



多媒体技术基础

第三章 多媒体数据压缩

§ 3.3 图像数据的压缩标准



2022年11月1日





第三章 多媒体数据压缩

- ◆ § 3.1 无损数据压缩
- ◆ § 3.2 音频数据的压缩标准
- ◆ § 3.3 图像数据的压缩标准
 - § 3.3.1 图像的冗余
 - § 3.3.2 变换编码
 - § 3.3.3 JPEG原理
 - § 3.3.4 JPEG示例
- ◆ § 3.4 视频数据的压缩标准



图像数据冗余

- ◆ 空间冗余
- ◆ 时间冗余
- ◆ 视觉冗余
- ◆ 知识冗余
- ◆ 结构冗余
- ◆ 熵冗余

音频数据时域冗余

- ◆ 幅度的非均匀分布
- ◆ 样本间的相关
- ◆ 长时自相关函数
- ◆ 静止系数



空间冗余

◆ 一幅图像表面上各采样点的颜色之间往往存在着空间连贯性，基于离散像素采样来表示物体表面颜色的像素存储方式可利用空间连贯性，达到减少数据量的目的。例如，**在静态图像中有一块表面颜色均匀的区域**，在此区域中所有点的光强和色彩以及饱和度都是相同的，因此数据有很大的空间冗余。

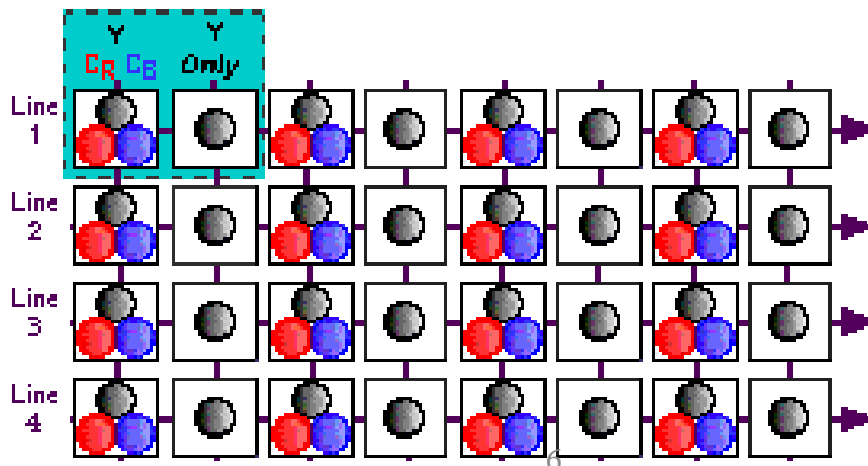


◆ 运动图像一般为位于一时间轴区间的一组连续画面，其中的相邻帧往往包含相同的背景和移动物体，只不过移动物体所在的空间位置略有不同，所以后一帧的数据与前一帧的数据有许多共同的地方，这种共同性是由于相邻帧记录了相邻时刻的同一场景画面，所以称为时间冗余。



◆人类的视觉系统对图像场的敏感度是非均匀的。但在记录原始的图像数据时，通常假定视觉系统近似线性的和均匀的，对视觉敏感和不敏感的部分同等对待，从而产生比理想编码（即把视觉敏感和不敏感的部分区分开来的编码）更多的数据，这就是视觉冗余。

◆例如，利用人眼对蓝光不敏感的视觉特性，在对彩色图像编码时，就可以用较低的精度对蓝色分量进行编码。





知识冗余

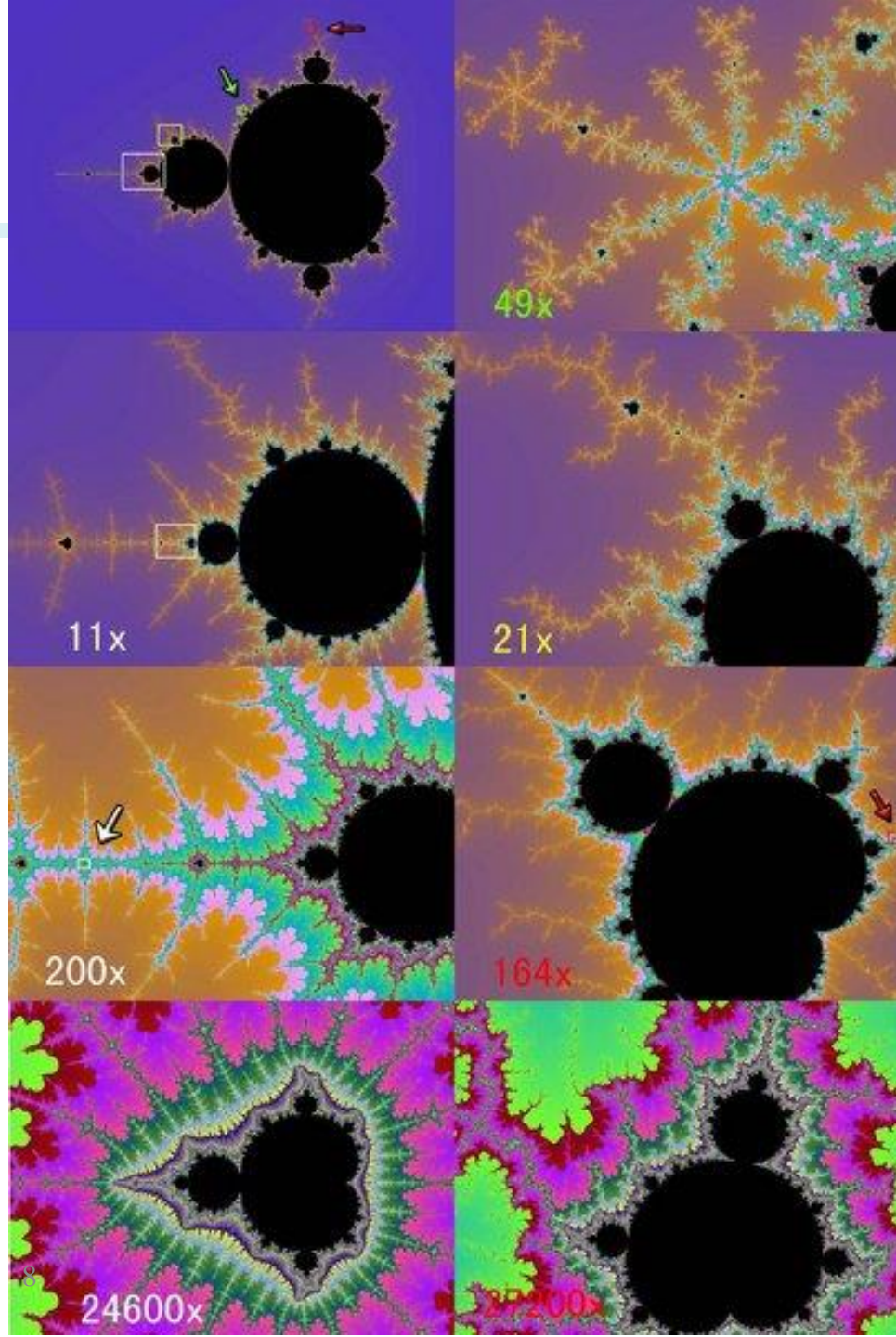
- ◆有些图像的理解与某些基础知识有相当大的相关性。
- ◆例如：人脸的图像有固定的结构。比如说嘴的上方有鼻子，鼻子的上方有眼睛，鼻子位于正脸图像的中线上，等等。这类规律性的结构可以由先验知识和背景知识中得到，我们称此类冗余为知识冗余。





结构冗余

◆有些图像的纹理区，图像的像素值存在着明显的分布模式。例如，方格状的地板图案等。我们称此为**结构冗余**。已知分布模式，可以通过某一过程生成图像。

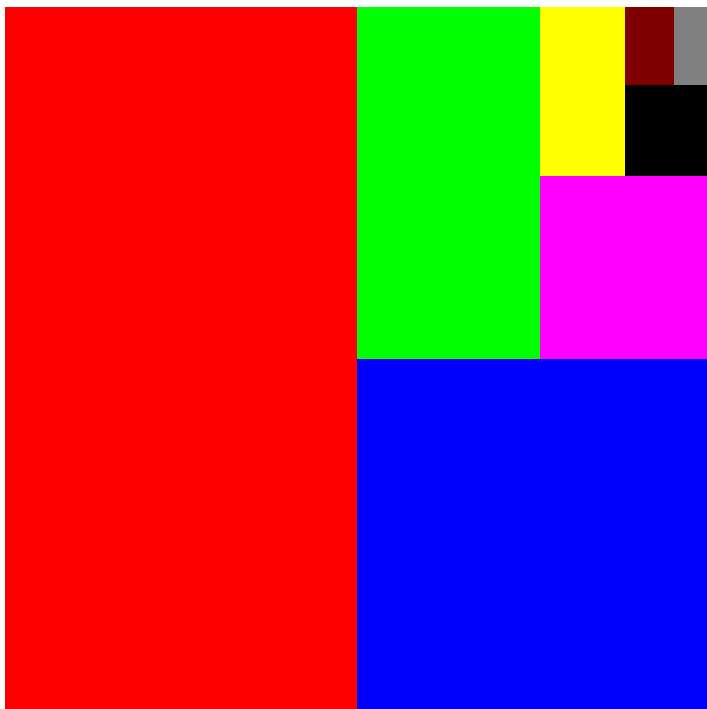




熵冗余

◆ 8色 ($1/2 + 1/4 + 1/8 + 1/16 + 1/32 + 1/64 + 1/128$)

◆ 二值





几个冗余有关的例子

- ◆ SubSampling/ReSampling
- ◆ R、G、B三个通道
- ◆ 灰度图的BitPlane
- ◆ Dither



Subsampling(下采样)

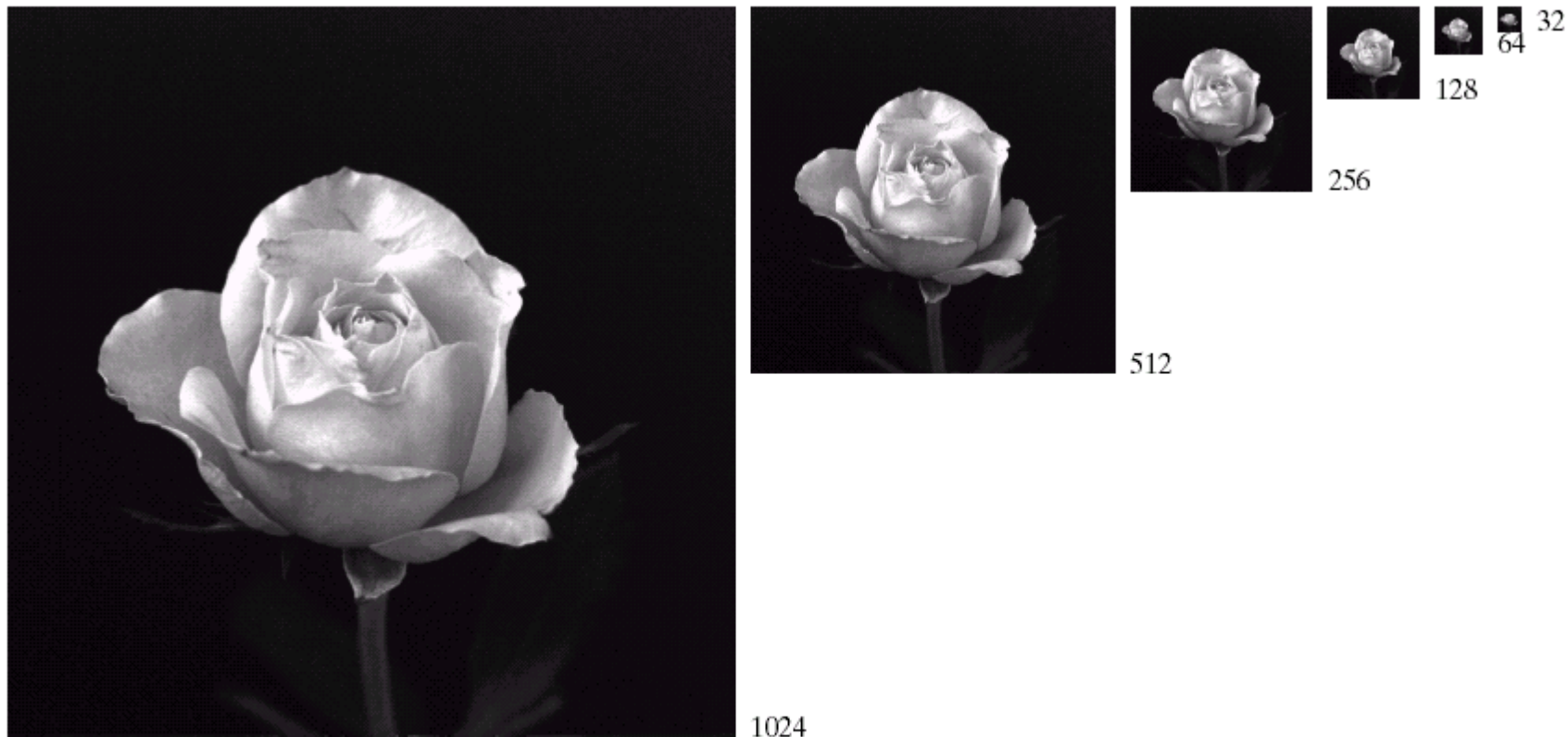
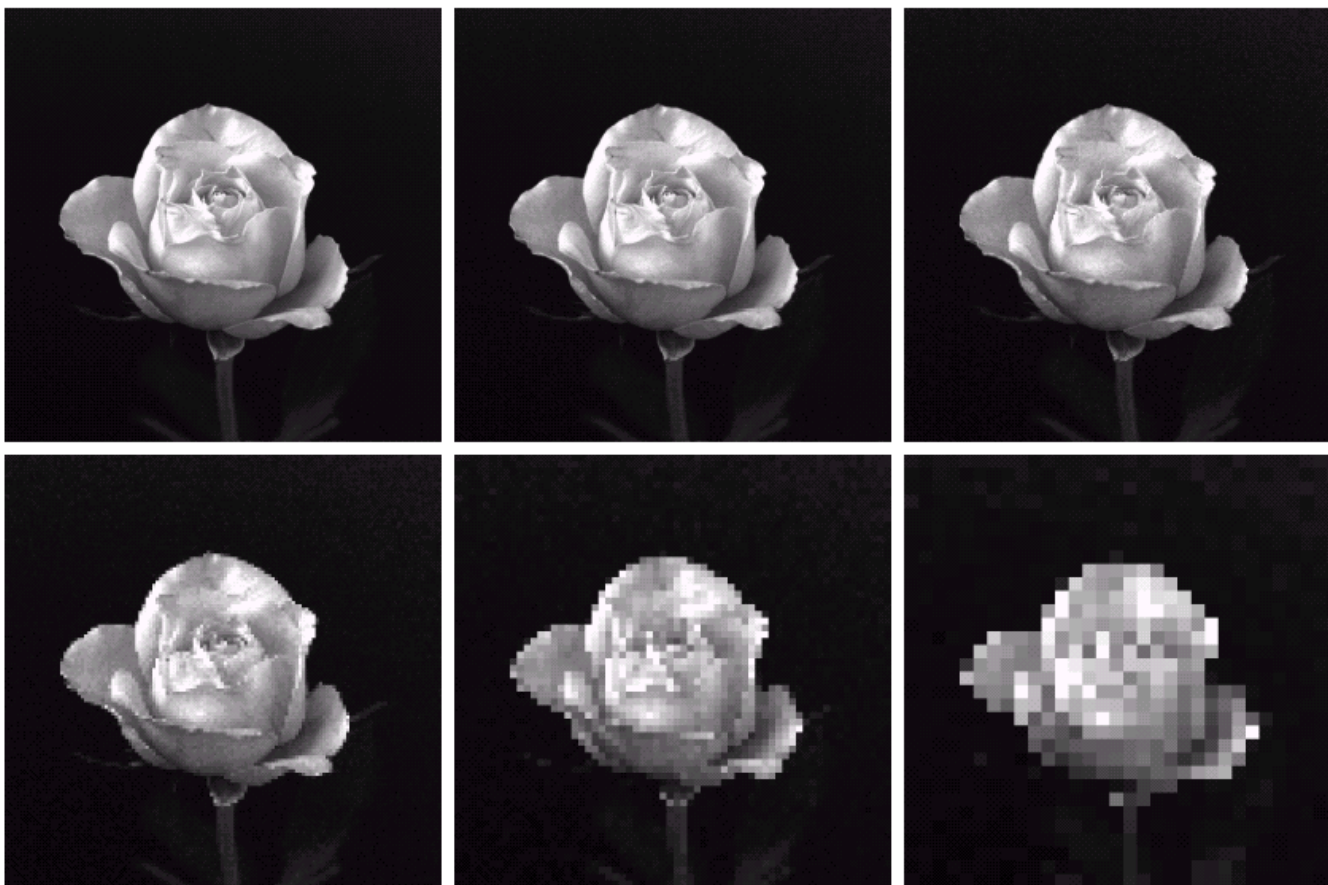


FIGURE 2.19 A 1024×1024 , 8-bit image subsampled down to size 32×32 pixels. The number of allowable gray levels was kept at 256.

