

فشرده‌سازی اطلاعات

۰۱-۷۰۲-۱۰-۱۴۰

بخش ششم



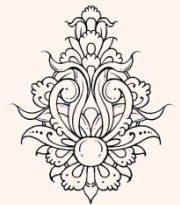
دانشگاه شهید بهشتی  
پژوهشکده‌ی فضای مجازی

بهار ۱۳۹۸

احمد محمودی ازناوه

# فهرست مطالب

- تبدیل کسینوسی صحیح
  - تصاویر پایه
  - ویژگی‌ها
- پردازش بلوکی تصاویر
- چندی کردن ضرائب (quantization)
- استاندارد فشرده‌سازی JPEG

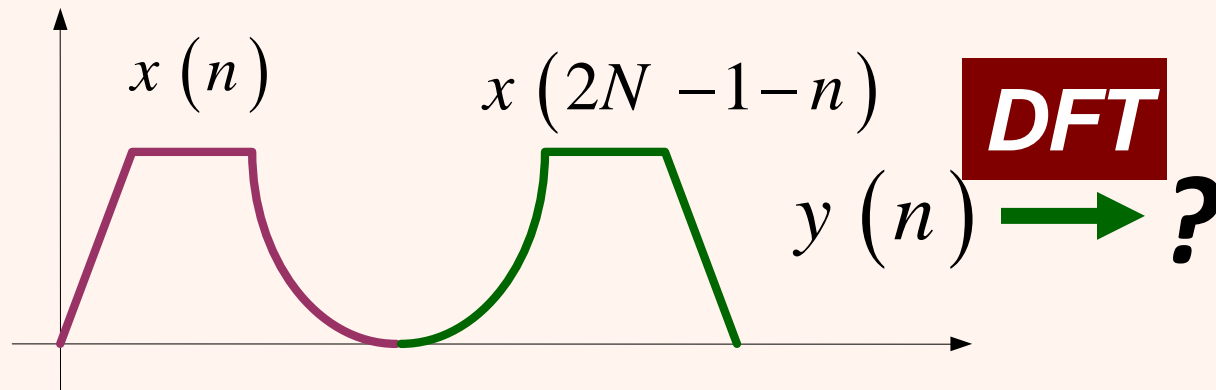


# تبدیل کسینوسی گسسته

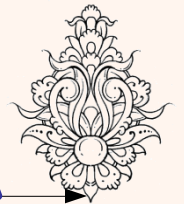
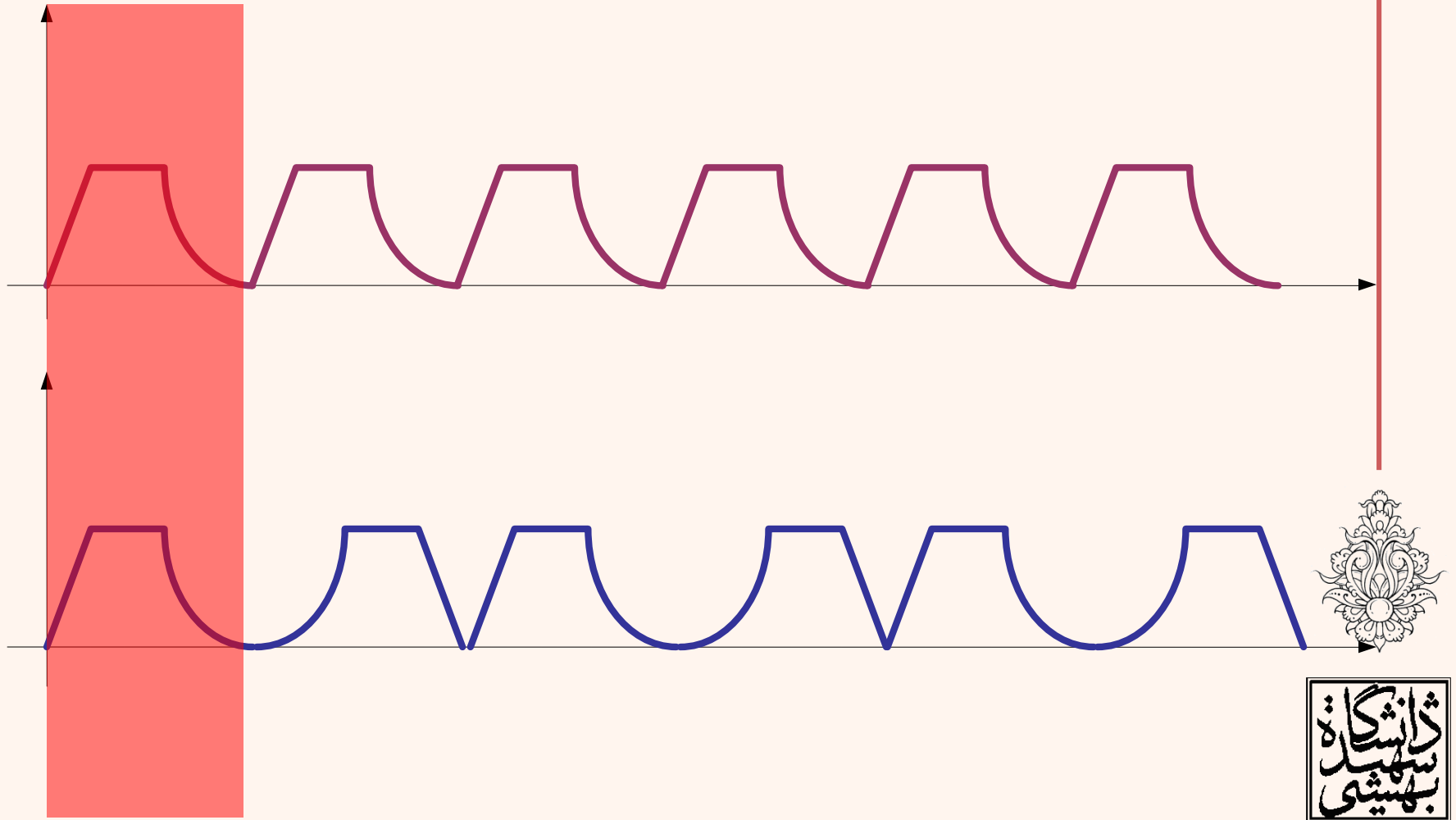
• یک سیگنال  $N$  نمونه‌ای،  $x(n)$  مفروض است:

$$y(n) = x(n) + x(2N - 1 - n)$$

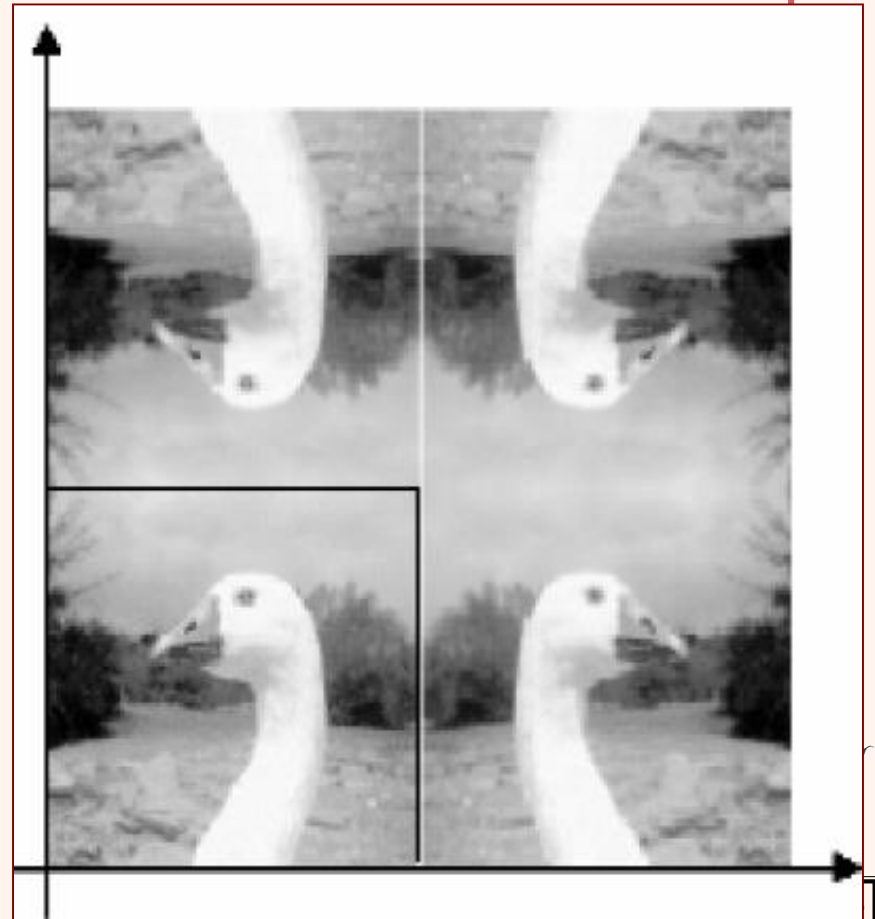
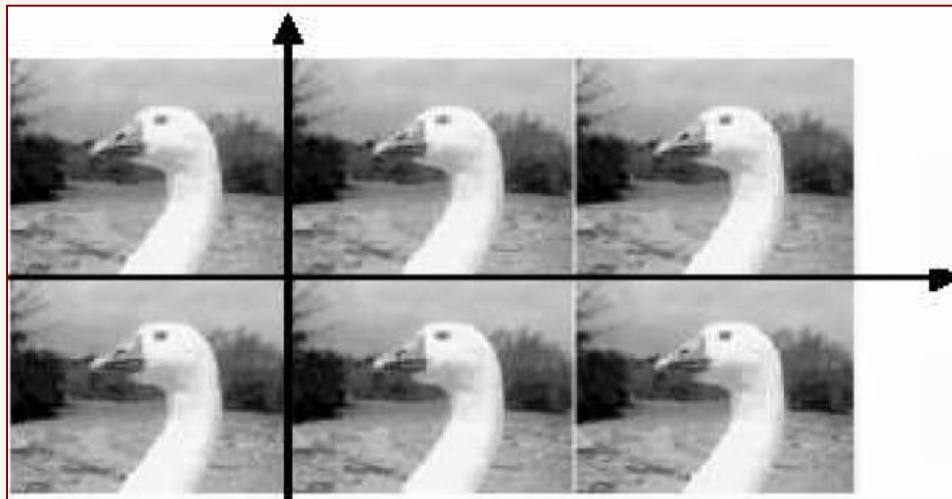
$$y(n) = \begin{cases} x(n) & 0 \leq n \leq N - 1 \\ x(2N - 1 - n) & N \leq n \leq 2N - 1 \\ 0 & \text{Otherwise} \end{cases}$$



