Note Title 21/09/2005

DCT in two dimensions

- Forward 2D-DCT

$$F(u,v) = \frac{2C(u)C(v)}{N} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i,j) \cos\frac{(2i+1)u\pi}{2N} \cos\frac{(2j+1)v\pi}{2N}$$

$$u, v = 0, 1, \dots, N-1, \quad C(\xi) = \begin{cases} \sqrt{2}/2, & \xi = 0 \\ 1, & \xi \neq 0 \end{cases}$$

and $f(i,j)$ is an image $N \times N$ pixels

- Inverse 2D-DCT (2D-IDCT)

$$f(i,j) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \frac{C(u)C(v)}{4} F(u,v) \cos\frac{(2i+1)u\pi}{2N} \cos\frac{(2j+1)v\pi}{2N}$$

$$i, j = 0, 1, \dots, N-1, \quad C(u), C(v) \text{ as before}$$

- For JPEG compression $N=8$ always

- Note that the 2D-DCT is a "separable Transform"

$$F(u,v) = \underbrace{\frac{c(u)}{\sqrt{N}} \sum_{i=0}^{N-1} \Bigg[\underbrace{\frac{c(v)}{\sqrt{N}} \sum_{j=0}^{N-1} f(i,j) \cos\frac{(2i+1)v\pi}{2N}}_{\text{1D-DCT across columns}} \Bigg] \cos\frac{(2j+1)u\pi}{2N}}_{\text{1D-DCT across rows}}$$