Resampling(重采样)



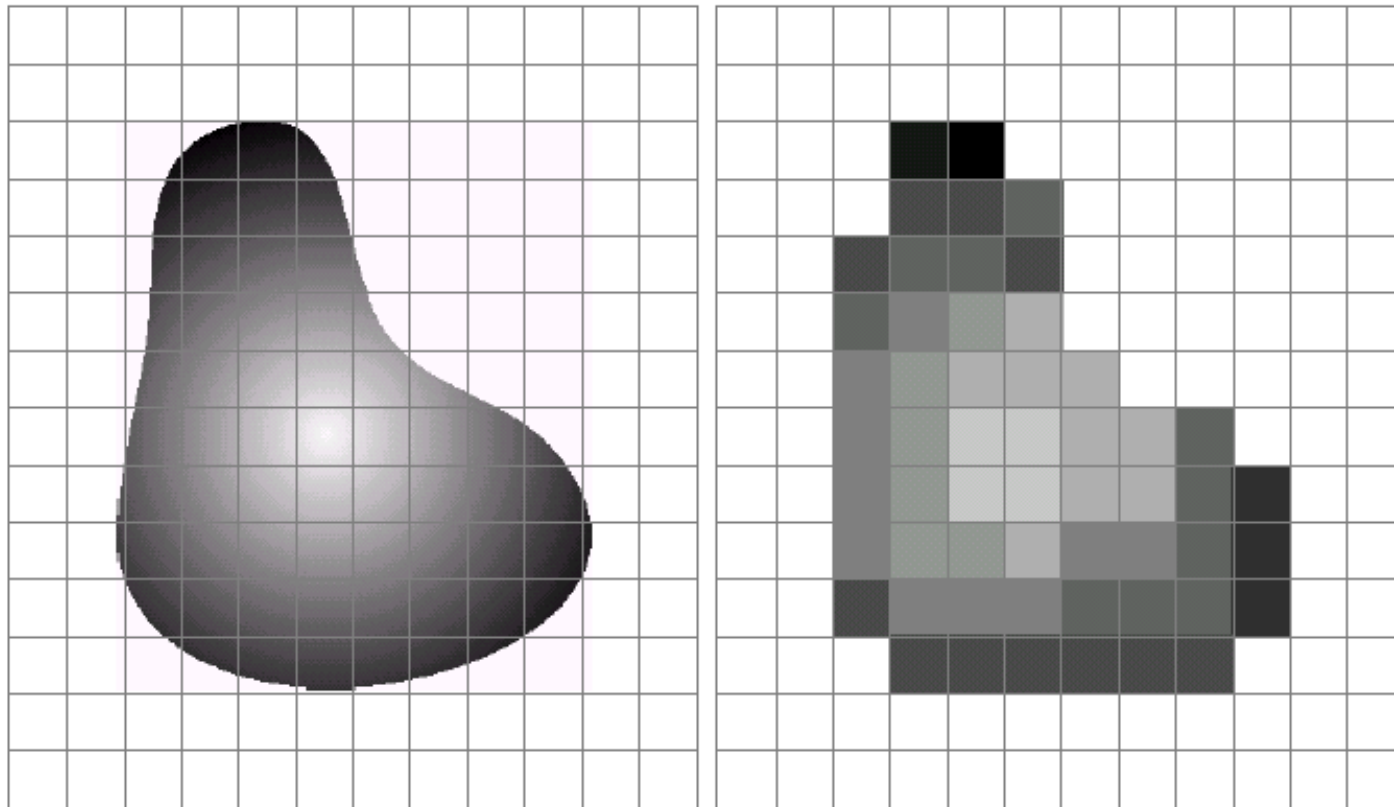
a	b	c
d	e	f

FIGURE 2.20 (a) 1024×1024 , 8-bit image. (b) 512×512 image resampled into 1024×1024 pixels by row and column duplication. (c) through (f) 256×256 , 128×128 , 64×64 , and 32×32 images resampled into 1024×1024 pixels.



Sampling and Quantization

空间域的采样和量化



a b

FIGURE 2.17 (a) Continuous image projected onto a sensor array. (b) Result of image sampling and quantization.





量化噪声示例

◆ Notice the quantization error, most noticeable in the 8 and 2 bpp images – large flat areas with no detail



256bpp



32bpp



8bpp

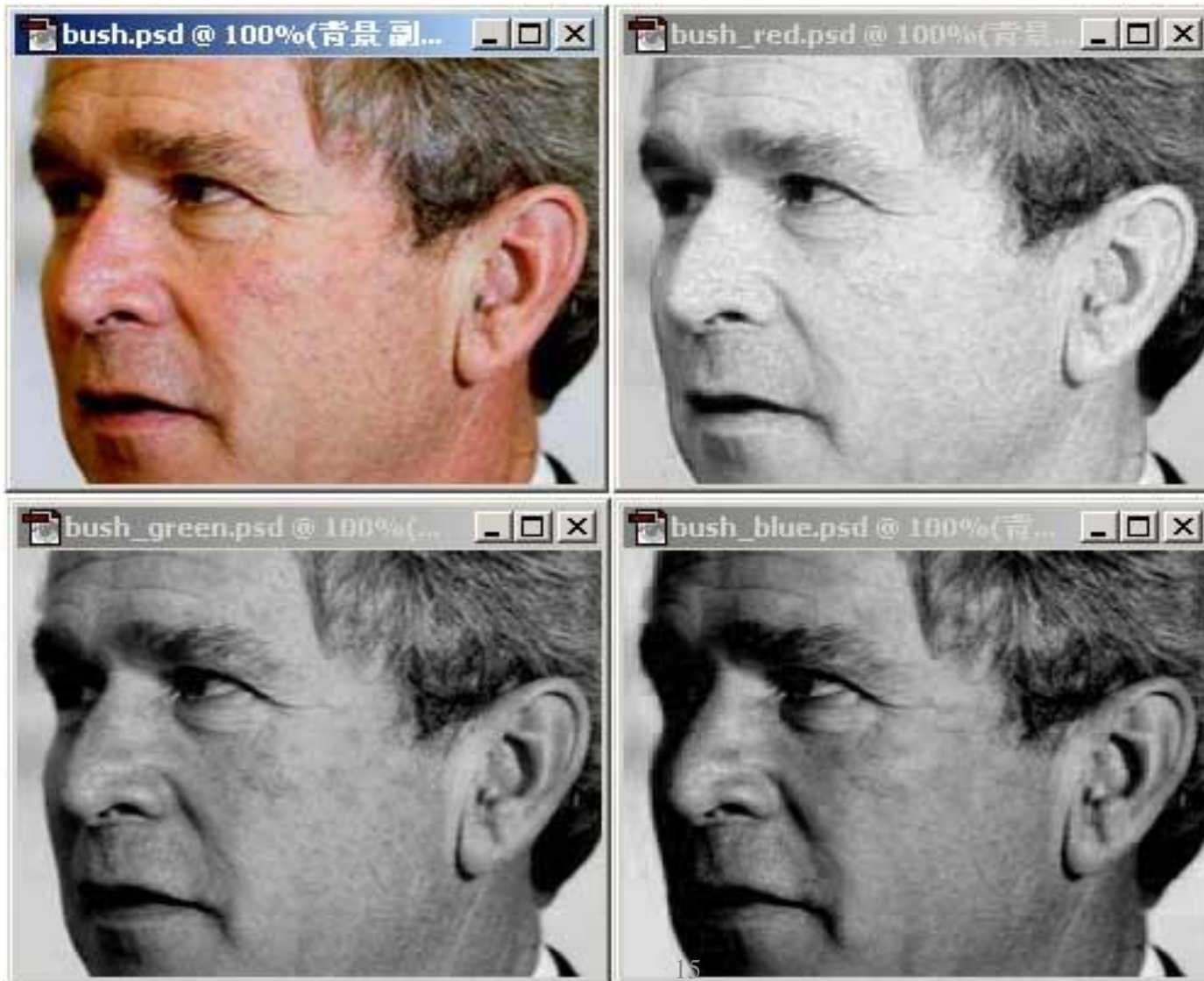


2bpp





Red、Green、Blue三个通道





Bitplanes(比特平面)



Original 8bits/pixel

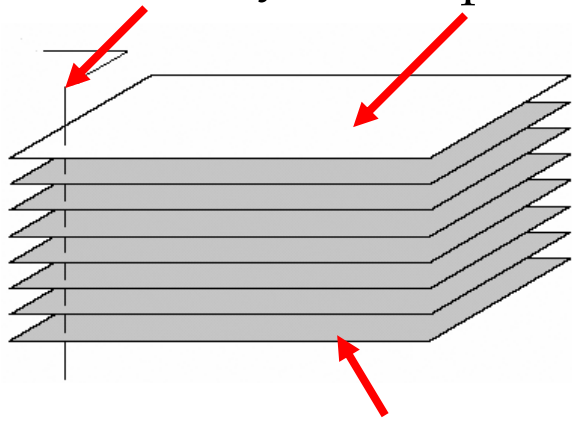


Bitplane 7



Bitplane 6

one 8-bit byte Bitplane 7



Bitplane 0



Bitplane 5



Bitplane 4

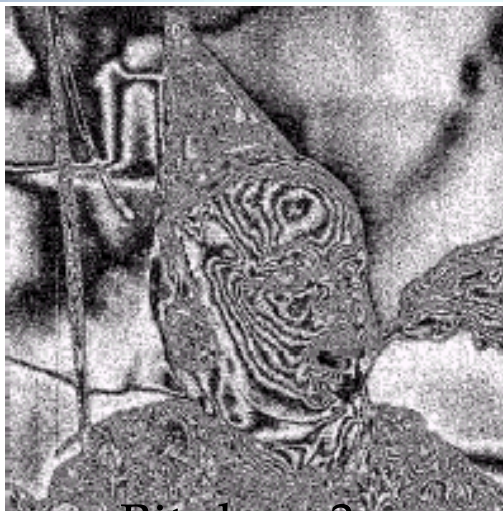




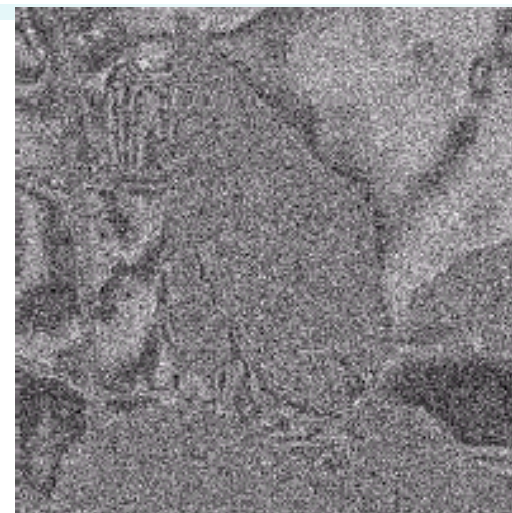
Bitplanes (比特平面)



Original 8bits/pixel

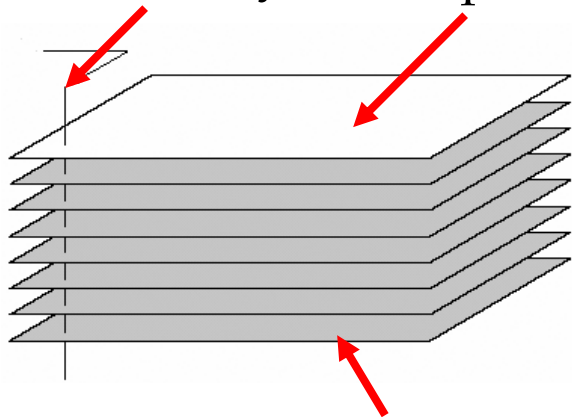


Bitplane 3



Bitplane 2

one 8-bit byte Bitplane 7



Bitplane 0



Bitplane 1

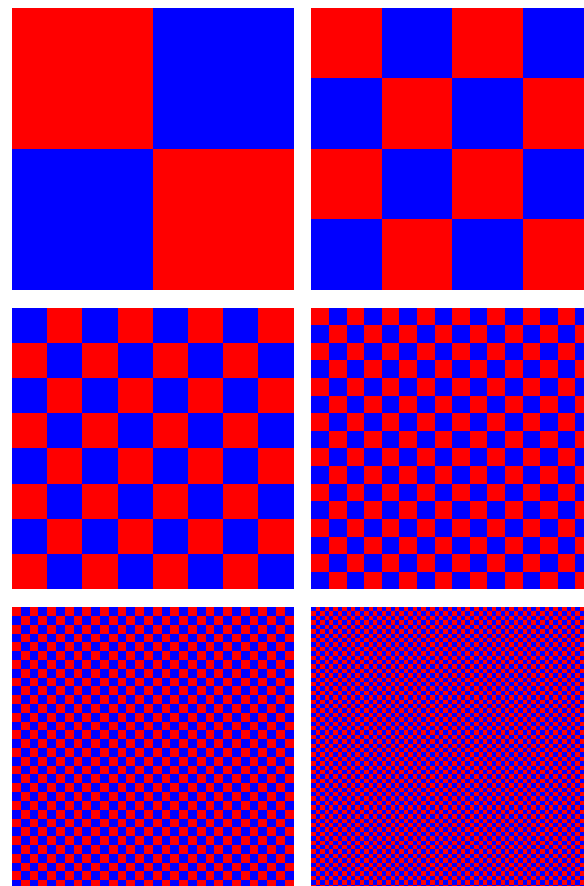
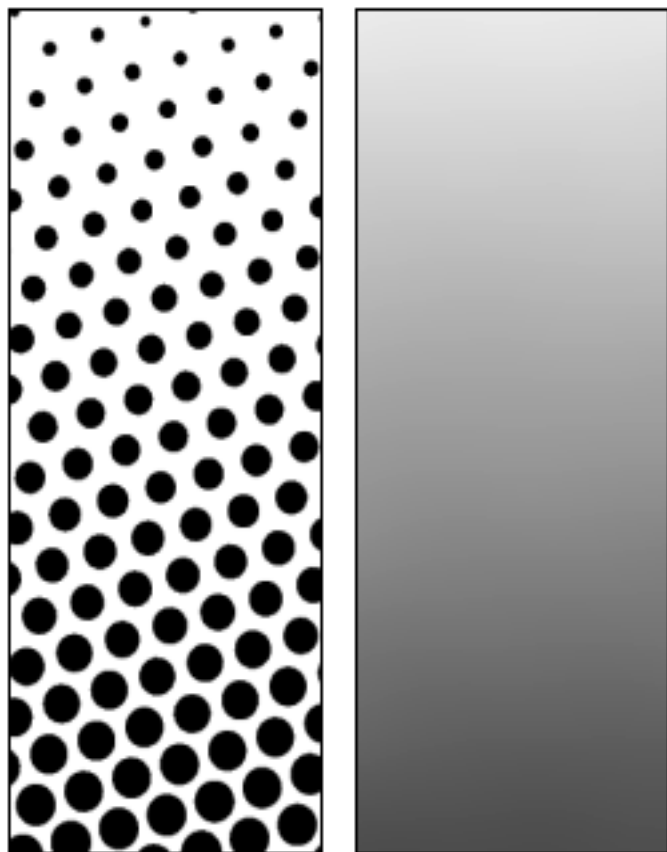


Bitplane 0

表现细节的比特
LSB (Least important Bit)?

