



كلية الهندسة - قسم الهندسة المعلوماتية

مقرر الوسائط المتعددة

القسم العملي

م. نور بلوط

م. أوشين داود

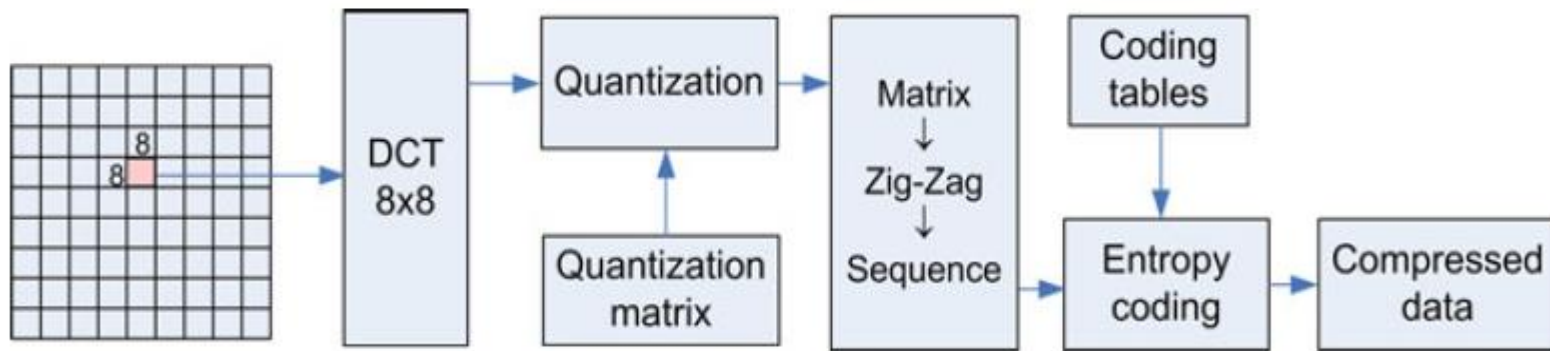
محاضرة الأسبوع ٧

الفصل الثاني - ٢٠٢٢/٢٠٢٣

## الضغط مع خسارة Lossy compression:

تعتمد تقنية الضغط مع خسارة على تخفيض حجم الملفات بالاعتماد على إزالة جزء من المعلومات الفائضة مما يخفض من جودة الملف بعد الضغط وعندما يفك الضغط لا يعاد الملف الأصلي لذا يسمى بالضغط غير القابل للعكس irreversible . يستخدم هذا النمط من الضغط في ملفات الصوت والصورة والفيديو وسندرس في محاضرتنا الحالية تقنية ضغط الصور. وسندرس بالتحديد خوارزمية JPEG Joint Photographic Experts Group من أجل الصور الرمادية والتي طورت لتصبح منها نسخة من أجل للصور الملونة ومن ثم تم تطوير JPEG2000 المعتمدة على التحويل المويجي Wavelet .

يبين الشكل التالي مراحل خوارزمية الضغط JPEG للصور الرمادية:



## التحويل التجيبي المتقطع DCT Discrete Cosine Transform

$$DCT(k_1, k_2) = \frac{C(k_1)}{2} \frac{C(k_2)}{2} \sum_{i=0}^7 \sum_{j=0}^7 a(i, j) \cos\left(\frac{(2i+1)k_1\pi}{16}\right) \cos\left(\frac{(2j+1)k_2\pi}{16}\right)$$

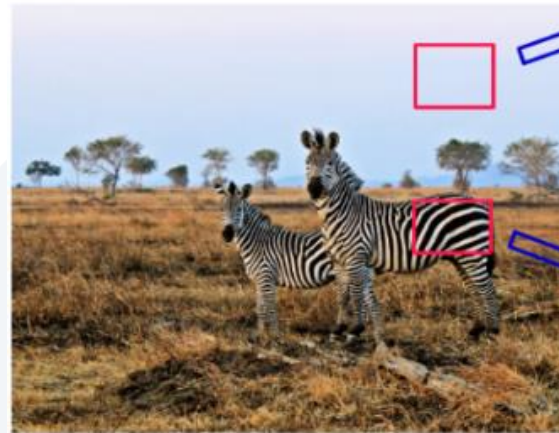
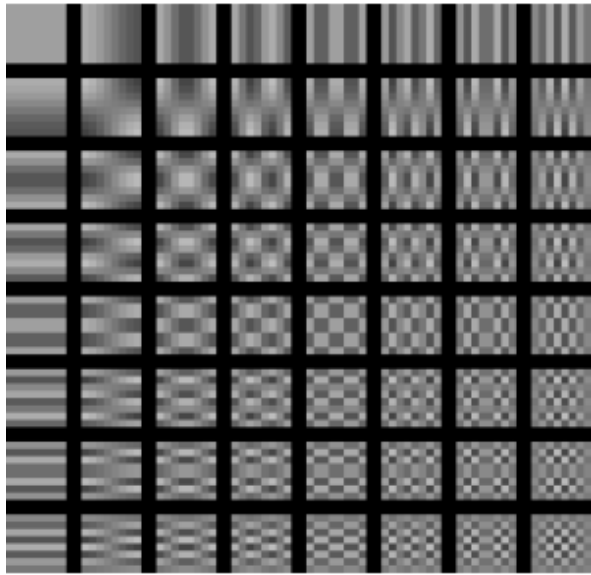
where:

