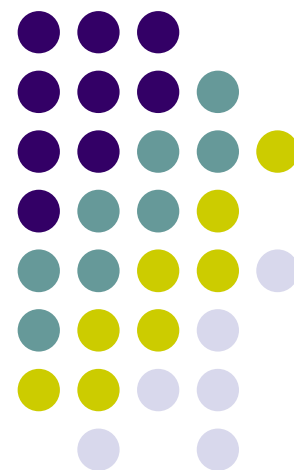




# 信息隐藏技术基础

王莘



哈爾濱工業大學  
HARBIN INSTITUTE OF TECHNOLOGY

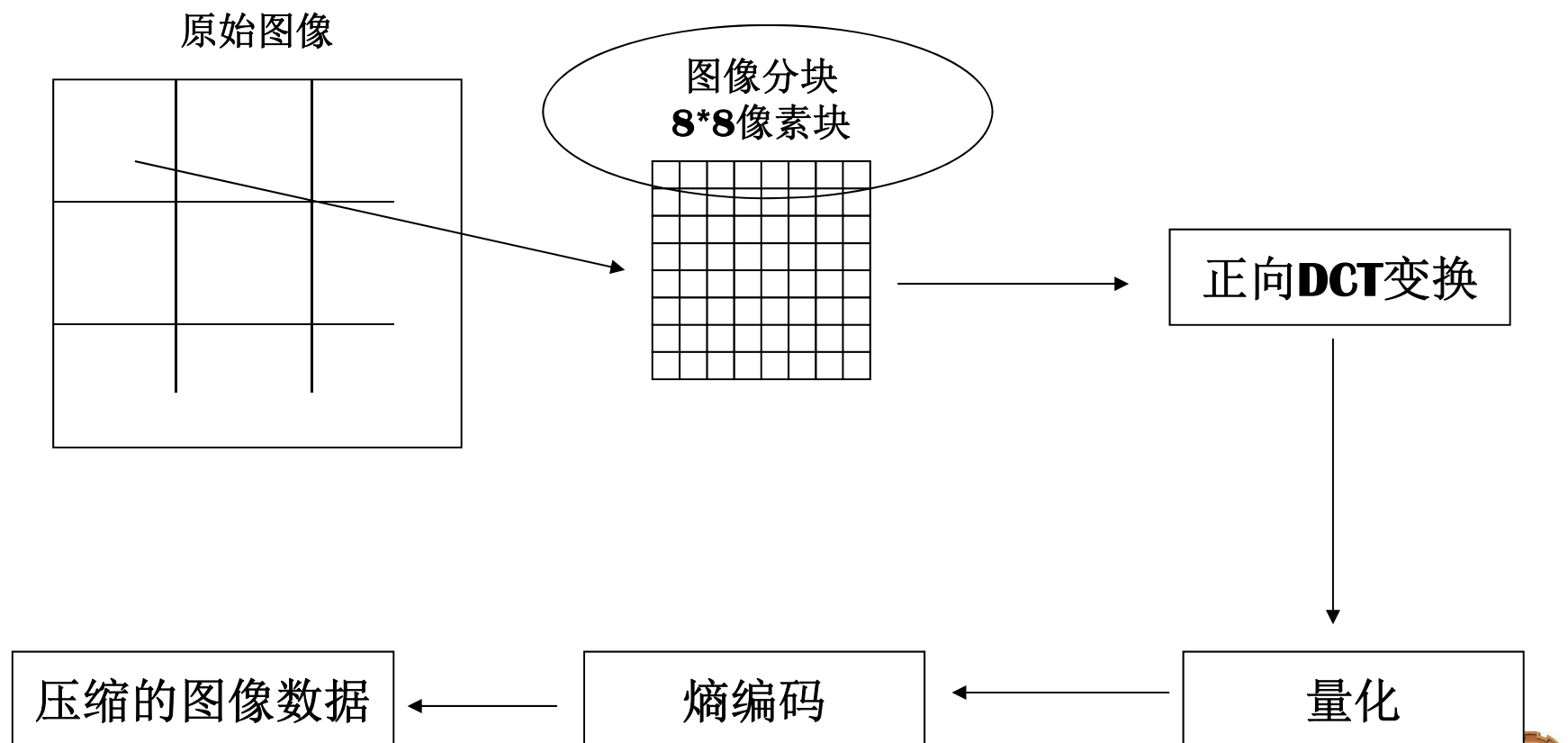
---

# 面向JPEG图像的隐写方法

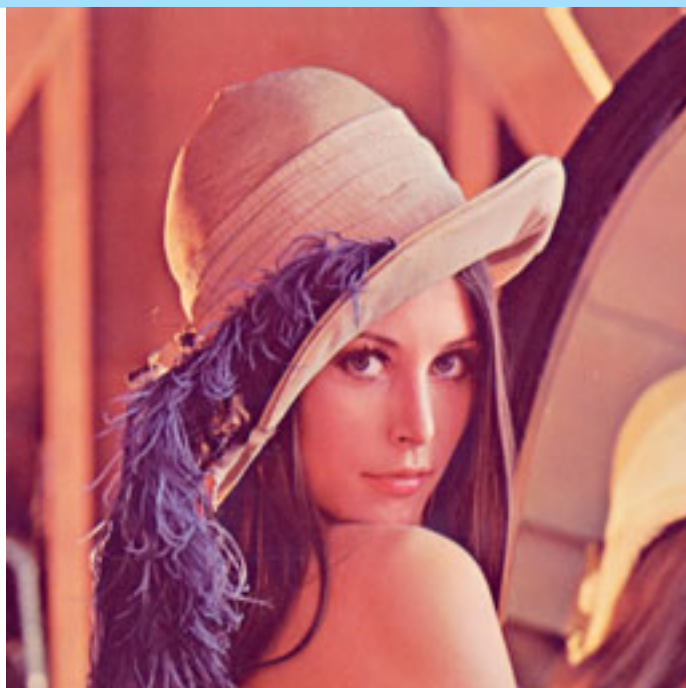


# JPEG图像压缩过程

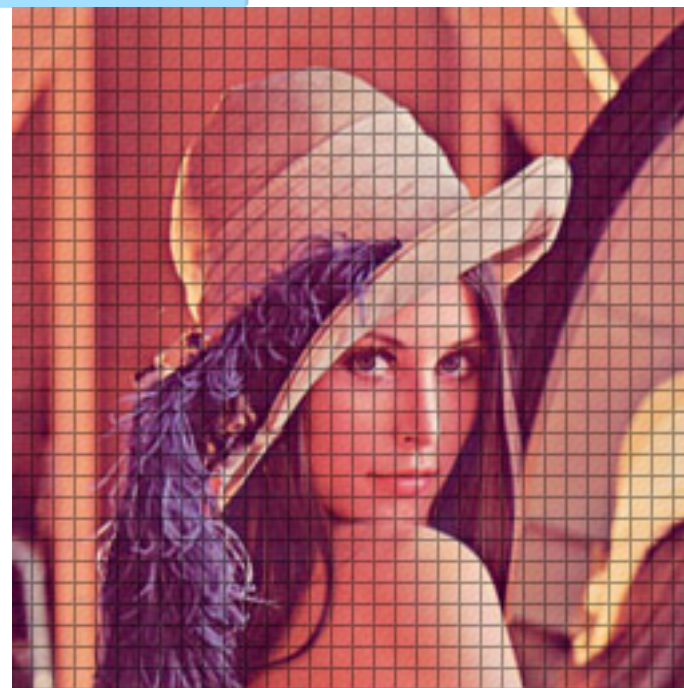
**JPEG** 是**Joint Photographic Experts Group**（联合图像专家小组）的缩写，是第一个国际图像压缩标准。