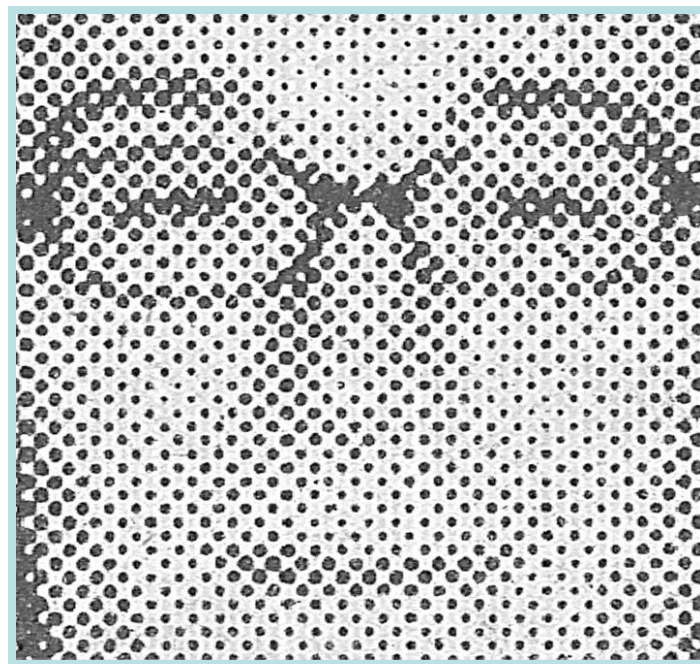
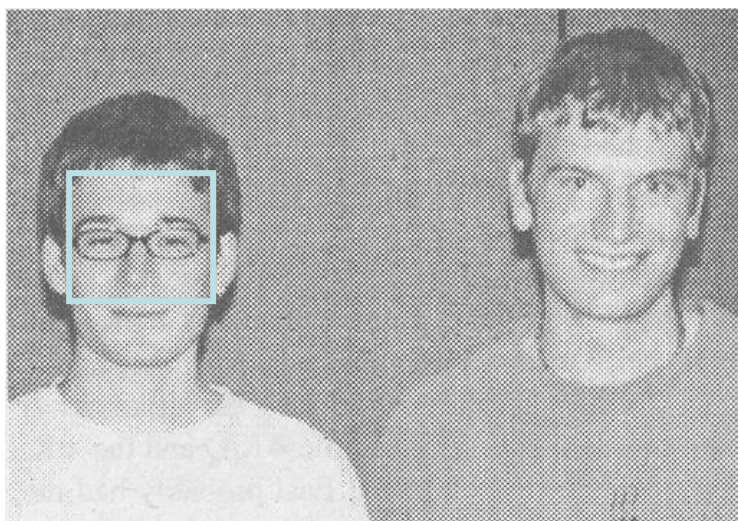


Dithering(抖动)





报纸上用黑白表示灰度





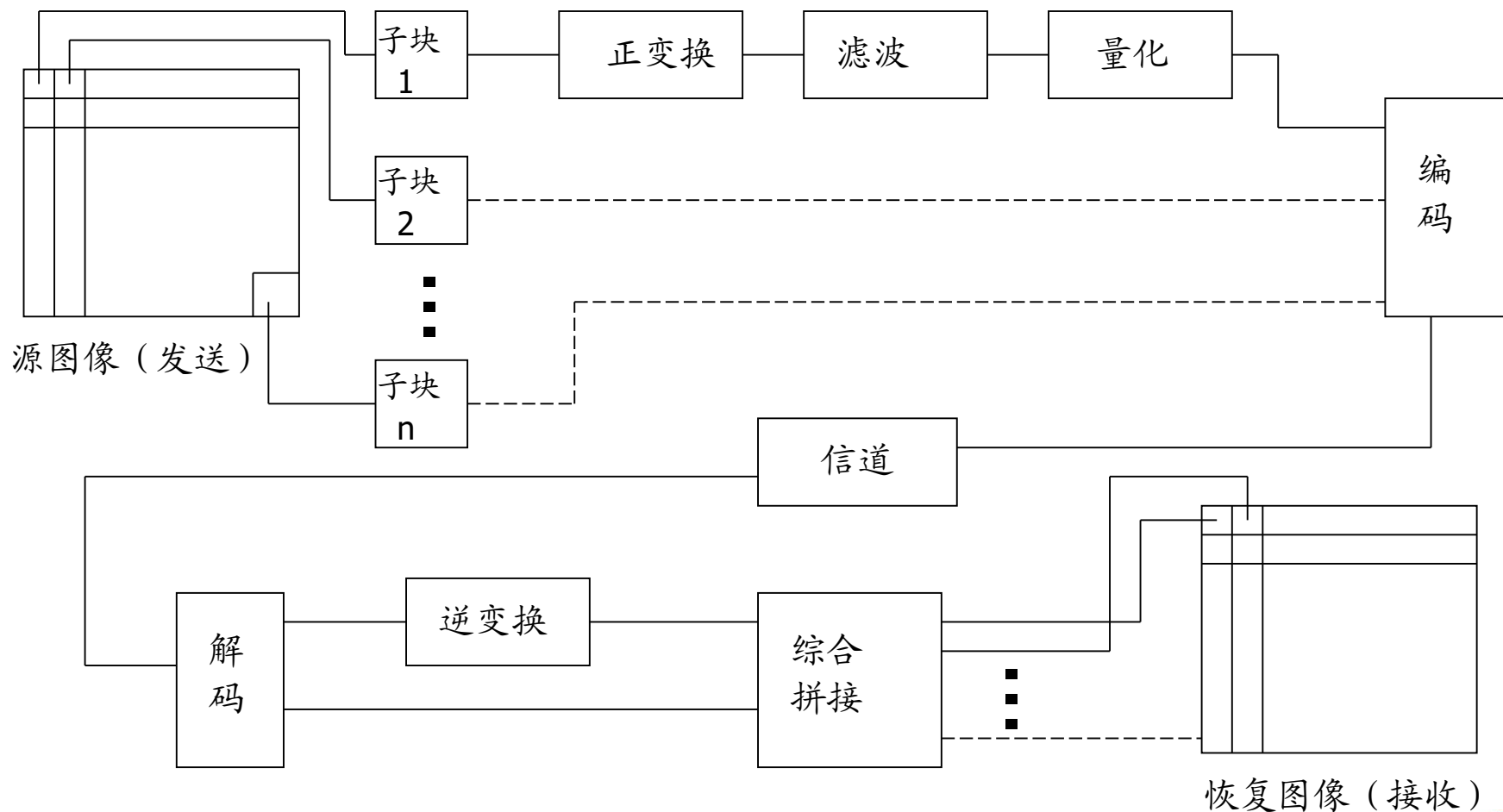
第三章 多媒体数据压缩

- ◆ § 3.1 无损数据压缩
- ◆ § 3.2 音频数据的压缩标准
- ◆ § 3.3 图像数据的压缩标准
 - § 3.3.1 图像的冗余
 - § 3.3.2 变换编码
 - § 3.3.3 JPEG原理
 - § 3.3.4 JPEG示例
- ◆ § 3.4 视频数据的压缩标准





分块变换编码原理示意





常见变换编码

- ◆ 傅立叶(Fourier)变换
- ◆ 沃尔什(Walsh)变换
- ◆ 离散正弦(DST)变换
- ◆ 离散余弦(DCT)变换
- ◆ 哈尔(Haar)变换
- ◆ 斜(Slant)变换
- ◆ K-L(Karhunen-Loeve)变换
- ◆ 小波(Wavelet)变换
- ◆

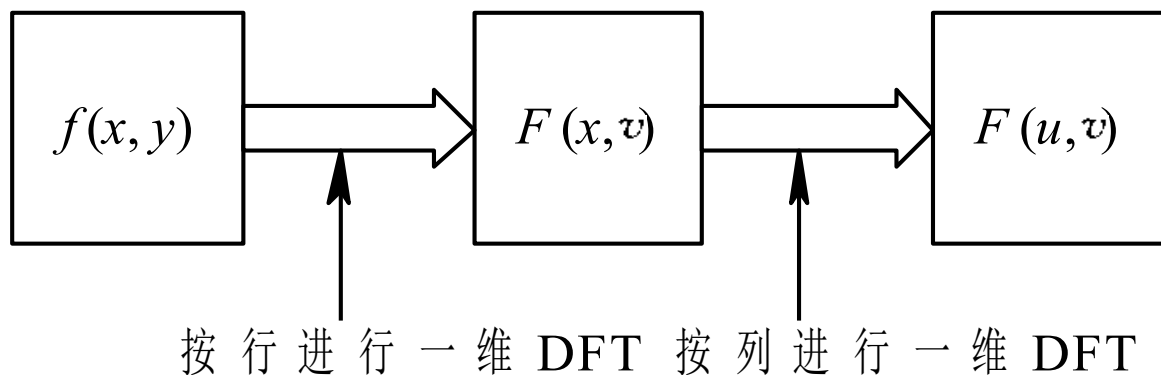
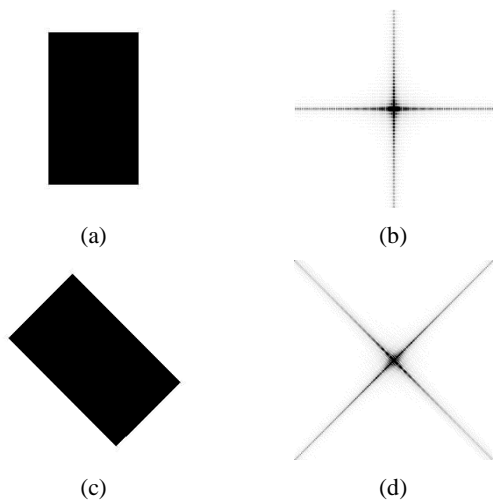




二维DFT

$$F[f(x, y)] = F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(\frac{ux}{M} + \frac{vy}{N})}$$

$$F^{-1}[F(u, v)] = f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(\frac{ux}{M} + \frac{vy}{N})}$$



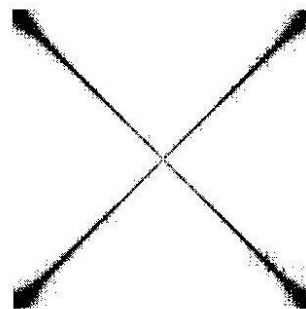


二维DCT

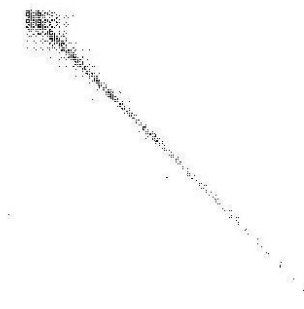
$$F(u) = C(u) \sqrt{\frac{2}{N}} \sum_{x=0}^{N-1} f(x) \cos \frac{(2x+1)u\pi}{2N}$$

$$F(u, v) = \frac{2}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) C(u) C(v) \cos \frac{(2x+1)u\pi}{2M} \cos \frac{(2y+1)v\pi}{2N}$$

与一维DCT变换类似，二维DCT变换后能量集中到了低频部分



(a)



(b)

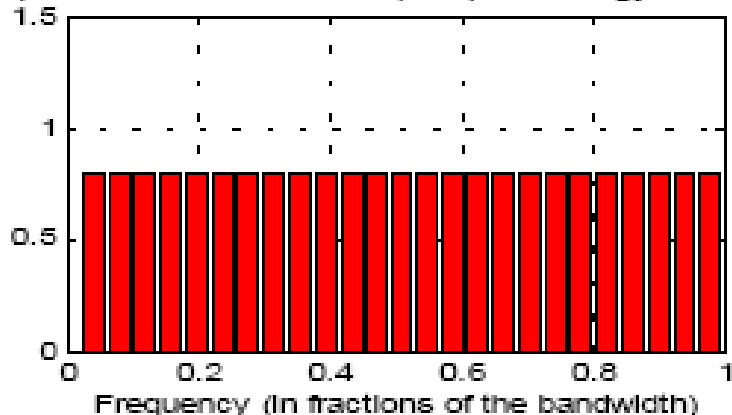




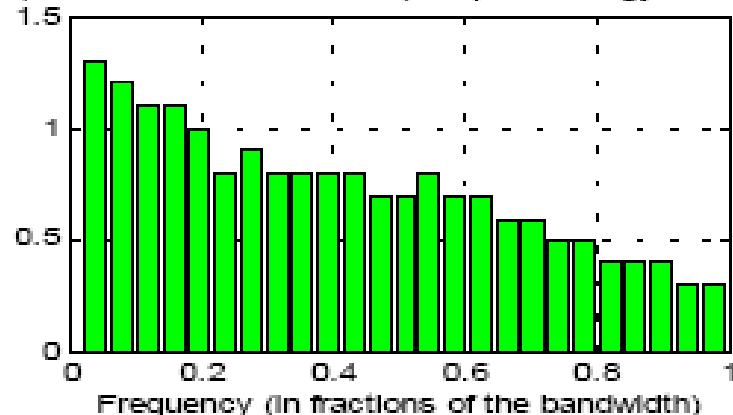
Spectral resolution

谱分辨率(不同变换在变换域的分辨率不同)

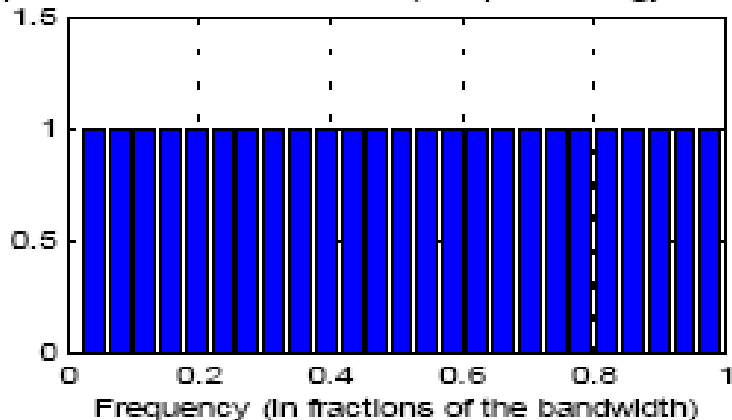
Spectral resolution of DFT (samples; energy level 0.5)



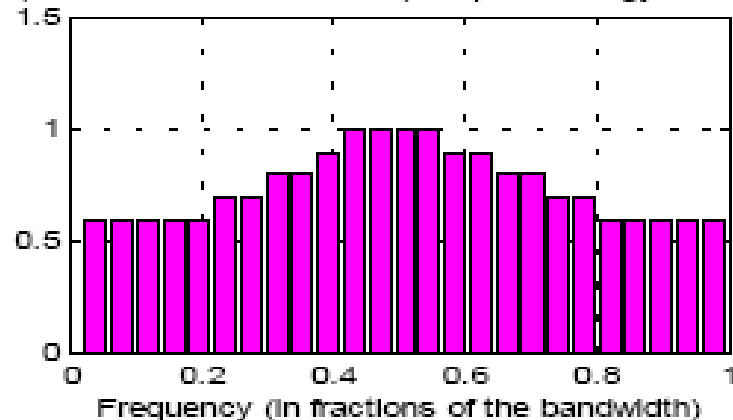
Spectral resolution of DCT (samples; energy level 0.5)



Spectral resolution of MDCT (samples; energy level 0.5)



Spectral resolution of DST (samples; energy level 0.5)



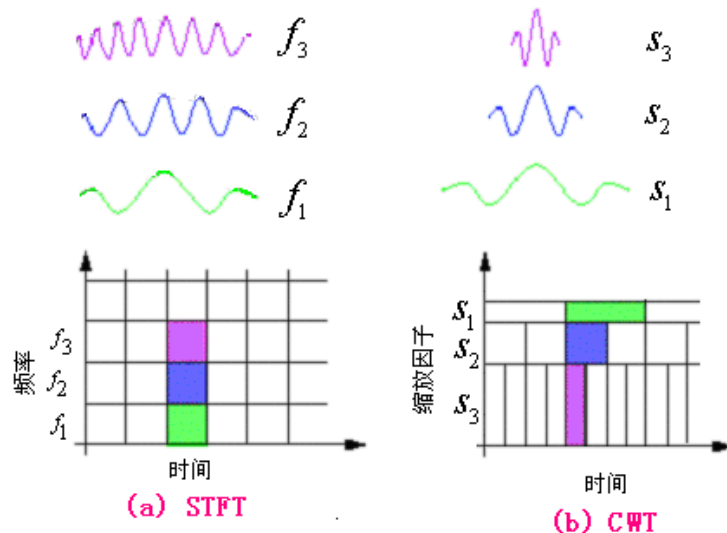
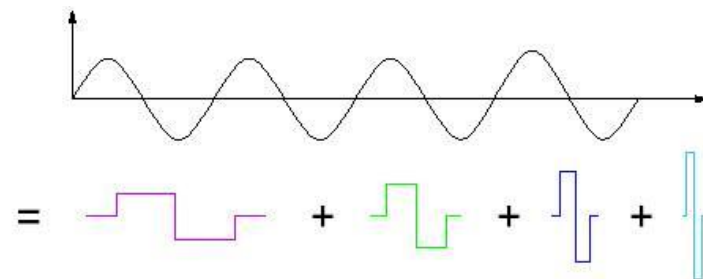
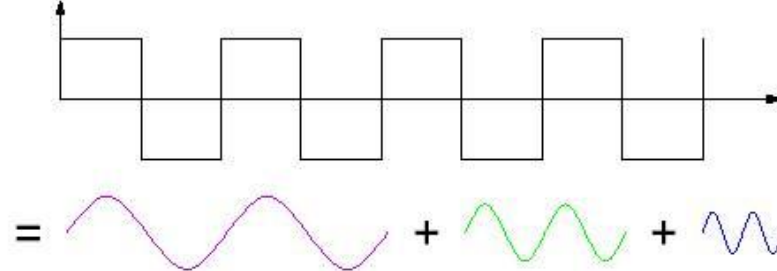


