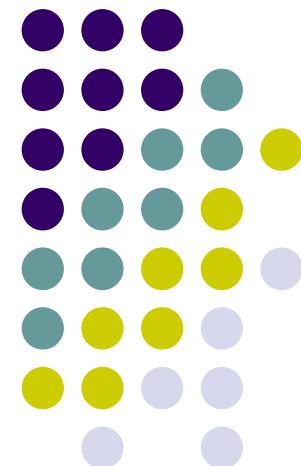


数字图像处理

第七讲
图像压缩





提纲

- 引言
- 基础知识
 - 图像冗余
 - 图像信息的度量
 - 保真度准则
 - 图像压缩模型
- 基本的压缩方法
 - 霍夫曼编码
 - 行程编码
 - 基于符号的编码
- 数字图像水印





引言

- 图像压缩

- 减少表示一张图像所需的数据量
- 数字图像处理领域商业上最成功的的技术之一
- 应用于生活中的方方面面
 - 数码相机、浏览互联网、观看视频

- 2小时标准画质的电视电影

$$30 \frac{\text{frames}}{\text{sec}} \times (720 \times 480) \frac{\text{pixels}}{\text{frame}} \times 3 \frac{\text{bytes}}{\text{pixel}} = 31,104,000 \text{ bytes/sec}$$

$$31,104,000 \frac{\text{bytes}}{\text{sec}} \times (60^2) \frac{\text{sec}}{\text{hr}} \times 2 \text{ hrs} \cong 2.24 \times 10^{11} \text{ bytes}$$

224G !





引言

- 互联网传输
 - 传输速率：电话线56Kbps，宽带12Mbps
 - 传输 128×128 彩色图片，需要7秒到0.03秒
 - 压缩可以缩短时间10倍以上
- 数码相机
 - 800万像素，一幅图像需要24M空间
- 视频会议
- 遥感、传真
- 文本和医学图像处理





提纲

- 引言
- 基础知识
 - 图像冗余
 - 图像信息的度量
 - 保真度准则
 - 图像压缩模型
- 基本的压缩方法
 - 霍夫曼编码
 - 行程编码
 - 基于符号的编码
- 数字图像水印





基础知识

- 数据压缩
 - 减少表示给定信息所需的数据量
 - 数据是传输信息所用的手段
- 冗余数据
 - 包含不相关或重复信息的表示
- b 和 b' 为两种不同表示方式的比特数
- 压缩比 $C = \frac{b}{b'}$
- 用 b 比特表示的相对数据冗余 $R = 1 - \frac{1}{C}$
 - $C = 10$ 意味着有90%的冗余





提纲

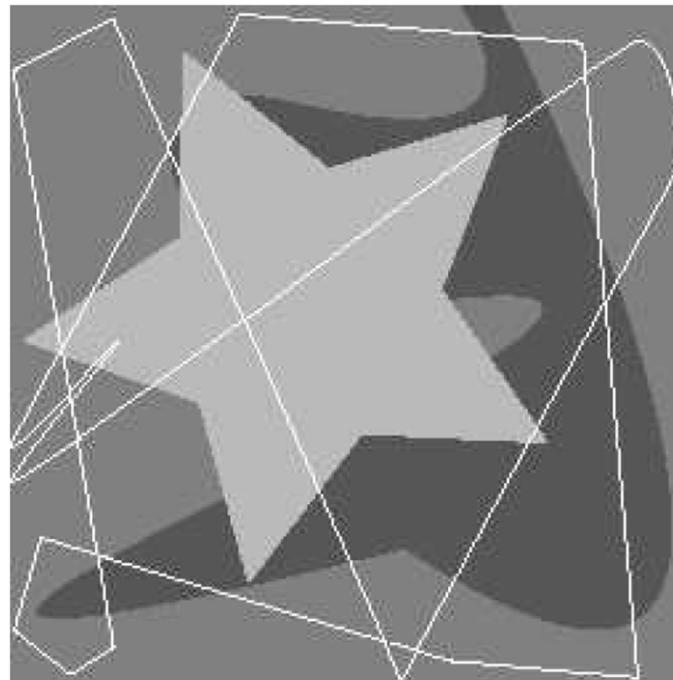
- 引言
- 基础知识
 - 图像冗余
 - 图像信息的度量
 - 保真度准则
 - 图像压缩模型
- 基本的压缩方法
 - 霍夫曼编码
 - 行程编码
 - 基于符号的编码
- 数字图像水印





图像冗余

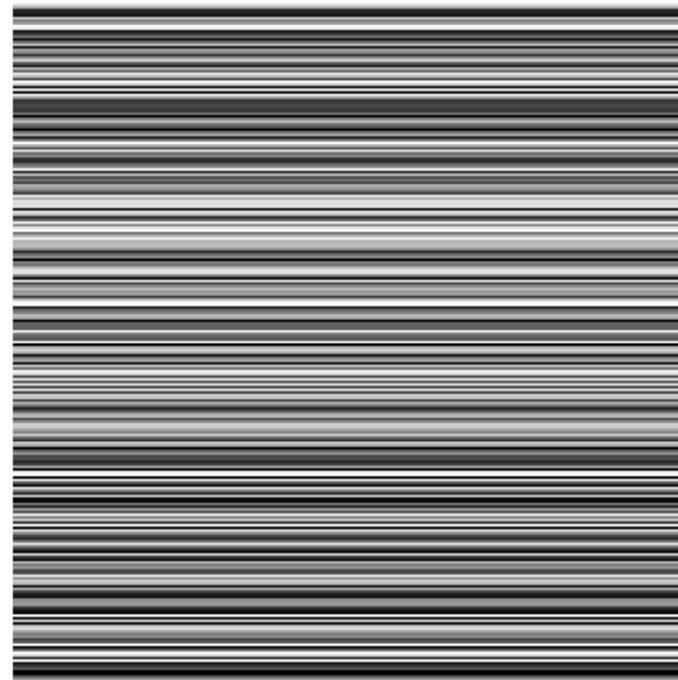
- 数字图像压缩
 - b 是以2维矩阵表示一幅图像所需的比特数
- 2维灰度矩阵包含三种冗余
 1. 编码冗余
 - 编码：表示信息的符号系统（字母、比特）
 - 码字：符号序列
 - 灰度图像的8位编码往往是冗余的





图像冗余

- 数字图像压缩
 - b 是以2维矩阵表示一幅图像所需的比特数
- 2维灰度矩阵包含三种冗余
 2. 空间和时间冗余
 - 图像中紧邻点是空间相关的
 - 视频中连续帧是时间相关的





图像冗余

- 数字图像压缩
 - b 是以2维矩阵表示一幅图像所需的比特数
- 2维灰度矩阵包含三种冗余
 - 3. 不相关的信息
 - 被视觉系统忽略的信息
 - 与图像用途无关的信息