# 图像分块

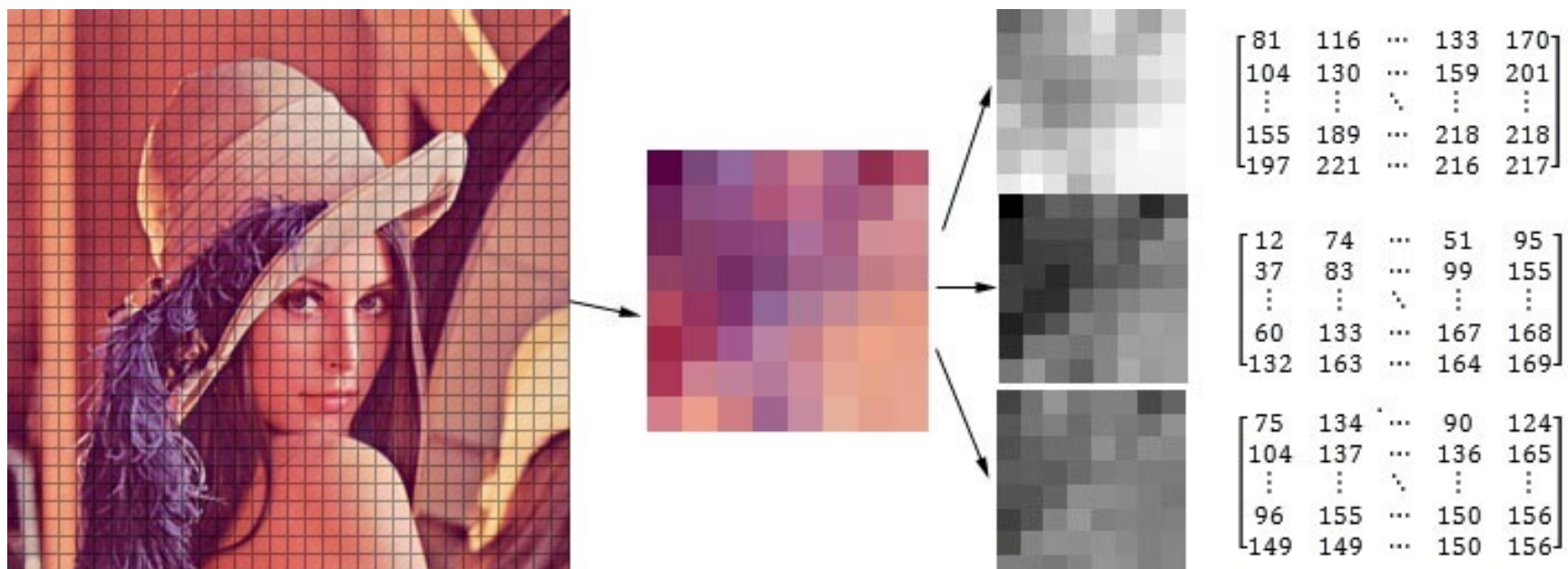


分块  
→



图像被分割成大小为 $8 \times 8$ 的小块，这些小块在整个压缩过程中都是单独被处理的。据说是世界上第一张JPEG图片，这张图片自从诞生之日开始，就和图像处理结下渊源，陪伴了无数理工男度过了的一个个不眠之夜，可谓功勋卓著。