# تبدیل کسینوسی گسسته (ادامه...)



# متناوب کردن تصویر



کتابخانه  
سپهر  
بهشتی

# تبدیل کسینوسی گسسته

$$C(k_1, k_2) = \alpha(k_1, k_2) \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f(m, n) \cos\left(\frac{\pi(2m+1)k_1}{2N}\right) \cdot \cos\left(\frac{\pi(2n+1)k_2}{2N}\right);$$

$$0 \leq k_1 \leq N-1, \quad 0 \leq k_2 \leq N-1,$$

تبدیل کسینوسی گسسته

$$f(m, n) = \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{N-1} \alpha(k_1, k_2) C(k_1, k_2) \cos\left(\frac{\pi(2m+1)k_1}{2N}\right) \cdot \cos\left(\frac{\pi(2n+1)k_2}{2N}\right);$$

$$0 \leq n \leq N-1, \quad 0 \leq m \leq N-1$$

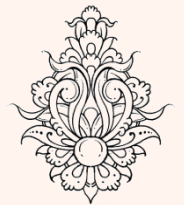
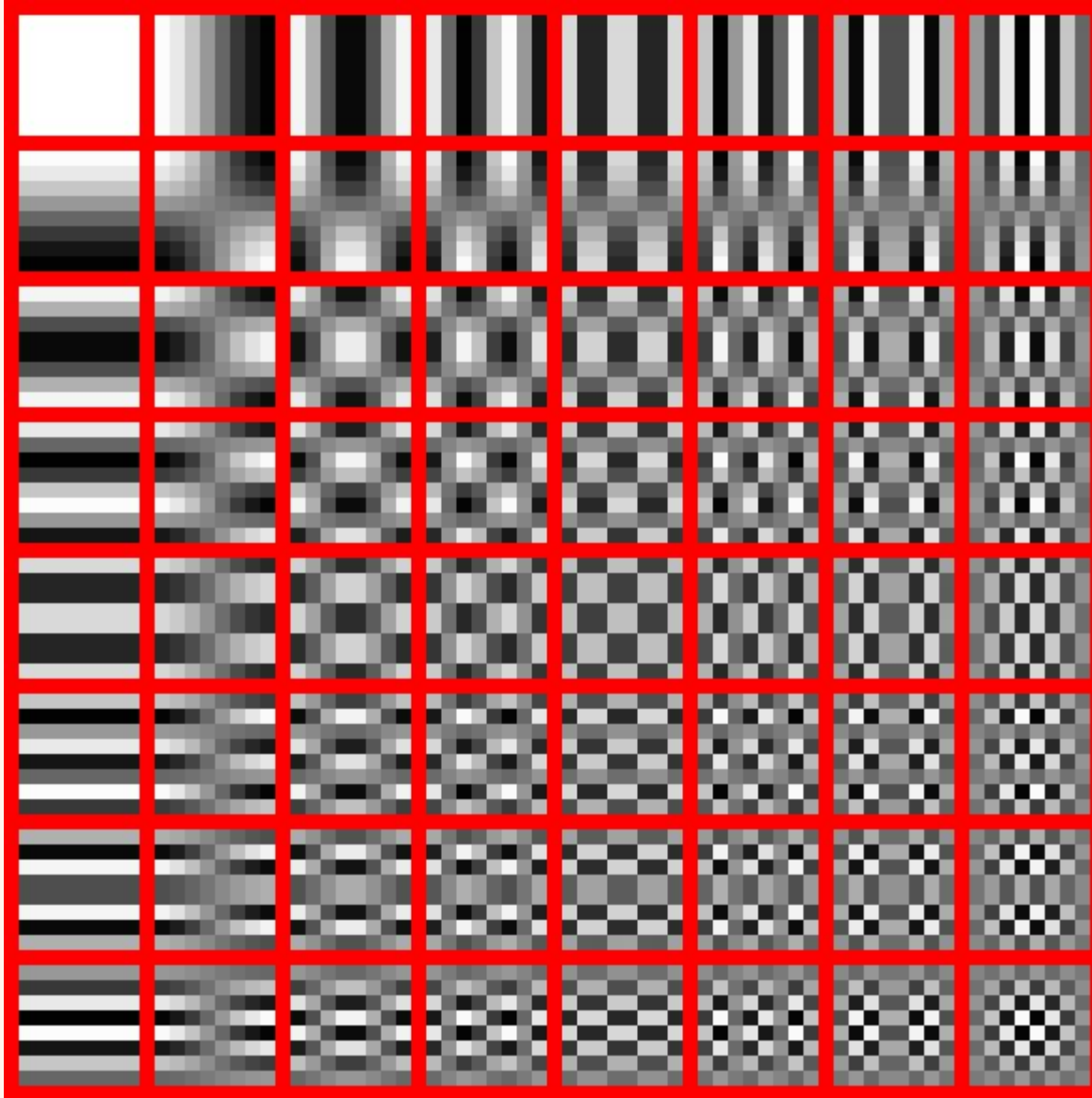
تبدیل معکوس

$$\alpha(k_1, k_2) = \begin{cases} \frac{1}{\sqrt{N}} & k_1 = 0 & 0 \leq k_2 \leq N-1 \\ \frac{1}{\sqrt{N}} & k_2 = 0 & 0 \leq k_1 \leq N-1 \\ \sqrt{\frac{2}{N}} & 0 < k_1 \leq N-1 & 0 < k_2 \leq N-1 \end{cases}$$

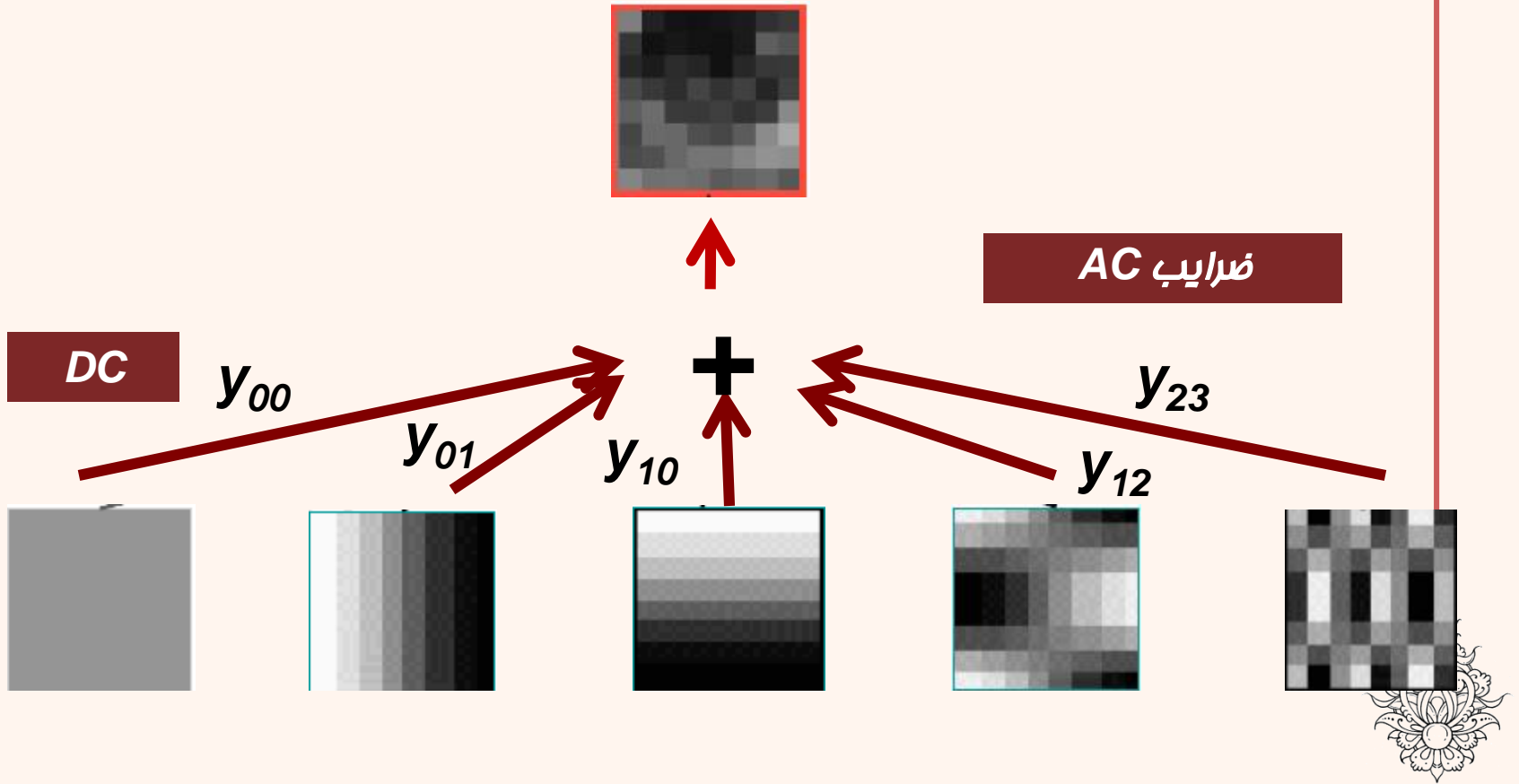


تصاویر پایه‌ی کسینوسی برای یک بلوک 8x8

تصاویر پایه



# مثال





# FFT و DCT

برای فشرده‌سازی تصویر ابتدا تصویر به «بلوک‌های ناهمپوشان» تقسیم شده و سپس مؤلفه‌های کم‌اهمیت هر بلوک در فضای تبدیل حذف می‌شوند.