1. 编码冗余

- 令 r_k 表示 $M \times N$ 大小图像的灰度数值
- r_k 为属于 $[0, L - 1]$ 的离散随机变量
 - 概率 $p_r(r_k) = \frac{n_k}{MN} \quad k = 0, 1, 2, \dots, L - 1$
 - n_k 为第 k 个灰度值在图像中出现的次数
- 表达 r_k 所需要的比特数记为 $l(r_k)$
- 平均比特数 $L_{\text{avg}} = \sum_{k=0}^{L-1} l(r_k)p_r(r_k)$
- 固定比特数

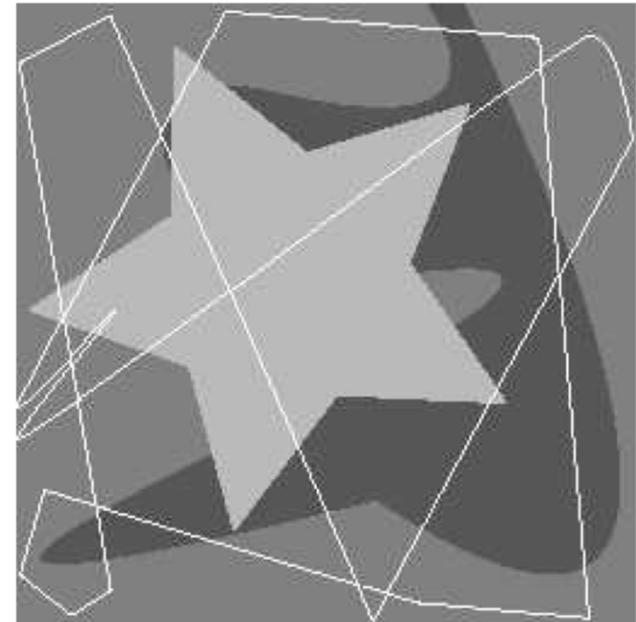
$$l(r_k) = m, \quad L_{\text{avg}} = m$$



举例

- 采用8位固定编码

- $L_{\text{avg}} = 8$



r_k	$p_r(r_k)$	Code 1	$I_I(r_k)$
$r_{87} = 87$	0.25	01010111	8
$r_{128} = 128$	0.47	10000000	8
$r_{186} = 186$	0.25	11000100	8
$r_{255} = 255$	0.03	11111111	8
r_k for $k \neq 87, 128, 186, 255$	0	—	8



举例

- 采用8位固定编码

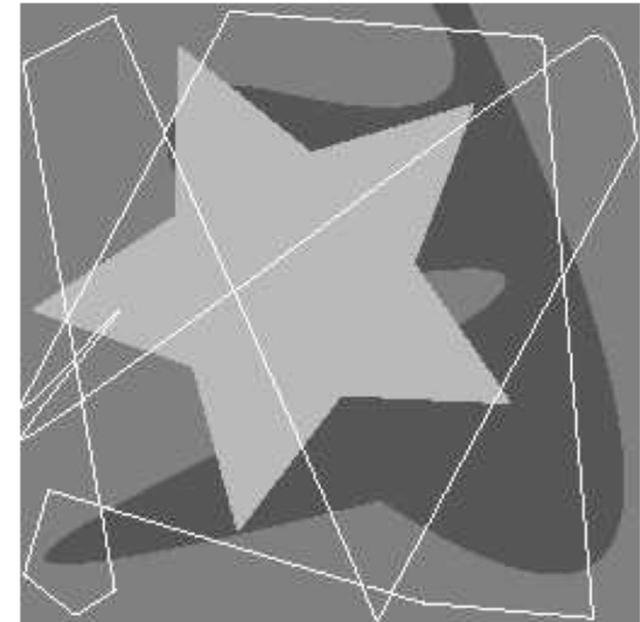
- $L_{\text{avg}} = 8$

概率大的灰度
用较少的比特

- 采用变长编码

$$L_{\text{avg}} = 0.25(2) + 0.47(1) + 0.25(3) + 0.03(3) = 1.81 \text{ bits}$$

- 总比特数 $MNL_{\text{avg}} = 256 \times 256 \times 1.81 = 118621$



r_k	$p_r(r_k)$	Code 1	$I_I(r_k)$	Code 2	$I_2(r_k)$
$r_{87} = 87$	0.25	01010111	8	01	2
$r_{128} = 128$	0.47	10000000	8	1	1
$r_{186} = 186$	0.25	11000100	8	000	3
$r_{255} = 255$	0.03	11111111	8	001	3
r_k for $k \neq 87, 128, 186, 255$	0	—	8	—	0



举例

- 采用8位固定编码

- $L_{\text{avg}} = 8$

- 采用变长编码

$$L_{\text{avg}} = 0.25(2) + 0.47(1) + 0.25(3) + 0.03(3) = 1.81 \text{ bits}$$

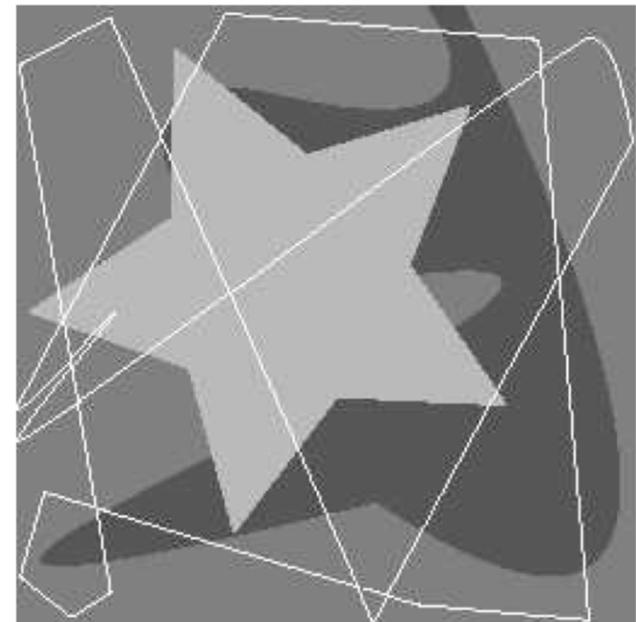
- 总比特数 $MNL_{\text{avg}} = 256 \times 256 \times 1.81 = 118621$

