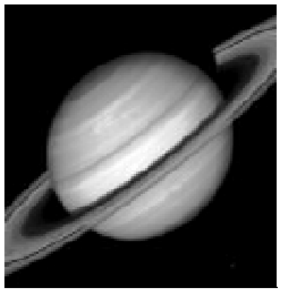
DCT coefficients



Original Saturn Image

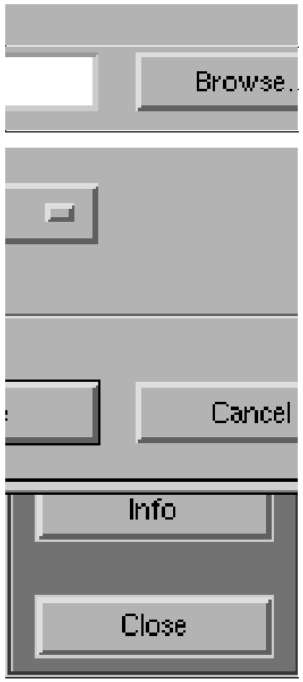


Reconstructed Image

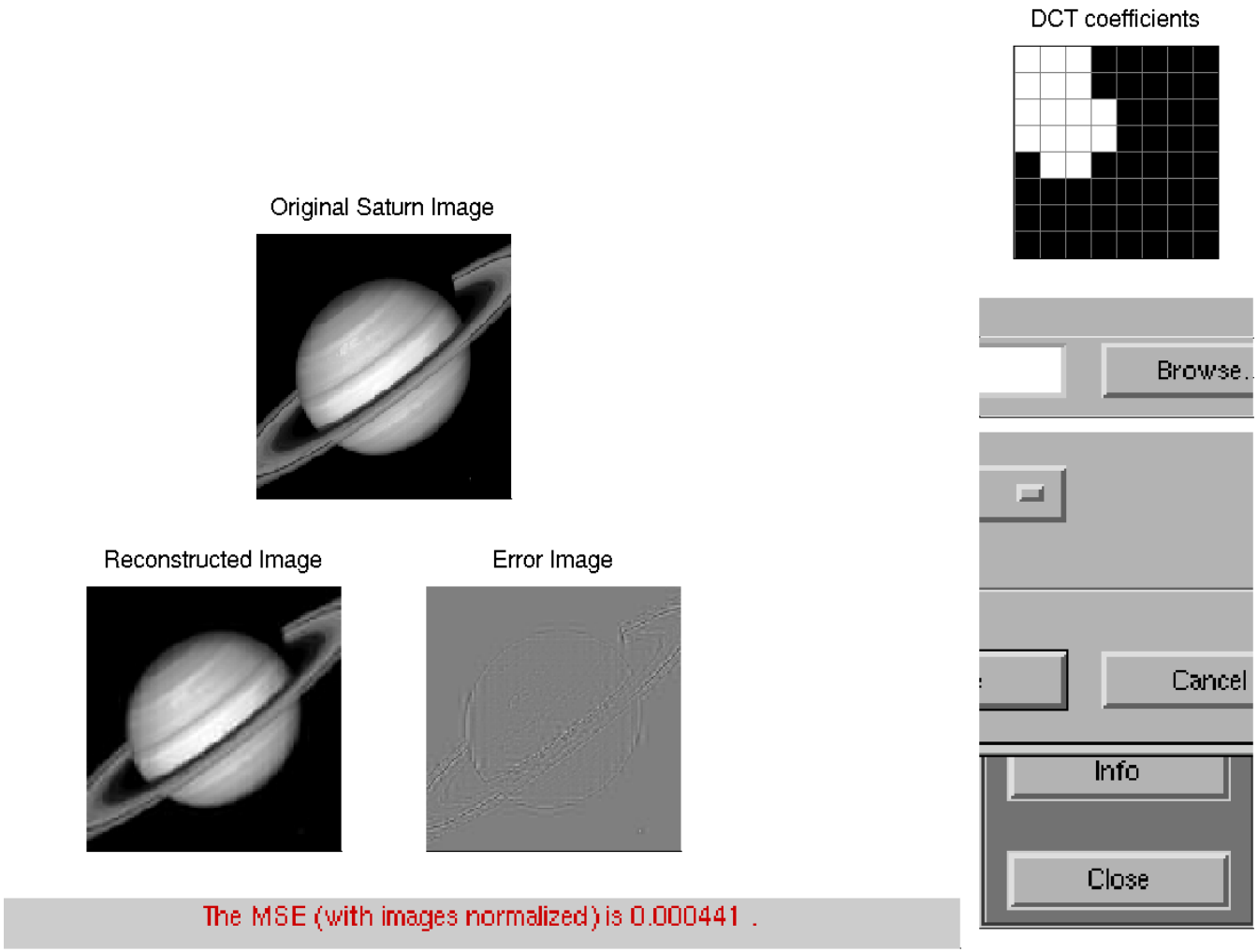


Error Image