FFT 25%  
saving



DCT 25%  
saving



# FFT و DCT



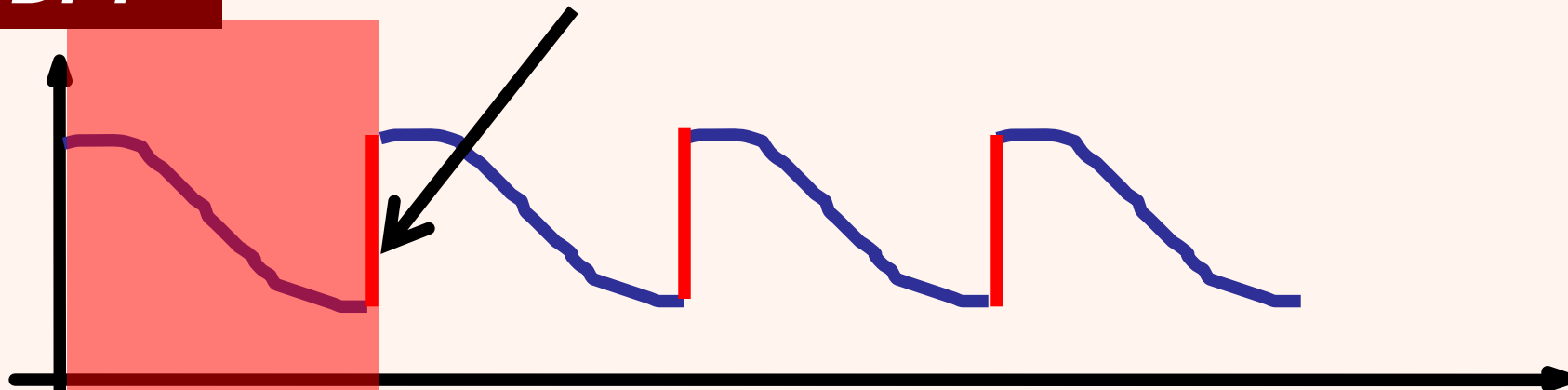
FFT 25%  
saving



DCT 25%  
saving

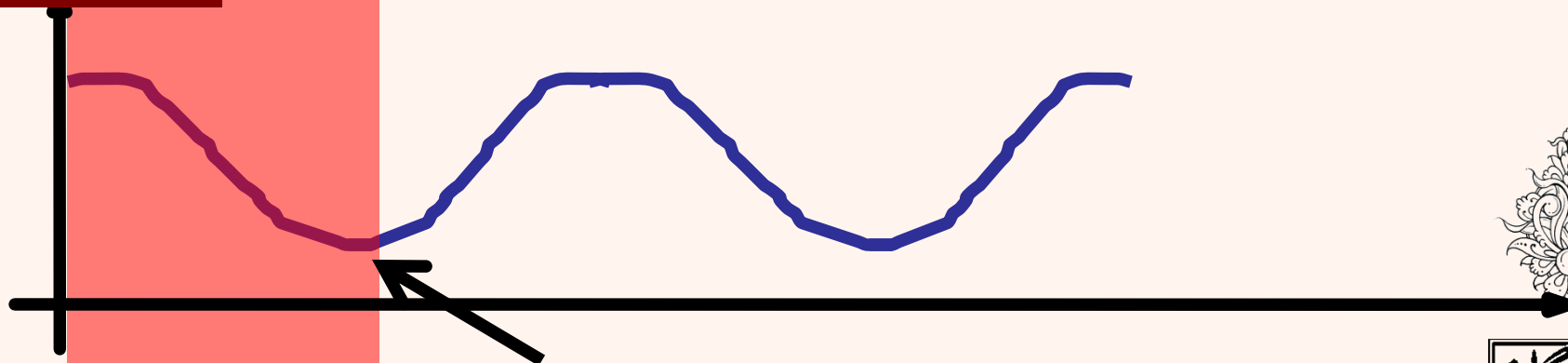
**DFT**

تغییرات ناگهانی لبه  
(معادل فرکانس بالا)



**DCT**

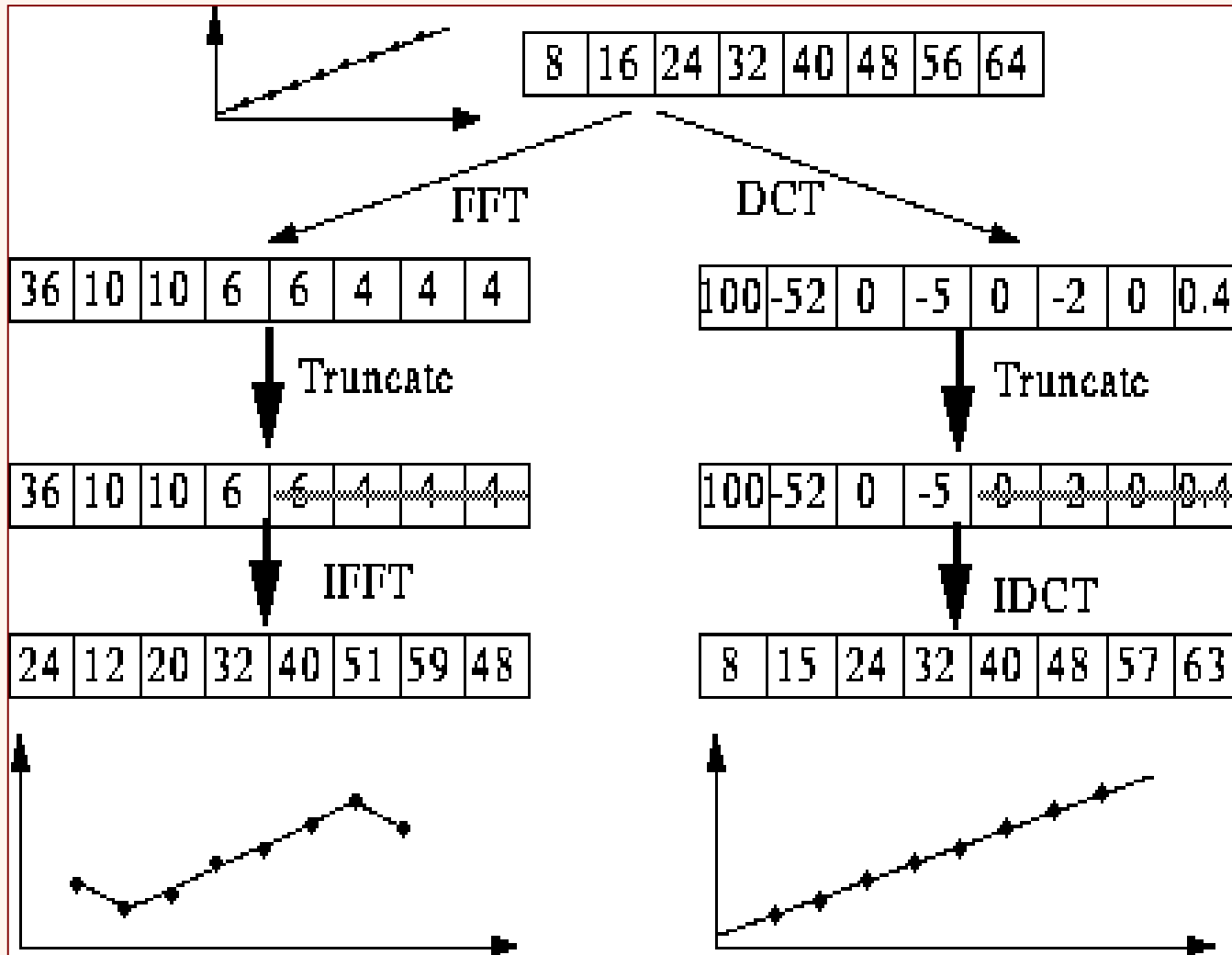
تغییرات نرم



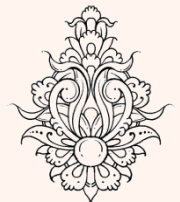
# FFT / DCT

مقادیر ضرایب در  $DCT$  دقیق‌تری ولی در  $DFT$  مختلطند

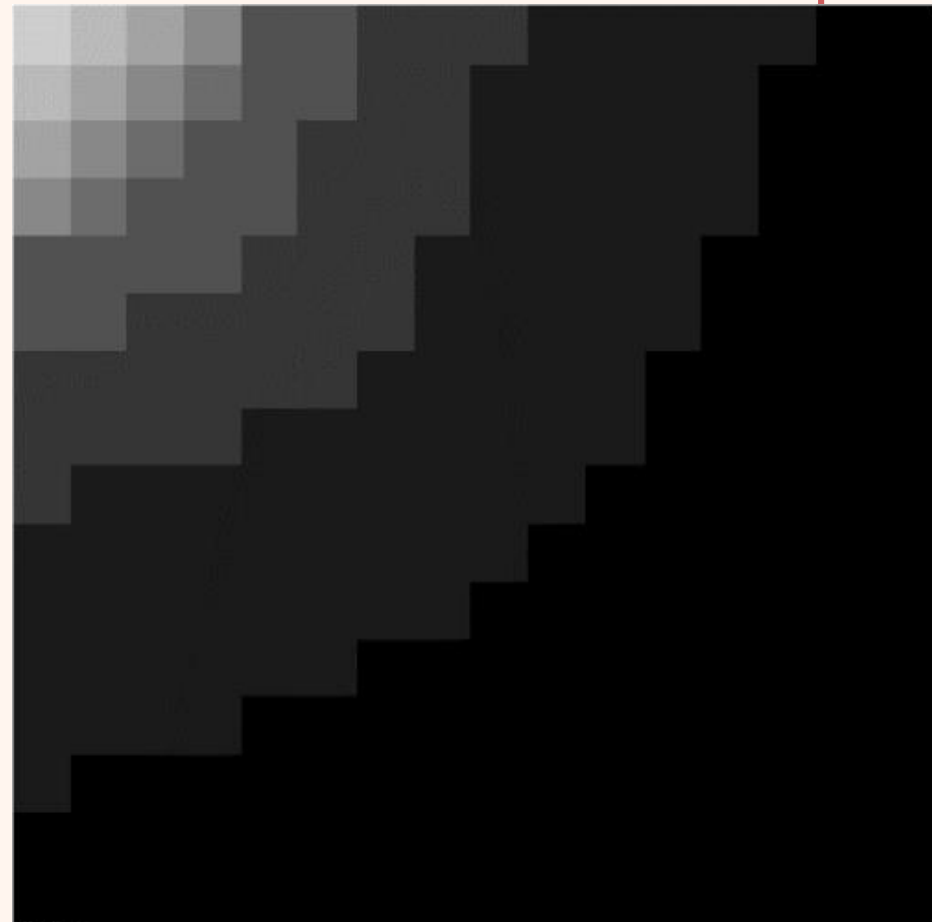
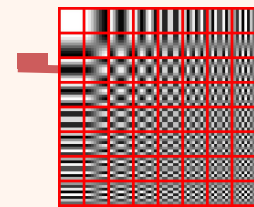
تقریب بهتر در  $DCT$  با استفاده از ضرایب کم‌تر



<http://www.cs.cf.ac.uk/Dave/Multimedia/node231.html>

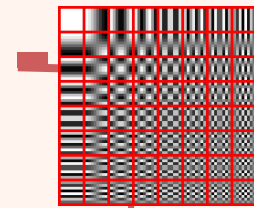


# اهمیت نسبی ضرایب تبدیل کسینوسی



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

# مثال



```
f = imread('cameraman.tif');  
imshow(f, [ ]);  
J = dct2(f);  
figure;  
imshow(log(abs(J)), [ ]), colormap(gray),  
colorbar;title('DCT2');  
figure  
imshow(log(abs(J)), [ ]), colormap(jet(64)),  
colorbar;title('DCT2');  
K = idct2(J);  
figure;imshow(K, [ ]);
```