$$C(k_1) = \begin{cases} \frac{1}{\sqrt{2}}, & \text{for } k_1 = 0 \\ 1, & \text{for } k_1 > 0 \end{cases}, \quad C(k_2) = \begin{cases} \frac{1}{\sqrt{2}}, & \text{for } k_2 = 0 \\ 1, & \text{for } k_2 > 0 \end{cases}.$$

يعمل هذا التحويل على نقل الصورة من المجال المكاني إلى المجال الترددي بحيث يعبر عن الصورة بتوابع تجيبية لها ترددات مختلفة و مطالات مختلفة.

## High frequency vs. Low frequency in Images

الأجزاء من الصورة ذات الترددات العالية هي الأجزاء التي يحدث فيها تغيرات سريعة في الشدة اللونية أما ذات الترددات المنخفضة فهي المناطق ذات تغيرات بطيئة في الشدة اللونية



Low Frequency



High Frequency

## التحويل التجيبي المتقطع DCT Discrete Cosine Transform

DC							
Low Frequency			AC				
	Mid Frequency						
				High Frequency			

يسمى المعامل  $DCT(0,0)$  بالمركبة المستمرة DC وهو يمثل القيمة المتوسطة ل 64 قيمة موجودة في كل بلوك أما المعاملات الـ 63 المتبقية فهي المعاملات المتناوبة AC.

تكون قيم المركبات الترددية الأخفض ذات قيم مطالية كبيرة أما ذات الترددات الأعلى فهي ذات قيم صغيرة ويمكن إهمالها بجعلها صفيرية في عملية التكميم (وهذا ما يسبب الخسارة في الصورة عند استردادها بفك الضغط مما يسبب بعض التشوه في الصورة مفكوكة الضغط)

يعمل التحويل DCT على الصور الرمادية (uint8) على مجال القيم من 0-255 وحتى 127 مما يستلزم إزاحة مجال قيم الصورة الرمادية (بعمق بت  $n=8$ ) كما يلي:

$$[-2^{n-1}, 2^{n-1} - 1]$$

## التكميم Quantization :

بعد حساب معاملات DCT على كل بلوك 8\*8 يتم تكميم هذه المعاملات من خلال قسمتها على معاملات مصفوفة التكميم ويتم تقريب النواتج :

$$DCT_q(k_1, k_2) = \text{round} \left\{ \frac{DCT(k_1, k_2)}{Q(k_1, k_2)} \right\}$$

وتحسب معاملات مصفوفة التكميم Q بالاعتماد على معامل الجودة (q) quality (q) والذي تحسب قيمته حسب مستوى الجودة المطلوب للضغط.

من أجل نسبة ضغط 50% تم حساب المصفوفة المجاورة ونستطيع حساب أي

$$Q = \text{round}(Q_{50} \cdot q),$$

مصفوفة لنسب ضغط أخرى بالعلاقة:

علماً أن q تحسب حسب قيمة نسبة الضغط المرغوبة بالعلاقات :