其他变换

◆ Alfred Haar对在函数空间中寻找一个与傅立叶类类似的基非常感兴趣。他最早发现和使用的小波。

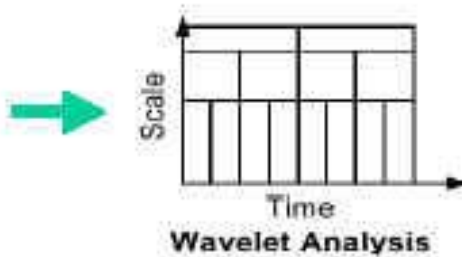
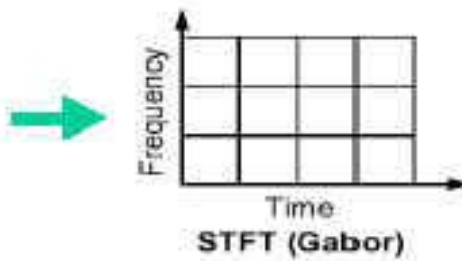
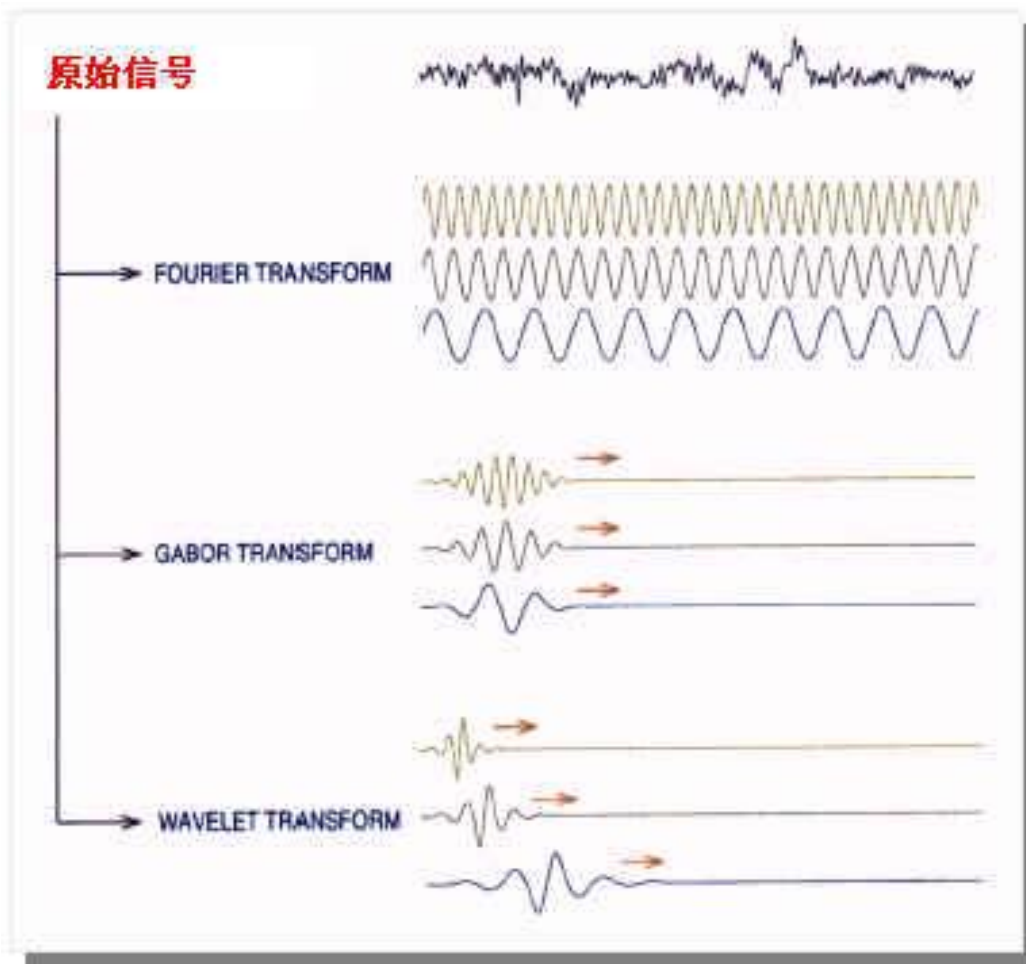
◆ 20世纪40年代Gabor开发了STFT (short time Fourier transform)

◆ 20世纪70年代, Jean Morlet提出了小波变换WT(wavelet transform)的概念。



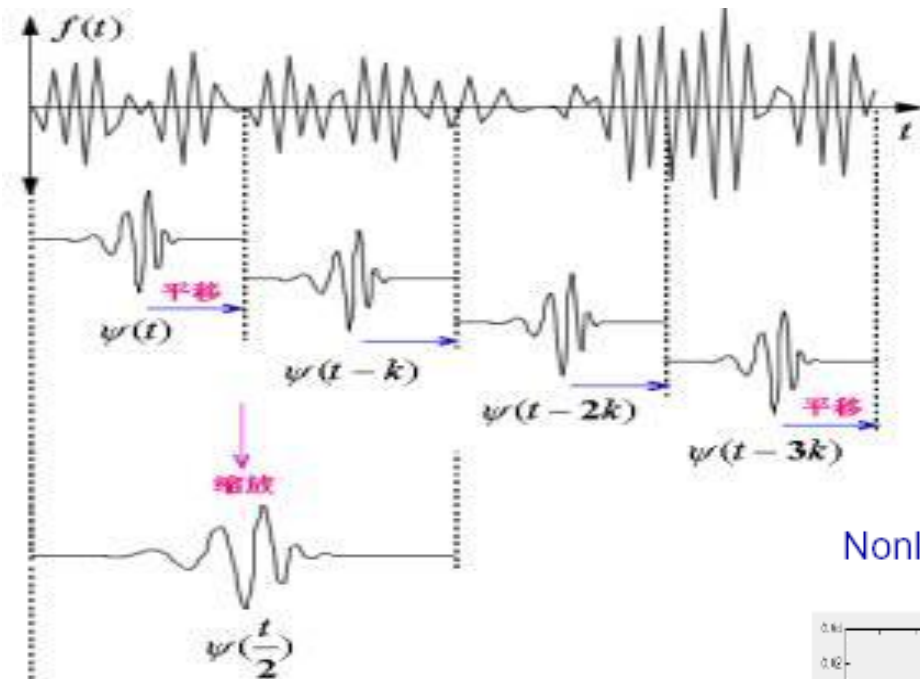


时间和频率分辨率

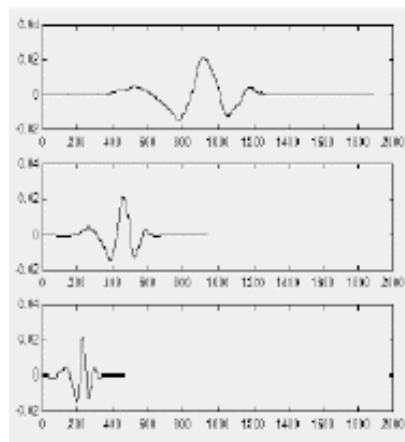




scaled & translation 尺度和平移量



Nonlinear mapping between scale factor and frequency



Scale
↓
 $f(t) = \psi(t) ; a = 1$

$f(t) = \psi(2t) ; a = \frac{1}{2}$

$f(t) = \psi(4t) ; a = \frac{1}{4}$





第三章 多媒体数据压缩

- ◆ § 3.1 无损数据压缩
- ◆ § 3.2 音频数据的压缩标准
- ◆ § 3.3 图像数据的压缩标准
 - § 3.3.1 图像的冗余
 - § 3.3.2 变换编码
 - § 3.3.3 JPEG原理
 - § 3.3.4 JPEG示例
- ◆ § 3.4 视频数据的压缩标准





JPEG简介

◆ JPEG(Joint Photographic Experts Group) 是一个由 ISO和CCITT两个组织机构联合组成的一个图像专家小组，负责制定静态的数字图像数据压缩编码标准，这个专家组开发的算法称为JPEG算法，并且成为国际上通用的标准，因此又称为JPEG标准。JPEG是一个适用范围很广的静态图像数据压缩标准，既可用于灰度图像又可用于彩色图像。JPEG不仅适于静止图像的压缩，电视图像的帧内图像的压缩编码，也常采用此算法。

◆ JPEG专家组开发了两种基本的压缩算法，一种是采用以离散余弦变换(Discrete Cosine Transform, DCT)为基础的有损压缩算法，另一种是采用以预测技术为基础的无损压缩算法。



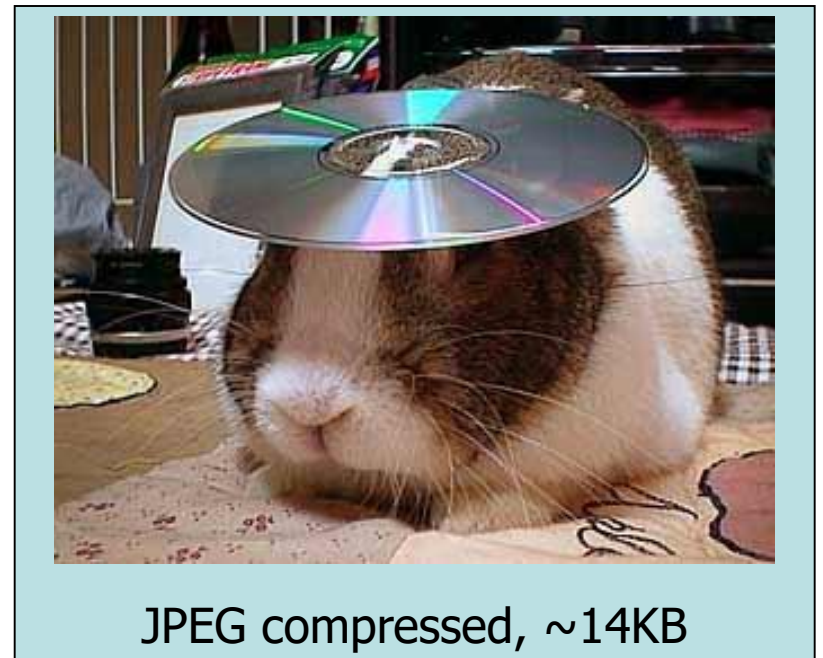
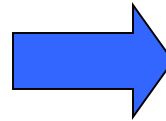


JPEG Compression: Basics

- ◆ Human vision is insensitive to high spatial frequencies
- ◆ JPEG Takes advantage of this by compressing high frequencies more coarsely and storing image as frequency data
- ◆ JPEG is a “lossy” compression scheme.



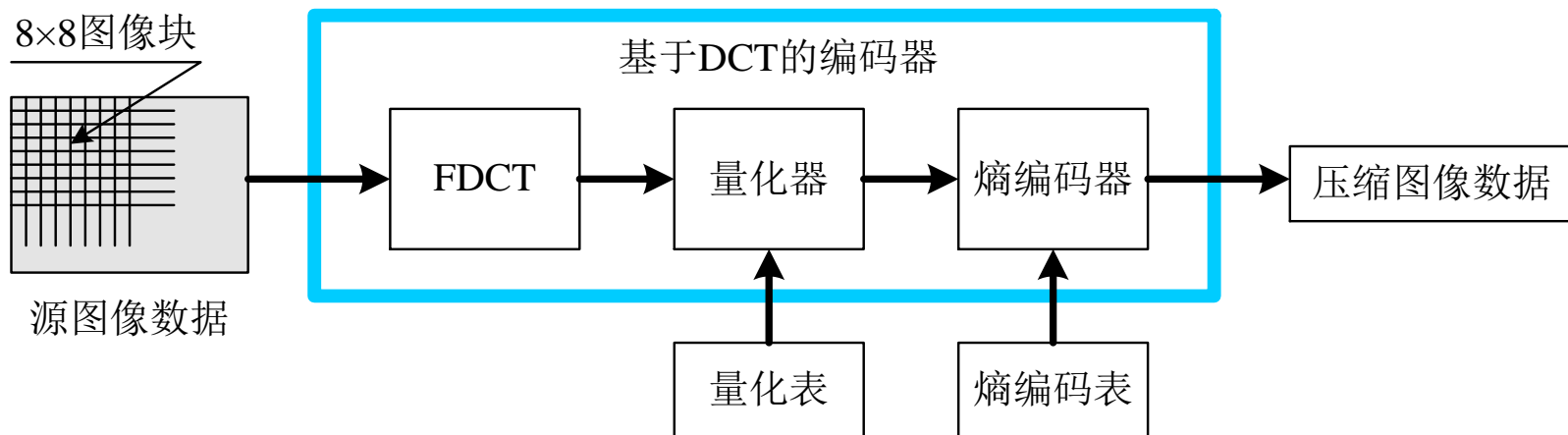
Losslessly compressed image, ~150KB



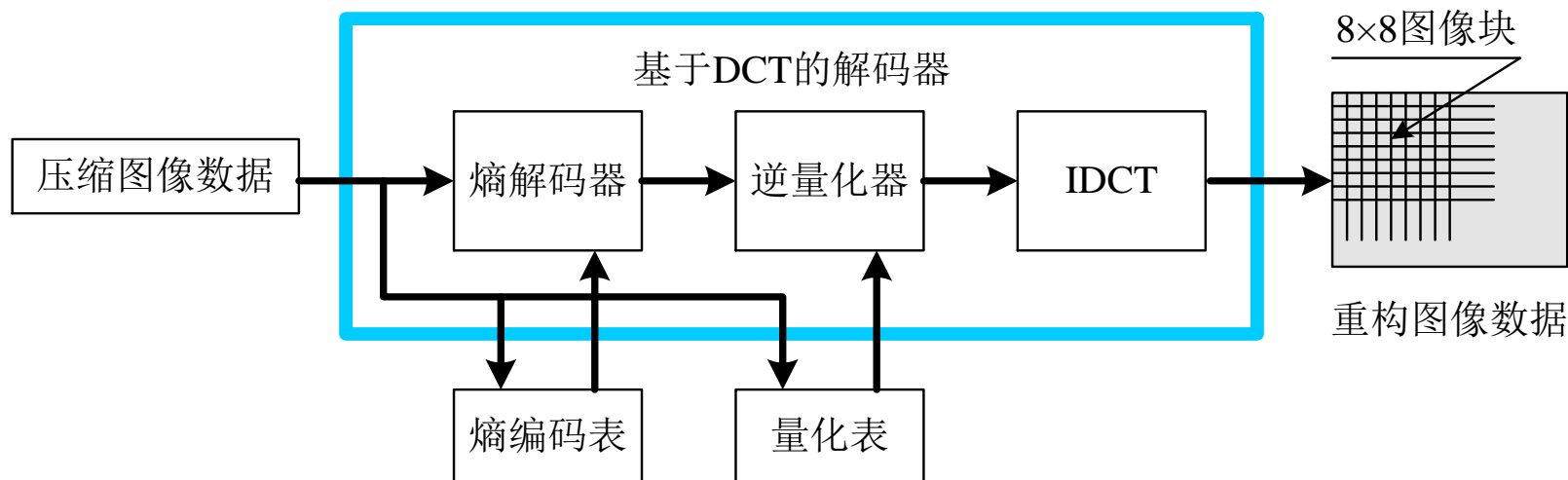
JPEG compressed, ~14KB



JPEG压缩-解压缩算法框图



(a) JPEG压缩算法框图



(b) JPEG解压缩算法框图





JPEG算法概要

- ◆ 1.正向离散余弦变换(FDCT)
- ◆ 2.量化(quantization)
- ◆ 3.Z字形编码(zigzag scan)
- ◆ 4.使用DPCM对直流系数(DC)进行编码
- ◆ 5.使用RLE对交流系数(AC)进行编码
- ◆ 6.熵编码(entropy coding)





正向离散余弦变换(FDCT)

