



# 多媒体技术

*JPEG* 压缩编码

# JPEG标准

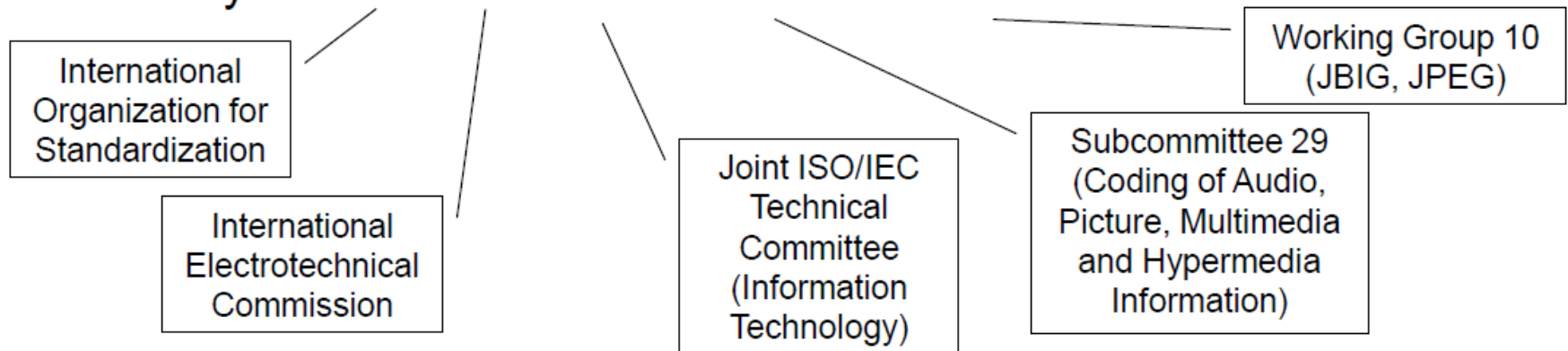
---

- 什么是JPEG
  - JPEG (Joint Photographic Expert Group) 是联合图像专家小组的英文缩写。其中“联合”的含义
    - 国际标准化组织 (ISO)
    - 国际电报电话咨询委员会 (CCITT)
  - 由联合专家组所开发的JPEG算法被确定为JPEG国际标准
    - 彩色、灰度、静止图像的第一个国际标准
    - 静止图像的压缩、电视图像的帧内图像的压缩编码

# JPEG标准

---

- JPEG: “Joint Photographic Experts Group”
- Formally: ISO/IEC JTC1/SC29/WG10



- Joint effort with CCITT (International Telephone and Telegraph Consultative Committee, now ITU-T) Study Group VIII
- Work commenced in 1986
- International standard ISO/IEC 10918-1 and CCITT Rec. T.81 in 1992
- Widely used for image exchange, WWW, and digital photography

# JPEG标准

---

- JPEG的目的是为了给出一个适合于连续色调图像的压缩方法，使之满足以下要求：
  - 1、达到或接近当前压缩比和图像保真度的技术水平，能覆盖一个较宽的图像质量等级范围，能达到“很好”到“极好”的评估，与原始图像相比，人的视觉难以区分
  - 2、能适合任何种类的连续色调的图像，长宽比不受限制，同时也不受限于景物内容、图像的复杂程度和统计特性等
  - 3、计算复杂度是可控制的，其软件可在各种CPU上完成，算法也可用硬件实现

# JPEG标准

---

- JPEG算法具有以下4种操作方式
  - 顺序编码(Sequential)
    - 从左到右，从上到下完成一次扫描
  - 累进编码(Progressive)
    - 图像编码在多次扫描中完成
  - 无失真编码(lossless)
  - 分层编码(Hierarchical)
    - 图像在多个空间分辨率进行编码