Rec

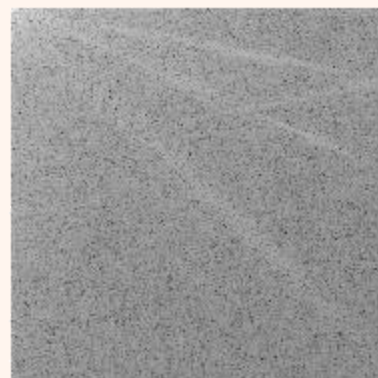


۱۴

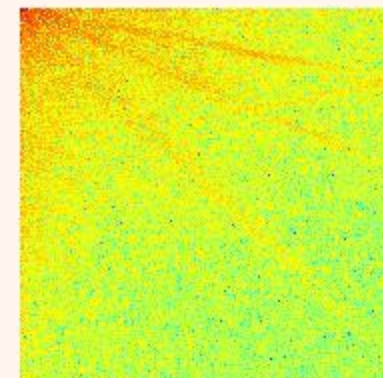


Org

DCT2



DCT2

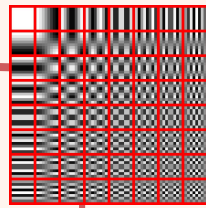


```

f = imread('cameraman.tif');
imshow(f, []);
J = dct2(f);
a=ones(size(f));
a(:,(size(f,2)/2):end)=0;
J2=J.*a;
figure;imshow(log(abs(J2)), []);
K = idct2(J2);
figure;imshow(uint8(K), []);
Diff=f-uint8(K);
figure;imshow(Diff, []);

```

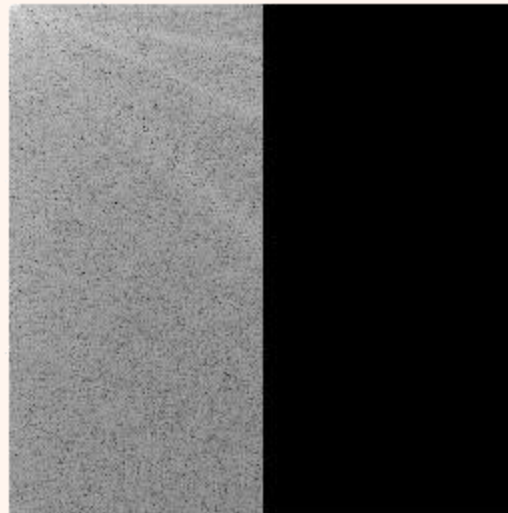
# مثال



*difference*

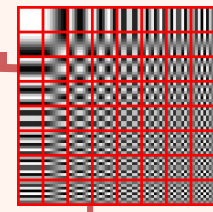


*Org*



*Rec*

# مثال



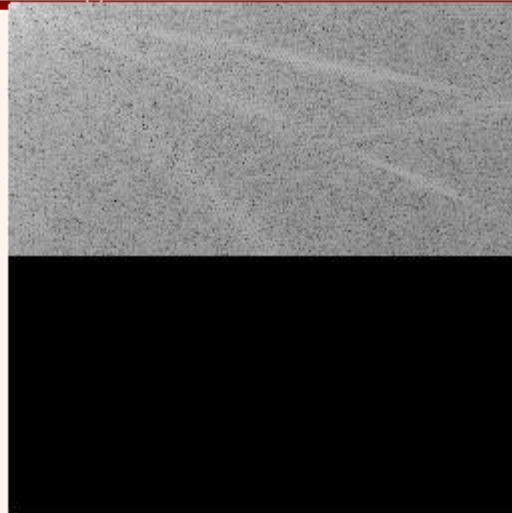
```
f = imread('cameraman.tif');  
imshow(f, []);  
J = dct2(f);  
a=ones(size(f));  
a((size(f,2)/2):end,:)=0;  
J2=J.*a;  
figure;imshow(log(abs(J2)), []);  
K = idct2(J2);  
figure;imshow(uint8(K), []);  
Diff=f-uint8(K);  
figure;imshow(Diff, []);
```



difference



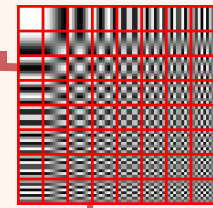
Org



Rec



# مثال



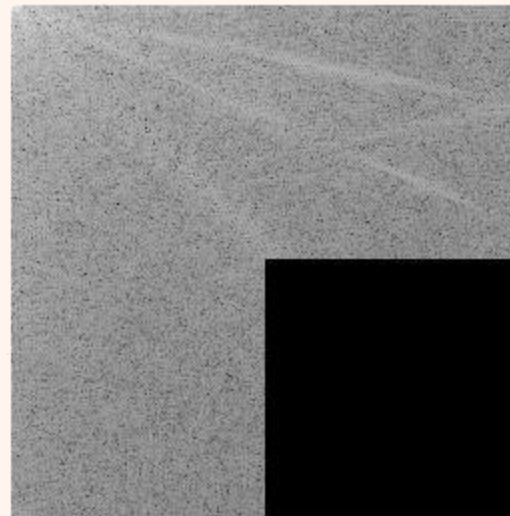
```
f = imread('cameraman.tif');
imshow(f, [ ]);
J = dct2(f);
a=ones(size(f));
a((size(f,2)/2):end, (size(f,2)/2):end)
=0;
J2=J.*a;
figure;imshow(log(abs(J2)), [ ]);
K = idct2(J2);
figure;imshow(uint8(K), [ ]);
Diff=f-uint8(K);
figure;imshow(Diff, [ ]);
```



difference

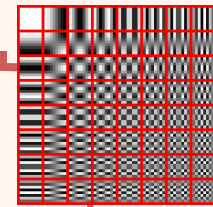


Org



Rec

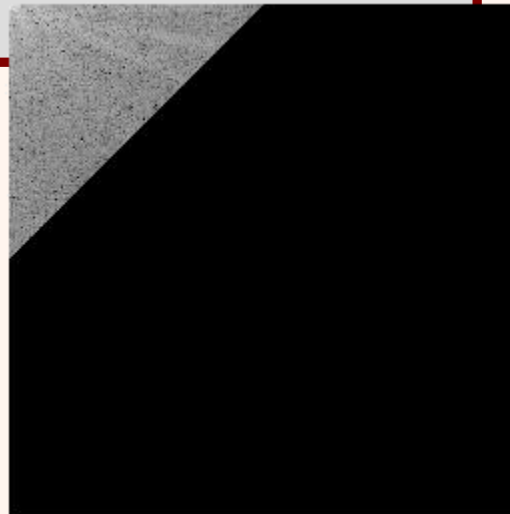
# مثال



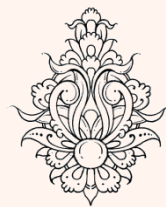
```
f = imread('cameraman.tif');  
imshow(f, []);  
J = dct2(f);  
figure; imshow(log(abs(J)), []);  
a=zeros(size(f));  
for i=1:size(f,2)/2  
    for j=1:((size(f,2)/2)-i)  
        a(i,j)=1;  
    end  
end  
J2=J.*a;  
figure; imshow(log(abs(J2)), []);  
K = idct2(J2);  
figure; imshow(uint8(K), []);  
Diff=f-uint8(K);  
figure; imshow(Diff, []);
```



*difference*

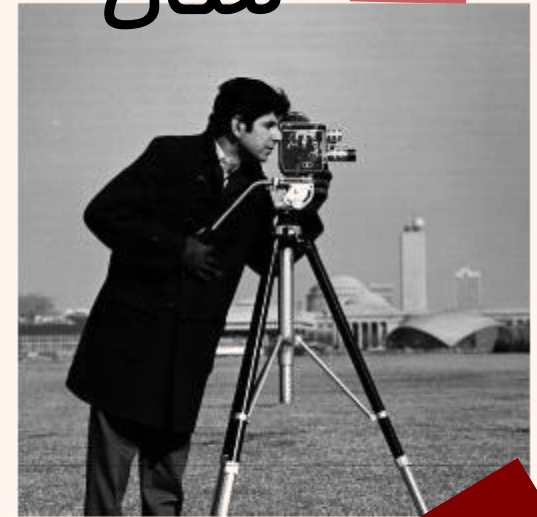


- در بسیاری از کاربردها در پردازش تصویر، بلوک بندی تصویر انجام شده و پس از آن هر بلوک مورد پردازش قرار می گیرد.
- این مسأله سبب می شود بتوان به صورت موازی به روی بخش های کوچک تری از تصویر پردازش همزمان صورت داد.
- بازده سیستم افزایش یافته، سرعت پردازش بالا می رود.
- معمولاً اندازهی بلوکها توانی از دو است.
  - بسته به کاربرد اندازه از  $2 \times 2$  و بالاتر متخیر است.
- در بیشتر موارد (استانداردها) اندازهی بلوکها  $8 \times 8$  در نظر گرفته می شود.

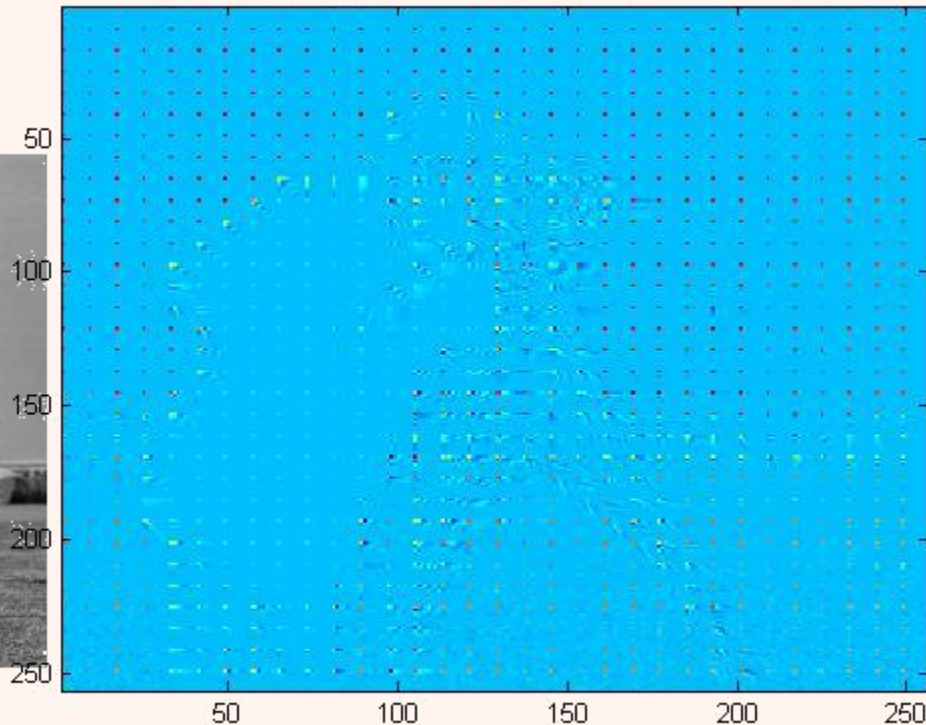
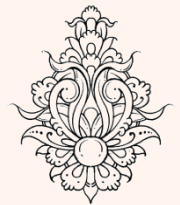