◆对每个单独的彩色图像分量，把整个分量图像分成 8×8 的图像块，作为两维离散余弦变换DCT的输入。通过DCT变换，把能量集中在少数几个系数上。

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

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

□ $C(u), C(v) = \frac{1}{\sqrt{2}}, u, v = 0$

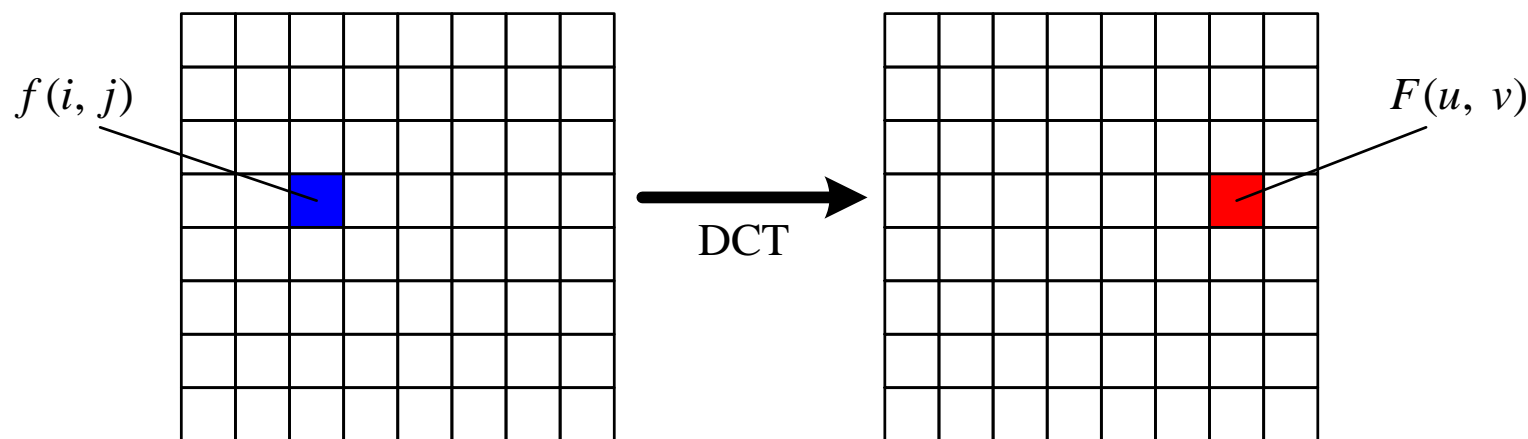
□ $C(u), C(v) = 1, \text{其他}$





DCT计算

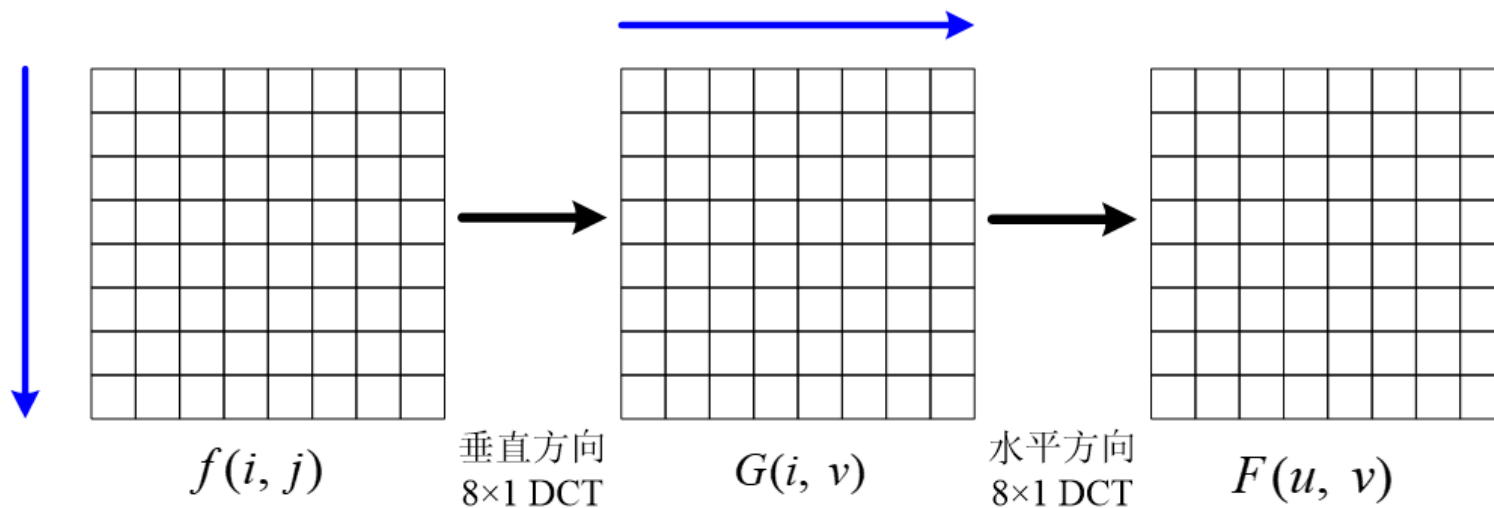
◆ 对每个单独的彩色图像分量，把整个分量图像分成 8×8 图像块，如下图所示，作为DCT的输入，通过DCT把能量集中在频率较低的少数几个系数上





DCT计算

◆ 在计算两维的DCT变换时，可把两维的DCT变换变成一维的DCT变换

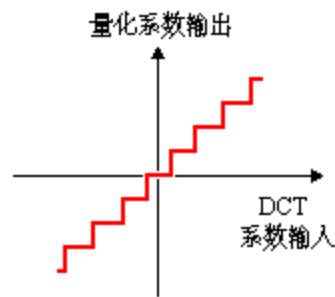


$$F(u, v) = \frac{1}{2} C(u) \left[\sum_{i=0}^7 G(i, v) \cos \frac{(2i+1)u\pi}{16} \right]$$

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



- ◆ 量化是对经过FDCT变换后的频率系数进行量化。量化的目的是减小非“0”系数的幅度以及增加“0”值系数的数目。量化是图像质量下降的最主要原因。
- ◆ 对于有损压缩算法，JPEG算法使用均匀量化器进行量化，量化步距是按照系数所在的位置和每种颜色分量的色调值来确定。因为人眼对亮度信号比对色差信号更敏感，因此使用了两种量化表。





量化表

哪一个亮度
量化表？

- ◆ 由于人眼对低频分量的图像比对高频分量的图像更敏感，因此图中的左上角的量化步距要比右下角的量化步距小。
- ◆ 表中的数值对CCIR 601标准电视图像已经是最佳的。还可以使用自己的量化表。

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

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





JPEG Example



**Original
Image**

8	5	5	8	12	20	25	30
6	6	7	9	13	29	30	27
7	6	8	12	20	28	34	28
7	8	11	14	25	43	40	31
9	11	18	28	34	54	51	38
12	17	27	32	40	52	56	46
24	32	39	43	51	60	60	50
36	46	47	49	56	50	51	49



Compression Ratio: 7.7

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



Compression Ratio: 12.3

Blocking artifact
(JPEG 2000 ?)

64	44	40	64	96	160	204	244
48	48	56	76	104	232	240	220
56	52	64	96	160	228	276	224
56	68	88	116	204	300	300	248
72	88	148	224	272	300	300	300
96	140	220	256	300	300	300	300
196	256	300	300	300	300	300	300
288	300	300	300	300	300	300	300



Compression Ratio: 33.9

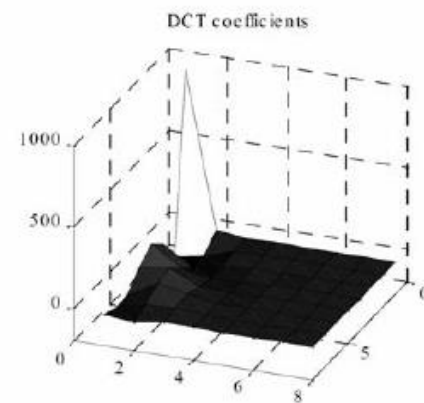
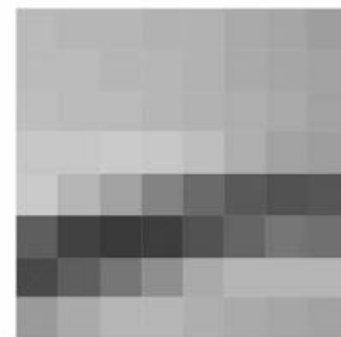
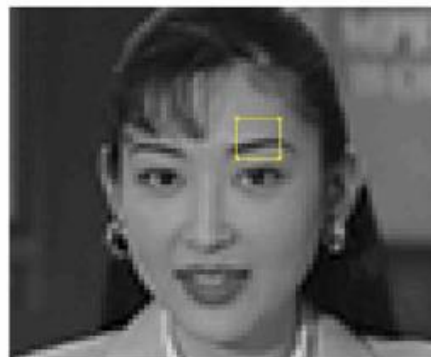
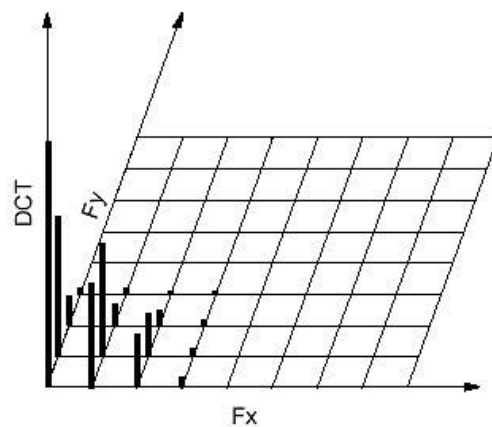
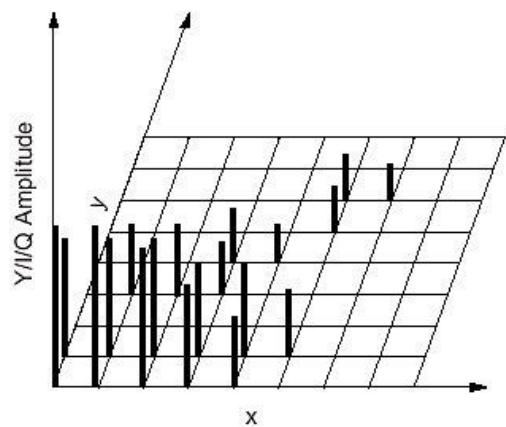
128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128



Compression Ratio: 60.1



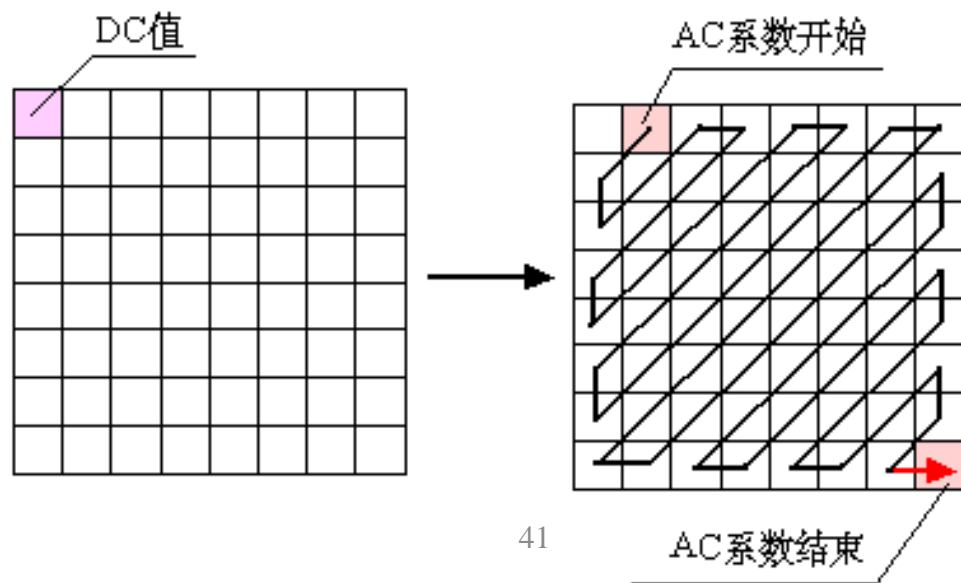
DCT结果示例





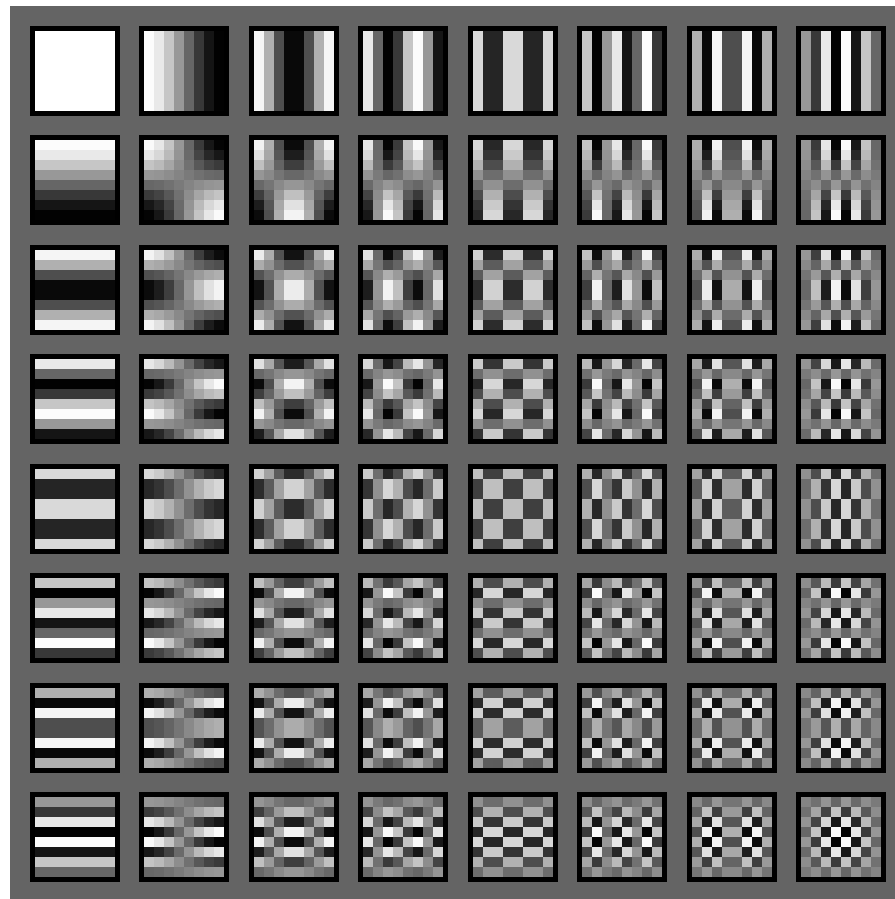
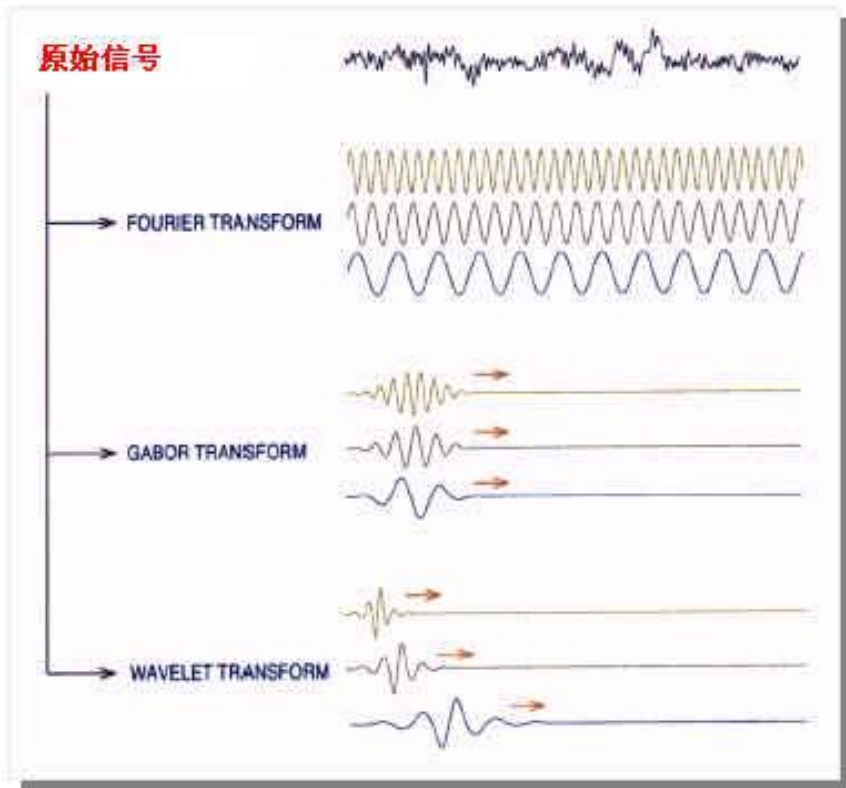
ZIGZAG