# JPEG标准

---

## Baseline Sequential



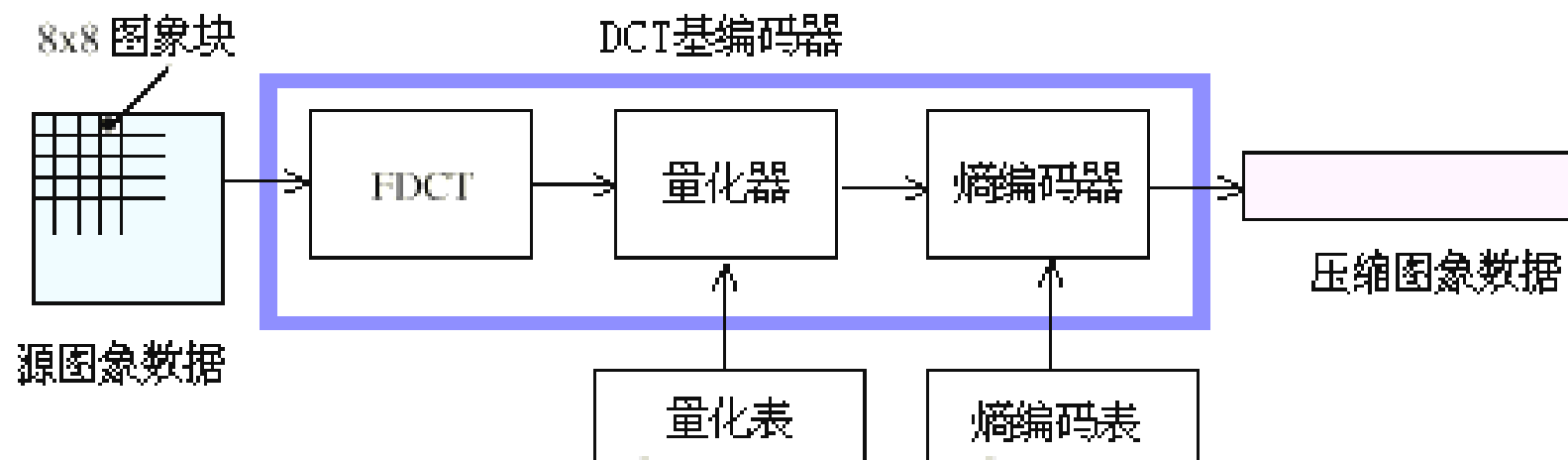
## Progressive



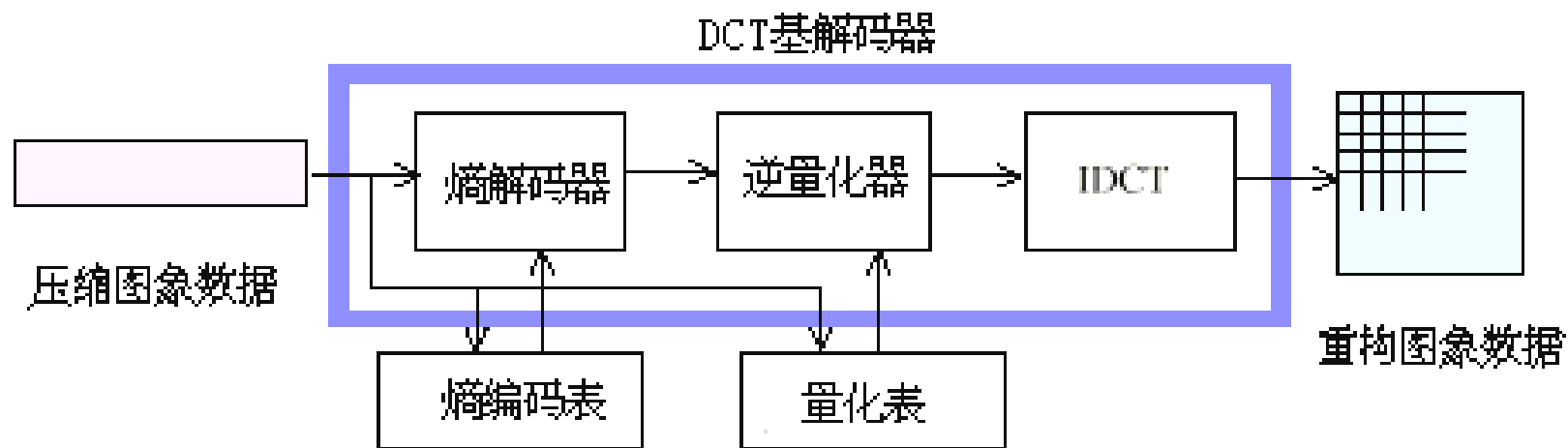
## Hierarchical



This is the most common mode and the only one we're going to talk about



(a) DCT基压缩编码步骤



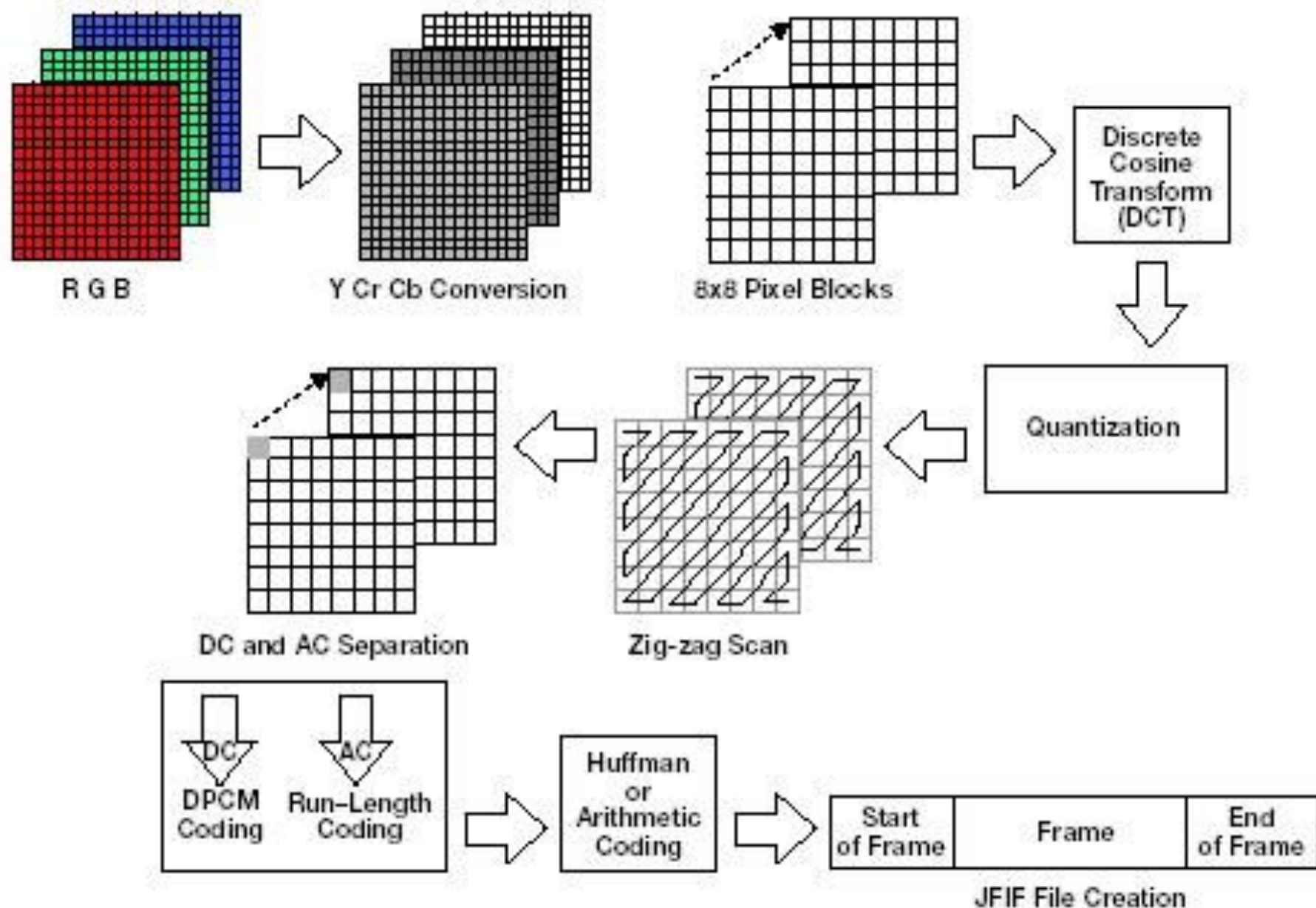
(b) DCT基解压缩步骤

# JPEG标准

---

- JPEG有损顺序编码算法的主要计算步骤：
  - 1. 将源图像分成几个颜色平面（彩色图像分量）
  - 2. 分成 $8 \times 8$ 数据块进行正向离散余弦变换(FDCT)
  - 3. 量化(quantization)
  - 4. Z字形排列量化结果(zigzag scan)
  - 5. 使用 DPCM对直流系数(DC)进行编码，再做熵编码
  - 6. 使用行程编码(RLE)对交流系数(AC)进行编码，再做熵编码





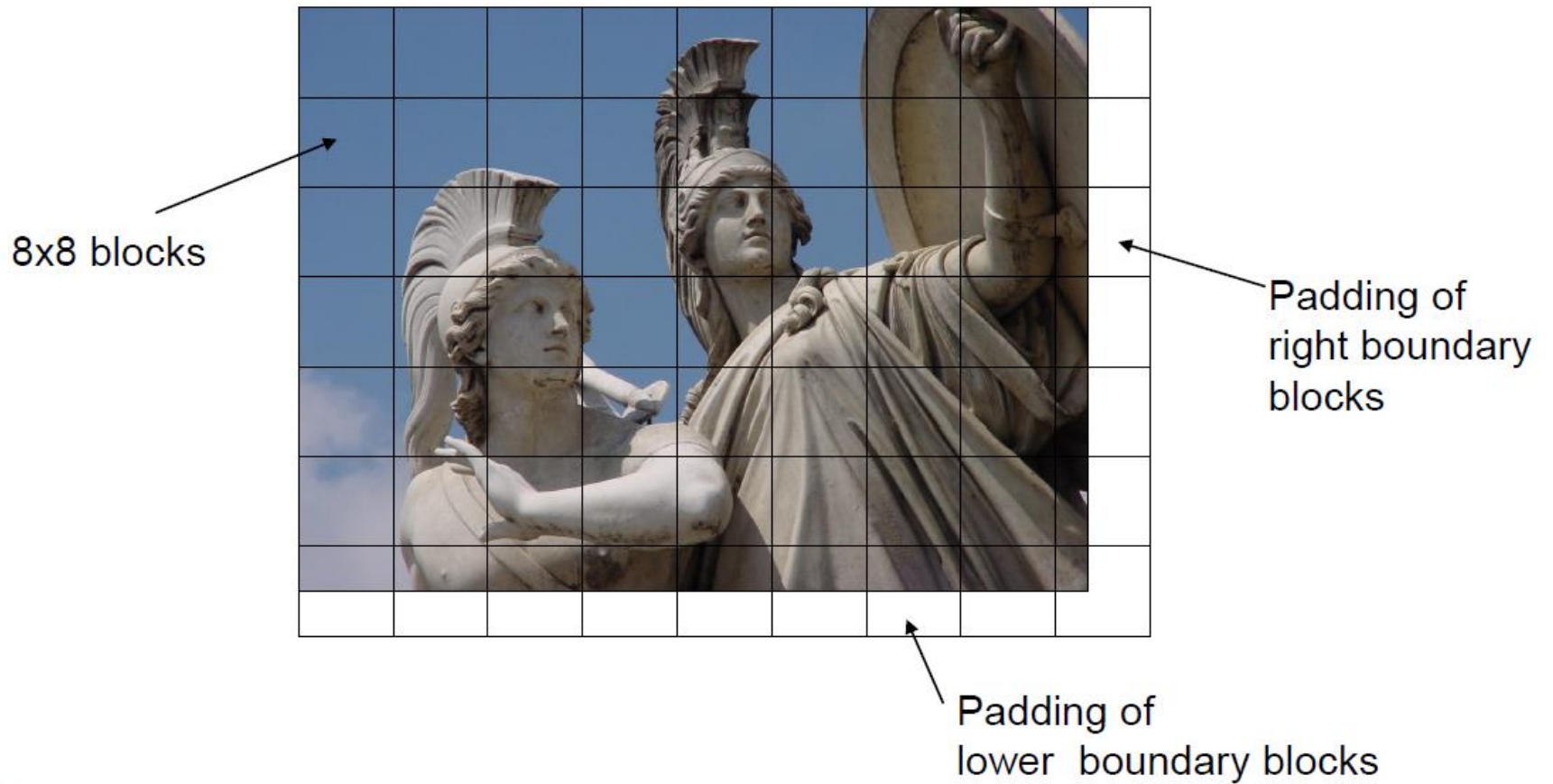
# JPEG标准

---

- JPEG采用的是 $8 \times 8$ 大小的子块的二维离散余弦变换（DCT）
  - 把原始图像分割成一系列 $8 \times 8$ 的子块
  - 设原始图像为P位，输入时将 $[0, 2^P-1]$  范围的无符号整数变成 $[-2^{P-1}, 2^{P-1}-1]$ 范围的有符号整数，以此作为正向离散余弦变换（FDCT）的输入
  - 对于解码器输出端的离散余弦逆变换（IDCT）的结果，需将其数值范围由 $[-2^{P-1}, 2^{P-1}-1]$ 再变回到 $[0, 2^P-1]$ ，来获得重构图像