# 图像预处理



JPEG压缩算法中，需要把图案转换成为YCbCr模型，这里的Y表示亮度(Luminance)，Cb和Cr分别表示绿色和红色的“色差值”。

# 图像预处理



$$Y = 0.299R + 0.587G + 0.114B$$

$$C_b = -0.1687R - 0.3313G + 0.5B$$

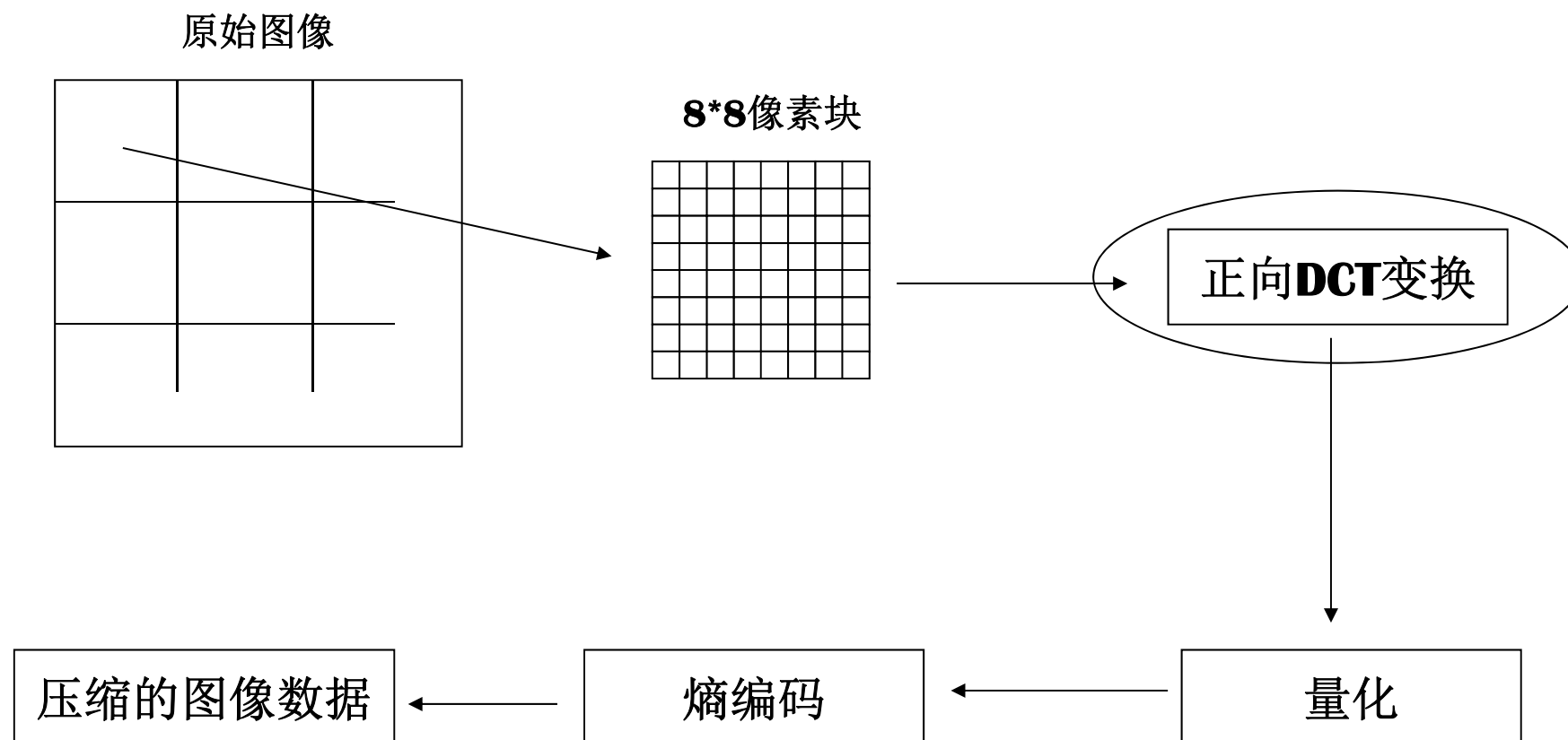
$$C_r = 0.5R - 0.4187G + 0.0813B$$

压缩原理：把重要的信息和  
不重要的信息分开

对于人眼来说，图像中明暗  
的变化更容易被感知到，这  
是由于人眼的构造引起的。  
视网膜上有两种感光细胞，  
能够感知亮度变化的视杆细  
胞，以及能够感知颜色的视  
锥细胞，由于视杆细胞在数  
量上远大于视锥细胞。



# 图像压缩过程





# DCT变换

---

通过**DCT**，空间表达式可以转换成频谱表达式或频率域，在频域中只用少量的数据就可以表示空间域中大量的数据，从而达到数据压缩的目的。

1. 可以将图像的像素空间转换到频域中，从而用少量的数据表示图像。
2. **DCT**产生的系数容易被量化，能获得较好的压缩快。
3. **DCT**算法的性能好，计算速度快。
4. **DCT**算法是可逆的，所以利用逆**DCT**变换可以解压图像。





# DCT变换

**DCT变换:** 
$$S_{vu} = \frac{1}{4} C_u C_v \sum_{x=0}^7 \sum_{y=0}^7 s_{yx} \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16}$$

**IDCT变换:** 
$$s_{yx} = \frac{1}{4} \sum_{u=0}^7 \sum_{v=0}^7 C_u C_v S_{vu} \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16}$$

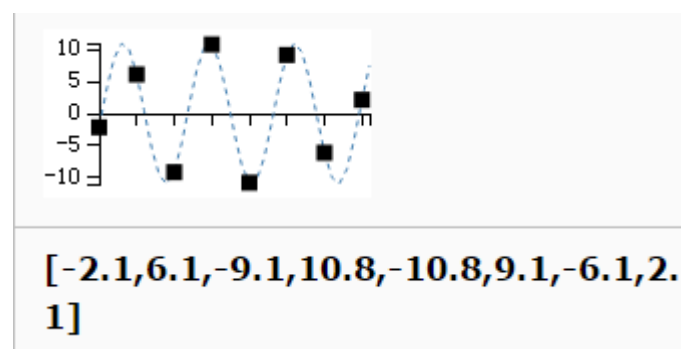
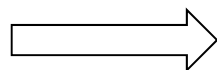
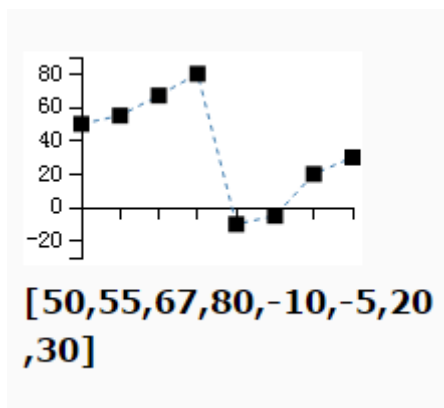
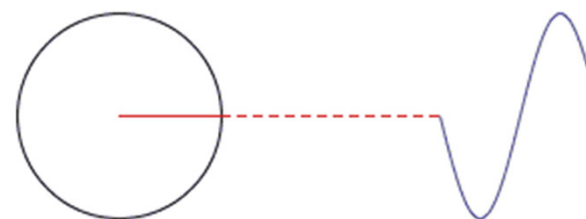
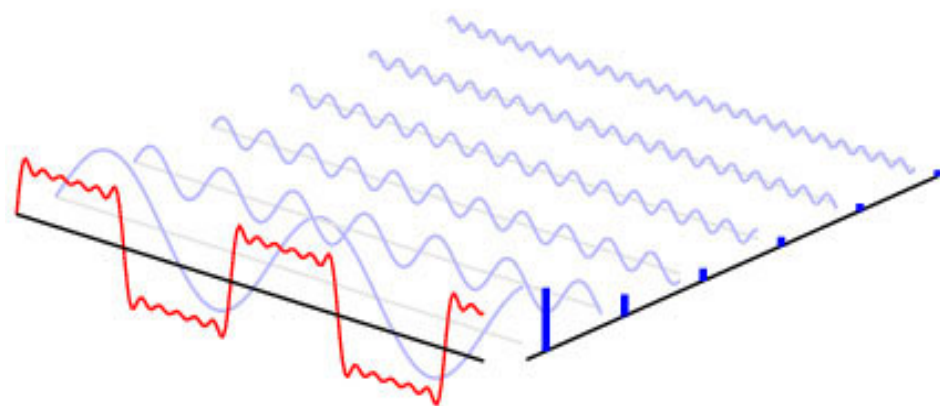
$s_{yx}$ 表示位于坐标 (y,x) 处的图像像素值