- 压缩比

$$C = \frac{256 \times 256 \times 8}{118,621} = \frac{8}{1.81} \approx 4.42$$

- 相对数据冗余

$$R = 1 - \frac{1}{4.42} = 0.774$$



举例

- 采用8位固定编码

- $L_{\text{avg}} = 8$

- 采用变长编码

$$L_{\text{avg}} = 0.25(2) + 0.47(1) + 0.25(3) + 0.03(3) = 1.81 \text{ bits}$$

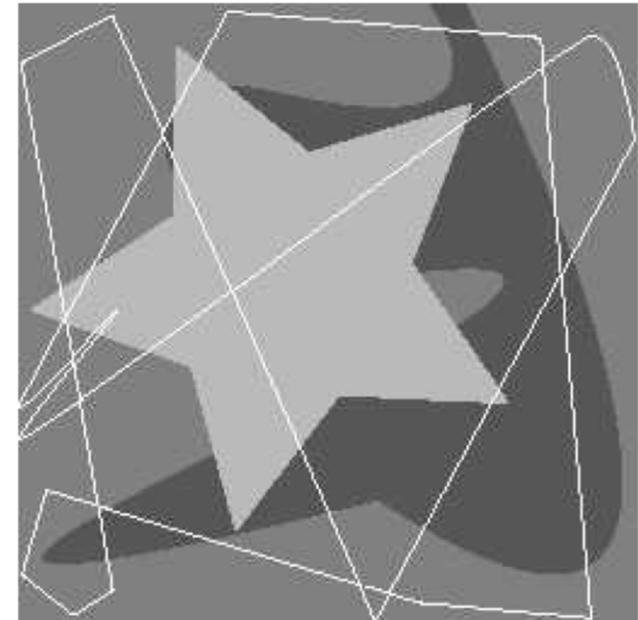
- 总比特数 $MNL_{\text{avg}} = 256 \times 256 \times 1.81 = 118621$

- 最优的固定长度编码

- $L_{\text{avg}} = 2 > 1.81$

- 采用固定长度编码普遍存在冗余

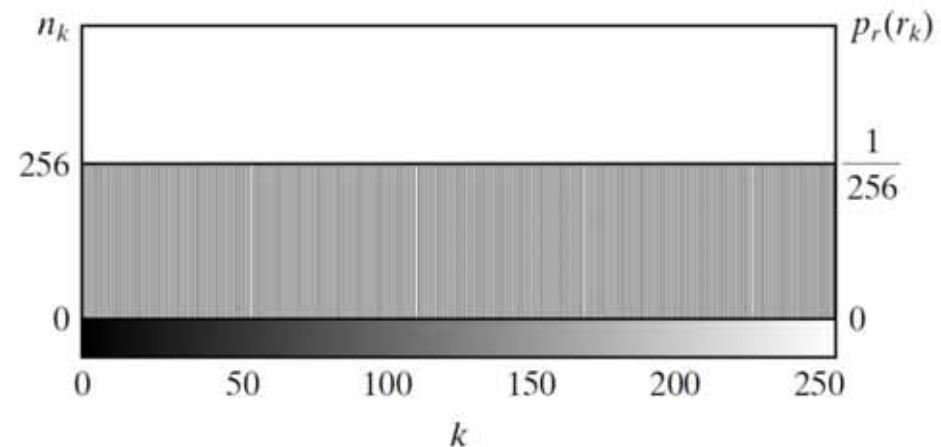
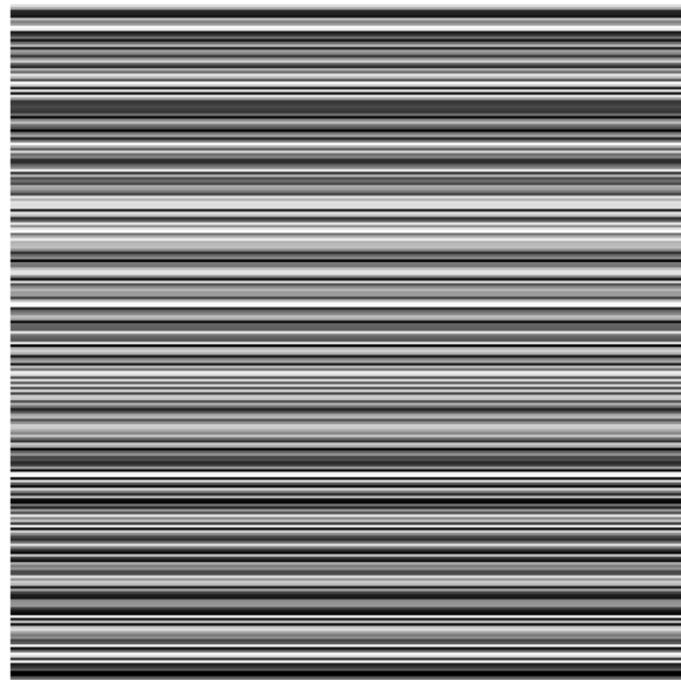
- 灰度直方图不是均匀分布





2. 空间和时间冗余

1. 所有灰度值出现的概率相同
 - 没有编码冗余



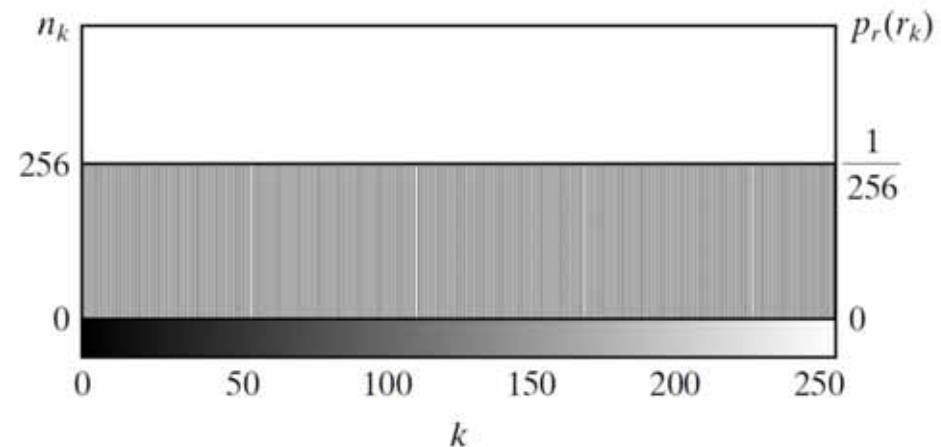
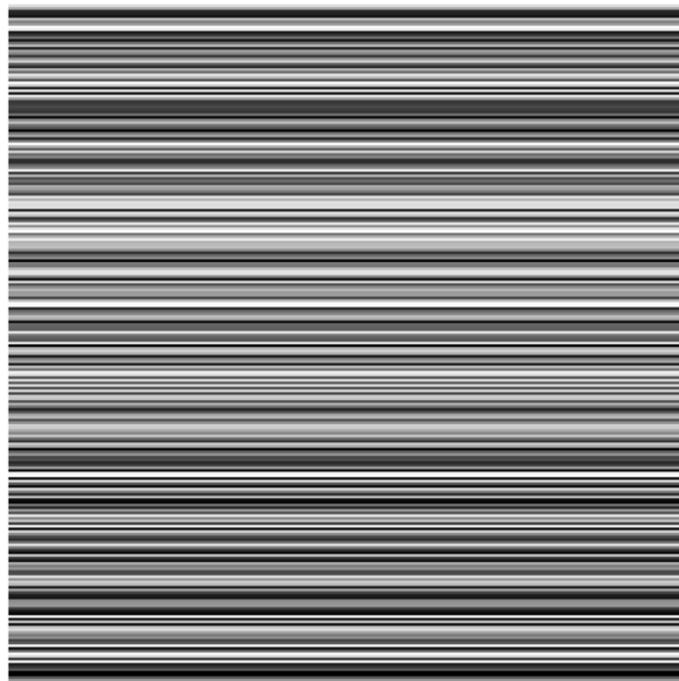
灰度直方图