- The DCT is a linear transform that is

$$DCT\left[\alpha p(i,j) + \beta q(i,j)\right] =$$

$$= \alpha\, DCT\left[p(i,j)\right] + \beta\, DCT\left[q(i,j)\right]$$

for any costants α, β and functions $p[\ ]$ and $q[\ ]$

- The DCT has always real values

MATRIX NOTATION FOR DCT

Given an 8×8 pixel image

$$f(i,j), \quad i,j = 0, \dots, 7$$

We can write the DCT as follows

$$F = M f M^T$$

and

$$M = \begin{bmatrix}
\frac{1}{\sqrt{8}} & \frac{1}{\sqrt{8}} & \frac{1}{\sqrt{8}} & \frac{1}{\sqrt{8}} & \frac{1}{\sqrt{8}} & \frac{1}{\sqrt{8}} & \frac{1}{\sqrt{8}} & \frac{1}{\sqrt{8}} \\
\frac{1}{2}\cos\frac{1}{16}\pi & \frac{1}{2}\cos\frac{3}{16}\pi & \frac{1}{2}\cos\frac{5}{16}\pi & \frac{1}{2}\cos\frac{7}{16}\pi & \frac{1}{2}\cos\frac{9}{16}\pi & \frac{1}{2}\cos\frac{11}{16}\pi & \frac{1}{2}\cos\frac{13}{16}\pi & \frac{1}{2}\cos\frac{15}{16}\pi \\
\frac{1}{2}\cos\frac{2}{16}\pi & \frac{1}{2}\cos\frac{6}{16}\pi & \frac{1}{2}\cos\frac{10}{16}\pi & \frac{1}{2}\cos\frac{14}{16}\pi & \frac{1}{2}\cos\frac{18}{16}\pi & \frac{1}{2}\cos\frac{22}{16}\pi & \frac{1}{2}\cos\frac{26}{16}\pi & \frac{1}{2}\cos\frac{30}{16}\pi \\
\frac{1}{2}\cos\frac{3}{16}\pi & \frac{1}{2}\cos\frac{9}{16}\pi & \frac{1}{2}\cos\frac{15}{16}\pi & \frac{1}{2}\cos\frac{21}{16}\pi & \frac{1}{2}\cos\frac{27}{16}\pi & \frac{1}{2}\cos\frac{33}{16}\pi & \frac{1}{2}\cos\frac{39}{16}\pi & \frac{1}{2}\cos\frac{45}{16}\pi \\
\frac{1}{2}\cos\frac{4}{16}\pi & \frac{1}{2}\cos\frac{12}{16}\pi & \frac{1}{2}\cos\frac{20}{16}\pi & \frac{1}{2}\cos\frac{28}{16}\pi & \frac{1}{2}\cos\frac{36}{16}\pi & \frac{1}{2}\cos\frac{44}{16}\pi & \frac{1}{2}\cos\frac{52}{16}\pi & \frac{1}{2}\cos\frac{60}{16}\pi \\
\frac{1}{2}\cos\frac{5}{16}\pi & \frac{1}{2}\cos\frac{15}{16}\pi & \frac{1}{2}\cos\frac{25}{16}\pi & \frac{1}{2}\cos\frac{35}{16}\pi & \frac{1}{2}\cos\frac{45}{16}\pi & \frac{1}{2}\cos\frac{55}{16}\pi & \frac{1}{2}\cos\frac{65}{16}\pi & \frac{1}{2}\cos\frac{75}{16}\pi \\
\frac{1}{2}\cos\frac{6}{16}\pi & \frac{1}{2}\cos\frac{18}{16}\pi & \frac{1}{2}\cos\frac{30}{16}\pi & \frac{1}{2}\cos\frac{42}{16}\pi & \frac{1}{2}\cos\frac{54}{16}\pi & \frac{1}{2}\cos\frac{66}{16}\pi & \frac{1}{2}\cos\frac{78}{16}\pi & \frac{1}{2}\cos\frac{90}{16}\pi \\
\frac{1}{2}\cos\frac{7}{16}\pi & \frac{1}{2}\cos\frac{21}{16}\pi & \frac{1}{2}\cos\frac{35}{16}\pi & \frac{1}{2}\cos\frac{49}{16}\pi & \frac{1}{2}\cos\frac{63}{16}\pi & \frac{1}{2}\cos\frac{77}{16}\pi & \frac{1}{2}\cos\frac{91}{16}\pi & \frac{1}{2}\cos\frac{105}{16}\pi
\end{bmatrix}$$

where T denotes matrix trasposition [columns are replaced by rows and vice versa]

- Notes $MM^T = M^T \cdot M = \begin{bmatrix} 1 & & & & & & & \\ & 1 & & & & 0 & & \\ & & 1 & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & 0 & & & & 1 & & \\ & & & & & & 1 & \\ & & & & & & & 1 \end{bmatrix} = I$

that is an "identity" matrix

This indicates that the DCT transform is an orthogonal transform.

- To obtain the inverse DCT in matrix form:

→ Forward 2D-DCT: $F = MfM^T$

→ Multiply by M^T from the left : $M^T F = M^T M f M^T = I f M^T = f M^T$

→ Multiply by M from the right : $M^T F M = f M^T M = f \cdot I = f$

So 2D-IDCT : $f = M^T F M$

• Using the 2D-DCT

Consider the image IRENE (Y component)

8x8 block of IRENE's Y component samples (near the eye)

58	45	29	27	24	19	17	20
62	52	42	41	38	30	22	18
48	47	49	44	40	36	31	25
59	78	49	32	28	31	31	31
98	138	116	78	39	24	25	27
115	160	143	97	48	27	24	21
99	137	127	84	42	25	24	20
74	95	82	67	40	25	25	19

8x8 DCT coefficients

−603	203	11	45	−30	−14	−14	−7
−108	−93	10	49	27	6	8	2
−42	−20	−6	16	17	9	3	3
56	69	7	−25	−10	−5	−2	−2
−33	−21	17	8	3	−4	−5	−3
−16	−14	8	2	−4	−2	1	1
0	−5	−6	−1	2	3	1	1
8	5	−6	−9	0	3	3	2

Note: while the image block coefficients are "size-wize" equally distributed the DCT coefficients have a higher value concentration towards the left upper corner.