# JPEG标准

---



# JPEG标准

---

- 离散余弦变换

二维离散余弦变换(2D DCT)也叫做正向离散余弦变换(2D FDCT), 用下式表示,

$$Y = AXA^T \quad (5-1)$$

2D DCT的逆变换(2D IDCT)为,

$$X = A^T Y A \quad (5-2)$$

$X$ 是输入样本矩阵,  $Y$ 是变换后的系数矩阵,  $A$ 是 $N \times N$ 变换矩阵。 $A$ 的元素是,

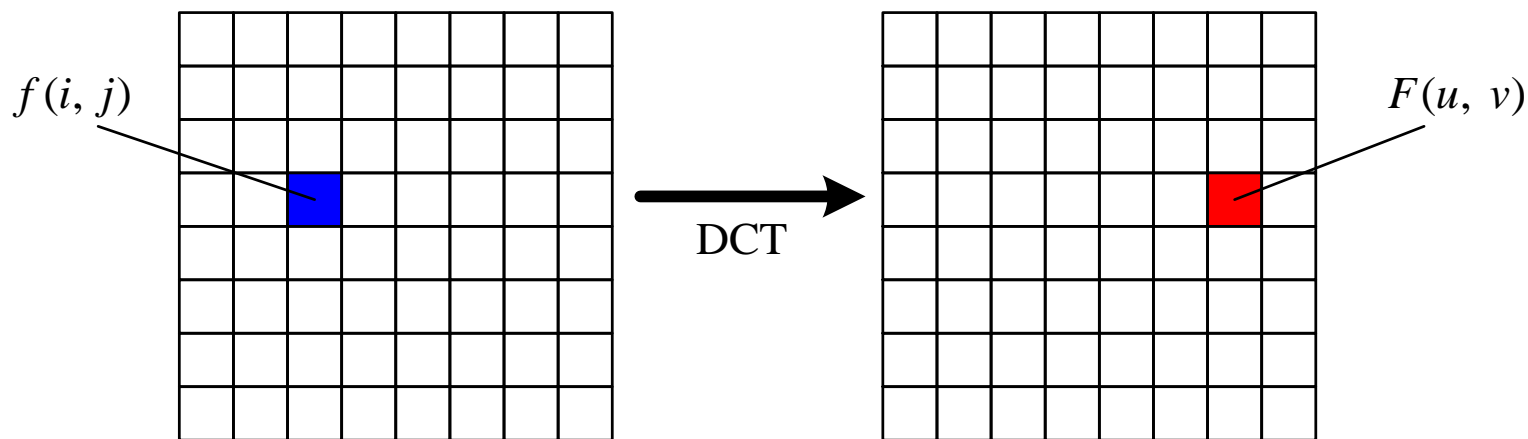
$$A_{ij} = C_i \cos \frac{(2j+1)i\pi}{2N}, \text{ 其中,} \quad (5-3)$$
$$C_i = \sqrt{\frac{1}{N}} \quad (i=0), \quad C_i = \sqrt{\frac{2}{N}} \quad (i>0)$$

# JPEG标准

---

## 正向离散余弦变换说明如下

(1) 对每个单独的彩色图像分量，把整个分量图像分成 $8 \times 8$ 图像块，如图5-8所示，作为DCT的输入，通过DCT把能量集中在频率较低的少数几个系数上



# JPEG标准

---

(2) 用和式表示时，式(5-1)的DCT变换使用下式计算，

$$F(u, v) = \frac{1}{4} C(u) C(v) \left[ \sum_{i=0}^7 \sum_{j=0}^7 f(i, j) \cos \frac{(2i+1)u\pi}{16} \cos \frac{(2j+1)v\pi}{16} \right] \quad (5-4)$$

逆变换式(5-2)使用下式计算，

$$f(i, j) = \frac{1}{4} C(u) C(v) \left[ \sum_{u=0}^7 \sum_{v=0}^7 f(u, v) \cos \frac{(2i+1)u\pi}{16} \cos \frac{(2j+1)v\pi}{16} \right] \quad (5-5)$$

在上面两式中，

$$C(u), C(v) = 1/\sqrt{2}, \text{ 当 } u, v = 0;$$

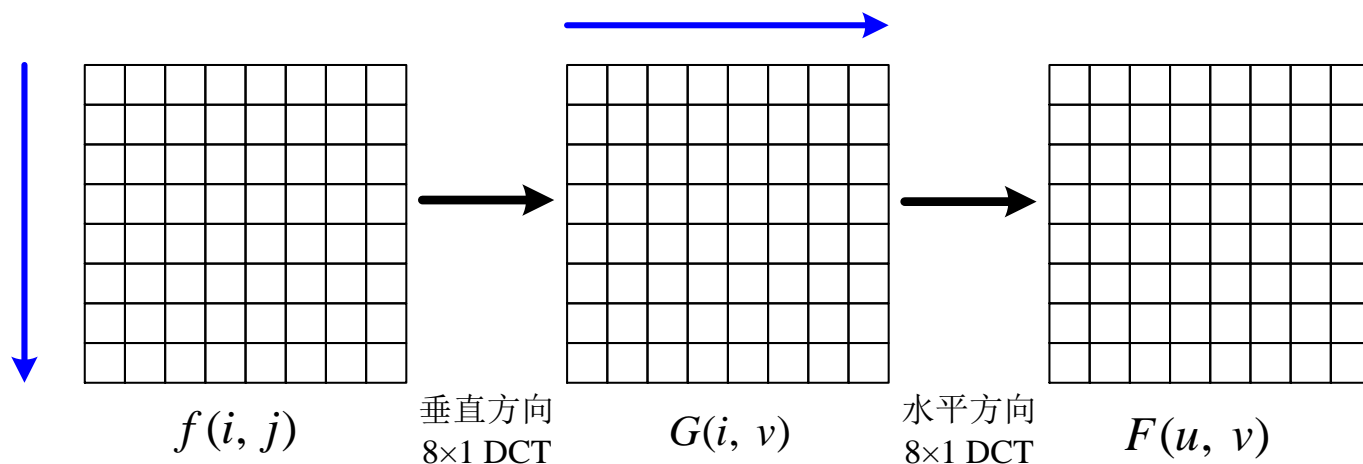
$$C(u), C(v) = 1, \text{ 其他。}$$

$f(i, j)$  是图像样本矩阵或样本的预测误差矩阵， $F(u, v)$  是  $f(i, j)$  经过DCT后的系数矩阵， $F(0, 0)$  是  $8 \times 8$  个像素值的平均值，称为直流系数(DC)，其他为交流系数(AC)

# JPEG标准

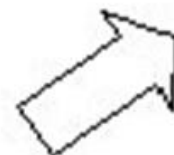
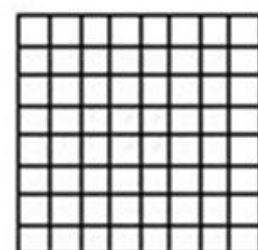
(3) 在计算二维DCT时，可用下面的计算式把二维DCT变成一维DCT，如图 5-9 所示，实际的快速计算方法可参看参考文献[4]，

$$\begin{aligned} F(u, v) &= \frac{1}{2} C(u) \left[ \sum_{i=0}^7 G(i, v) \cos \frac{(2i+1)u\pi}{16} \right] \\ G(i, v) &= \frac{1}{2} C(v) \left[ \sum_{j=0}^7 f(i, j) \cos \frac{(2j+1)v\pi}{16} \right] \end{aligned} \quad (5-6)$$



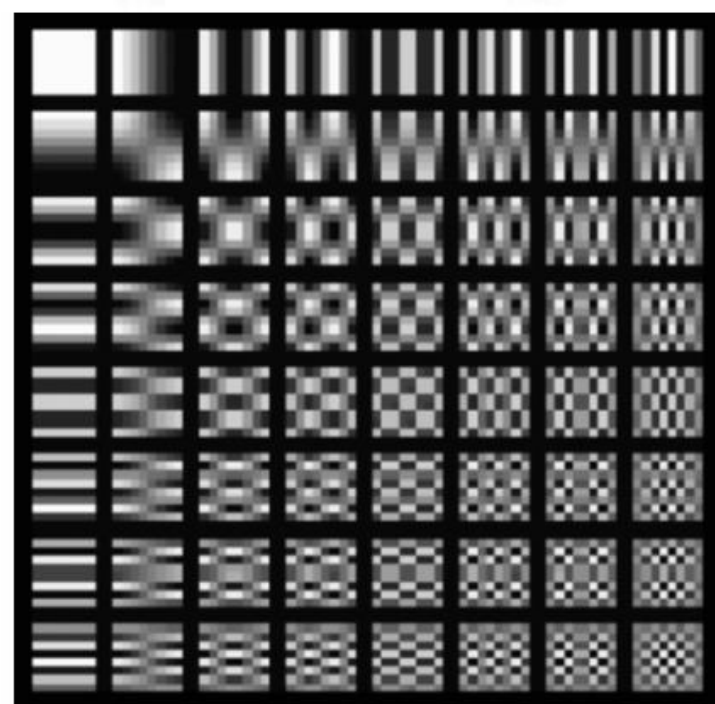
Input to DCT:  
8x8 pixel block

Output from DCT:  
8x8 coefficient block



■ DC coefficient

■ AC coefficient



DCT Basis Functions



# JPEG标准

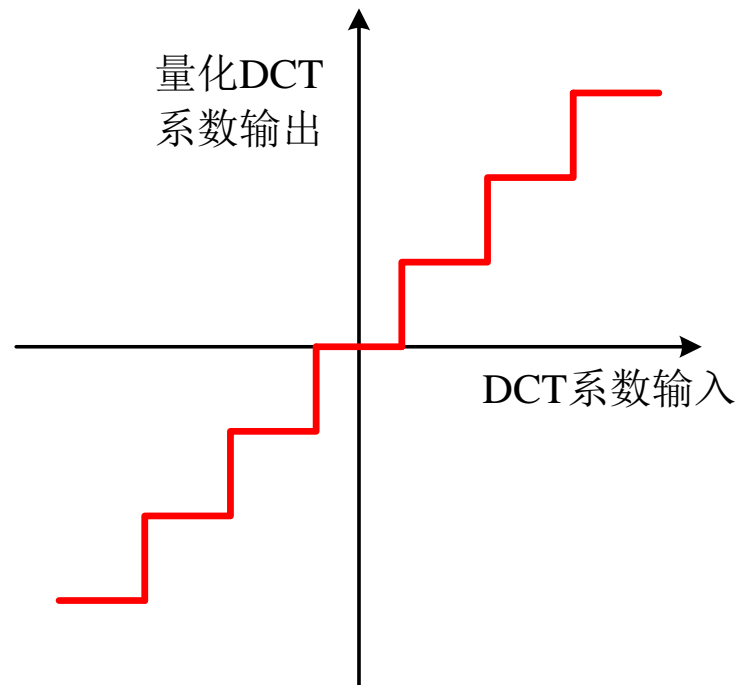
---

- 量化
  - 对经过FDCT变换后的频率系数进行量化
  - 目的：降低非0系数的幅度，增加0值系数的数目
  - 量化是造成图像质量下降的最主要原因
  - 使用如图所示的均匀量化器进行量化
  - 量化步距按系数所在位置和每种颜色分量的色调值确定