2. 空间和时间冗余

1. 所有灰度值出现的概率相同
2. 垂直方向的灰度没有任何关联
3. 水平方向的灰度值完全一样



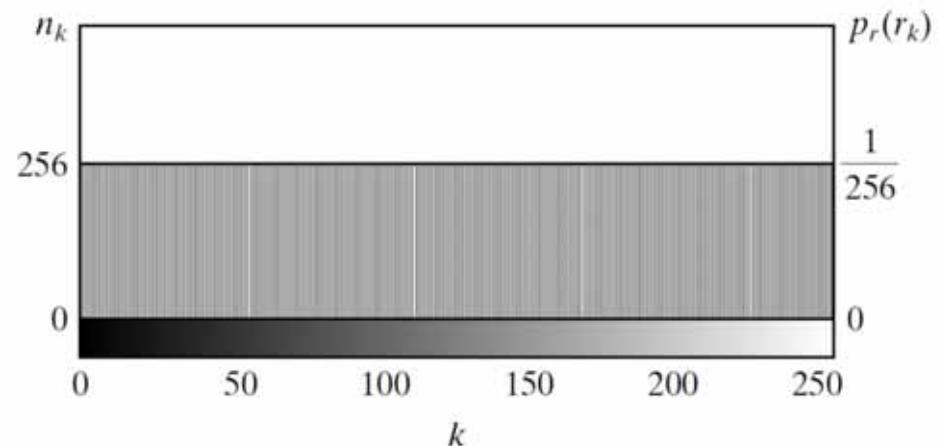
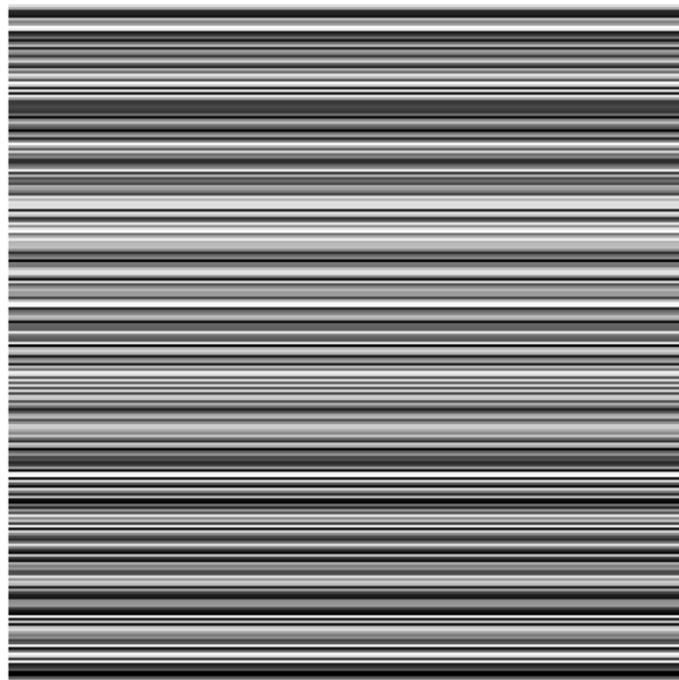
灰度直方图





2. 空间和时间冗余

- 行程对 (run-length pairs)
 - 灰度值，该灰度连续出现的次数
 - 压缩比 $\frac{256 \times 256 \times 8}{(256 + 256) \times 8} = \frac{128}{1}$



灰度直方图





2. 空间和时间冗余

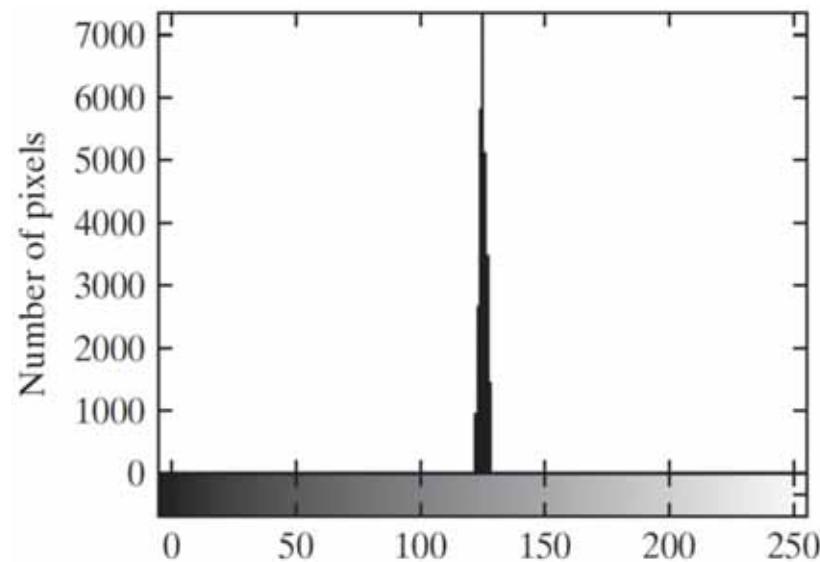
- 图像中的像素往往又是空间和时间相关的
 - 可以通过邻近像素预测该像素的值
- 更加高效但是视觉不可见的表示
 - 行程 (run-length)
 - 相邻像素的灰度差
 - 灰度差值具有规律性
- 映射
 - 可逆映射：可以完美还原
 - 不可逆映射：存在还原误差





3. 不相关的信息

- 不相关的信息
 - 被视觉系统忽略的信息
 - 与图像用途无关的信息



灰度直方图 (125 到 131 非零)





3. 不相关的信息

- 不相关的信息

- 被视觉系统忽略的信息
- 与图像用途无关的信息



平均灰度值

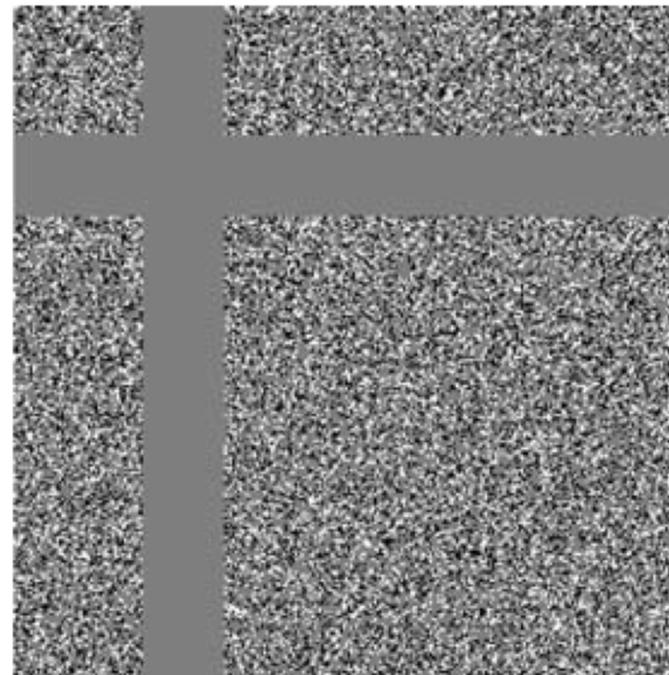
压缩比 $\frac{256 \times 256 \times 8}{8} = \frac{65536}{1}$