$S_{vu}$ 表示DCT系数

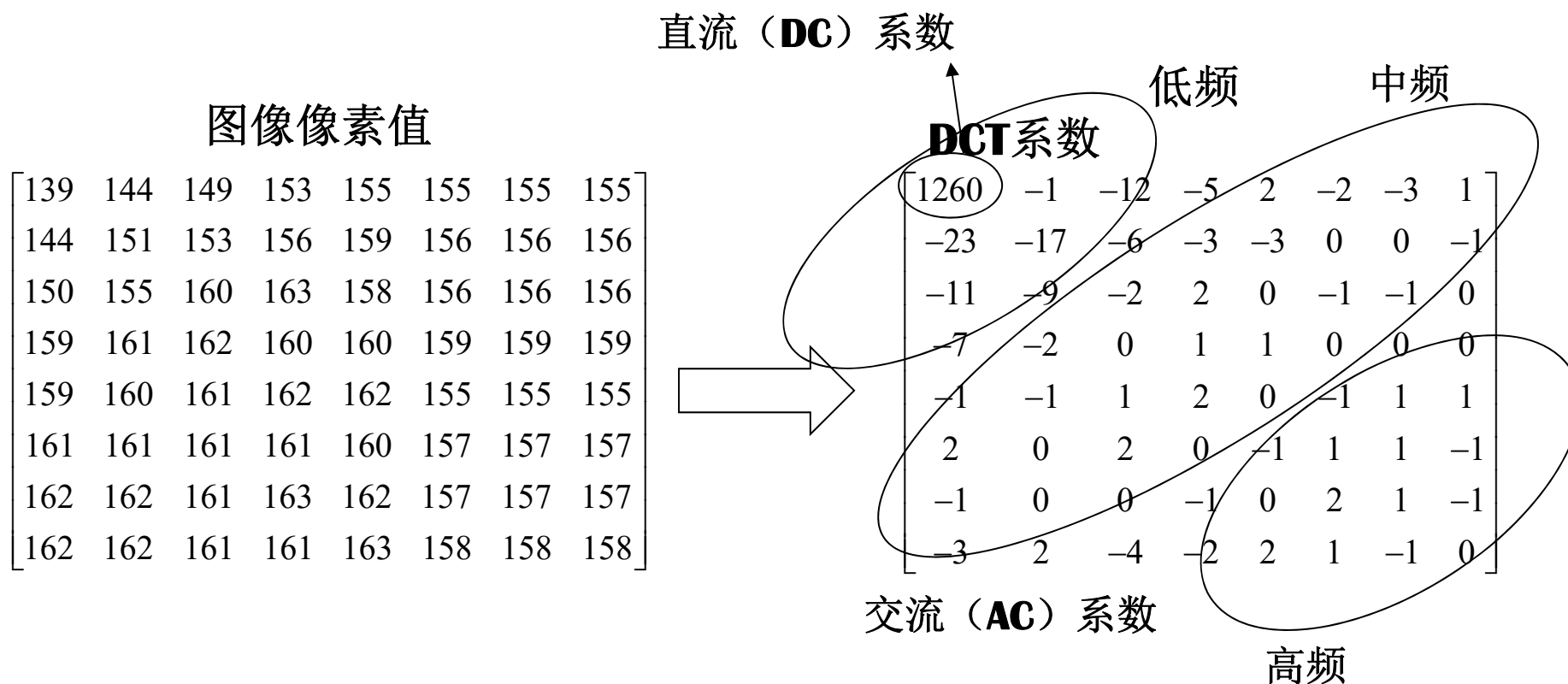
$$\begin{cases} C_u, C_v = \frac{1}{\sqrt{2}} & u, v = 0 \\ C_u, C_v = 1 & \text{其它} \end{cases}$$



# DCT变换



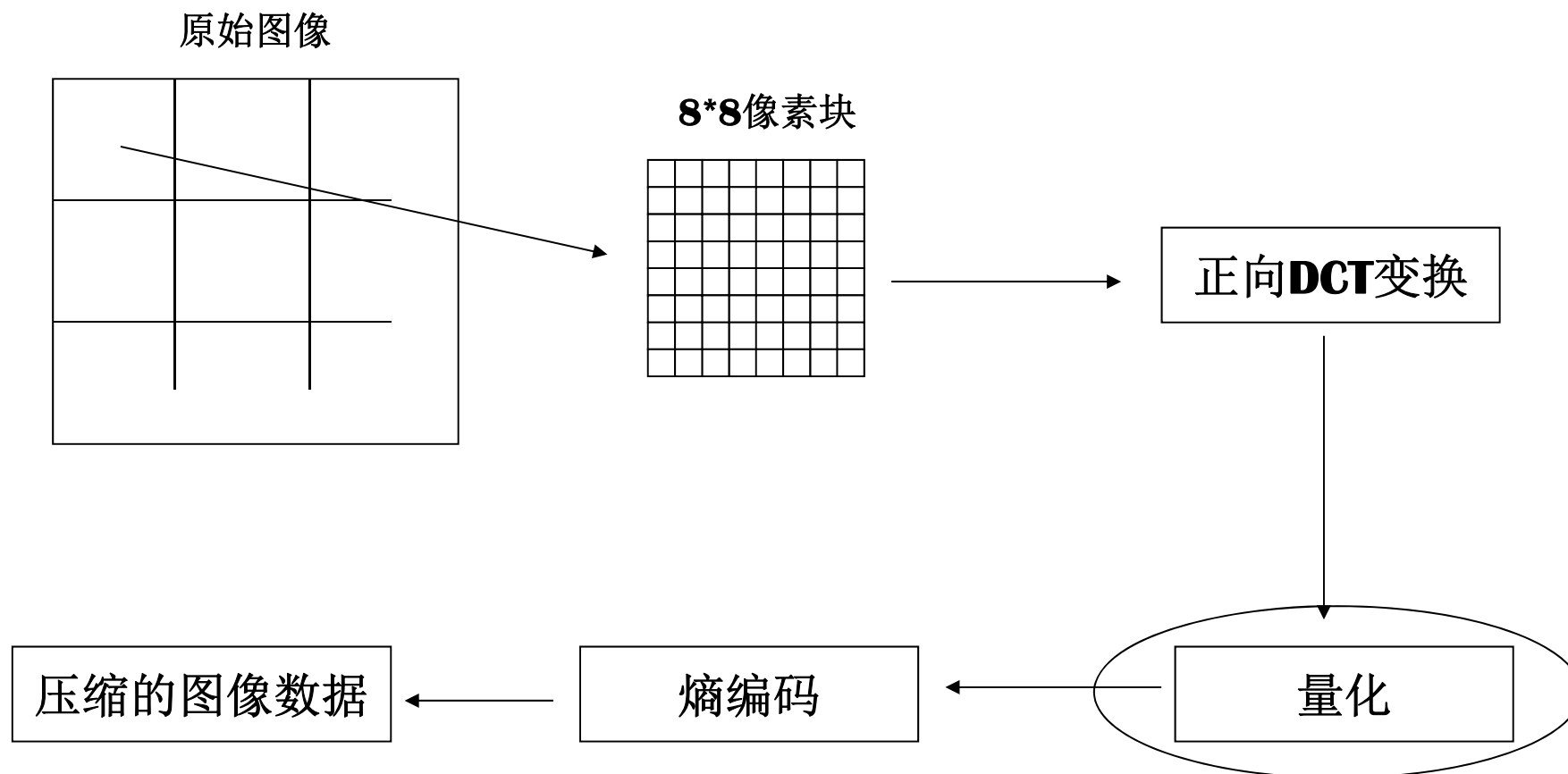
# DCT变换



对图像进行DCT变换后，图像重要的可视信息都集中在了中低频部分。



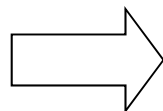
# 图像压缩过程



# 量化

139	144	149	153	155	155	155	155
144	151	153	156	159	156	156	156
150	155	160	163	158	156	156	156
159	161	162	160	160	159	159	159
159	160	161	162	162	155	155	155
161	161	161	161	160	157	157	157
162	162	161	163	162	157	157	157
162	162	161	161	163	158	158	158

原始图像灰度值



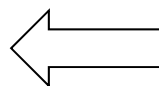
DCT系数

1260	-1	-12	-5	2	-2	-3	1
-23	-17	-6	-3	-3	0	0	-1
-11	-9	-2	2	0	-1	-1	0
-7	-2	0	1	1	0	0	0
-1	-1	1	2	0	-1	1	1
2	0	2	0	-1	1	1	-1
-1	0	0	-1	0	2	1	-1
-3	2	-4	-2	2	1	-1	0

( ÷ )

78	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-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
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