# JPEG标准

---

- 量化



均匀量化器

# JPEG标准

- 量化
  - 使用两种量化表
  - 人眼对低频分量图像比高频分量图像更敏感，因此表中的左上角的量化步距要比右下角的量化步距小
  - 表中数值对CCIR 601标准电视图像是最佳的

表 5-6 亮度量化表

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

表 5-7 色差量化表

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

# JPEG标准

---

- 量化

量化用下式计算,

$$\hat{F}(u,v) = round(\frac{F(u,v)}{Q(u,v)}) \quad (5-7)$$

其中,  $F(u,v)$  表示 DCT 系数,  $Q(u,v)$  表示“量化矩阵”,  $\hat{F}(u,v)$  表示量化(后的)系数,  $F(u,v)/Q(u,v)$  表示两个矩阵的对应元素相除,  $round$  表示四舍五入。

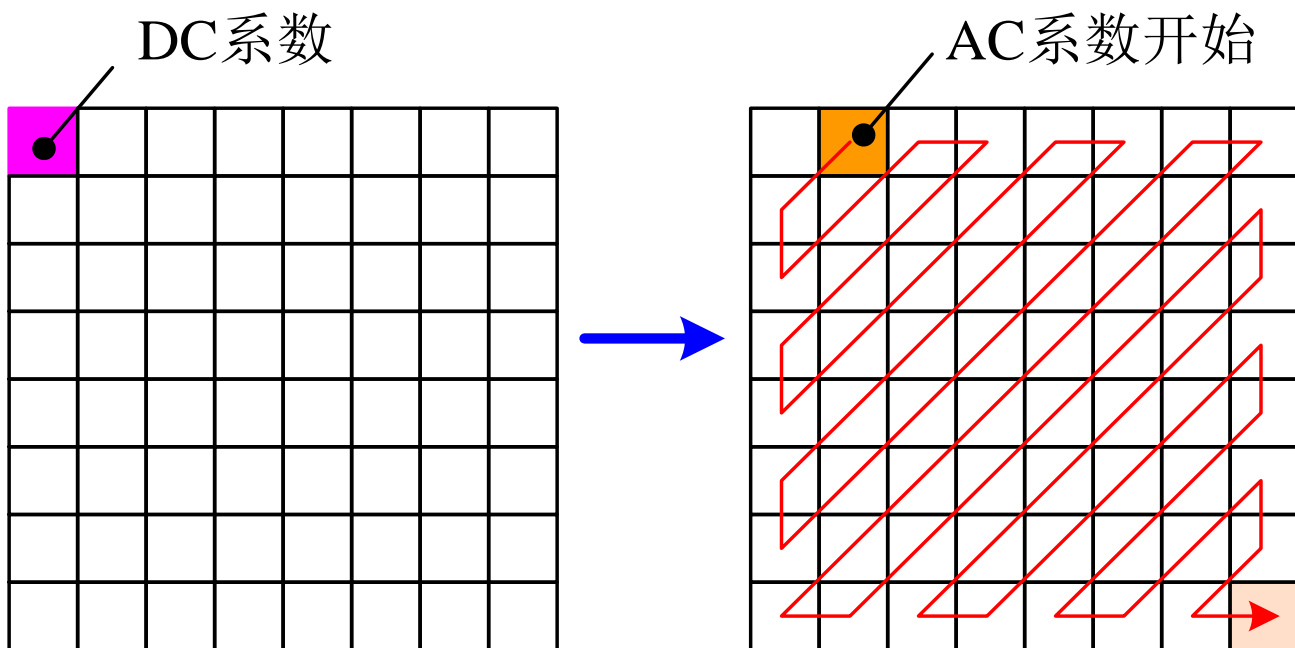
# JPEG标准

---

- 量化
  - $Q(u,v)$ 是量化器步长，它是量化表的元素，量化表元素随DCT变换系数的位置和彩色分量的不同有不同值。量化表的尺寸为 $8 \times 8$ ，与64个变换系数一一对应
  - 这个量化表应该由用户规定，在JPEG标准中给出参考值，并作为编码器的一个输入。量化表中的每个元素值为1~255之间的任意整数，其值规定了它所对应DCT系数的量化器步长

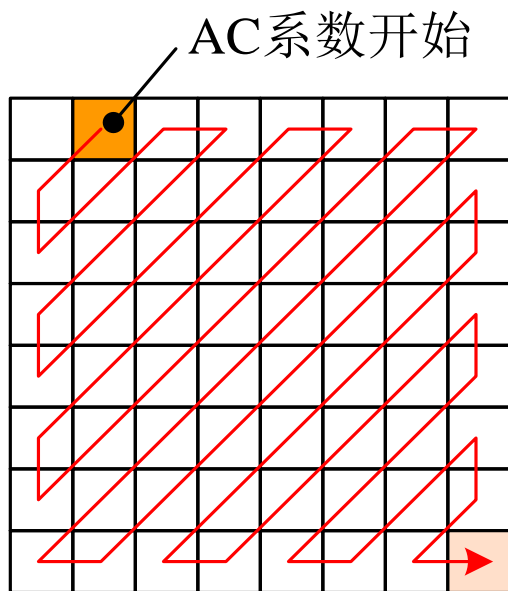
# JPEG标准

- Z字形编排
  - 重新编排量化后的系数，增加**连续的0值**系数数目
  - 排列方法：按Z字形排列

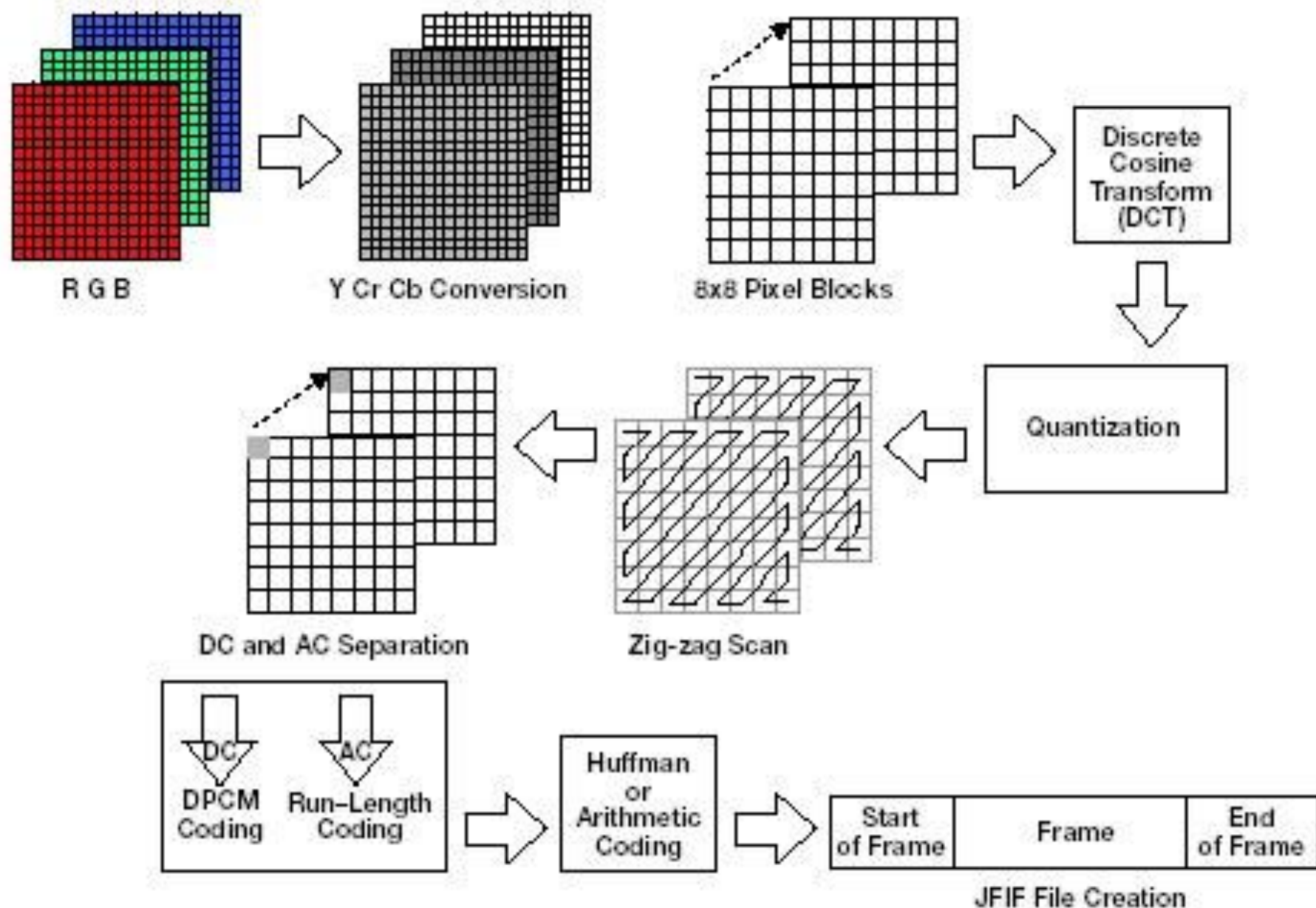


# JPEG标准

- Z字形编排
  - DCT系数序号见下图，序号小的位置表示频率较低，用  $zz(0)$ ,  $zz(1)$ , ...,  $zz(63)$ 表示
  - $8 \times 8$ 的矩阵变成  $1 \times 64$ 的矢量