3. 不相关的信息

- 不相关的信息
 - 被视觉系统忽略的信息
 - 与图像用途无关的信息



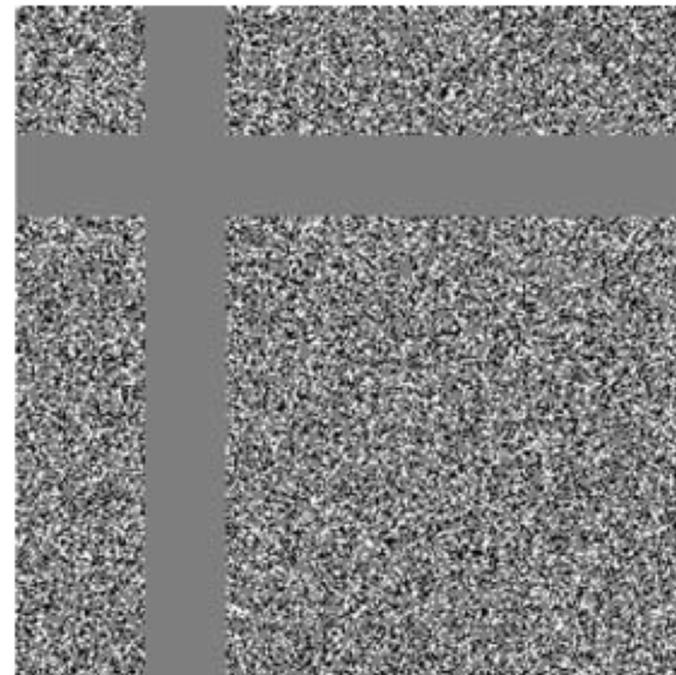
直方图均衡





3. 不相关的信息

- 不相关的信息
 - 被视觉系统忽略的信息
 - 与图像用途无关的信息
- 量化 (quantization)
 - 丢失信息
 - 不可逆映射



直方图均衡





提纲

- 引言
- 基础知识
 - 图像冗余
 - 图像信息的度量
 - 保真度准则
 - 图像压缩模型
- 基本的压缩方法
 - 霍夫曼编码
 - 行程编码
 - 基于符号的编码
- 数字图像水印





度量图像的信息

- 表达一幅图像最少的比特数是多少？
- 信息论 (Information Theory)
 - 信息的产生用概率过程来建模
 - 随机事件 E 蕴含的信息

$$I(E) = \log \frac{1}{P(E)} = -\log P(E)$$

- $P(E)$ 表示 E 发生的概率
- $P(E) = 1, I(E) = 0$
- 对数的底数决定了度量信息的单位
 - 底数为 m ，单位是 m -ary；2对应于比特；

$$P(E) = \frac{1}{2}, I(E) = -\log_2 \frac{1}{2} = 1$$





度量图像的信息

- 信源

- 生成统计上独立的随机事件
- 取值 $\{a_1, a_2, \dots, a_J\}$ ，概率 $\{P(a_1), P(a_2), \dots, P(a_J)\}$

- 信源的熵 (Entropy)

$$H = - \sum_{j=1}^J P(a_j) \log P(a_j)$$

- 每个输出的平均信息量
- 信源符号： a_j
- 事件独立：0记忆信源





度量图像的信息

- 虚构的灰度信源

- 输出灰度值，生成一幅图像
- 用直方图估计每个符号（灰度）的概率

- 灰度信源的熵

$$\tilde{H} = - \sum_{k=0}^{L-1} p_r(r_k) \log_2 p_r(r_k)$$

- $p_r(r_k)$ 表示 r_k 出现的概率
- 每个灰度输出的平均信息
- 不可能使用长度少于 \tilde{H} 的编码

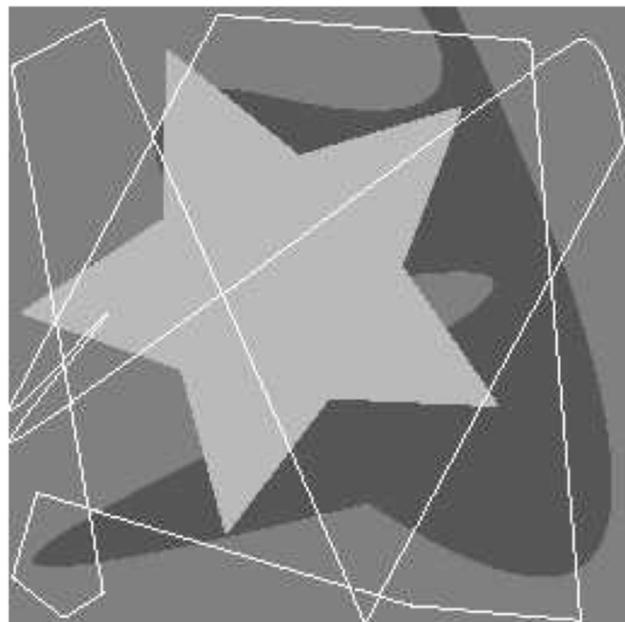




举例

● 图像的熵

$$\begin{aligned}\tilde{H} &= -[0.25 \log_2 0.25 + 0.47 \log_2 0.47 + 0.25 \log_2 0.25 + 0.03 \log_2 0.03] \\ &\approx -[0.25(-2) + 0.47(-1.09) + 0.25(-2) + 0.03(-5.06)] \\ &\approx 1.6614 \text{ bits/pixel}\end{aligned}$$



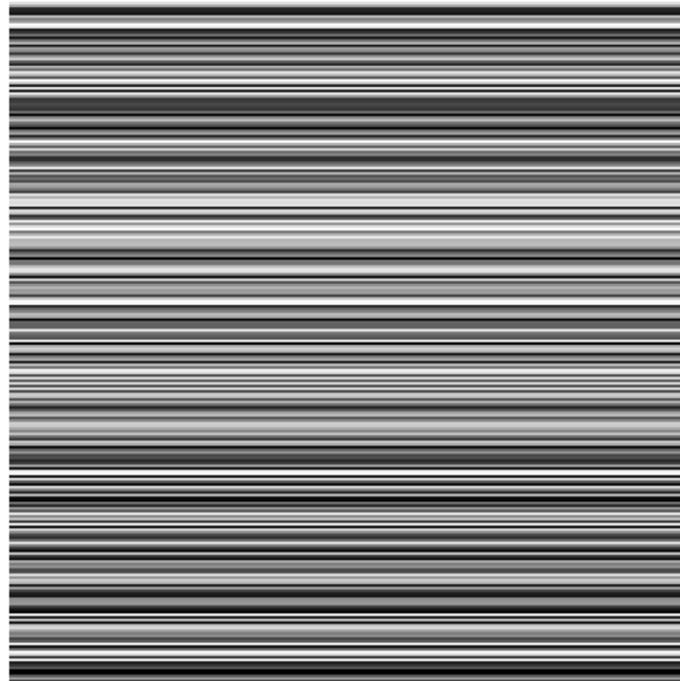
r_k	$p_r(r_k)$
$r_{87} = 87$	0.25
$r_{128} = 128$	0.47
$r_{186} = 186$	0.25
$r_{255} = 255$	0.03
r_k for $k \neq 87, 128, 186, 255$	0