13 量化后的DCT系数

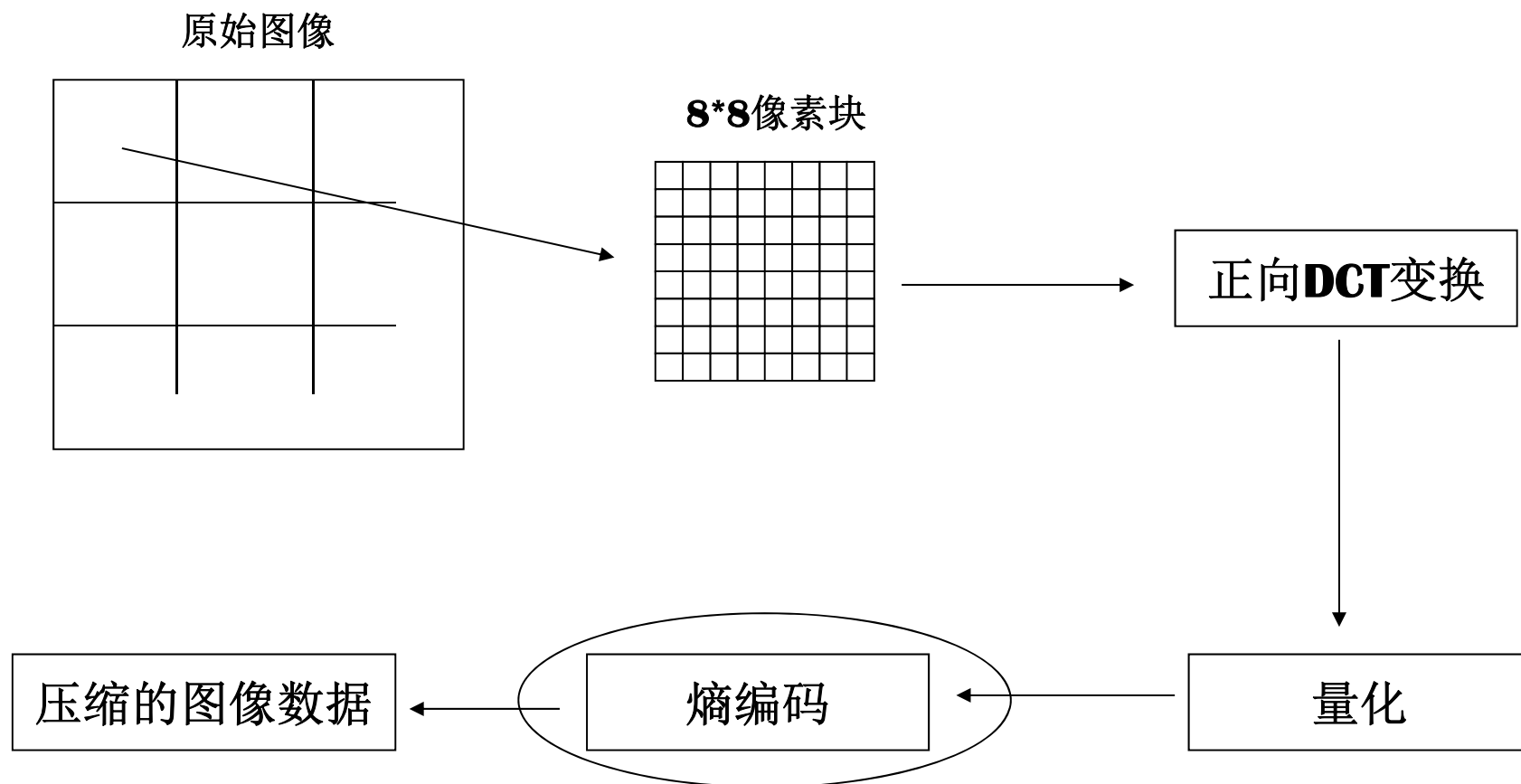


16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

量化表



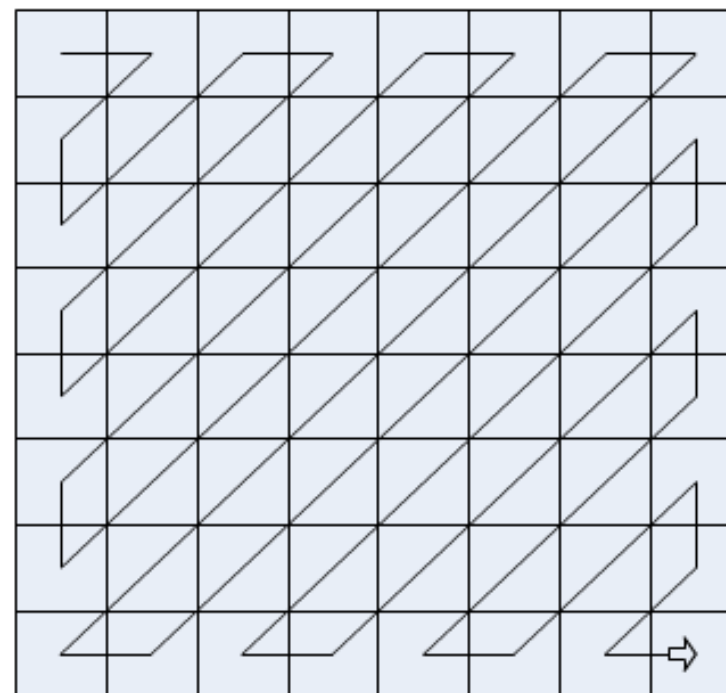
# 图像压缩过程



# Z字形排序

$$\begin{bmatrix} 78 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

量化后的DCT系数



排列顺序



# 熵编码-霍夫曼编码

---

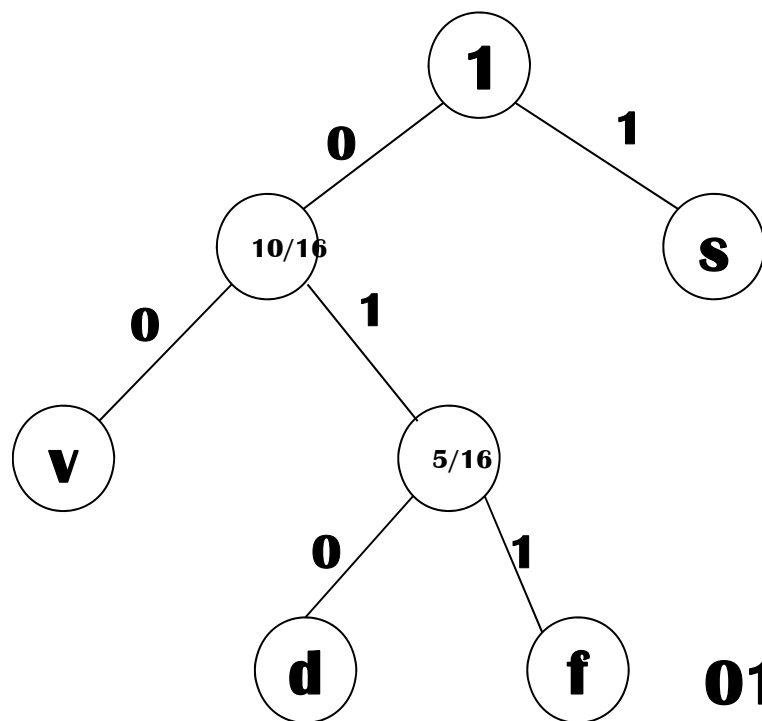
**Huffman**编码时常用的压缩方法之一，它是通过用更有效的代码代替数据来工作的。基本的方式是对每个符号赋予一个二进制码，每个符号的二进制码长度可以不同。每个符号码的码长度与这个符号的使用频率有关，使用越频繁的符号，它的码越短。