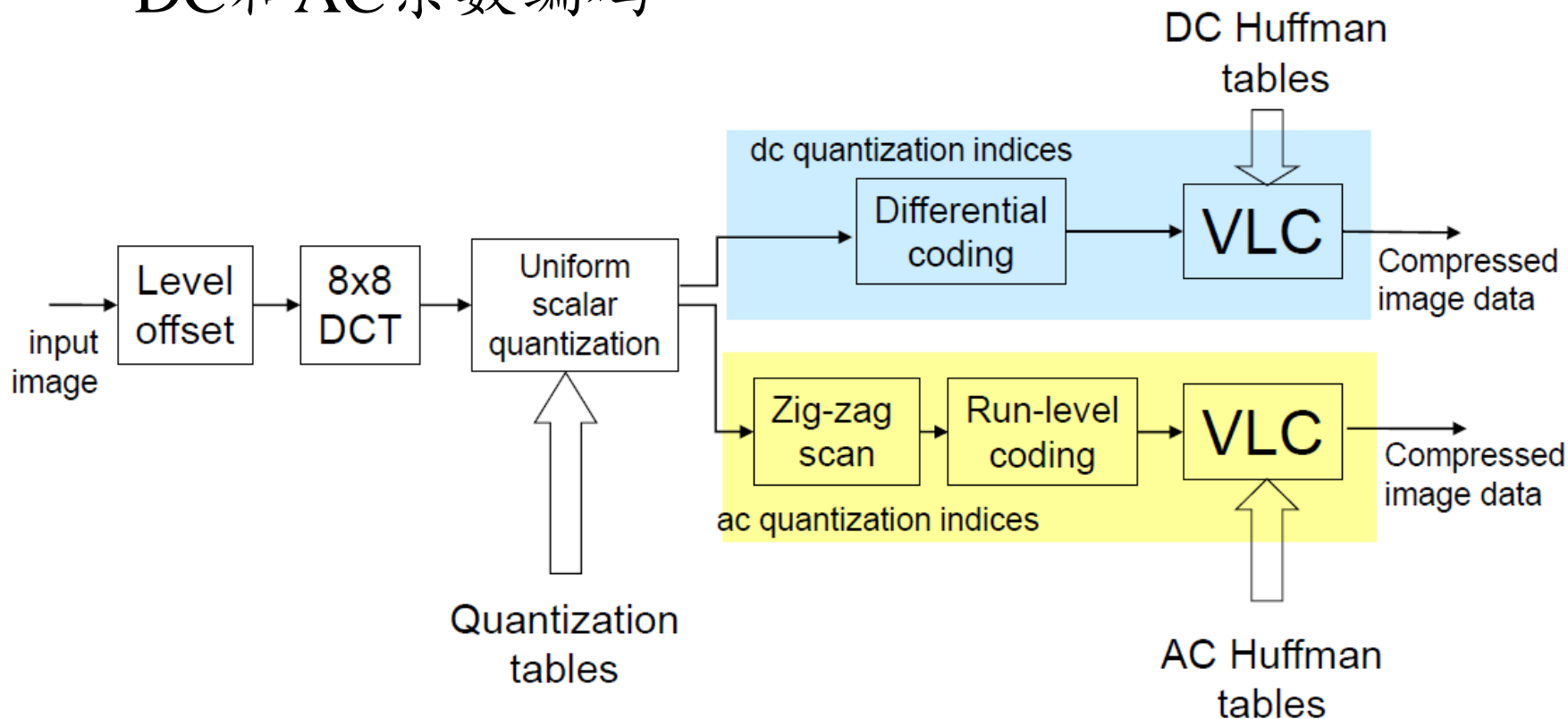
0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63





# JPEG标准

- DC和AC系数编码



# JPEG标准

---

- DC和AC系数编码
  - 将8x8图像块的有1个DC系数和63个AC系数
  - DC系数和AC系数要先转化为中间符号（intermediate symbol），再做熵编码
  - 用中间符号表示DC和AC时，它们都是由两个符号组成：
    - 表示数据大小的可变长度码（Variable-Length Code, VLC），用的代码是霍夫曼码
    - 表示实际幅度的可变长度整数（Variable-Length Integer, VLI），用的代码是补码

# JPEG标准

---

- DC和AC系数编码可分成三步进行
  - 1、DPCM编码（对DC系数）或者行程编码（对AC系数）
  - 2、把DC差值或者行程码字转换成一个中间符号序列
  - 3、给这些符号赋以变长码字，即：使不同符号的编码长度与其出现概率大小成反比。Huffman变长码表必须作为JPEG编码器的输入

# JPEG标准

---

- DC系数的编码
  - 相邻 $8 \times 8$ 块之间的DC系数有两个特点
    - 数值比较大
    - 相邻 $8 \times 8$ 块的DC系数值变化不大
  - 主要包括三个步骤
    - **DPCM编码**
    - **生成中间符号**
    - **符号编码（熵编码）**

# JPEG标准

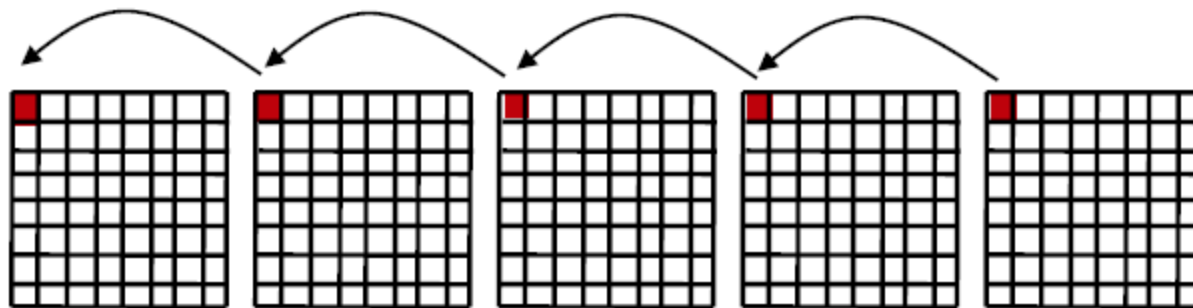
---

- DPCM编码

- 对相邻图像块之间量化DC系数的差值(Delta)进行编码

$$\Delta = DC(0, 0)_i - DC(0, 0)_{i-1}$$

- 例如：亮度分量的前5个图像块的DC系数分别是150、155、149、152、144，使用DPCM编码时产生的输出为150、5、-6、3、-8



# JPEG标准

---

- 生成中间符号
  - DC差值的中间符号由符号1和符号2表示
    - 符号1: (尺寸) **SSSS**
    - 符号2: (幅值) **DIFF**
  - “尺寸”表示DC差值的幅值编码所需的比特数, 而“幅值”表示DC差值的幅值, 范围为  $[-2^{11}, 2^{11}-1]$ 。

# JPEG标准

---

- 生成中间符号

- JPEG标准将DIFF值分成12类，DIFF和SSSS之间的对应关系如表所示
- 例如，可将DC的差值150、5、-6、3、-8转换为中间符号：  
(8, 150)、(3, 5)、(3, -6)、  
(2, 3)、(4, -8)

SSSS	DIFF values
0	0
1	-1,1
2	-3,-2,2,3
3	-7..-4,4..7
4	-15..-8,8..15
5	-31..-16,16..31
6	-63..-32,32..63
7	-127..-64,64..127
8	-255..-128,128..255
9	-511..-256,256..511
10	-1 023..-512,512..1 023
11	-2 047..-1 024,1 024..2 047

# JPEG标准

---

- 符号编码（熵编码）
  - SSSS采用Huffman表中的变长码编码(VLC)，可查表（亮度DC差值码和色度DC差值码表）直接得到。Huffman表必须作为JPEG编码器输入。
  - DIFF用变长整数VLI码编码、即补码表示



# JPEG标准

---

**Table K.3 – Table for luminance DC coefficient differences**

Category	Code length	Code word
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

**Table K.4 – Table for chrominance DC coefficient differences**

Category	Code length	Code word
0	2	00
1	2	01
2	2	10
3	3	110
4	4	1110
5	5	11110
6	6	111110
7	7	1111110
8	8	11111110
9	9	111111110
10	10	1111111110
11	11	11111111110