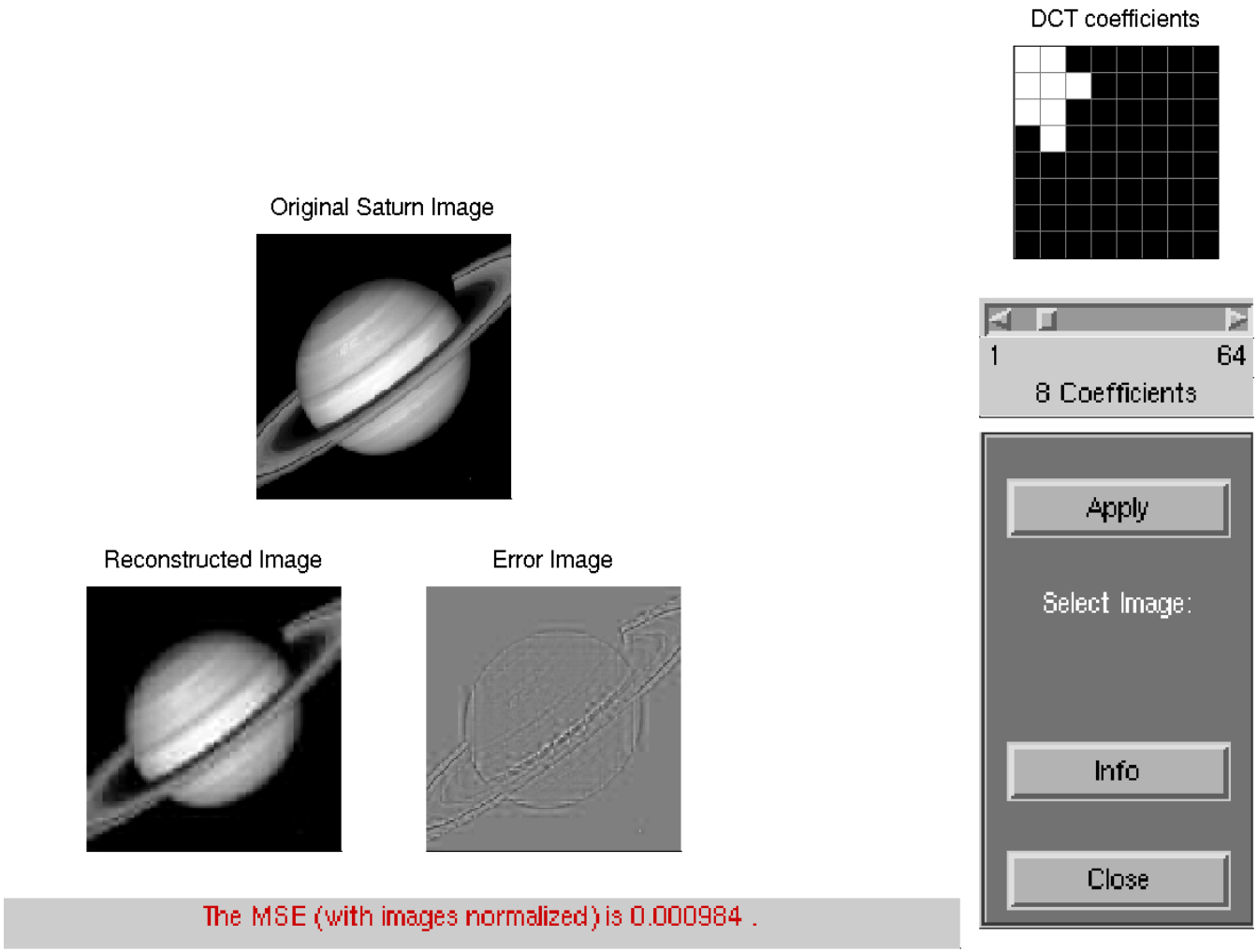
The MSE (with images normalized) is 0.000169 .



JPEG Demo: 16 DCT coefficients retained



JPEG Demo: 8 DCT coefficients retained



课堂测试

(写上姓名学号，现场递交)

1. 请解释什么是图像压缩。