# 霍夫曼编码

**ddddfvvvvssssss**

**s:6(6/16) v:5(5/16) d:4(4/16) f:1(1/16)**



编 1.根据符号概率大小由大到小排  
码 2.把概率最小的两个符号组成一个新符号  
规 3.重复第二步，直到形成一个符号为止  
则 4.从根开始将两个分支分别赋予0,1

**s: 1**

**v: 00**

**d: 010**

**f: 011**

**长度: 128 → 31**

**01001001001011010101010111111**

17 符号的使用频率越高，代表符号的叶子节点越接近二叉树的根



# JPEG数据存储

在头部有 **FFD8**，表示图像的开始；

结束部分有 **FFD9**，表示图像的结束。

在中间有两个量化表 **DQT** 对应的标记 **FFDB**；

还有图像大小信息对应的 **FFC0**

再后面有四个编码表对应的 **FFC4**；

SOI:  
Start Of Image

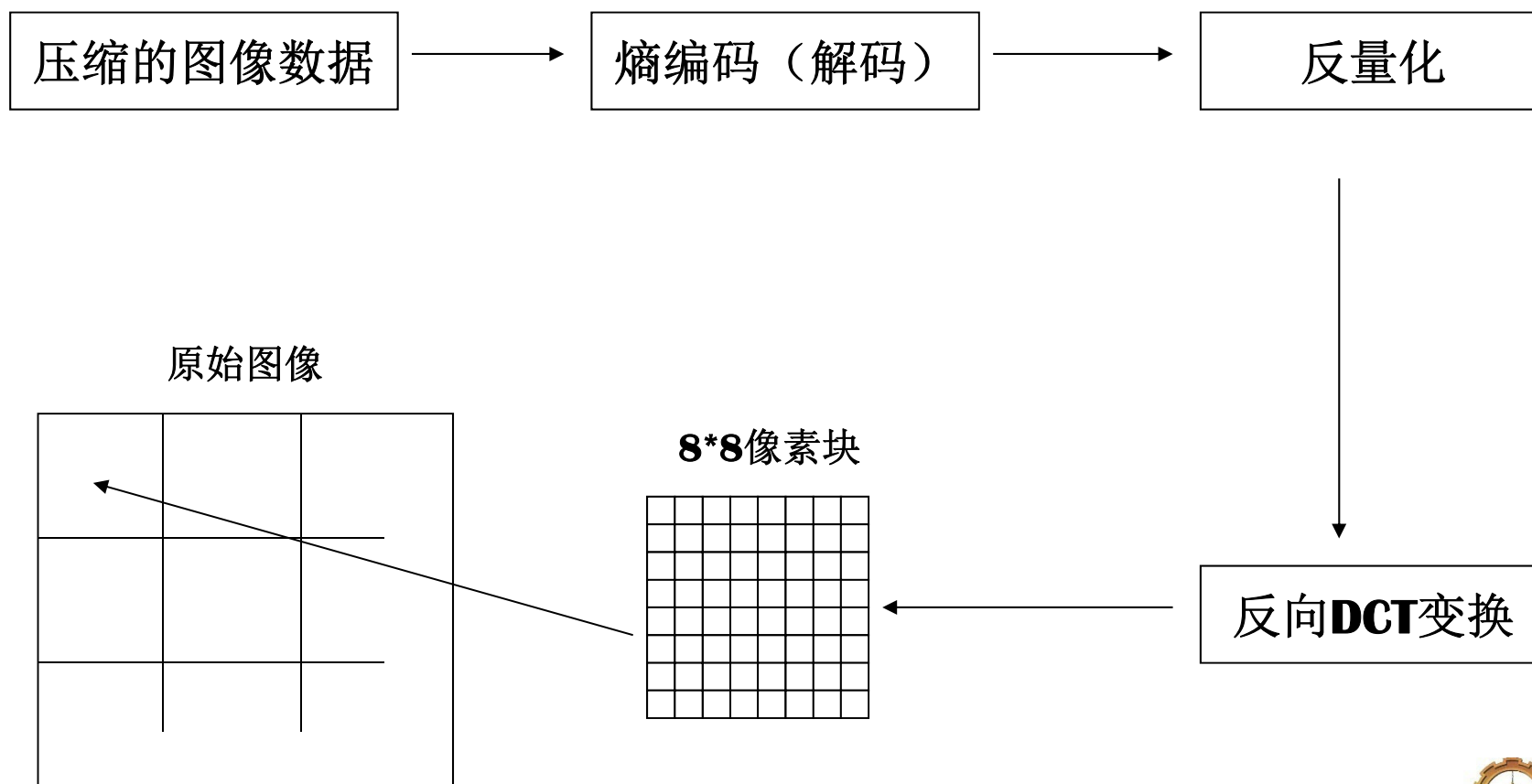
DQT:  
Define Quantization Table

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	
00000000h	FF	D8	FF	EO	00	10	4A	46	49	46	00	01	01	01	00	60	; JFIF.....
00000010h	00	60	00	00	FF	DB	00	43	00	08	06	06	07	06	05	08	; ... ?C.....
00000020h	07	07	07	09	09	08	0A	0C	14	0D	0C	0B	0B	0C	19	12	; .....
00000030h	13	0F	14	1D	1A	1F	1E	1D	1A	1C	1C	20	24	2E	27	20	; ..... \$.
00000040h	22	2C	23	1C	1C	28	37	29	2C	30	31	34	34	34	1F	27	; ",#.. (7),01444.
00000050h	39	3D	38	32	3C	2E	33	34	32	FF	DB	00	43	01	09	09	; 9=82<.342 ?C... SOS:
00000060h	09	0C	0B	0C	18	0D	0D	18	32	21	1C	21	32	32	32	32	; .....2!..12222-Start Of Frame 0
00000070h	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	; 2222222222222222
00000080h	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	; 2222222222222222
00000090h	32	32	32	32	32	32	32	32	32	32	32	32	32	32	FF	CO	; 2222222222222222
000000a0h	00	11	08	00	08	00	08	03	01	22	00	02	11	01	03	11	; .....".
000000b0h	01	FF	C4	00	1F	00	00	01	05	01	01	01	01	01	01	00	; . ?.....
000000c0h	00	00	00	00	00	00	00	01	02	03	04	05	06	07	08	09	; .....
000000d0h	0A	0B	FF	C4	00	B5	10	00	02	01	03	03	02	04	03	05	; .. ??.....
000000e0h	05	04	04	00	00	01	7D	01	02	03	00	04	11	05	12	21	; .....)!.....!
000000f0h	31	41	06	13	51	61	07	22	71	14	32	81	91	A1	08	23	; 1A..Qa."q.2△?#
00000100h	42	B1	C1	15	52	D1	F0	24	33	62	72	82	09	0A	16	17	; B綑..R佯\$3br?... SOS:
00000110h	18	19	1A	25	26	27	28	29	2A	34	35	36	37	38	39	3A	; ...\$&'( *456789;
00000120h	43	44	45	46	47	48	49	4A	53	54	55	56	57	58	59	5A	; CDEFGHIJSTUVWXYZ
00000130h	63	64	65	66	67	68	69	6A	73	74	75	76	77	78	79	7A	; cdefghijstuvwxyz
00000140h	83	84	85	86	87	88	89	8A	92	93	94	95	96	97	98	99	; 傲厚哈塌振欧松掩
00000150h	9A	A2	A3	A4	A5	A6	A7	A8	A9	AA	B2	B3	B4	B5	B6	B7	; 殺丫々;商吹斗
00000160h	B8	B9	BA	C2	C3	C4	C5	C6	C7	C8	C9	CA	D2	D3	D4	D5	; 腹部媚牌桥墙矣哉
00000170h	D6	D7	D8	D9	DA	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	; 肿到登杆漠晓恬露
00000180h	F2	F3	F4	F5	F6	F7	F8	F9	FA	FF	C4	00	1F	01	00	03	; 娘嫩嫩 ??...
00000190h	01	01	01	01	01	01	01	01	01	01	00	00	00	00	00	01	; .....
000001a0h	02	03	04	05	06	07	08	09	0A	0B	FF	C4	00	B5	11	00	; ..... ??.
000001b0h	02	01	02	04	04	03	04	07	05	04	04	00	01	02	77	00	; .....v.
000001c0h	01	02	03	11	04	05	21	31	06	12	41	51	07	61	71	13	; .....1!..AQ.ag.
000001d0h	22	32	81	08	14	42	91	A1	B1	C1	09	23	33	52	F0	15	; "2?.B綑綑.#3R?
000001e0h	62	72	D1	0A	16	24	34	E1	25	F1	17	18	19	1A	26	27	; br?.\$4??...&'
000001f0h	28	29	2A	35	36	37	38	39	3A	43	44	45	46	47	48	49	; ()*56789:CDEFGHI
00000200h	4A	53	54	55	56	57	58	59	5A	63	64	65	66	67	68	69	; JSTUVWXYZcdefghi
00000210h	6A	73	74	75	76	77	78	79	7A	82	83	84	85	86	87	88	; jstuvwxyz们刺啫?
00000220h	89	8A	92	93	94	95	96	97	98	99	9A	A2	A3	A4	A5	A6	; 塌振欧松掩殺丫々
00000230h	A7	A8	A9	AA	B2	B3	B4	B5	B6	B7	B8	B9	BA	C2	C3	C4	; X;商吹斗腹部媚
00000240h	C5	C6	C7	C8	C9	CA	D2	D3	D4	D5	D6	D7	D8	D9	DA	E2	; 牌桥墙矣哉肿到
00000250h	E3	E4	E5	E6	E7	E8	E9	EA	F2	F3	F4	F5	F6	F7	F8	F9	; 沉三站腿嫩嫩
00000260h	FA	FF	DA	00	0C	03	01	00	02	11	03	11	00	3F	00	E8	; ??.....??.?
00000270h	AC	7C	31	7B	27	88	1D	C3	C7	0E	82	6E	A7	83	EC	68	; 瑞1('?)们.倪 啫
00000280h	3E	E4	64	3E	6A	38	75	8B	6A	70	10	E7	71	F9	54	2C	; >院>j8u綑p.鑽綑,
00000290h	79	25	36	D1	45	15	B5	3C	55	78	37	C9	36	AF	D9	B3	; y\$6移.2Ux?? ?
000002a0h	5A	DC	B3	E5	7C	A9	69	D1	25	E5	F7	BE	A7	FF	D9	;	; 2++增 ?繁晶 0