# JPEG标准

---

编码步骤	亮度DC系数举例				
	$DC(0,0)_0$	$DC(0,0)_1$	$DC(0,0)_2$	$DC(0,0)_3$	$DC(0,0)_4$
DC系数值	150	155	149	152	144
DC差值	150	5	-6	3	-8
中间符号	(8,150)	(3,5)	(3,-6)	(2,3)	(4,-8)
熵和幅度编码	(111110,10010110)	(100,101)	(100,001)	(011,11)	(101,0111)

# JPEG标准

---

- AC系数的编码

- 量化AC系数的特点包含有许多“0”系数，并且许多“0”是连续的。因此使用非常简单和直观的游程长度编码(RLE)对它们进行编码，得到中间符号，最后进行符号编码。

- 主要步骤：

- RLE编码
- 生成中间符号
- 符号编码（熵编码）

# JPEG标准

---

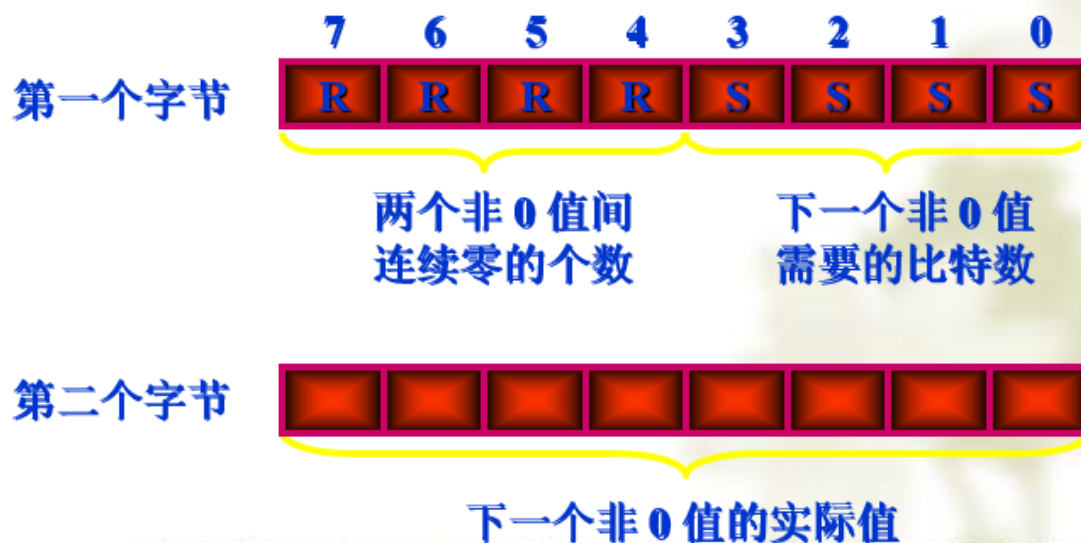
- AC系数的编码
  - 中间符号由符号1和符号2组成
    - 符号1总共8位，高4位表示连续0值系数的个数**RRRR**，低4位**SSSS**表示编码下一个非0值所需要的位数
    - 符号2总共8位，表示下一个非0值的**AC系数**的实际值

# JPEG标准

- AC系数的编码
  - 中间符号由符号1和符号2组成

☞ 符号1: (行程, 尺寸) (RRRRSSSS) R/S

☞ 符号2: (幅值) AC coefficients



# JPEG标准

---

- AC系数的编码
  - 生成中间符号
    - JPEG标准将SSSS值分成11类，SSSS和AC系数之间的对应关系如表所示

SSSS	AC coefficients
1	-1,1
2	-3,-2,2,3
3	-7..-4,4..7
4	-15..-8,8..15
5	-31..-16,16..31
6	-63..-32,32..63
7	-127..-64,64..127
8	-255..-128,128..255
9	-511..-256,256..511
10	-1 023..-512,512..1 023

# JPEG标准

---

- AC系数的编码
  - 生成中间符号
    - 行程RRRR取值范围为1~15, 如果连续0值大于15
      - ✓ (15/0) 表示连续15个0值系数后面跟一个0值系数, 总共连续16个0
      - ✓ (0/0) 表示其后所有系数均为0值系数

# JPEG标准

---

- AC系数的编码

- 生成中间符号

- 例如经过Z字形排列后的系数为

- $15, 0, -2, -1, -1, -1, 0, 0, -1, 0, \dots, 0$

- 中间符号为：

$(3, -5) (1/2, -2) (0/1, -1) (0/1, -1) (0/1, -1) (2/1, -1) (0/0)$



# JPEG标准

---

- AC系数的编码
  - 生成中间符号
    - 例如AC系数 $zz(11) \sim zz(29)$ 都为0,  $zz(30)=3$

中间符号为:  $(15/0) (3/2, 3)$

# JPEG标准

---

- AC系数的编码
  - 符号编码（熵编码）
    - 符号1，即RRRRSSSS或R/S，查霍夫曼码表可得。
    - 符号2，即AC系数，用补码表示

Table K.5 – Table for luminance AC coefficients (sheet 1 of 4)

Run/Size	Code length	Code word
0/0 (EOB)	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	1111111110000010
0/A	16	1111111110000011
1/1	4	1100
1/2	5	11011
1/3	7	1111001
1/4	9	111110110
1/5	11	11111110110
1/6	16	1111111110000100
1/7	16	1111111110000101
1/8	16	1111111110000110
1/9	16	1111111110000111
1/A	16	1111111110001000
2/1	5	11100
2/2	8	11111001
2/3	10	1111110111
2/4	12	111111110100
2/5	16	1111111110001001
2/6	16	1111111110001010
2/7	16	1111111110001011
2/8	16	1111111110001100
2/9	16	1111111110001101
2/A	16	1111111110001110
3/1	6	111010
3/2	9	111110111
3/3	12	111111110101
3/4	16	1111111110001111
3/5	16	1111111110010000
3/6	16	1111111110010001
3/7	16	1111111110010010
3/8	16	1111111110010011
3/9	16	1111111110010100
3/A	16	1111111110010101

Table K.5 (sheet 2 of 4)

Run/Size	Code length	Code word
4/1	6	111011
4/2	10	1111111000
4/3	16	1111111110010110
4/4	16	1111111110010111
4/5	16	1111111110011000
4/6	16	1111111110011001
4/7	16	1111111110011010
4/8	16	1111111110011011
4/9	16	1111111110011100
4/A	16	1111111110011101
5/1	7	1111010
5/2	11	11111110111
5/3	16	1111111110011110
5/4	16	1111111110011111
5/5	16	1111111110100000
5/6	16	1111111110100001
5/7	16	1111111110100010
5/8	16	1111111110100011
5/9	16	1111111110100100
5/A	16	1111111110100101
6/1	7	1111011
6/2	12	1111111101110
6/3	16	1111111110100110
6/4	16	1111111110100111
6/5	16	1111111110101000
6/6	16	1111111110101001
6/7	16	1111111110101010
6/8	16	1111111110101011
6/9	16	1111111110101100
6/A	16	1111111110101101
7/1	8	11111010
7/2	12	1111111101111
7/3	16	1111111110101110
7/4	16	1111111110101111
7/5	16	1111111110110000
7/6	16	1111111110110001
7/7	16	1111111110110010
7/8	16	1111111110110011
7/9	16	1111111110110100
7/A	16	1111111110110101
8/1	9	111111000
8/2	15	111111111000000

Table K.5 (sheet 3 of 4)

Run/Size	Code length	Code word
8/3	16	111111110110110
8/4	16	111111110110111
8/5	16	111111110111000
8/6	16	111111110111001
8/7	16	111111110111010
8/8	16	111111110111011
8/9	16	111111110111100
8/A	16	111111110111101
9/1	9	11111001
9/2	16	111111110111110
9/3	16	111111110111111
9/4	16	111111111000000
9/5	16	111111111000001
9/6	16	111111111000010
9/7	16	111111111000011
9/8	16	111111111000100
9/9	16	111111111000101
9/A	16	111111111000110
A/1	9	11111010
A/2	16	111111111000111
A/3	16	111111111001000
A/4	16	111111111001001
A/5	16	111111111001010
A/6	16	111111111001011
A/7	16	111111111001100
A/8	16	111111111001101
A/9	16	111111111001110
A/A	16	111111111001111
B/1	10	111111001
B/2	16	111111111010000
B/3	16	111111111010001
B/4	16	111111111010010
B/5	16	111111111010011
B/6	16	111111111010100
B/7	16	111111111010101
B/8	16	111111111010110
B/9	16	111111111010111
B/A	16	111111111011000
C/1	10	111111010
C/2	16	111111111011001
C/3	16	111111111011010
C/4	16	111111111011011

Table K.5 (sheet 4 of 4)

Run/Size	Code length	Code word
C/5	16	111111111011100
C/6	16	111111111011101
C/7	16	111111111011110
C/8	16	111111111011111
C/9	16	111111111100000
C/A	16	111111111100001
D/1	11	11111111000
D/2	16	111111111100010
D/3	16	111111111100011
D/4	16	111111111100100
D/5	16	111111111100101
D/6	16	111111111100110
D/7	16	111111111100111
D/8	16	111111111101000
D/9	16	111111111101001
D/A	16	111111111101010
E/1	16	111111111101011
E/2	16	111111111101100
E/3	16	111111111101101
E/4	16	111111111101110
E/5	16	111111111101111
E/6	16	111111111110000
E/7	16	111111111110001
E/8	16	111111111110010
E/9	16	111111111110011
E/A	16	111111111110100
F/0 (ZRL)	11	1111111001
F/1	16	111111111110101
F/2	16	111111111110110
F/3	16	111111111110111
F/4	16	111111111111000
F/5	16	111111111111001
F/6	16	111111111111010
F/7	16	111111111111011
F/8	16	111111111111100
F/9	16	111111111111101
F/A	16	111111111111110