# مثال

```
siz=8;  
I = imread('cameraman.tif');  
fun = @dct2;  
J = blkproc(I,[siz siz],fun);  
figure;imshow(I,[ ]);  
figure;imagesc(J), colormap(jet);  
fun = @idct2;  
I2 = blkproc(J,[siz siz],fun);  
figure;imshow(I2,[ ]);
```



Rec



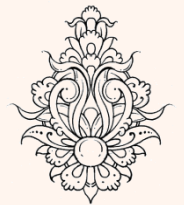
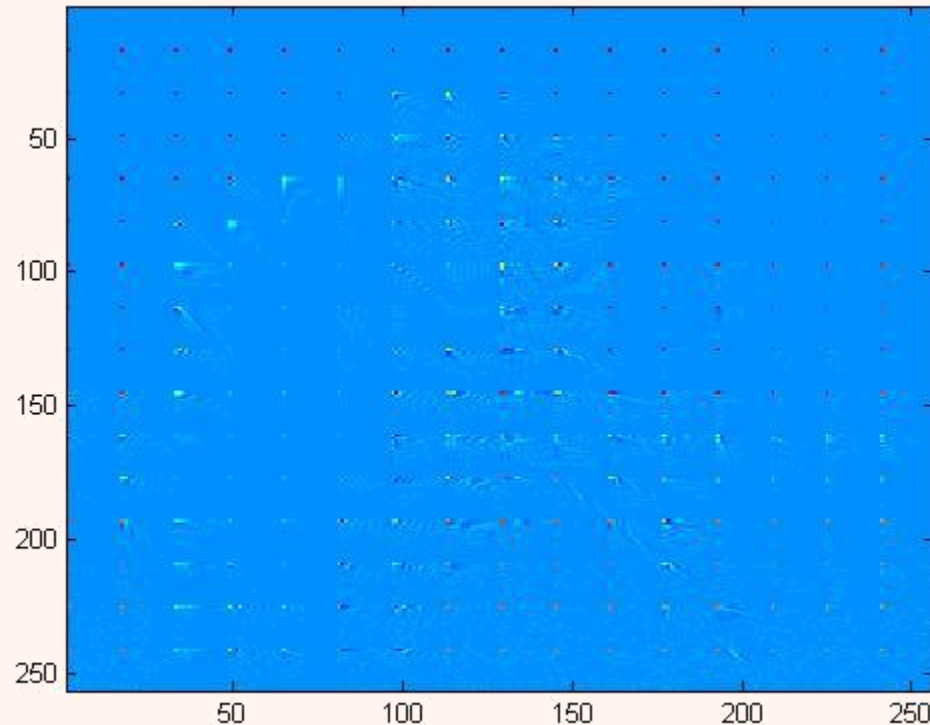
```
siz=16;  
I = imread('cameraman.tif');  
fun = @dct2;  
J = blkproc(I,[siz siz],fun);  
figure;imshow(I,[ ]);  
figure;imagesc(J), colormap(jet);  
fun = @idct2;  
I2 = blkproc(J,[siz siz],fun);  
figure;imshow(I2,[ ]);
```



Rec



Org



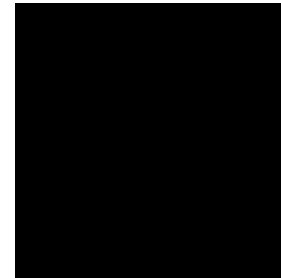
```

siz=16;
I = imread('cameraman.tif');
fun = @dct2;
J = blkproc(I,[siz siz],fun);
figure;imshow(I, []);
figure;imagesc(J), colormap(hot);
a=ones(siz);a(3:siz,3:siz)=0;
figure;imshow(a, []);
fun = @(x) (x).*a;
J2 = blkproc(J,[siz siz],fun);
figure;imagesc(J2), colormap(hot);
fun = @idct2;
I2 = blkproc(J2,[siz siz],fun);
figure;imshow(I2, []);

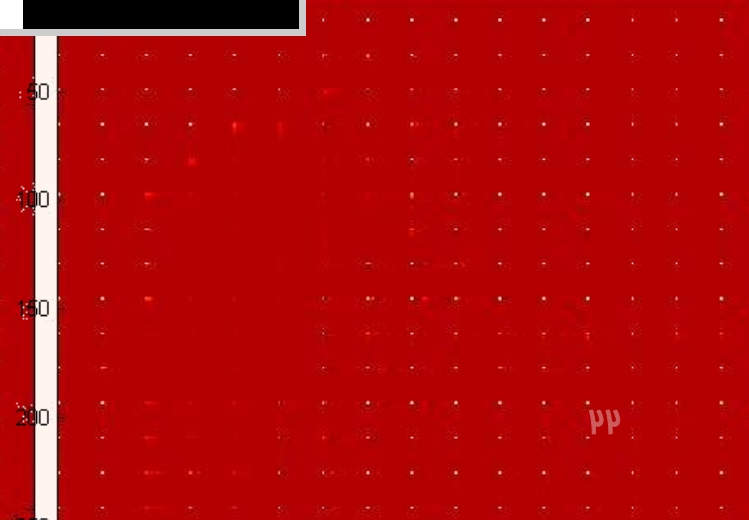
```



Rec



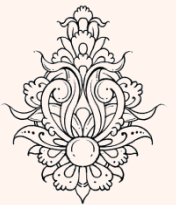
Org



# مثال



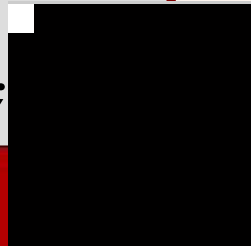
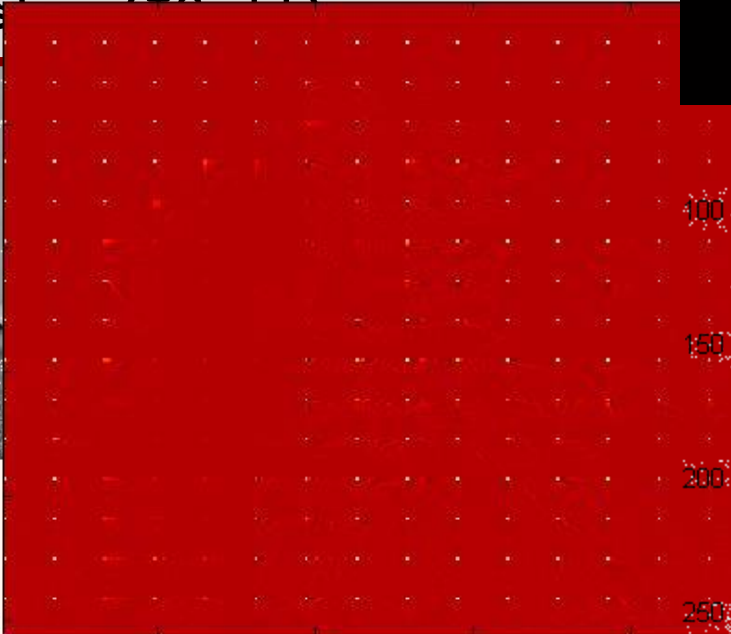
*8x8 block implementation*



```
siz=16;I = imread('cameraman.tif');
fun = @dct2;
J = blkproc(I,[siz siz],fun);
figure;imshow(I,[]);
figure;imagesc(J), colormap(hot);
a=zeros(siz);a(1:2,1:2)=1;
figure;imshow(a,[]);
fun = @(x) (x).*a;
J2 = blkproc(J,[siz siz],fun);
figure;imagesc(J2), colormap(hot);
fun = @idct2;
I2 = blkproc(J2,[siz siz],fun);
figure;imshow(I2,[]);
```



Rec





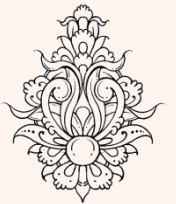
# کدگذاری دامنه‌ی تبدیل

## • مراحل

- اعمال تبدیل به روی تصویر
- کدگذاری ضرایب تبدیل
- ارسال داده‌ها
- بازسازی تصویر با استفاده از داده‌های ارسالی

## • مزایا

- انرژی در ضرایب محدودی فشرده‌شده است.
- مقاومت در برابر خطاهای کانال ارسالی



# کدگذاری

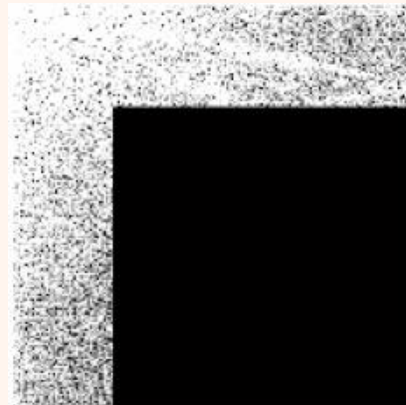
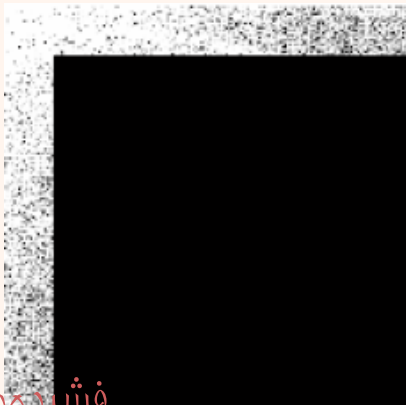
## • کدگذاری

– ناحیه‌ای (Zonal Coding)

• تنها ناحیه‌ای از ضرایب ارسال می‌شود.

– آستانه‌ای (Threshold Coding)

• تنها ضرایبی ارسال می‌شود که از میزان آستانه بالاتر باشد.