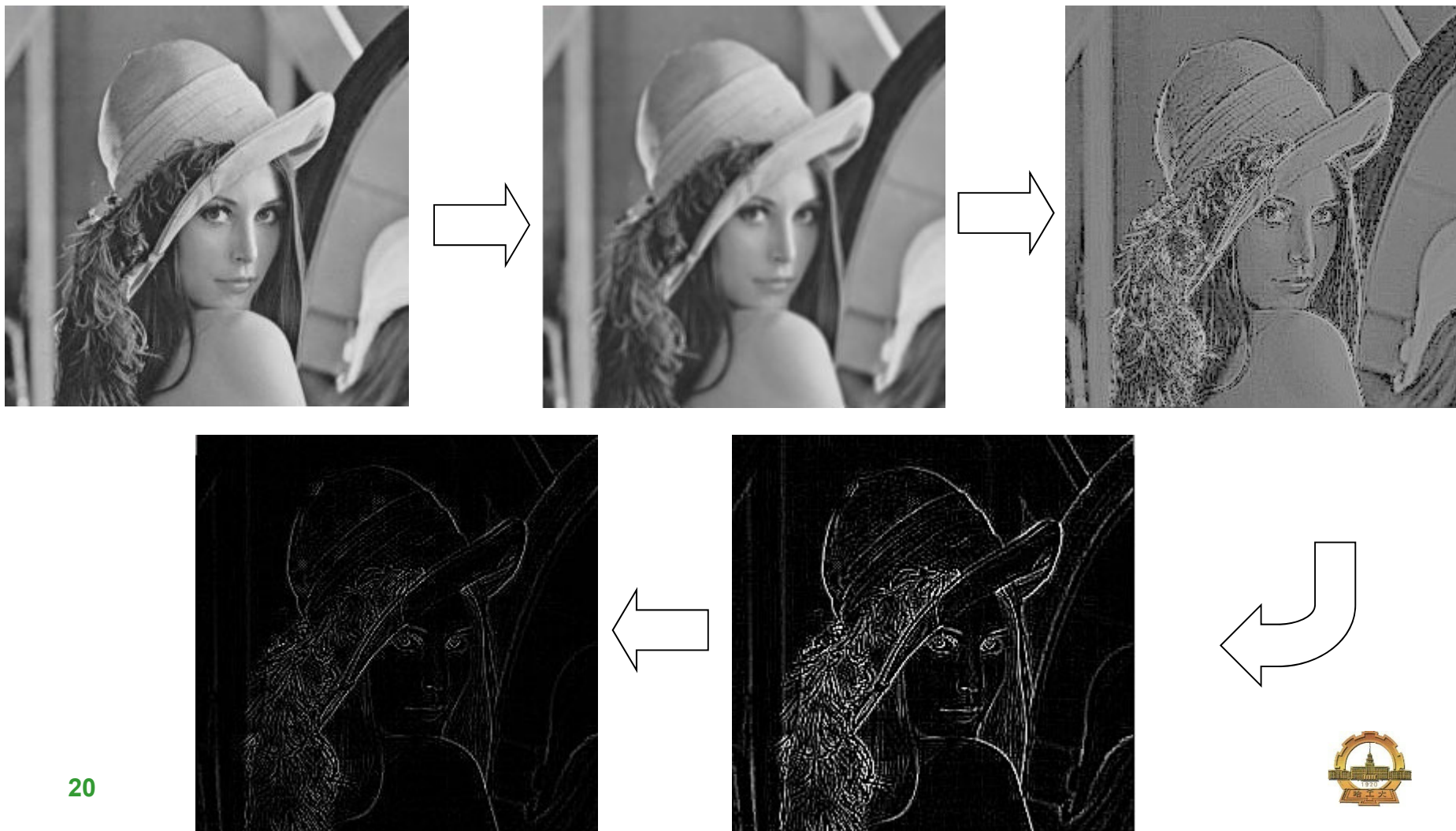
SOS:  
Start Of Scan

EOI:  
End Of Image

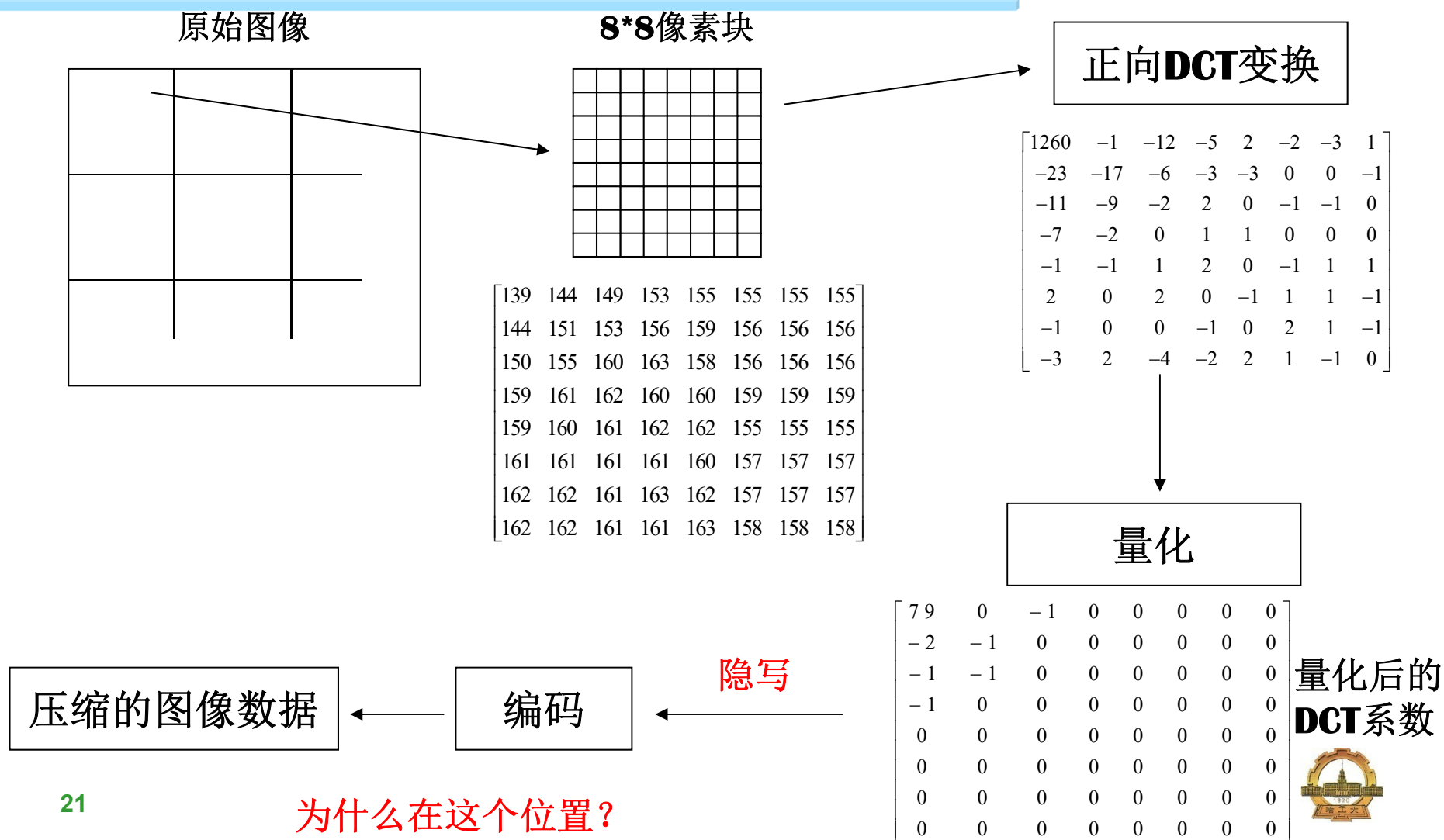
# JPEG图像解压缩过程



# 图像压缩



# JPEG图像隐写





# Jsteg隐写方法

**Jsteg**隐写是将秘密信息嵌入在量化后的**DCT**系数的**LSB**上，但原始值为**-1,0, +1**的**DCT**系数除外。

秘密信息: **0 1**

79	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-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
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

**DCT**系数

**0** → **79** → **78**  
**1** → **-2** → **-3**

78	0	-1	0	0	0	0	0
-3	-1	0	0	0	0	0	0
-1	-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
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

隐写后的**DCT**系数





# Jsteg隐写方法



# F3隐写方法

---

## 嵌入规则

- 每个非**0**的**DCT**数据用于隐藏**1**比特秘密信息，为**0**的**DCT**系数不负载秘密信息。
- 如果秘密信息与**DCT**的**LSB**相同，便不作改动；如果不同，将**DCT**系数的绝对值减小**1**，符号不变。
- 当原始值为**+1**或**-1**且预嵌入秘密信息为**0**时，将这个位置归**0**并视为无效，在下一个**DCT**系数上重新嵌入。



# F3隐写方法

DCT系数

79	0	-1	0	0	0	0	0