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

源图象样本

144	146	149	152	154	156	156	156
148	150	152	154	156	156	156	156
155	156	157	158	158	157	156	155
160	161	161	162	161	159	157	155
163	163	164	163	162	160	158	156
163	164	164	164	162	160	158	157
160	161	162	162	162	161	159	158
158	159	161	161	162	161	159	158

重构图象样本

235.6	-1.0	-12.1	-5.20	2.1	-1.7	-2.7	1.3
-22.6	-18.5	-6.2	-3.2	-2.9	-0.1	0.4	-1.2
-10.9	-9.3	-1.6	1.5	0.2	-0.9	-0.6	-0.1
-7.1	-1.9	0.2	1.5	0.9	-0.1	0.0	0.3
-0.6	-0.8	1.5	1.6	-0.1	-0.7	0.6	1.3
1.8	-0.2	-1.6	-0.3	-0.8	1.5	1.0	-1.0
-1.3	-0.4	-0.3	-1.5	-0.5	1.7	1.1	-0.8
-2.6	1.6	-3.8	-1.8	1.9	1.2	-0.6	-0.4

FDCT系数

15	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-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

规格化量化系数

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

240	0	-10	0	0	0	0	0
-24	-12	0	0	0	0	0	0
-14	-13	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

逆量化后的系数

15	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-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

规格化量化系数

# JPEG标准

---

- 压缩比和图像质量

压缩效果(比特/像素)	质 量
0.25~0.50	中~好, 满足某些应用
0.50~0.75	好~很好, 满足多数应用
0.75~1.5	极好, 满足大多数应用
1.2~2.0	与原始图像分不出来

# JPEG标准

---

- JPEG文件格式

- JPEG文件大体上可以分成标记码(tag)和压缩数据两个部分。标记码部分给出了JPEG图像的所有信息，如图像的宽、高、Huffman表、量化表等
- JPEG文件使用的颜色空间是电视图像信号数字化标准ITU-R BT 601推荐标准规定的 $YC_bC_r$ 彩色空间。从RGB转换成 $YC_bC_r$ 的计算公式如下：

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

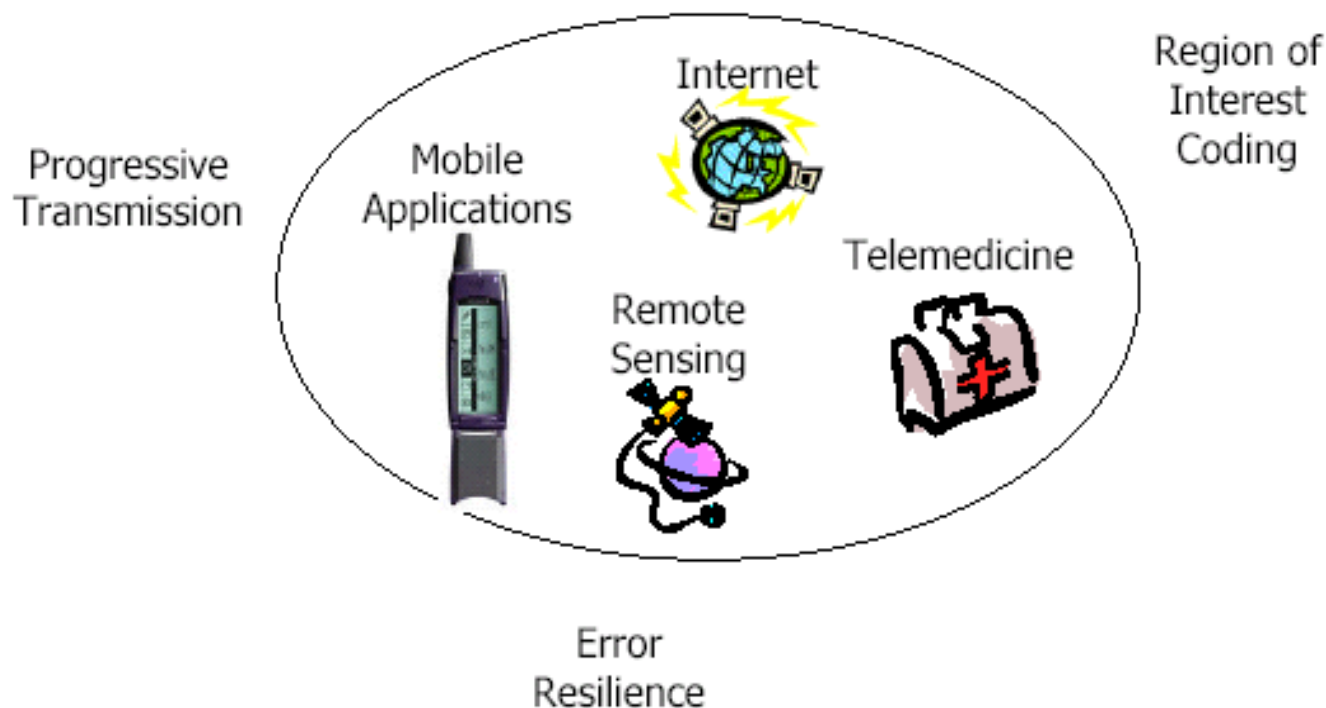
$$Cb = -0.1687 R - 0.3313 G + 0.5 B + 128$$

$$Cr = 0.5 R - 0.4187 G - 0.0813 B + 128$$

# JPEG 2000简介

---

JPEG 2000标准旨在进一步改进目前压缩算法的性能，以适应低带宽、高噪声环境，以及医疗图像、电子图书馆、传真、Internet网上客户/服务器，保安等方面的应用



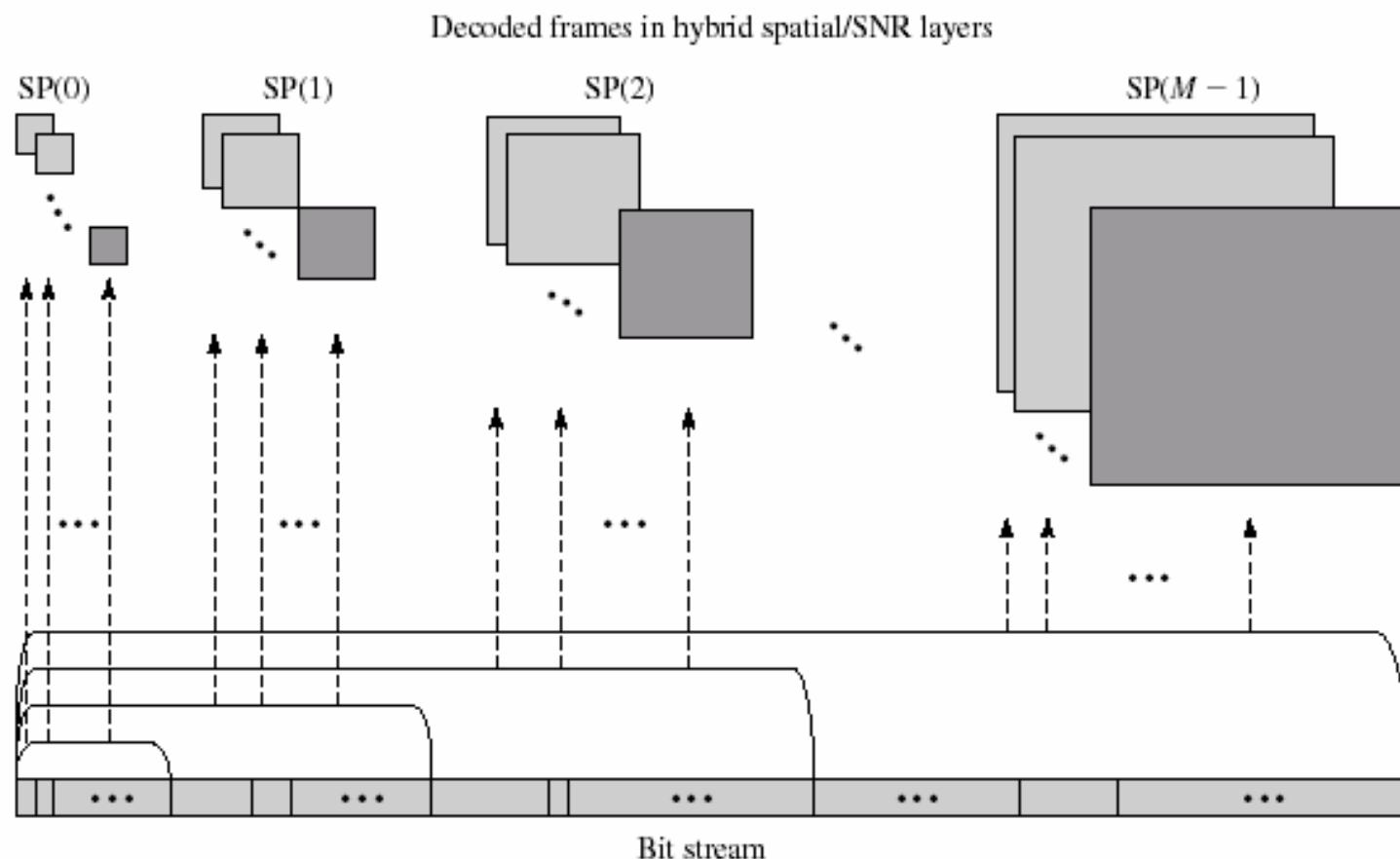


# JPEG 2000简介

---

- 与过去的图像压缩标准相比，JPEG2000标准既提高了性能又增加了功能
  - 在相同质量的前提下与JPEG标准相比，JPEG 2000标准的压缩比可提高20%以上
  - JPEG 2000能实现渐进传输（progressive transmission）
  - JPEG 2000另一个重要特性是支持感兴趣区域（ROI）的编码