举例

- 图像的熵

8比特



1.566比特





香农第一定理

- 无噪声编码定理
- 表示 n 个连续的符号

$$\lim_{n \rightarrow \infty} \left[\frac{L_{\text{avg},n}}{n} \right] = H$$

- $L_{\text{avg},n}$ 是表示 n 个符号的平均编码长度
- 每个符号的平均长度为 H
- 当图像的像素存在相关性
 - 该定理不再成立
 - 马尔科夫信源、有限记忆信源





提纲

- 引言
- 基础知识
 - 图像冗余
 - 图像信息的度量
 - 保真度准则
 - 图像压缩模型
- 基本的压缩方法
 - 霍夫曼编码
 - 行程编码
 - 基于符号的编码
- 数字图像水印





保真度准则

- 量化信息的损失

1. 客观保真度准则

- $f(x, y)$, 输入图像
- $\hat{f}(x, y)$, 压缩 , 解压缩之后的图像
- 误差

$$e(x, y) = \hat{f}(x, y) - f(x, y)$$

- 总误差

$$\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x, y) - f(x, y)]$$





保真度准则

1. 客观保真度准则

- 均方根误差 (root-mean-square error)

$$e_{\text{rms}} = \left[\frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x, y) - f(x, y)]^2 \right]^{1/2}$$

- 均方信噪比 (mean-square signal-to-noise ratio)

$$\text{SNR}_{\text{ms}} = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \hat{f}(x, y)^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x, y) - f(x, y)]^2}$$





保真度准则

2. 主观保真度准则

- 用户对解压缩后图像的主观评价

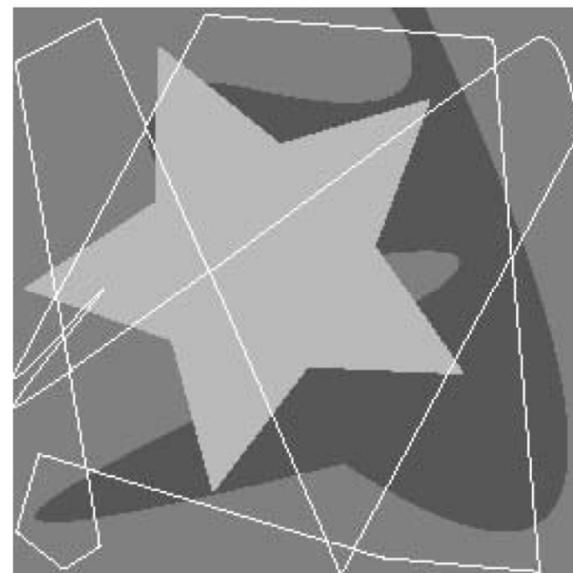
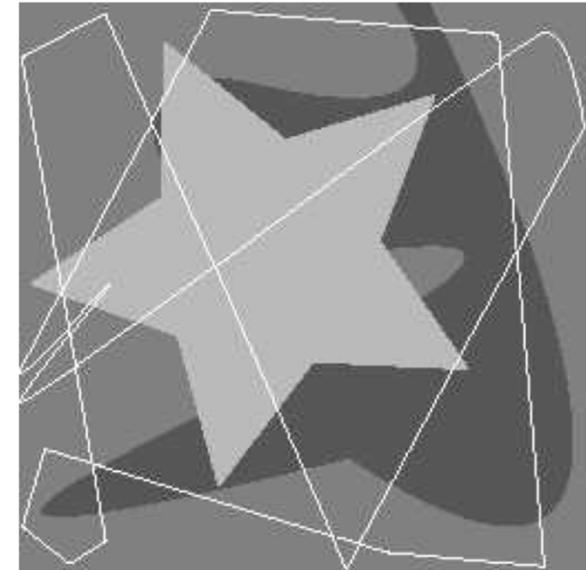
Value	Rating	Description
1	Excellent	An image of extremely high quality, as good as you could desire.
2	Fine	An image of high quality, providing enjoyable viewing. Interference is not objectionable.
3	Passable	An image of acceptable quality. Interference is not objectionable.
4	Marginal	An image of poor quality; you wish you could improve it. Interference is somewhat objectionable.
5	Inferior	A very poor image, but you could watch it. Objectionable interference is definitely present.
6	Unusable	An image so bad that you could not watch it.