-1	-2	0	0	0	0	0	0
-1	-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
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

隐写后的DCT系数

78	0	0	0	0	0	0	0
0	-1	0	0	0	0	0	0
-1	-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
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

秘密信息: **0 0 1 1 1**

<b>0</b>	→	<b>79</b>	→	<b>78</b>
<b>0</b>	→	<b>-2</b>	→	<b>-1</b>
<b>1</b>	→	<b>-1</b>	→	<b>-1</b>

嵌入规则:

1. 每个非0的DCT数据用于隐藏1比特秘密信息，为0的DCT系数不负载秘密信息。
2. 如果秘密信息与DCT的LSB相同，便不作改动；如果不同，将DCT系数的绝对值减小1，符号不变。
3. 当原始值为+1或-1且预嵌入秘密信息为0时，将这个位置归0并视为无效，在下一个DCT系数上重新嵌入。



---

# 谢谢！



1. JPEG图像与BMP图像相比较的优缺点是什么？
2. DCT变换前图像进行哪些预处理，目的是什么？
3. JPEG压缩分为哪几个步骤？哪些位置适合隐写，为什么？
4. JPEG压缩过程中使用DCT变换的目的是什么？
5. 相比于Jsteg隐写F3隐写算法进行了哪些方面的改进？
6. 计算使用Jsteg算法和F3算法嵌入秘密信息后的DCT系数

$$\begin{bmatrix} 97 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ -3 & -2 & 4 & 0 & 0 & 0 & 0 & 0 \\ -2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

原始DCT系数

**0 0 1 1 0**

秘密信息