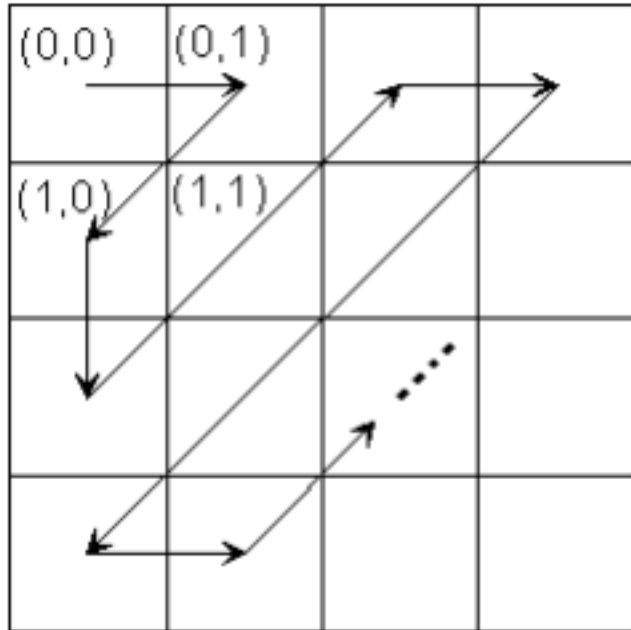
$$q = \begin{cases} 2 - 0.02QF & \text{for } QF \geq 50, \\ \frac{50}{QF} & \text{for } QF < 50. \end{cases}$$

Q\_50=

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

## : ZIG ZAG scan

يتم ترتيب المعاملات المكعبة لتتحول من مصفوفة 2D الى شعاع  $1*64$  بترتيب ZIGZAG كما هو مبين جانباً بحيث تكون القيم الترددية الأخفض في بداية الشعاع والقيم الترددية الأعلى في نهاية الشعاع



a	177 153 151 143 142 145 150 158 159 148 148 145 144 153 147 149 154 155 151 154 157 146 156 146 152 159 157 155 151 147 159 151 142 154 144 143 144 144 168 156 160 150 136 136 148 153 161 153 150 142 142 142 156 152 161 161 135 131 150 154 151 144 146 160	b	49 25 23 15 14 17 22 30 31 20 20 17 16 25 19 21 26 27 23 26 29 18 28 18 24 31 29 27 23 19 31 23 14 26 16 15 16 16 40 28 32 22 8 8 20 25 33 25 22 14 14 14 28 24 33 33 7 3 22 26 23 16 18 32	c	180 -10 21 4 3 6 -6 8 10 28 9 4 -2 3 3 -3 -7 0 2 0 15 1 19 0 0 0 26 14 -3 0 -1 0 0 1 0 -17 0 3 0 0 13 5 3 3 3 0 -2 4 0 6 1 3 4 1 -2 1 0 -6 0 -3 0 0 -7 4
d	10 7 6 10 14 24 31 37 7 7 8 11 16 35 36 33 8 8 10 14 24 34 41 34 8 10 13 17 31 52 48 37 11 13 22 34 41 65 62 46 14 21 33 38 49 62 68 55 29 38 47 52 62 73 72 61 43 55 57 59 67 60 62 59	e	18 -1 4 0 0 0 0 0 1 4 1 0 0 0 0 0 -1 0 0 0 1 0 0 0 0 0 2 1 0 0 0 0 0 0 0 -1 0 0 0 0 1 0		

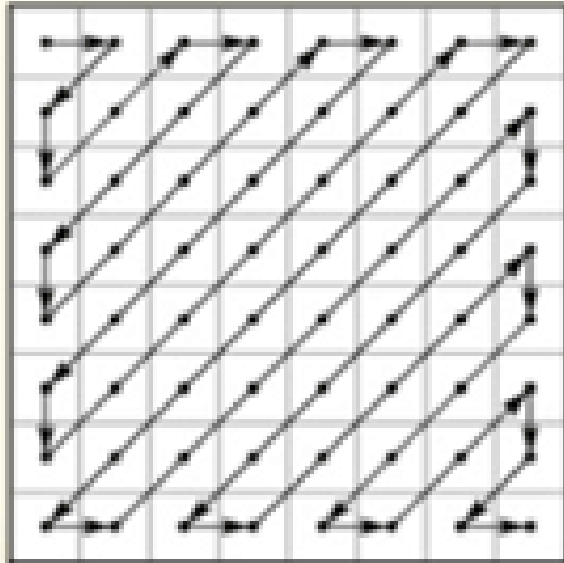
**Fig. 4.23** (a)  $8 \times 8$  image block, (b) values of pixels after shifting to the range  $[-128, 127]$ , (c) DCT coefficients (rounded to integers) for the given block, (d) quantization matrix  $QF = 70$ , (e) DCT coefficients after quantization

مثال :

يبين المثال المصفوفة النهائية e وتصبح بعد ترتيبها zigzag شعاع :