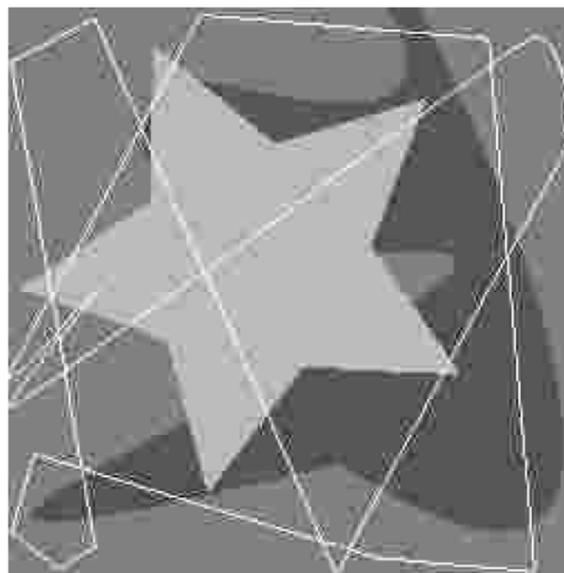
举例

- 三种近似

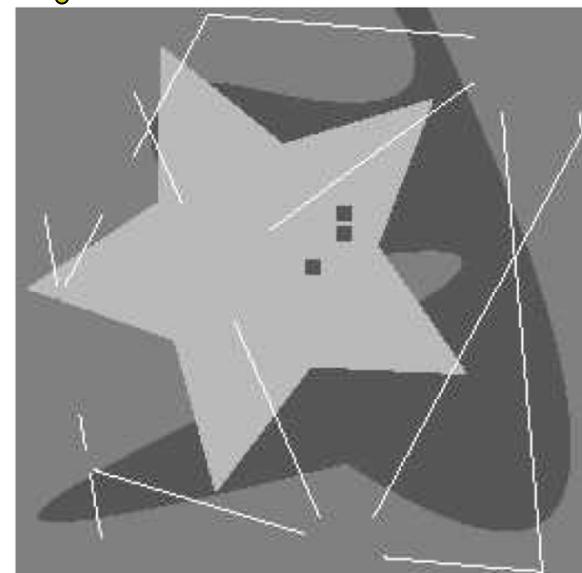
视觉上是质量最差的



均方根误差 5.17



15.67



14.17





提纲

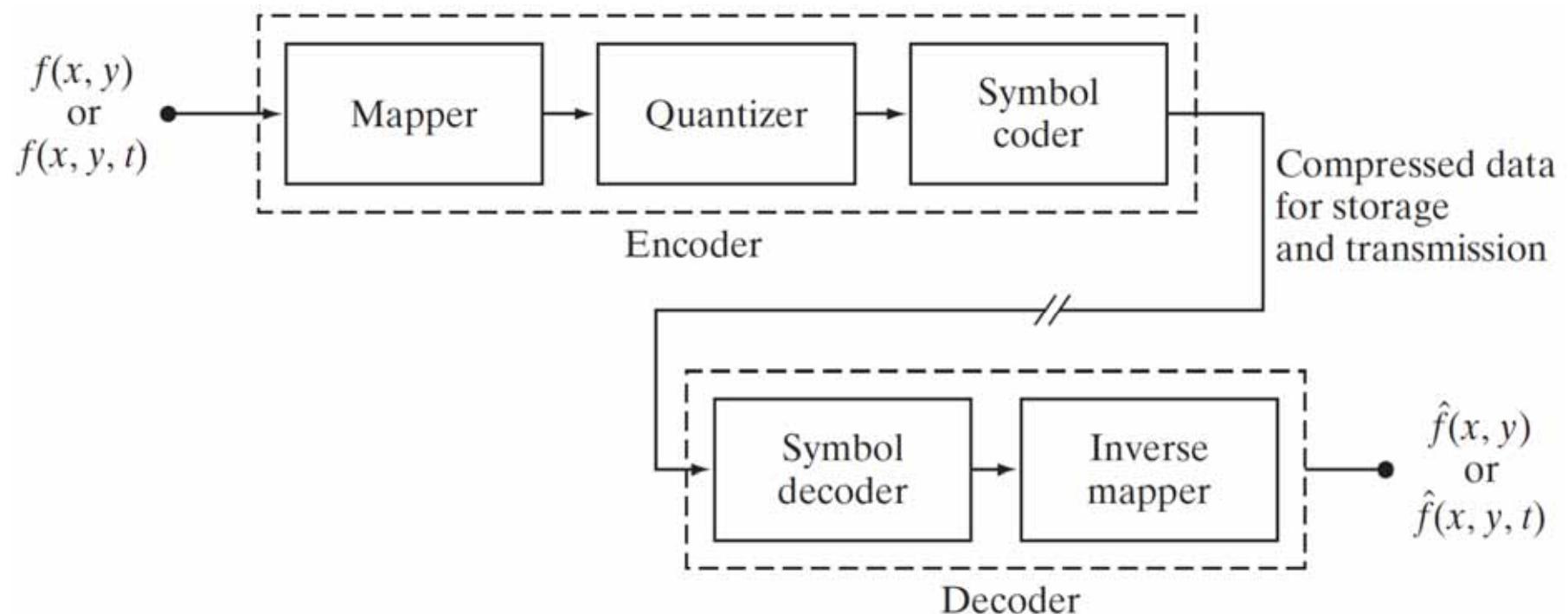
- 引言
- 基础知识
 - 图像冗余
 - 图像信息的度量
 - 保真度准则
 - 图像压缩模型
- 基本的压缩方法
 - 霍夫曼编码
 - 行程编码
 - 基于符号的编码
- 数字图像水印





图像压缩模型

- 编码器 (encoder)
- 解码器 (decoder)



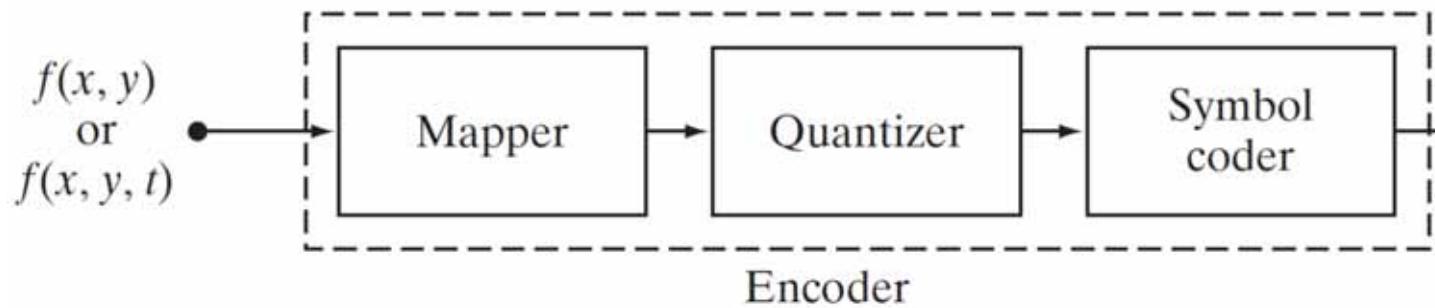
- 无损压缩、有损压缩





编码器

- 去掉三种冗余



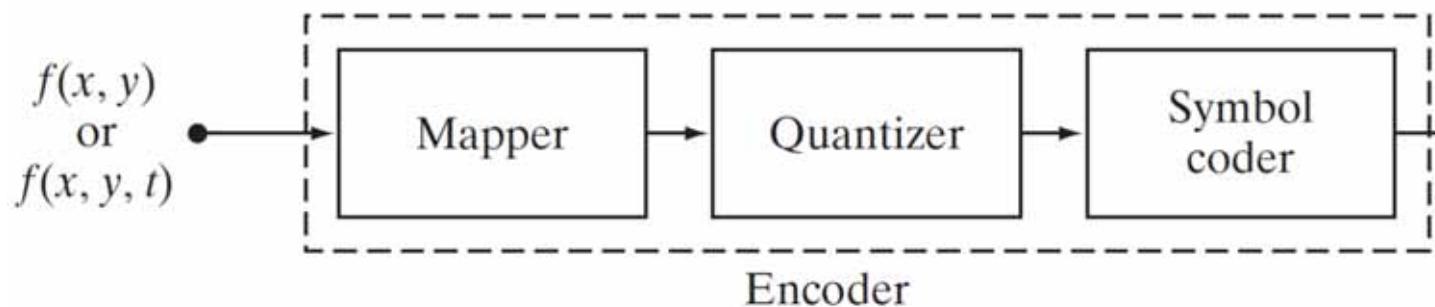
- 映射器（Mapper）
 - 转换为便于去掉空间和时间冗余的形式
 - 可逆的，未必改变数据量
- 量化器（quantizer）
 - 根据预设的保真度准则降低精度
 - 去除不相关的信息（不可逆）