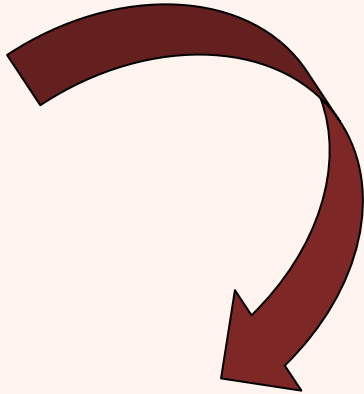


Zonal Coding



تازشکا  
بهبیتو

|     |     |     |    |
|-----|-----|-----|----|
| 120 | 134 | 24  | 17 |
| 145 | 145 | 230 | 25 |
| 16  | 234 | 23  | 18 |
| 23  | 24  | 28  | 19 |



**Compression!!!**

|           |          |           |           |
|-----------|----------|-----------|-----------|
| 306.2500  | 104.8835 | -114.7500 | -45.3384  |
| 100.0207  | 62.2961  | -1.1577   | 30.1075   |
| -111.7500 | -6.2990  | 99.2500   | 5.0445    |
| -55.7716  | 26.1075  | -7.3678   | -160.7961 |



# چندی کردن

## Quantization

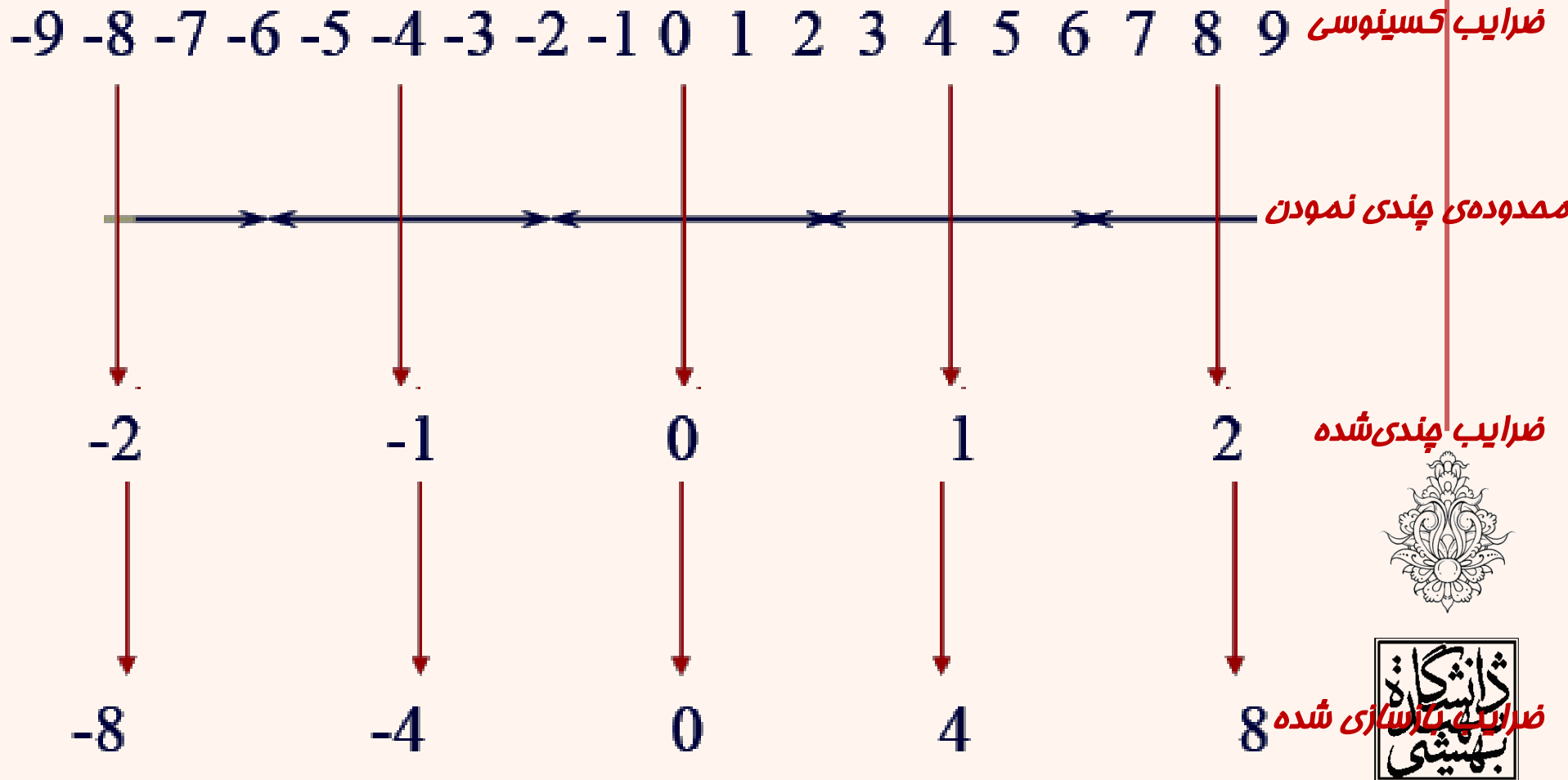
- به وسیله‌ی چندی کردن از دقت ضرایب کسینوسی کاسته می‌شود به گونه‌ای که تبدیل به مقادیر صحیح گردند.
- این مساله باعث می‌شود که بسیاری از ضرایب فرکانس‌های بالا به صفر تبدیل گردند.
- آستانه‌ای که با استفاده از آن چندی شدن صورت می‌پذیرد به گونه‌ای انتخاب می‌شود که نتیجه‌ی بازسازی تا حد ممکن برای چشم انسان قابل رویت نباشد.



# مثال

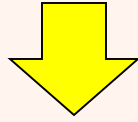
Quantization =

4



|      |     |     |     |     |     |     |     |
|------|-----|-----|-----|-----|-----|-----|-----|
| -415 | -33 | -58 | 35  | 58  | -51 | -15 | -12 |
| 5    | -34 | 49  | 18  | 27  | 1   | -5  | 3   |
| -46  | 14  | 80  | -35 | -50 | 19  | 7   | -18 |
| -53  | 21  | 34  | -20 | 2   | 34  | 36  | 12  |
| 9    | -2  | 9   | -5  | -32 | -15 | 45  | 37  |
| -8   | 15  | -16 | 7   | -8  | 11  | 4   | 7   |
| 19   | -28 | -2  | -26 | -2  | 7   | -44 | -21 |
| 18   | 25  | -12 | -44 | 35  | 48  | -37 | -3  |

**ضرایب DCT**

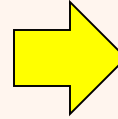


$$\text{round}\left(\frac{-415}{16}\right) = \text{round}(-25.9375) = -26$$

**ماتریس پندی کننده**

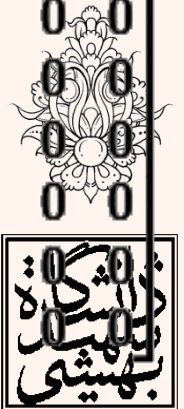
|    |    |    |    |     |     |     |     |
|----|----|----|----|-----|-----|-----|-----|
| 16 | 11 | 10 | 16 | 24  | 40  | 51  | 61  |
| 12 | 12 | 14 | 19 | 26  | 58  | 60  | 55  |
| 14 | 13 | 16 | 24 | 40  | 57  | 69  | 56  |
| 14 | 17 | 22 | 29 | 51  | 87  | 80  | 62  |
| 18 | 22 | 37 | 56 | 68  | 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  |

**ماتریس پندی کننده  
استاندارد برای بازسازی  
استفاده خواهد شد.**



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

**ماتریس نتیجه‌ی پندی  
شده**



# مراحل الگوریتم JPEG

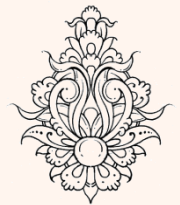
- بلوک بندی تصاویر

– تصویر  $f(m, n)$  ← زیر تصاویرهای  $8 \times 8$   $f_i(m, n)$

– فرض می‌کنیم مقادیر روشنایی  $[0, 2^L - 1]$

- هر  $f_i$  با رابطه‌ی زیر به  $g_i$  تبدیل می‌شود:

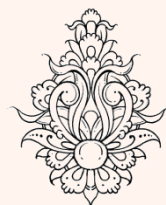
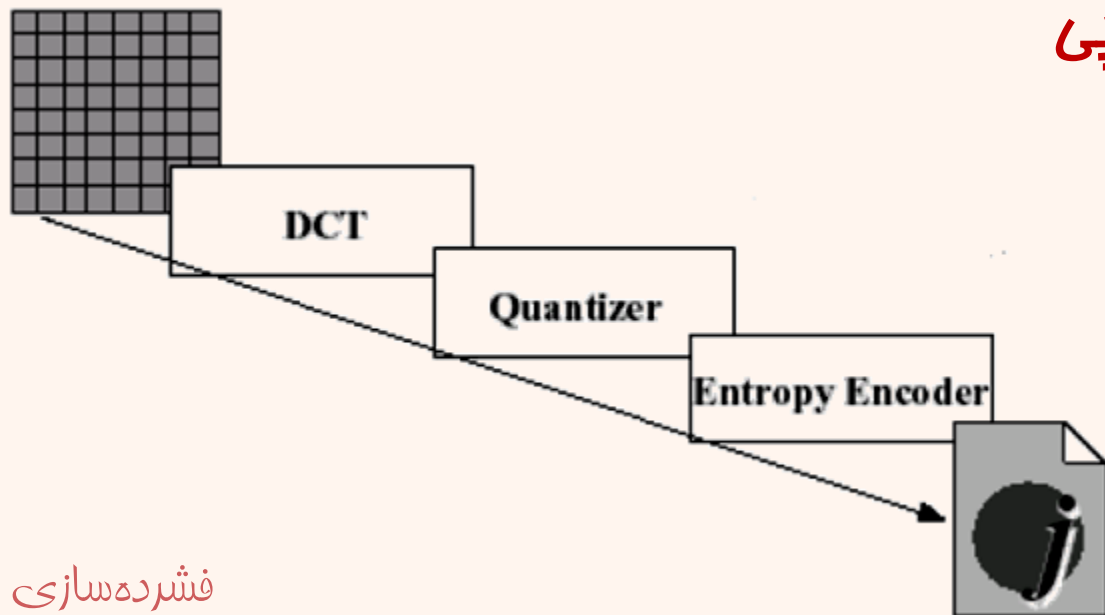
$$g_i(m, n) = f_i(m, n) - 2^{L-1}$$



# مراحل الگوریتم JPEG (ادامه...)

• چگونگی الگوریتم فشرده‌سازی

- بلوک‌بندی تصویر
- اعمال تبدیل گسسته‌ی کسینوسی
- چندی نمودن ضرایب
- پویش زیگزاگ ضرایب
- کدگذاری آنتروپی
- ارسال





# مراحل الگوریتم jpeg (ادامه...)

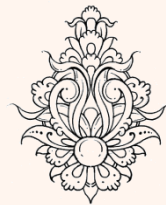
- $g_i(m, n)$  ورودی تبدیلی DCT هستند که نتیجه‌ای همانند  $D_i(k_1, k_2)$  دارند.
- ماتریس چندی‌کننده‌ای در نظر گرفته شده رابطه‌ی زیر به دست می‌آید:

$$D_{iq}(k_1, k_2) = \text{round}\left[\frac{D_i(k_1, k_2)}{T(k_1, k_2) \times \alpha}\right]$$