مثال :



e

18	-1	4	0	0	0	0	0
1	4	1	0	0	0	0	0
-1	0	0	0	1	0	0	0
0	0	2	1	0	0	0	0
0	0	0	-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

يبين المثال المصفوفة النهائية e  
وتصبح بعد ترتيبها zigzag شعاع :

18, -1, 1, -1, 4, 4, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0,  
-1, 0,

## : JPEG Quantization in MATLAB

```
I = imread('cameraman.tif'); %
256*256
%I = rgb2gray(I);
I = double(I)-128;
QF=70; % desired quality factor
q=2-0.02*QF; %q=50/QF;
Q50 = [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] ;

Q=round(Q50*q);
```

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

b_vector = zeros(1,64);
a = I(1:8, 1:8);
dct_block = dct2 (a);
dct_quantized = round (dct_block./Q) ;
b_vector = dct_quantized(order);
```

## الترميز encoding

Amplitude range	Size
-1,1	1
-3,-2,2,3	2
-7,-6,-5,-4,4,5,6,7	3
-15 ,... ,-8,8 ,... ,15	4
-31 ,... ,-16,16 ,... ,31	5
-63 ,... ,-32,32 ,... ,63	6
-127 ,... ,-64,64 ,... ,127	7
-255 ,... ,-128,128 ,... ,255	8
-511 ,... ,-256,256 ,... ,511	9
-1023 ,... ,-512,512 ,... ,1023	10

SIZE	Code Length	Code
0	2	00
1	3	010
2	3	011
3	3	100
4	3	101
5	3	110
6	4	1110
7	5	11110
8	6	111110
9	7	1111110
10	8	11111110
11	9	111111110

ترميز المركبة المستمرة DC:

لكوننا في هذا المثال نتعامل مع بلوك واحد فنقوم بترميز قيمته المستمرة باستخدام الجدول المرافق كمايلي:

يتم التمثيل بخانتين الأولى عدد البتات والثانية القيمة المستمرة (18)(5)

أما بحال وجود أكثر من بلوك فيتم ترميز

الفرق بين القيم المستمرة للبلوكات المتجاورة

ومن ثم يتم ترميز الفروق باستخدام جداول

ترميز كما هو مبين جانباً

Ex: 18 is represented with 5 bits  
(size )(value)->(5)(18)

5 is represented with 3 bits

The final bit stream: 1 1 0 1 0 0 1 0

## الترميز

ترميز المركبات المتناوبة AC :

ترمز الازيفار وفق ترميز RLE حيث يتم ترميزها وفق الثنائية :

(skip , value) تمثل skip عدد الازيفار التي تسبق القيمة value

وعند التمثيل النهائي لكل القيم يأخذ التمثيل الشكل التالي:

(skip, size) value

حيث يمثل size عدد البتات المطلوبة لتمثيل القيمة value

18, -1, 1, -1, 4, 4, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0,  
-1, 0

(5)(18), (0,1)(-1), (0,1)(1), (0,1)(-1) (0,3)(4), (0,3)(4), (1,1)(1), (10,2)(2), (1,1)(1),  
(3,1)(1), (0,1)(1), (6,1)(-1), (0,0)

## الترميز Encoding

ترميز المركبات المتناوبة AC :

وفق الجدول المبين جانباً يجب البحث عن كل ثنائية (skip, size) في النظام الثنائي:

Run/ SIZE	Code Length	Code
0/0	4	1010
0/1	2	00
0/2	2	01
0/3	3	100
0/4	4	1011
0/5	5	11010
0/6	7	1111000
0/7	8	11111000
0/8	10	1111110110
0/9	16	111111110000010
0/A	16	111111110000011

Run/ SIZE	Code Length	Code
1/1	4	1100
1/2	5	11011
1/3	7	1111001
1/4	9	111110110
1/5	11	11111110110
1/6	16	111111110000100
1/7	16	111111110000101
1/8	16	111111110000110
1/9	16	111111110000111
1/A	16	111111110001000
.. 15/A	More	Such rows

تراميز الثنائيات الموجودة في مثالنا المدروس

(0,1)	00
(0,3)	100
(1,1)	1100
(3,1)	111010
(6,1)	1111011
(10,2)	111111111000111
(0,0) EOB	1010

The final  
bitstream

(101)(10010) (00)(0) (00)(1) (00)(0) (100)(100) (100)(100) (1100)  
(1) (111111111000111)(10) (1100)(1) (111010)(1) (00)(1) (1111011)(0) (1010)

EOB -> (0,0) -> 1010

# انتهت تمارين الأسبوع ٧