# JPEG 2000简介



**Figure 11.7**  $N \times M$  layers of combined spatial/quality scalability. Reprinted from I. Sodagar, H.-J. Lee, P. Hatrack, and Y.-Q. Zhang, Scalable wavelet coding for synthetic/natural hybrid images, *IEEE Trans. Circuits Syst. for Video Technology* (March 1999), 9:244–54. Copyright 1999 IEEE.

# JPEG 2000简介

---



▲ 17. Example of SNR scalability. Part of the decompressed image "bike" at (a) 0.125 b/p, (b) 0.25 b/p, and (c) 0.5 b/p.

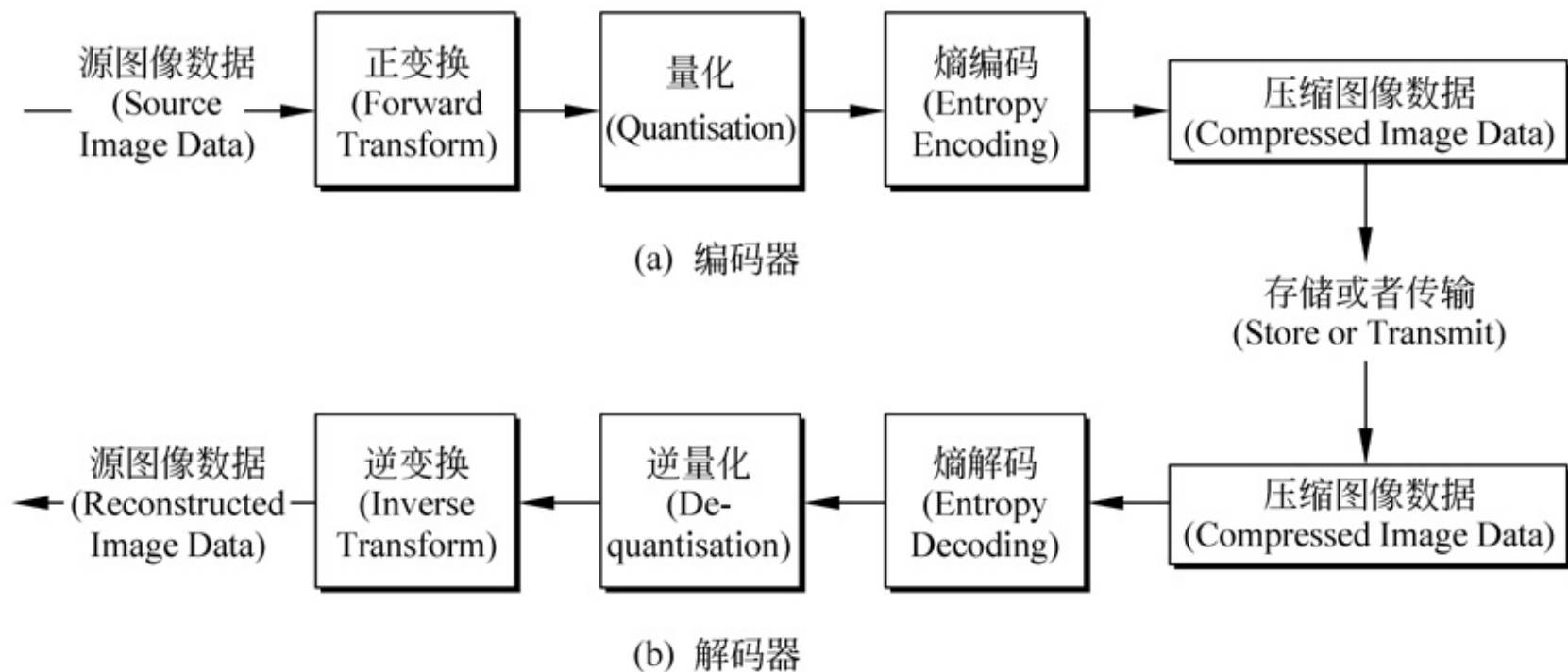
# JPEG 2000简介



▲ 18. Example of the progressive-by-resolution decoding for the color image "bike."

# JPEG 2000简介

- JPEG2000的基本结构





# JPEG 2000简介

- 图像的小波分解方法

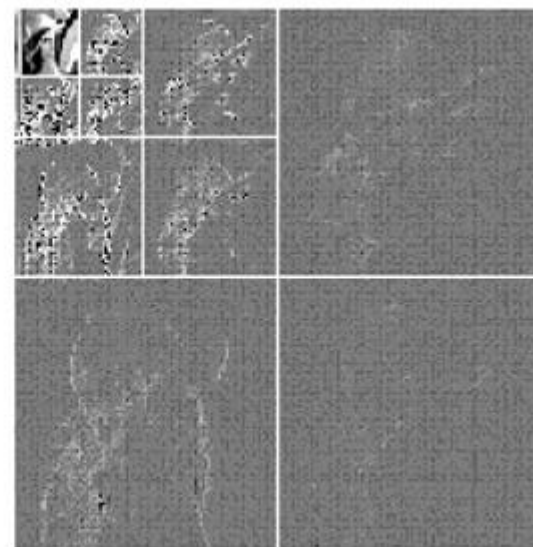


(a) 一级分解

...



(b) 三级分解



(c) Lena 三级分解图



**Original Carol Image (512 x 512 Pixels, 24-Bit RGB, Size 786K)**



75:1, 10.6 Kbyte

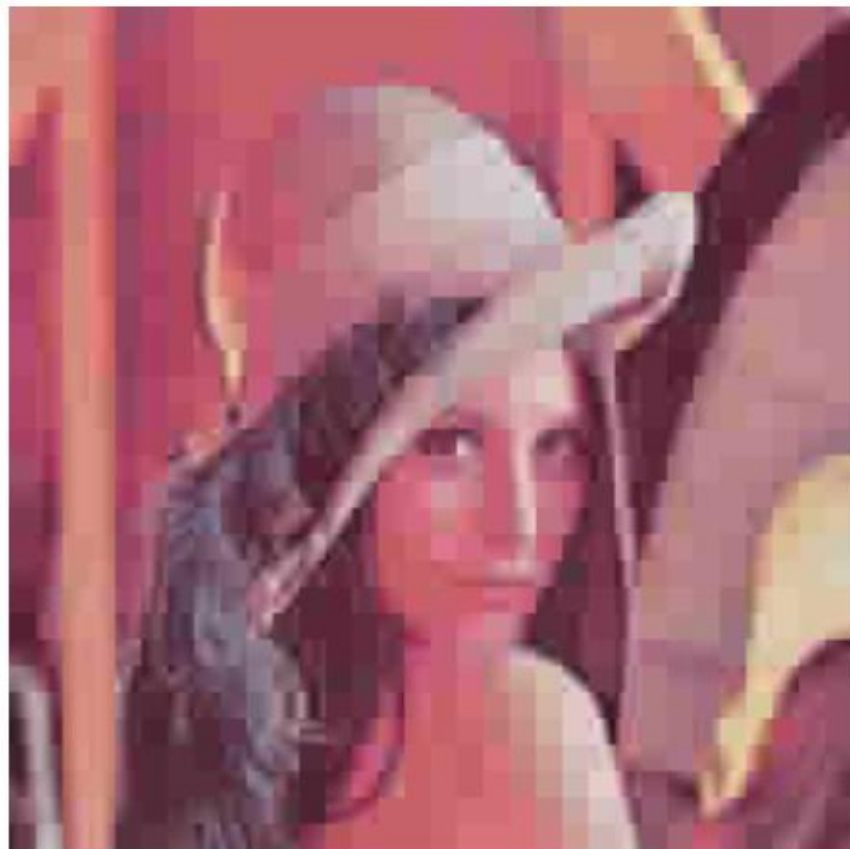




150:1, 5.3 Kbyte



300:1, 2.6 Kbyte



Lenna, 256x256 RGB  
Baseline JPEG: 4572 bytes



Lenna, 256x256 RGB  
JPEG-2000: 4572 bytes

# Web Links

---

- JPEG home

<http://www.jpeg.org/>

- MPEG home

<http://www.chiariglione.org/mpeg/>

- ISO

<http://www.iso.ch/>

- ITU-T

<http://www.itu.int/ITU-T/>

- AVS

<http://www.avs.org.cn/>