◆ 量化后的系数要重新编排，目的是为了增加连续的“0”系数的个数，就是“0”的游程长度，方法是按照Z字形的式样编排。这样就把一个 8×8 的矩阵变成一个 1×64 的矢量，频率较低的系数放在矢量的顶部。



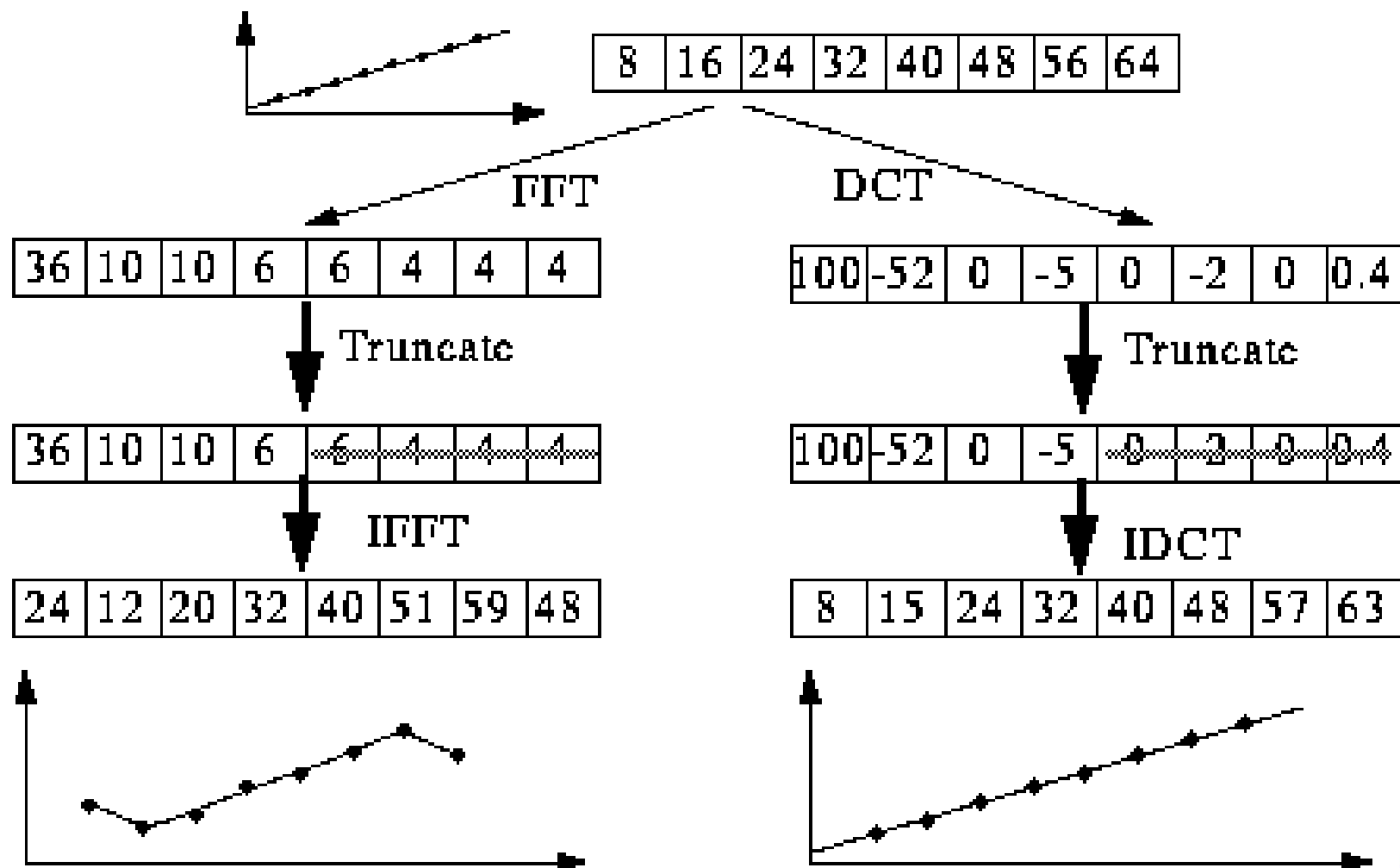


64 (8 x 8) DCT basis functions





Why DCT not FFT?

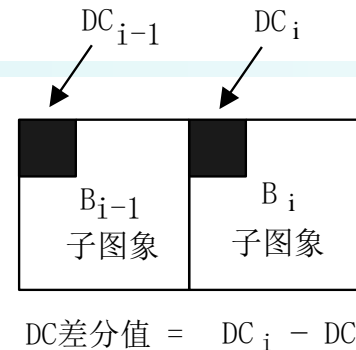


◆ 直流系数的编码: DPCM

◆ 交流系数的编码: RLE

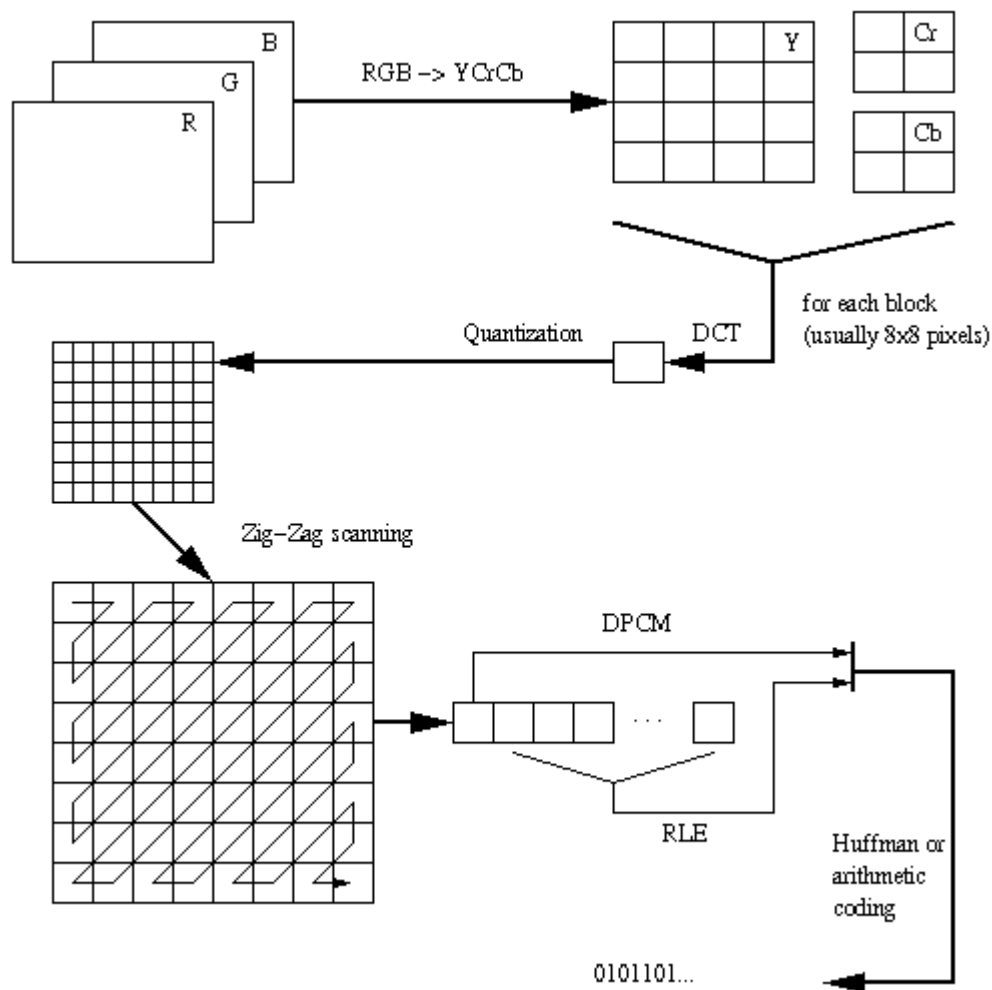
◆ 熵编码

□ 使用熵编码还可以对DPCM编码后的直流DC系数和RLE编码后的交流AC系数作进一步的压缩。在JPEG有损压缩算法中，使用霍夫曼编码器来减少熵。使用霍夫曼编码器时可以使用很简单的查表(lookup table)方法进行编码。





彩色图像 JPEG压缩流程



RGB→YUV时，**空间**采样对Y、U、V三个分量采用不同的**采样频率**。

YUV 4:2:2

YUV 4:2:0

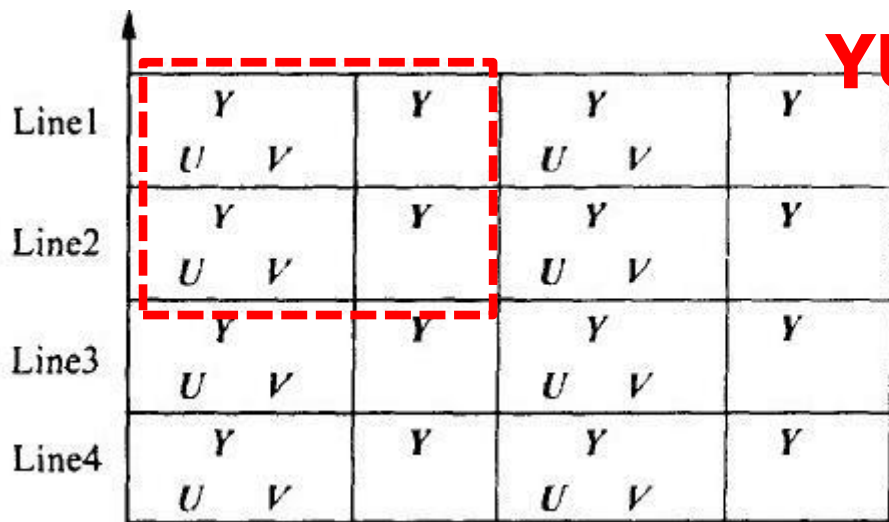
YUV 4:1:1



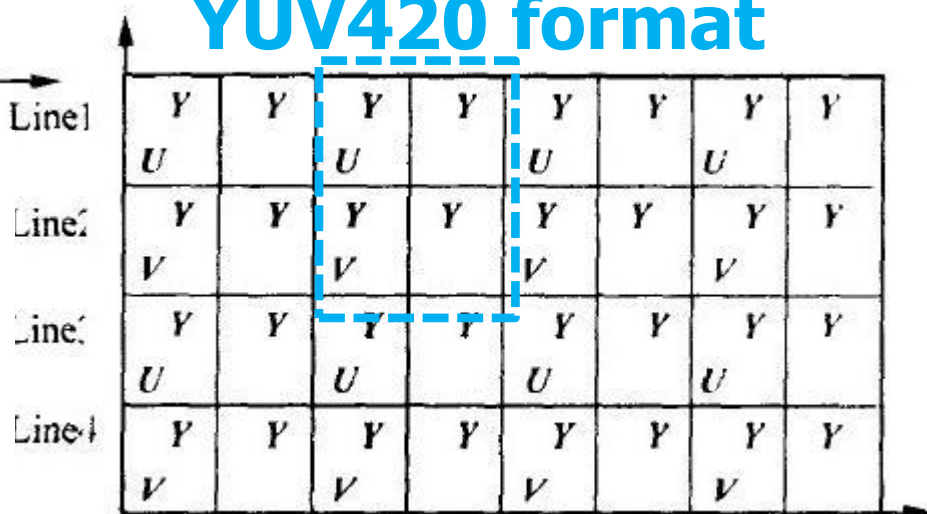


彩色图像的采样格式

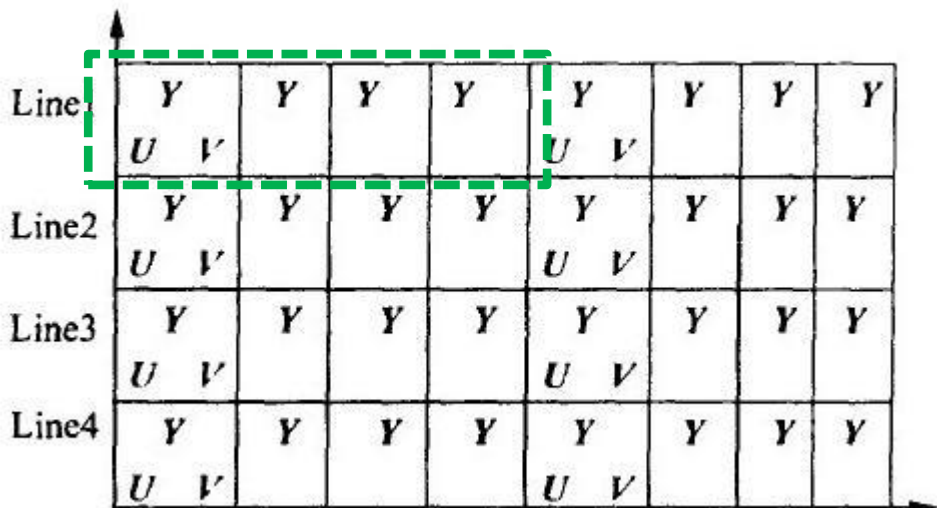
YUV422 format



YUV420 format



YUV411 format





JPEG小结

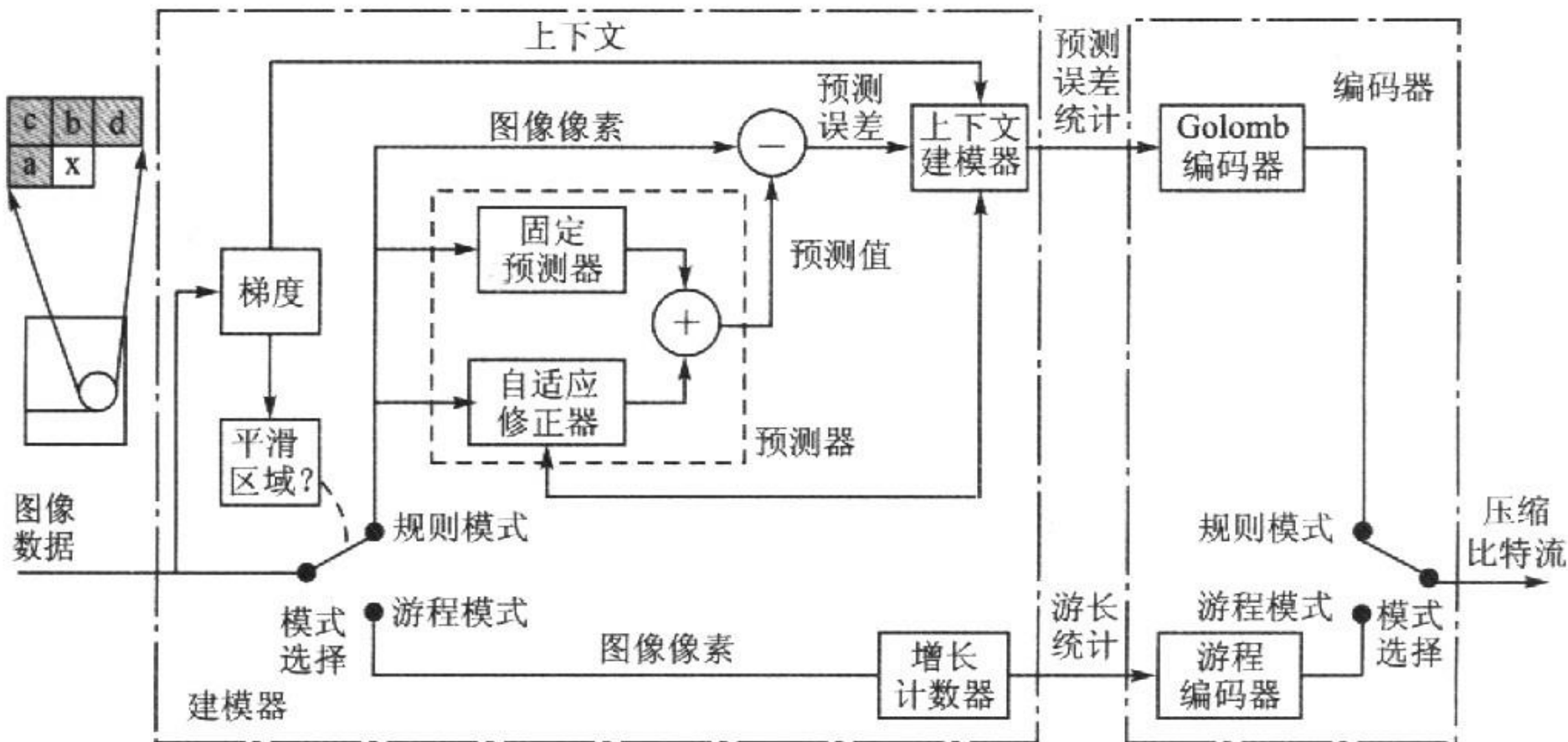
◆ JPEG标准中以DCT为基础的编码过程，实际上是在消除图象中的相关性，或者说消除图象中的冗余度，这种冗余度包括下列三种。