- Quantization of DCT coefficients

→ $$\text{Quantized value} = \text{Round}\left(\frac{\text{Coefficient}}{\text{Quantum value}}\right)$$

→ $$\text{Coefficient} = \text{Quantized value} \times \text{Quantum value}$$

* Highly nonlinear operation
* Quantum values are provided by quantization tables obtained "empirically"

Y component Quantization table

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

C_b, C_r Component quantization table

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

(From Compressed image File Formats by J. Miano)

back to our example:

8x8 image coefficients (signal domain) Y - component

58	45	29	27	24	19	17	20
62	52	42	41	38	30	22	18
48	47	49	44	40	36	31	25
59	78	49	32	28	31	31	31
98	138	116	78	39	24	25	27
115	160	143	97	48	27	24	21
99	137	127	84	42	25	24	20
74	95	82	67	40	25	25	19

8X8 DCT coefficients (frequency domain)

" signal power is concentrated towards low frequencies"

−603	203	11	45	−30	−14	−14	−7
−108	−93	10	49	27	6	8	2
−42	−20	−6	16	17	9	3	3
56	69	7	−25	−10	−5	−2	−2
−33	−21	17	8	3	−4	−5	−3
−16	−14	8	2	−4	−2	1	1
0	−5	−6	−1	2	3	1	1
8	5	−6	−9	0	3	3	2

zing-zang ordering

Quantized DCT coefficients using quantization table

" Most coefficients are eliminated "

−38	18	1	−3	−1	0	0	0
−9	−8	1	3	1	0	0	0
−3	−2	0	1	0	0	0	0
4	4	0	−1	0	0	0	0
−2	−1	0	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

" The dynamic range of non-zero coefficients has been reduced significantly ⟶ ⟶ lower bit representation is needed

* To group as many of the quantized zero value coefficients together a zing-zang path is chosen ⟶ −38, 18, −9, −3, −8, 1, - - - - - - -

Problem 5, chapter 8 from text

Note Title 21/09/2005

a) 8×8 gray scale image has dynamic range 0→255
what is the largest value a DCT coefficient could be, and for what input image?
Also, state all the DCT coefficient values for that image:

Ans. The largest DCT value is obtained for an image with all 64 pixels taking the value 255. Then all the energy is concentrated on the DC DCT coefficient $F(0,0)$
Then from the DCT formula

$$F(0,0) = \frac{1}{8} \sum_{i,j=0}^{7}\sum^{7} 255 = \frac{64 \cdot 255}{8} = 2040$$

Also, $F(u,v) = 255\,\frac{c(u)c(v)}{8} \underbrace{\sum_{i=0}^{7} \cos\frac{(2i+1)u\pi}{16} \sum_{j=0}^{7} \cos\frac{(2j+1)v\pi}{16}}_{0}$

$(u,v) \neq (0,0)$

b) If we subtract the value 128 from the whole image and then take the DCT, what is the exact effect on the value $F[2,3]$?

Ans

$$DCT\left[f(i,j)-f_{128}(i,j)\right] = DCT\left(f(i,j)\right) - DCT\left(f_{128}(i,j)\right)$$

where $f_{128}(i,j) = 128$ all $i = 0,\dots 7$, $j = 0,\dots,7$

From part (a) we know that

$$DCT\left[f_{128}(i,j)\right] = \begin{cases} \frac{64 \cdot 128}{8}, & u=v=0 \\ 0, & \text{otherwise} \end{cases}$$

Thus, $F[2,3]$ remains unaffected

Actually only the DC $(F(0,0))$ coefficient of the DCT will change.

c) Why may we need to subtract 128 from an image block before taking the DCT? Does it help?

Ans As we have established this action is going to reduce significantly only the DC component. There will be some bit reduction in the representation of this component and the quantization error in representing it. JPEG standard requires this operation.

d) Would it be possible to invert the subtraction of 128 during IDCT?

Ans. Yes, We calculate the IDCT and then add 128 to the pixels of the obtained image.