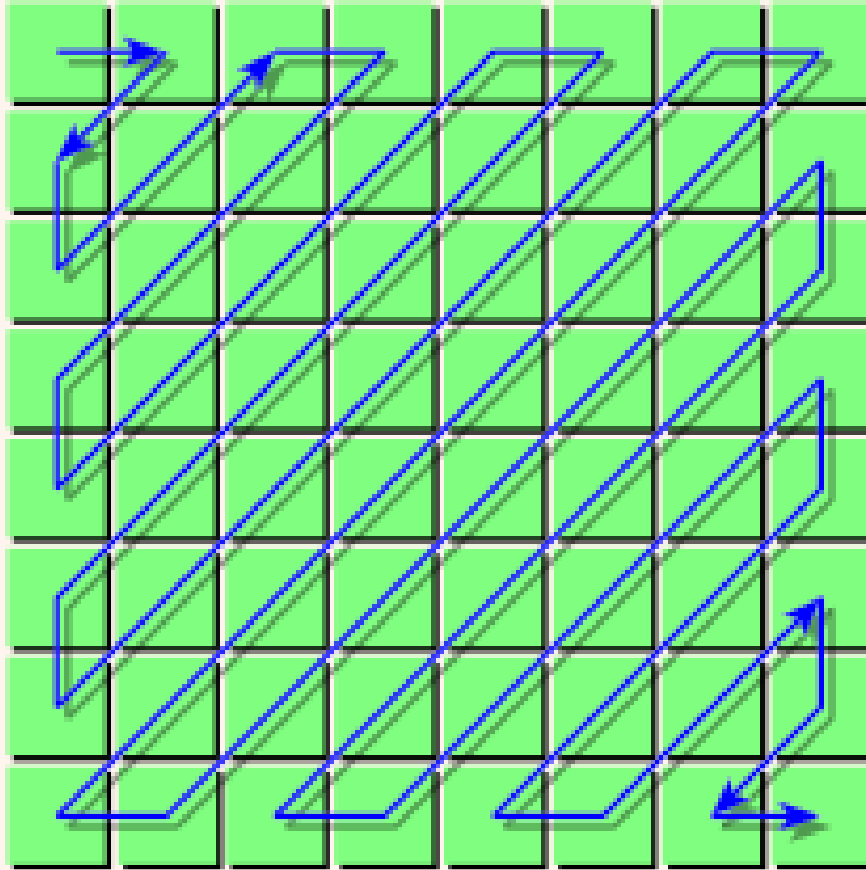
**میزان ریزش** →

- مرتب نمودن زیگزاگ
- رمزگذاری مقادیر

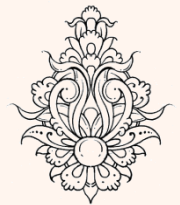
DC –  
AC –



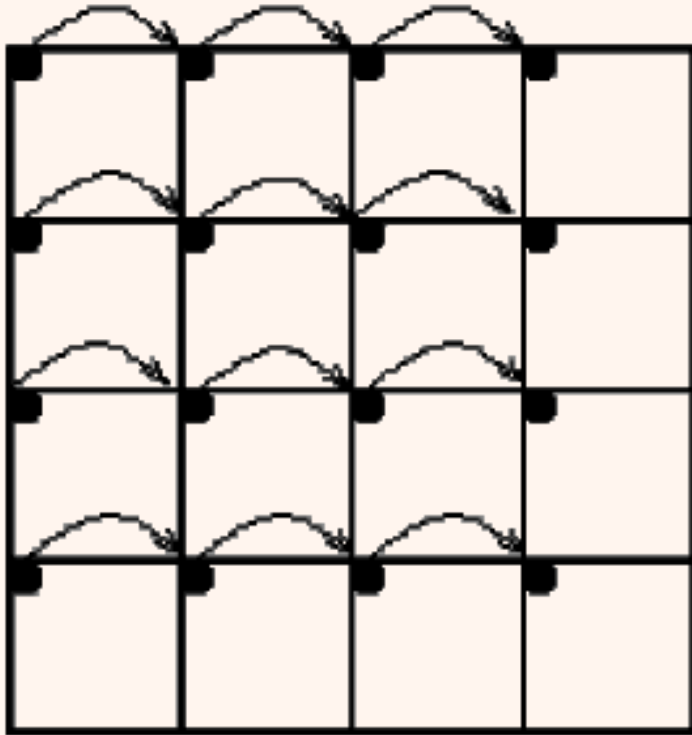
# پویش ضرایب



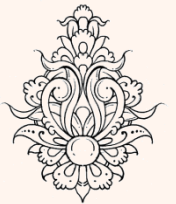
پویش ضرایب به صورت زیگزاگ صورت می‌پذیرد



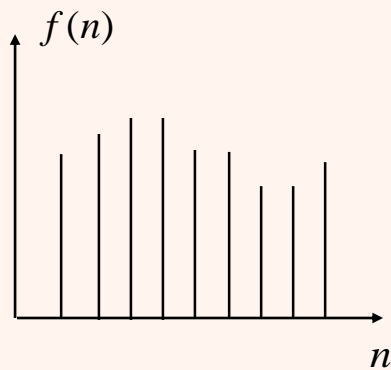
# کدگذاری مؤلفه‌های DC



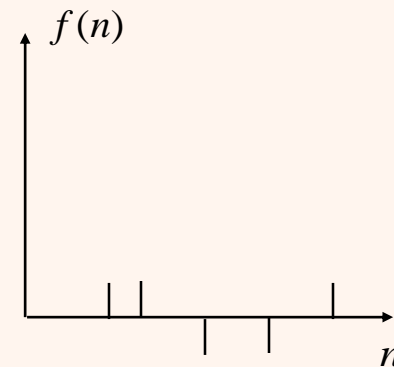
اختلاف مؤلفه‌های DC بلوک‌های متوالی کد می‌شوند



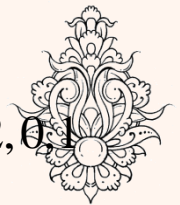
## Differential Pulse Code Modulation (DPCM)



$$f(n) = 156, 157, 158, 158, 156, 156, 154, 154, 155$$



$$\delta f(n) = 156, 1, 1, 0, -2, 0, -2, 0$$

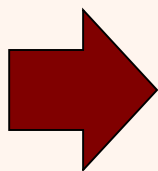


# Run-Level coding

- در این شیوهی کد نمودن از تعداد صفرها و عناصر غیر صفر استفاده می‌شود:

|     |     |    |    |    |    |   |   |
|-----|-----|----|----|----|----|---|---|
| 102 | -33 | -3 | -4 | -2 | -1 | 0 | 0 |
| 21  | -2  | -3 | 0  | -1 | 0  | 0 | 0 |
| -3  | 0   | 1  | 0  | 0  | 0  | 0 | 0 |
| 2   | 0   | 0  | 0  | 1  | 0  | 0 | 0 |
| 1   | 0   | 0  | 0  | 0  | 0  | 0 | 0 |
| -2  | 0   | 0  | 0  | 0  | 0  | 0 | 0 |
| 0   | 0   | 0  | 0  | 0  | 0  | 0 | 0 |
| 0   | 0   | 0  | 0  | 0  | 0  | 0 | 0 |

**$[-33, 21, -3, -2, -3, -4, -3, 0,$   
 $2, 1, 0, 1, 0, -2, -1, -1, 0, 0, 0,$   
 $-2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,$   
 $1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,$   
 $0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,$   
 $0, 0, 0, 0, 0, 0]$**



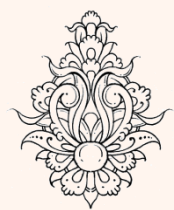
**$(0, -33) (0, 21) (0, -3) (0, -2) (0, 3) (0, -4) (0, -3)$   
 $(1, 2) (0, 1) (1, 1) (1, -2) (0, -1) (0, -1) (3, -2)$   
 $(11, 1)$**



| Coefficient to be transmitted | etc.<br>↑<br>Coefficient code | Size parameter |
|-------------------------------|-------------------------------|----------------|
| 15                            | 1 1 1 1                       | 4              |
| 14                            | 1 1 1 0                       |                |
| 13                            | 1 1 0 1                       |                |
| 12                            | 1 1 0 0                       |                |
| 11                            | 1 0 1 1                       |                |
| 10                            | 1 0 1 0                       |                |
| 9                             | 1 0 0 1                       |                |
| 8                             | 1 0 0 0                       |                |
| 7                             | 1 1 1                         | 3              |
| 6                             | 1 1 0                         |                |
| 5                             | 1 0 1                         |                |
| 4                             | 1 0 0                         |                |
| 3                             | 1 1                           | 2              |
| 2                             | 1 0                           |                |
| 1                             | 1                             | 1              |
| -1                            | 0                             |                |
| -2                            | 0 1                           | 2              |
| -3                            | 0 0                           |                |
| -4                            | 0 1 1                         | 3              |
| -5                            | 0 1 0                         |                |
| -6                            | 0 0 1                         |                |
| -7                            | 0 0 0                         |                |
| -8                            | 0 1 1 1                       | 4              |
| -9                            | 0 1 1 0                       |                |
| -10                           | 0 1 0 1                       |                |
| -11                           | 0 1 0 0                       |                |
| -12                           | 0 0 1 1                       |                |
| -13                           | 0 0 1 0                       |                |
| -14                           | 0 0 0 1                       |                |
| -15                           | 0 0 0 0                       |                |
|                               | etc.<br>↓                     |                |

**(RUN, CAT)**

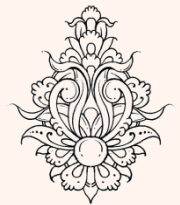
*CAT is the category for the amplitude of a nonzero coefficient in the zigzag order, and RUN is the number of zeros preceding this nonzero coefficient*



# گروه‌بندی مقادیر

| 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         |

Table is from slides at Gonzalez/ Woods DIP book website (Chapter

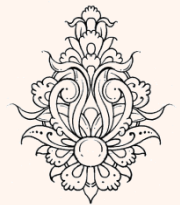


| Category | Base Code | Length | Category | Base Code | Length |
|----------|-----------|--------|----------|-----------|--------|
| 0        | 010       | 3      | 6        | 1110      | 10     |
| 1        | 011       | 4      | 7        | 11110     | 12     |
| 2        | 100       | 5      | 8 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

ضرایب DC

ضرایب AC

| 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          | 11111111000000   | 17     |
| 0/3          | 100             | 6      | 8/3          | 111111110110111  | 19     |
| 0/4          | 1011            | 8      | 8/4          | 111111110111000  | 20     |
| 0/5          | 11010           | 10     | 8/5          | 111111110111001  | 21     |
| 0/6          | 111000          | 12     | 8/6          | 111111110111010  | 22     |
| 0/7          | 1111000         | 14     | 8/7          | 111111110111011  | 23     |
| 0/8          | 1111110110      | 18     | 8/8          | 111111110111100  | 24     |
| 0/9          | 111111110000010 | 25     | 8/9          | 111111110111101  | 25     |
| 0/A          | 111111110000011 | 26     | 8/A          | 111111110111110  | 26     |
| 1/1          | 1100            | 5      | 9/1          | 111111000        | 10     |
| 1/2          | 111001          | 8      | 9/2          | 111111110111111  | 18     |
| 1/3          | 1111001         | 10     | 9/3          | 111111111000000  | 19     |
| 1/4          | 111110110       | 13     | 9/4          | 111111111000001  | 20     |
| 1/5          | 11111110110     | 16     | 9/5          | 1111111111000010 | 21     |