- 第一种是**编码冗余度**。例如，DCT变换，哈夫曼编码，是消除编码冗余度。
- 第二种是**像素间冗余度**。例如，直流系数用差分编码就是消除相邻子图象间的灰度(或亮度)冗余度。
- 第三种是**心理视觉冗余度**。例如，用量化过程，就是利用人眼对各种空间频率，包括亮度、色度、纵、横方向的高频、低频的敏感程度不同，从而降低和消除一部分数据，达到数据压缩的目的，或降低传输位率，同时又不损害心理视觉对图象的主观评价。这就是充分利用心理视觉冗余度对图象数据进行压缩。





JPEG-LS编码, 1998



- 三邻域预测结果与边缘检测结果相关
- 根据梯度计算是否为平滑区域（平滑区域熵编码效果不好）



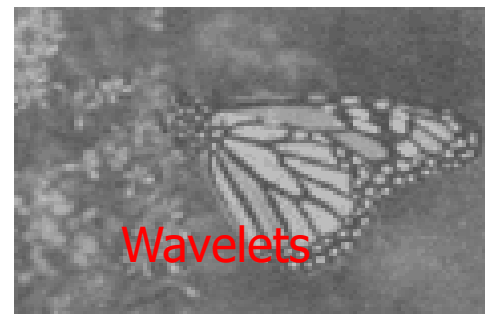
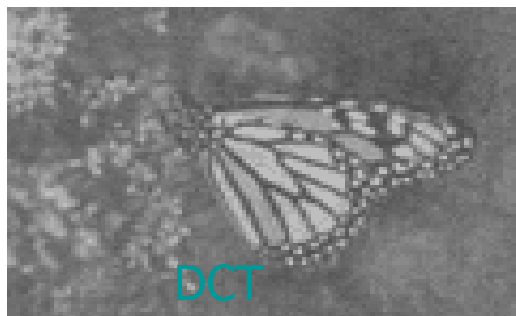


JPEG 2000

◆ JPEG 2000是**基于小波变换**的图像压缩标准。JPEG2000的推出是准备取代JPEG图像压缩标准（事实上并没有）。扩展名通常为*.jp2

◆ 目标：提高对连续色调图像的压缩效率、管理和传输性能，图像质量无明显下降

◆ 在高压缩比的情形下，JPEG 2000图像失真程度一般会比传统的JPEG图像要小。JPEG 2000同时支持有损压缩和无损压缩，也支持更复杂的渐进式显示和下载。





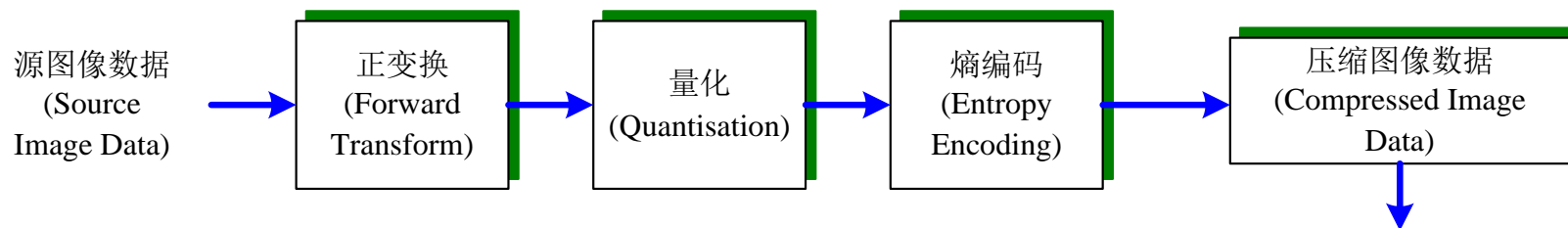
JPEG 2000的基本结构

◆ 编码器

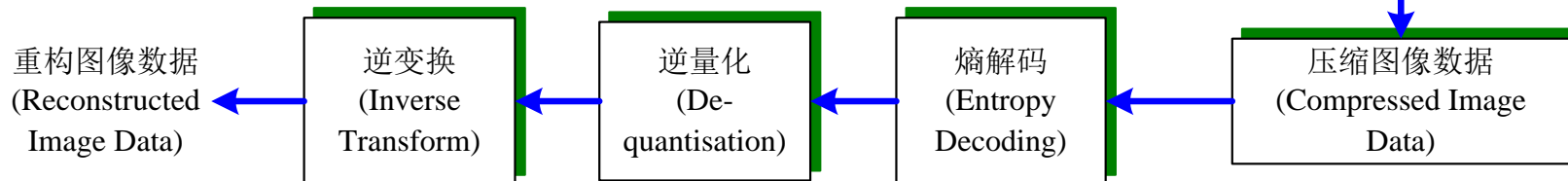
□ 源图像小波变换⇒系数量化⇒熵编码⇒代码流/位流

◆ 解码器

□ 码流熵解码⇒逆量化⇒逆向变换⇒重构图像



(a) 编码器



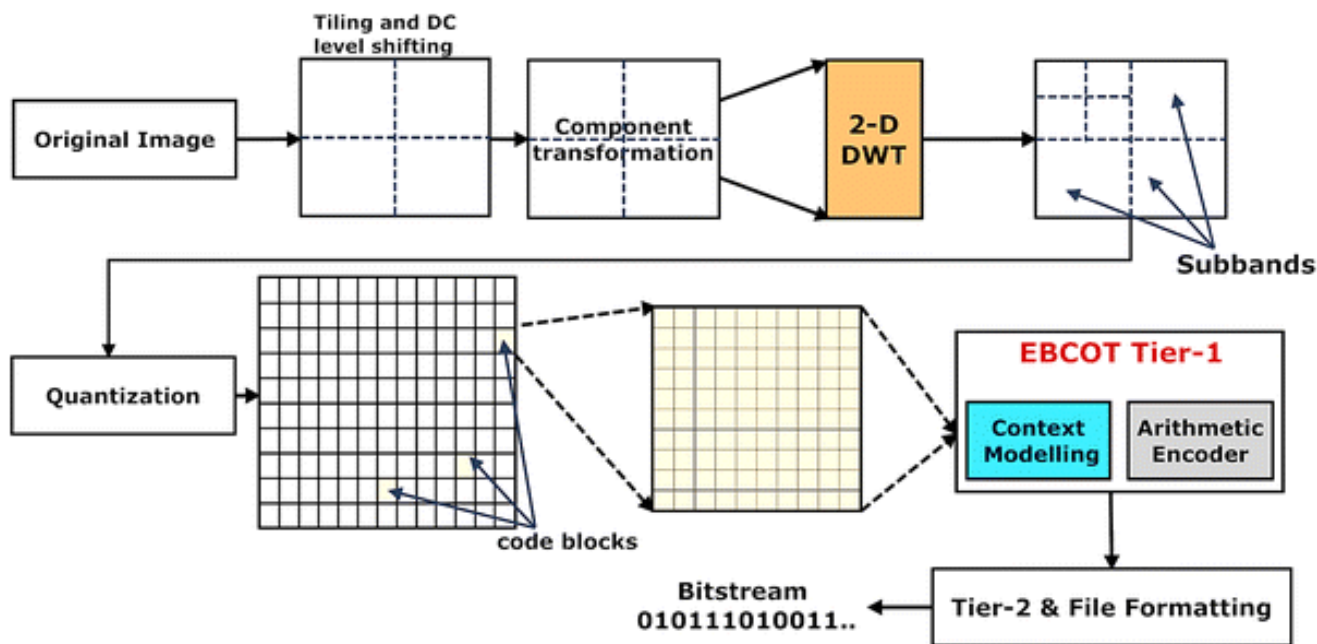
(b) 译码器





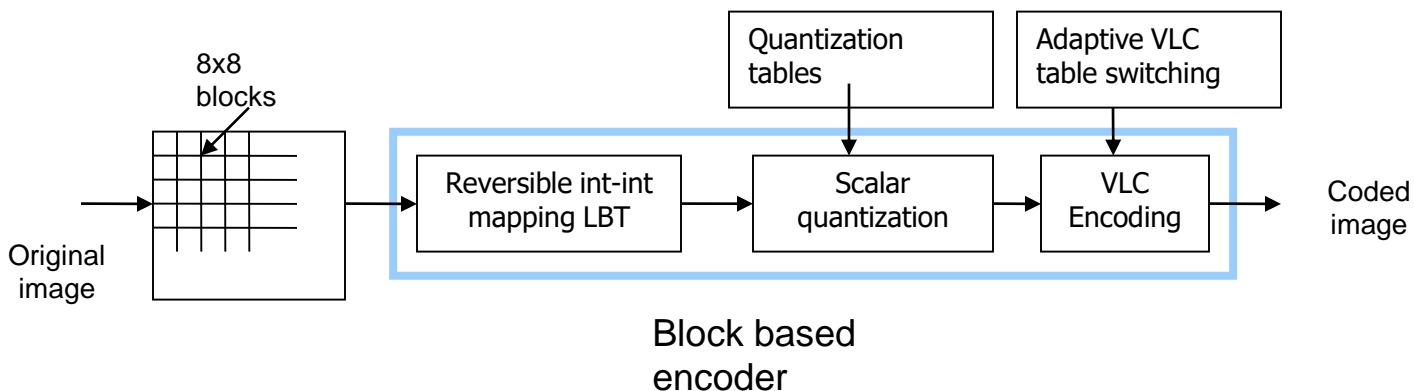
JPEG 2000的编码：EBCOT

- ◆ 以图像块为单位处理：变换、量化、编码
 - 图像块进行二维DWT变换
 - 用EBCOT算法对DWT变换产生的系数进行量化和编码
 - EBCOT: embedded block coding with optimized truncation

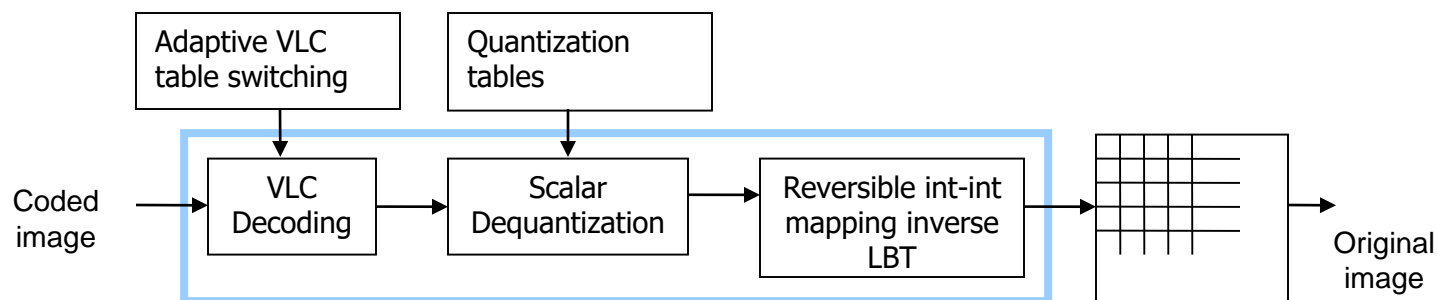




JPEG-XR, 2009



- **lapped bi-orthogonal transform (LBT)**
- **VLC Encoding**





JPEG1992之后

- JPEG XL专注于可伸缩的Web分发和有效压缩高质量图像。
- JPEG XS目标是流式传输，应VR虚拟实境而设计，压缩编码更简单、还原速度快
- JPEG XT: HDR (High Dynamic Range)、WCG (Wide Color Gamut)





JPEG1992之后：标准已定稿

◆ JPEG XT

- ISO/IEC 18477-1:2015 → ISO/IEC 18477-1:2020
- JPEG XT (ISO/IEC 18477) specifies a series of backwards compatible **extensions** to the legacy JPEG standard (ITU Recommendation T.81 | ISO/IEC 10918-1).
 - HDR (High Dynamic Range)
 - WCG (Wide Color Gamut)
 - ...

◆ JPEG XS

- ISO/IEC 21122-1:2019
- The JPEG XS (ISO/IEC 21122) standard specifies a compression technology with an **end-to-end latency** of a few lines, at a **low implementation complexity**.





JPEG1992之后：标准未定稿

◆ JPEG AI

- The scope of JPEG AI is the creation of a learning-based image coding standard.

◆ JPEG XL

- JPEG XL is designed to meet the needs of image delivery on the web and professional photography.

◆ JPEG Pleno

- JPEG Pleno aims to provide a standard framework for representing new imaging modalities, such as texture-plus-depth, light field, point cloud, and holographic imaging.

◆ JPEG AIC

- The purpose of the Advanced Image Coding and Evaluations work is to locate and evaluate new scientific developments and advancements in image coding research.





第三章 多媒体数据压缩

- ◆ § 3.1 无损数据压缩
- ◆ § 3.2 音频数据的压缩标准
- ◆ § 3.3 图像数据的压缩标准
 - § 3.3.1 图像的冗余
 - § 3.3.2 变换编码
 - § 3.3.3 JPEG原理
 - § 3.3.4 JPEG示例
- ◆ § 3.4 视频数据的压缩标准





JPEG实例

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

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





位长表

位长 (size)	DC 差分值 (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	-1023..-512, 512..1023
11	-2047..-1024, 1024..2047

DC差分值的位长表

位长 (size)	AC 差分值 (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	-1023..-512, 512..1023

AC系数的位长表





霍夫曼编码实现-符号序列

(15 0 -2 -1 -1 -1 0 0 -1 EOB)

假设上个block DC=12

(2)(3), (1,2)(-2), (0,1)(-1), (0,1)(-1), (0,1)(-1), (2,1)(-1), (0,0)

◆ DC coefficient.

◆ skip the number of

◆ Value the

◆ AC coefficient

◆ skip

(1,2)(-2) (0,1)(-1)

1: 1个0 0: 0个0

2: 由AC 1: 由AC系数的位长

3: (2,1)(-1)

对应的位长

值-1对应的位长

(2,3)

2: 由DC系数的位长表查得值3对应的位长

3: 需要编码的数值



码字表

位长 (Size)	位数 (Code Length)	码字 (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

亮度DC系数表

位长 (Size)	位数 (Code Length)	码字 (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

色度DC系数表

位长 (Size)	振幅 (Amplitude)	码字 (Code Word) (binary)
0	0	-
1	-1, 1	0, 1
2	-3, -2, 2, 3	00, 01, 10, 11
3	-7, ..., -4, ..., 7	000, ..., 011, 100, ..., 111
4	-15, ..., -8, 8, ..., 15	0000, ..., 0111, 1000, ..., 1111
...
16	32768	

正负值表

行程/位长 (Run/Size)	位数 (Code Length)	码字 (Code Word)	行程/位长 (Run/Size)	位数 (Code Length)	码字 (Code Word)
0/0 (ZOB)	4	1010	2/1	5	11100
0/1	2	00	2/2	8	11111001
0/2	2	01	2/3	10	1111110111
0/3	3	100	2/4	12	11111110100
0/4	4	1011	2/5	16	111111110001001
0/5	5	11010	2/6	16	1111111110001010
0/6	7	1111000	2/7	16	1111111110001011
0/7	8	11111000	2/8	16	1111111110001100
0/8	10	1111110110	2/9	16	1111111110001101
0/9	16	1111111110000010	2/A	16	1111111110001110
0/A	16	1111111110000011	3/1	6	111010
1/1	4	1100	3/2	9	11110111
1/2	5	11011	3/3	12	11111110101
1/3	7	1111001	3/4	16	111111110001111
1/4	9	111110110	3/5	16	1111111110010000
1/5	11	11111110110	3/6	16	1111111110010001
1/6	16	11111110000100	3/7	16	1111111110010010
1/7	16	11111110000101	3/8	16	1111111110010011
1/8	16	11111110000110	3/9	16	1111111110010100
1/9	16	11111110000111	3/A	16	1111111110010101
1/A	16	11111110001000	4/1	6	111011

亮度AC系数表

(2)(3) -> 011 11

(1,2)(-2)->11011 01

(0,1)(-1)->00 0

(0,1)(-1)->00 0

(0,1)(-1)->00 0

(2,1)(-1)->11100 0

(0,0)->1010



霍夫曼编码实现过程

◆ Zig-zag sequence

□ (15 0 -2 -1 -1 0 0 -1 EOB)

◆ Intermediate symbol sequence

□ (2)(3), (1,2)(-2), (0,1)(-1), (0,1)(-1), (0,1)(-1), (2,1)(-1), (0,0)

◆ Encoded bit sequence (total 26 bits)

□ (01111) (1101101) (000) (000) (000) (111000) (1010)





媒体信息表示的两条线索

◆ 媒体信息的本质

□ 时域

- 针对声音的波形的采样量化
- 针对亮度和色度信号的采样量化

□ 频域

- 子带编码/MP3中的MDCT
- JPEG中变换编码（DCT）的使用
- JPEG2000中的Wavelets变换

◆ 媒体信息的知识

□ MIDI音乐/矢量图

有没有新的思路？

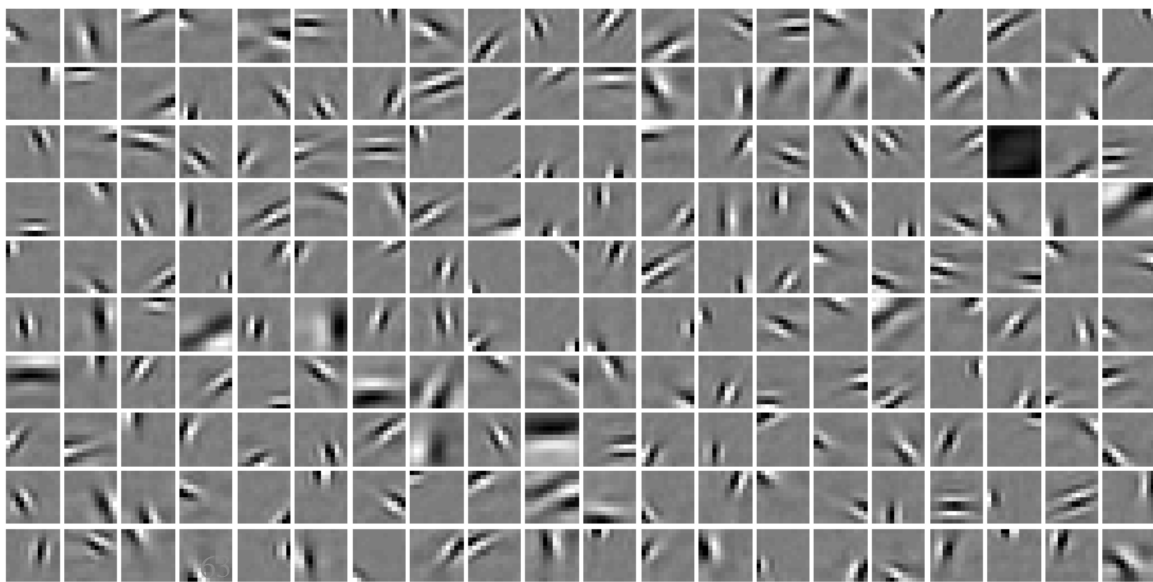


基于稀疏模型的图像压缩

稀疏编码模型

Olshausen 1996 年基于对人类神经系统的实验结果提出稀疏编码模型：The receptive fields of simple cells in mammalian primary visual cortex can be characterized as being spatial localized, oriented and bandpass(selective to structure at different spatial scales), comparable to the basis functions of wavelet transforms. 稀疏编码模型的目标是找到一组基，用于对信号进行稀疏表示。

Basis functions learned from static natural images. Shown is a set of 200 basis functions which were adapted to 12*12 pixel image patches





基于稀疏模型的图像压缩 稀疏表示

◆ 信号表示，就是将原始信号在某一已知函数或者矢量集上来进行表示。

◆ 稀疏表示，尽量少的使用基函数对原始信号进行表示的过程。

- 如何确定基函数（即字典的获取）
- 如何进行稀疏表示（即稀疏分解）

◆ 传统图像表示的方法

- 傅里叶表示、离散余弦变换、小波变换等

◆ 图像的稀疏表示

- 过完备字典的稀疏表示，能够以更简洁更稀疏的方式来表示原有图像
 - 1993年，由S.Mallat和Z.Zhang首次提出字典的概念并用其代替传统小波变换。
 - 1995年和2001年，S.Chen和D.Donoho等人进一步探讨了稀疏解的求解方法，并就稀疏解的唯一性、移不变特性、收敛性等问题给出了理论分析。





基于稀疏模型的图像压缩

字典如何得到？分析字典和学习字典

◆信号稀疏表示的两大主要任务就是字典的生成和信号的稀疏分解，对于字典的选择，一般有分析字典和学习字典两大类。

◆常用的分析字典有小波字典、过完备DCT字典和Curvelets等，用分析字典进行信号的稀疏表示时，简单易实现，但信号的表达形式单一且不具备自适应性。

◆而学习字典的自适应能力强，能够更好的适应不同的图像数据，在目前的研究中，常用的学习字典的方法包括：Engan于1999年提出的最优方向算法（method of directions, MOD），该算法是学习字典的鼻祖，它的字典更新方式简单，但与此同时它的收敛速度很慢，在该算法的基础上，一些研究人员同时还提出了一些其他的字典学习算法，如FOCUSS字典学习算法，广义PCA算法等等，Michael Elad也于2006年提出了基于超完备字典稀疏分解的K-SVD算法，该算法相较于之前算法，收敛速度有了很大的提高，但是随着噪声的逐渐增大，使用该算法进行去噪后的图像因纹理细节的丢失会产生模糊的效果。



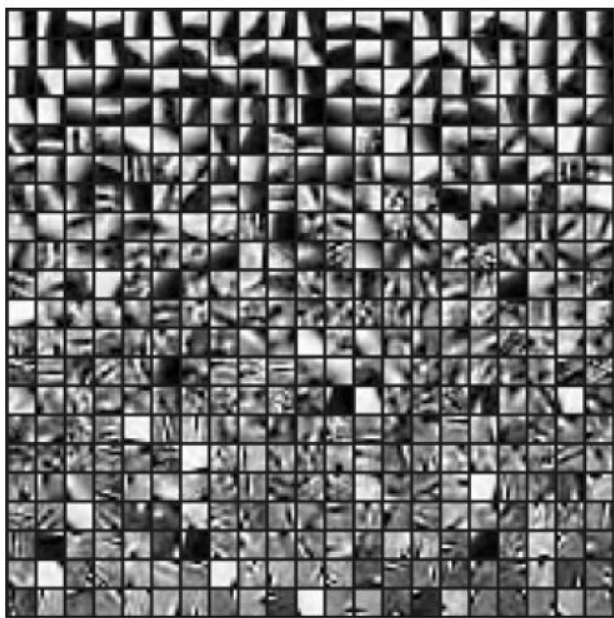


基于稀疏模型的图像压缩

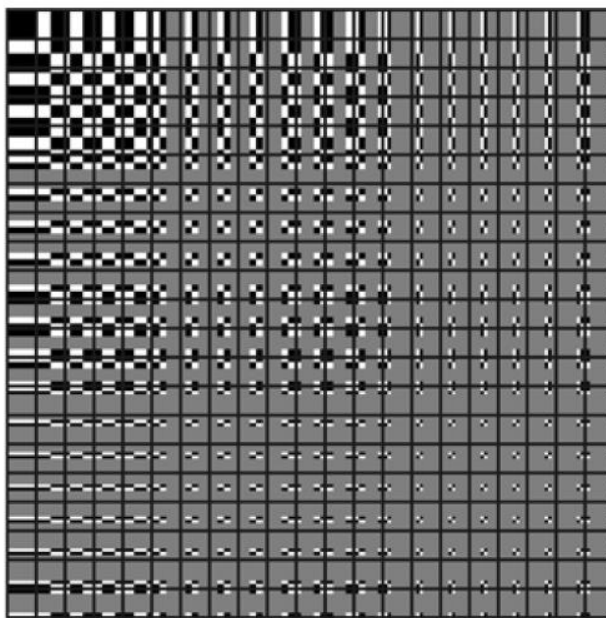
字典如何得到？分析字典或学习字典

K-SVD的字典是学习字典。K-SVD 算法通过迭代过程实现**原有过完备字典**在图像样本下的训练，通过稀疏分解系数不断调整原子库中的原子，获得更能有效反映图像特征的过完备原子库。

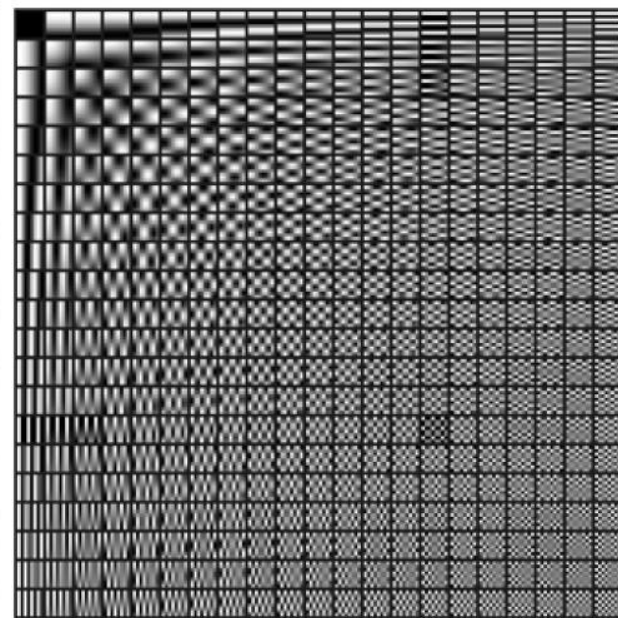
The **learned** dictionary



The overcomplete separable **Haar** dictionary



The overcomplete **DCT** dictionary

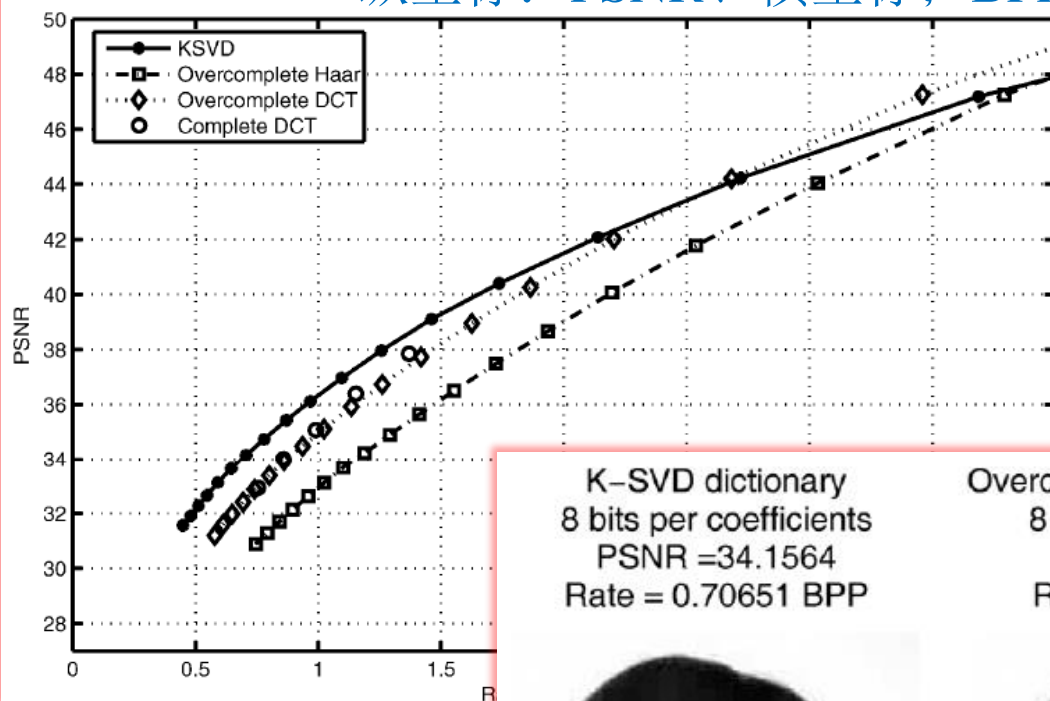




基于稀疏模型的图像压缩

基于学习字典的图像压缩低码率时质量更好

纵坐标: PSNR、横坐标: BPP



K-SVD dictionary
8 bits per coefficients
PSNR = 34.1564
Rate = 0.70651 BPP

Overcomplete DCT dictionary
8 bits per coefficients
PSNR = 32.4021
Rate = 0.69419 BPP

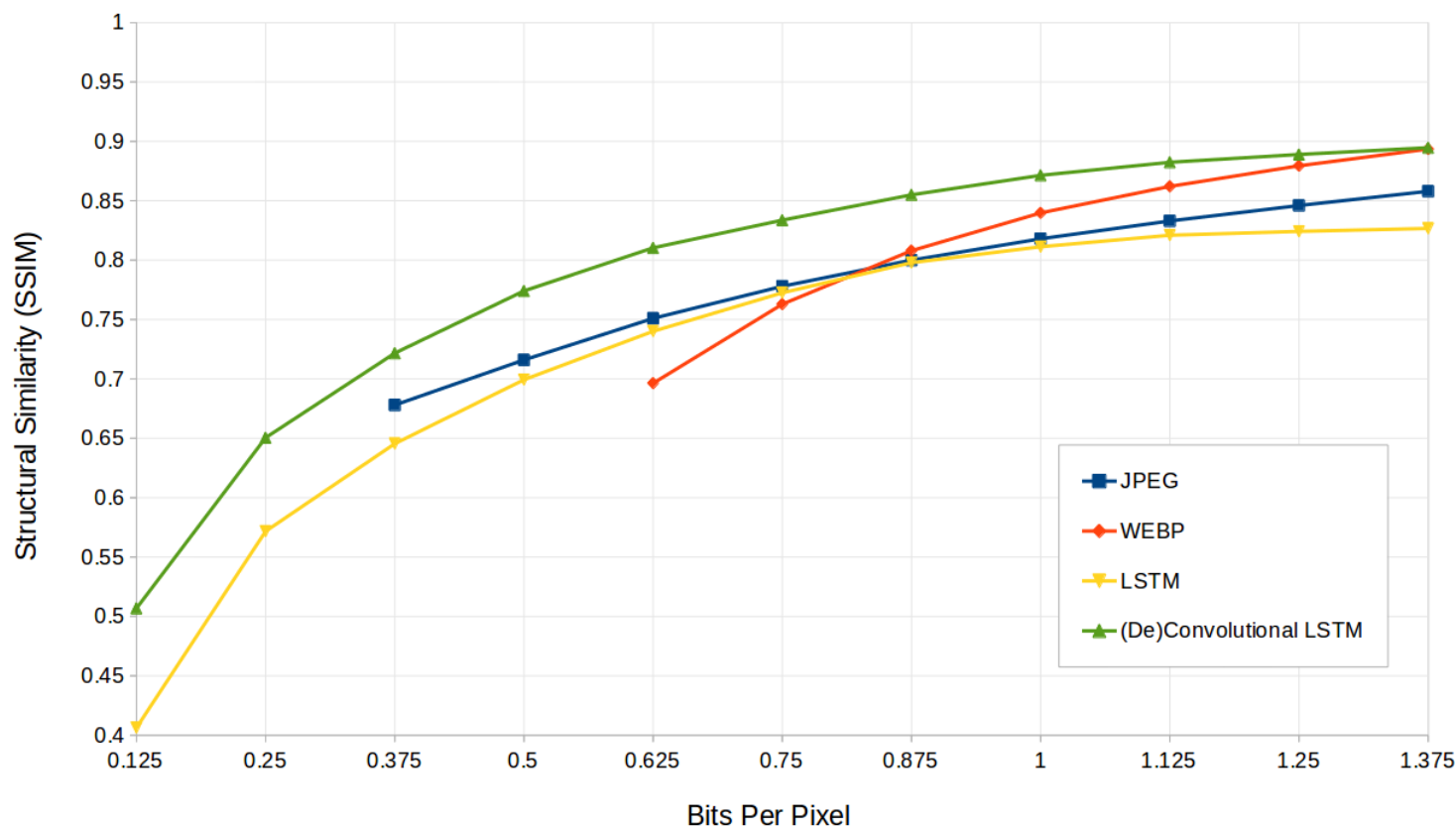
Complete DCT dictionary
8 bits per coefficients
PSNR = 32.3917
Rate = 0.70302 BPP





基于神经网络的图像压缩

WebP: Google在2010年提出的一种图像压缩方法
LSTM(Long Short-Term Memory): 一种循环神经网络





小结

- ◆ 矢量量化/词典编码/伪彩色
- ◆ 空间/时间/视觉/知识/结构/熵冗余
- ◆ DFT/DCT/DST/MDCT/STFT/Wavelet
- ◆ BMP/GIF/PNG/JPEG/JP2
- ◆ Sampling/DCT/ quantization
/DPCM/ZigZag/RLE/Huffman



多媒体技术基础

第三章 多媒体数据压缩

§ 3.3 图像数据的压缩标准

谢谢！