## Codeword for the DC coefficient

| Category | Base Code | Length |
|----------|-----------|--------|
| 0        | 010       | 3      |
| 1        | 011       | 4      |
| 2        | 100       | 5      |
| 3        | 00        | 5      |
| 4        | 101       | 7      |
| 5        | 110       | 4      |

DC

$DIFF = 31 - 28 = 3$



$CAT = 2$

CAT is 100



$DIFF = 3 > 0$



the DC coefficient is 10011

## Codeword for the AC coefficients

$AC=18$

the codeword for (0, 5) is 11010



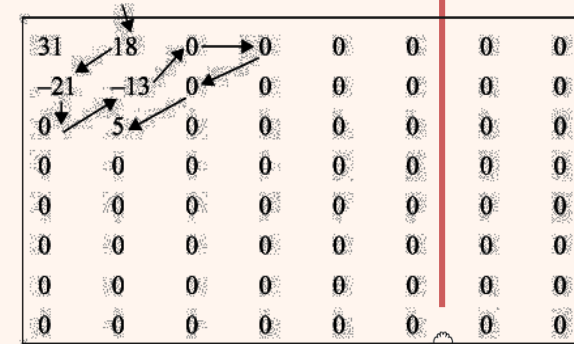
1101010010

$AC=-21$

the codeword for (0, 5) is 11010



1101001010



AC

| Run/Category | Base      |
|--------------|-----------|
| 0/0          | 1010 (=1) |
| 0/1          | 00        |
| 0/2          | 01        |
| 0/3          | 100       |
| 0/4          | 1011      |
| 0/5          | 11010     |

| 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           |

مدان

-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 EOB

|    |              |
|----|--------------|
| -3 | (0/2)=0100   |
| 1  | (0/1)=001    |
| -3 | (0/2)=0100   |
| -2 | (0/2)=0101   |
| -6 | (0/3)=100001 |
| 2  | (0/2)=0110   |

|    |              |
|----|--------------|
| -4 | (0/3)=100011 |
| 1  | (0/1)=001    |
| -4 | (0/3)=100011 |
| 1  | (0/1)=001    |
| 1  | (0/1)=001    |
| 5  | (0/3)=100101 |

|     |                |
|-----|----------------|
| 2   | (1/2)=11100110 |
| -1  | (2/1)=110110   |
| 2   | (0/2)=0110     |
| -1  | (5/1)=11110100 |
| -1  | (0/1)=000      |
| EOB | 1010           |

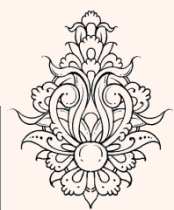
قطار بیت ارسال برای ضرایب AC

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

تعداد بیت‌های ارسال 85 بیت

| 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         |

| Run/Category | Base Code    | Length |
|--------------|--------------|--------|
| 0/0          | 1010 (= EOB) | 4      |
| 0/1          | 00           | 3      |
| 0/2          | 01           | 4      |
| 0/3          | 100          | 6      |
| 0/4          | 1011         | 8      |
| 0/5          | 11010        | 10     |



تراشه‌نگاره  
به‌پیشی



Original

bpp = 8.00 mse = 0.00



JPEG

bpp = 1.00 mse = 17.26



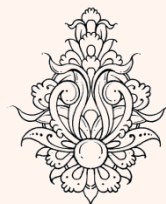
JPEG

bpp = 0.50, mse = 33.08

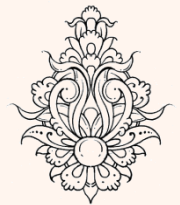
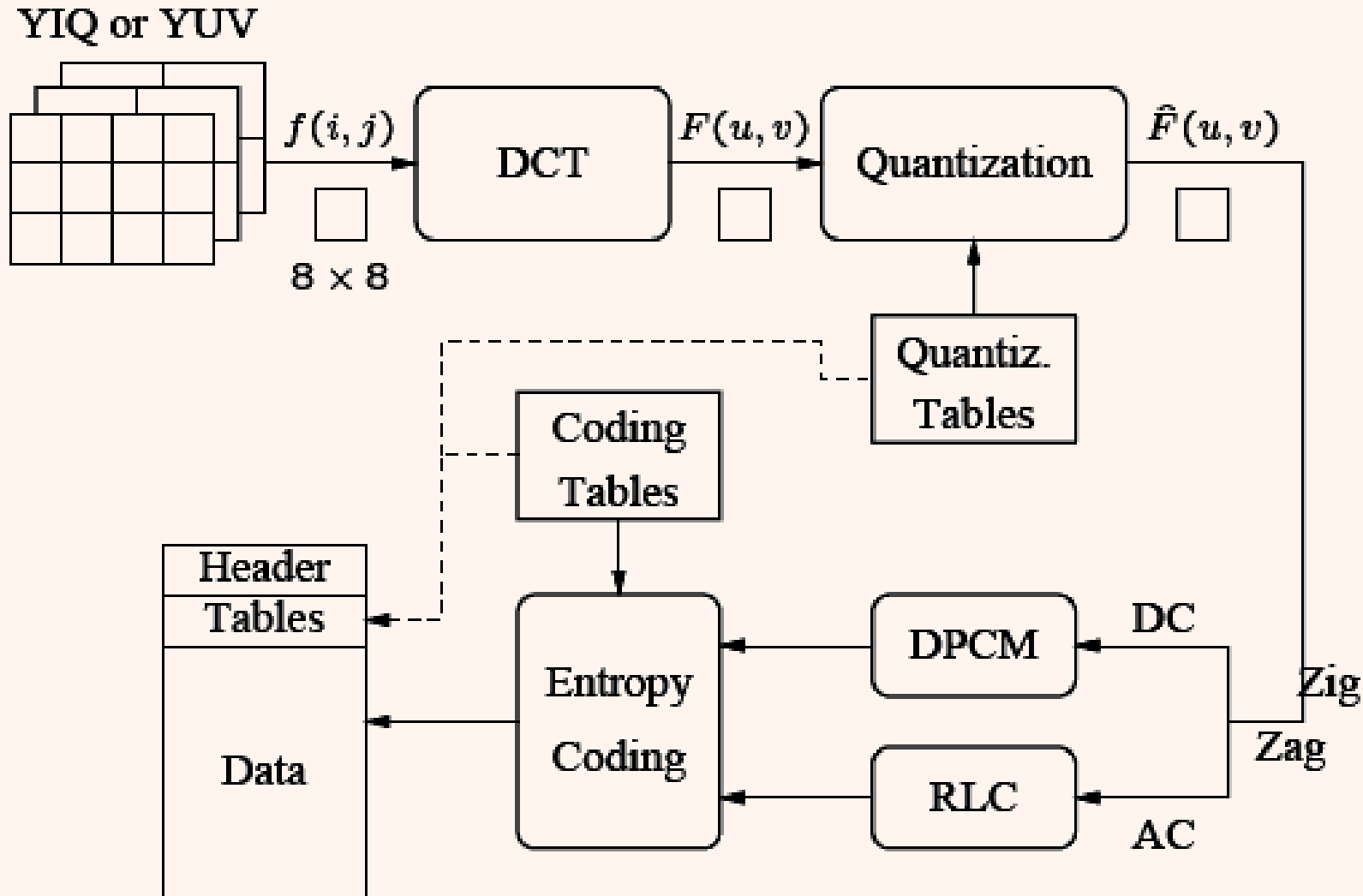


JPEG

bpp = 0.25, mse = 79.11



Block diagram for JPEG encoder.



# مقایسه‌ی تصویر

• برای مقایسه‌ی تصویر بهترین معیار مشاهده‌ی آن است

– معیار کیفی است

– وابسته به احساس بشر

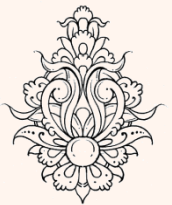
• معیار کمی

– مجموع مربعات خطا (MSE)

– نسبت توان سیگنال به نویز (SNR)

$$MSE = \frac{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} [f(m,n) - \hat{f}(m,n)]^2}{MN}$$

$$SNR = \frac{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n)^2}{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} [f(m,n) - \hat{f}(m,n)]^2}$$



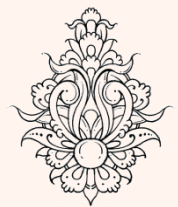
peak signal-to-noise ratio

• پاسخ بر پایه‌ی db است

$$PSNR = 10 \times \log\left(\frac{255^2}{MSE}\right)$$

• به صورت کلی اگر هر پیکسل را با B بیت بتوان نشان داد، فرمول مذکور به صورت زیر تغییر می‌نماید

$$PSNR = 10 \times \log\left(\frac{(2^B - 1)^2}{MSE}\right)$$



# آیا PSNR معیار خوبی است؟

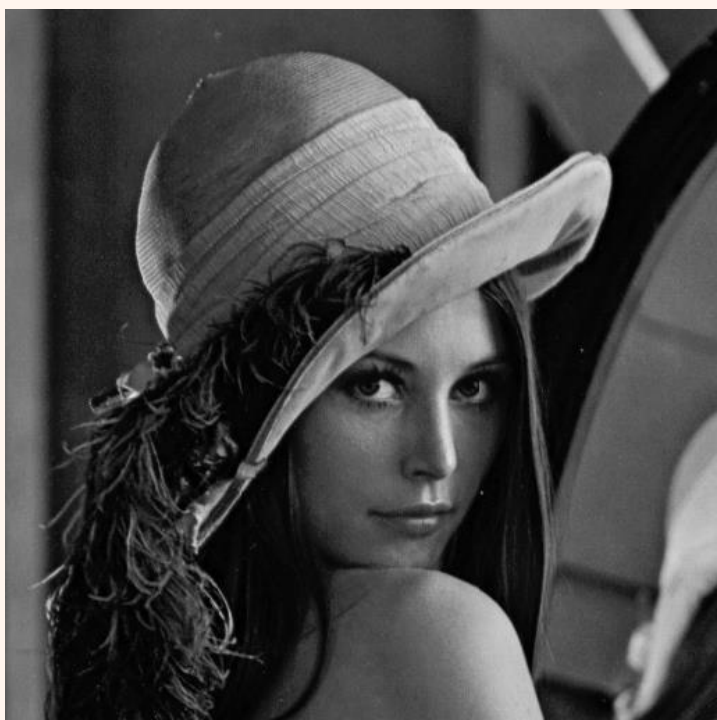


$MSE=100.006$



$MSE=48.82$

دانشگاه  
تهران  
پیشین



***PSNR=22.78***



***PSNR=22.97***



***PSNR=22.87***



تراشگاه  
سپیدی  
بهشتی



# استانداردهای جدید

JPEG

BPG

39.2 KB ← → 40.3 KB



# استاندارهای جدید

JPEG

BPG



**The blocking, color banding and aliasing artifacts of heavily compressed JPEG (left), while the heavily compressed BPG (right) looks much**

*smoother*