编码器

- 去掉三种冗余



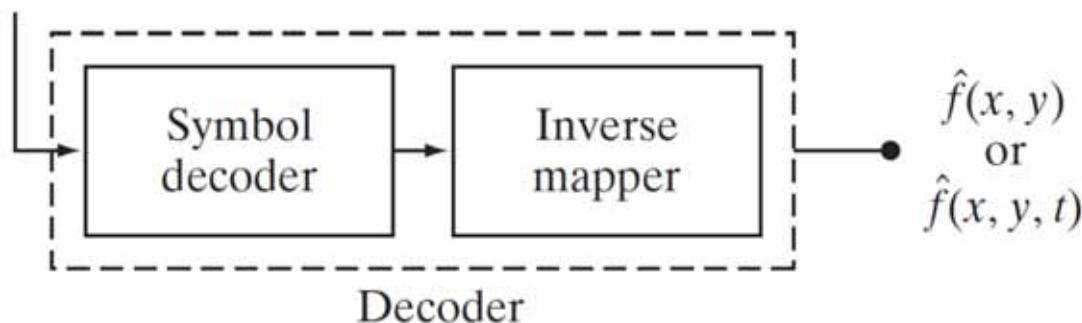
- 符号编码器 (symbol coder)
 - 生成**定长编码**表示量化器输出
 - 生成**变长编码**表示量化器输出
 - 最频繁的输出用最短的编码
 - 可逆的





解码器

- 恢复图像



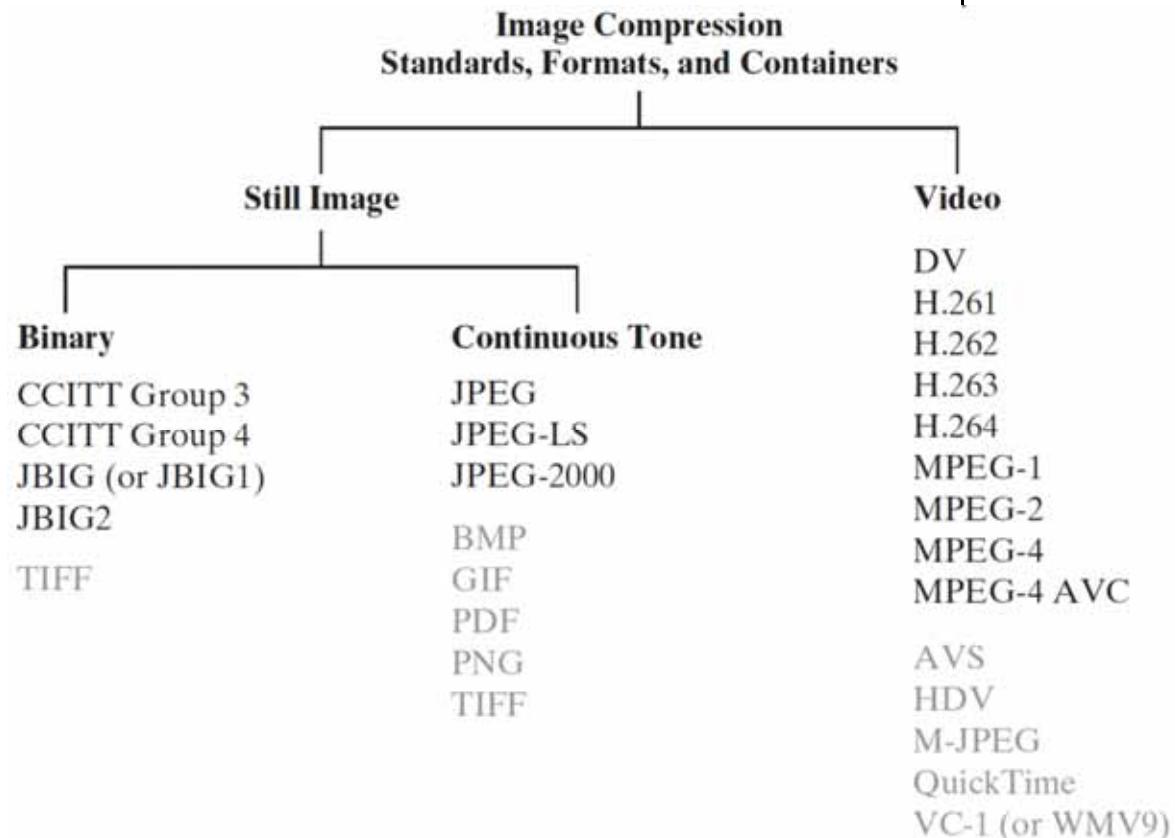
- 符号解码器 (symbol decoder)
 - 符号编码器的逆操作
- 反映射器 (inverse mapper)
 - 映射器的逆操作





图像格式和压缩标准

- 图像格式
 - 定义数据的排列方式、压缩的类型
- 压缩标准
 - 定义压缩和解压缩的流程



黑色为国际公认的标准





提纲

- 引言
- 基础知识
 - 图像冗余
 - 图像信息的度量
 - 保真度准则
 - 图像压缩模型
- **基本的压缩方法**
 - 霍夫曼编码
 - 行程编码
 - 基于符号的编码
- 数字图像水印





霍夫曼编码

- 单独对信源符号编码
 - 霍夫曼编码是最短的编码
 - 对于固定的 n ，该编码是最优的
- 广泛应用于各种压缩标准中
 - CCITT
 - JBIG2
 - JPEG
 - MPEG-1,2,4
 - H.261, H.262, H.263, H.264
- 并不局限于灰度值
- 并不局限于数字图像处理





霍夫曼编码

1. 简化信源

- 对符号的概率进行排序，合并低概率符号
- 重复该过程，直到剩余两个符号

Original source		Source reduction			
Symbol	Probability	1	2	3	4
a_2	0.4	0.4	0.4	0.4	0.6
a_6	0.3	0.3	0.3	0.3	0.4
a_1	0.1	0.1	0.2	0.3	
a_4	0.1	0.1	0.1		
a_3	0.06	0.1			
a_5	0.04				





霍夫曼编码

2. 对简化后的信息源编码

- 从最小信源开始，返回到原信源
- 为每一个分支分配符号

熵是
2.14

Original source			Source reduction							
Symbol	Probability	Code	1	2	3	4				
a_2	0.4	1	0.4	1	0.4	1	0.6	0		
a_6	0.3	00	0.3	00	0.3	00	0.4	1		
a_1	0.1	011	0.1	011	0.2	010	0.3	01		
a_4	0.1	0100	0.1	0100	0.1	011				
a_3	0.06	01010	0.1	0101						
a_5	0.04	01011								

$$\begin{aligned}L_{\text{avg}} &= (0.4)(1) + (0.3)(2) + (0.1)(3) + (0.1)(4) + (0.06)(5) + (0.04)(5) \\&= 2.2 \text{ bits/pixel}\end{aligned}$$





霍夫曼编码的实现

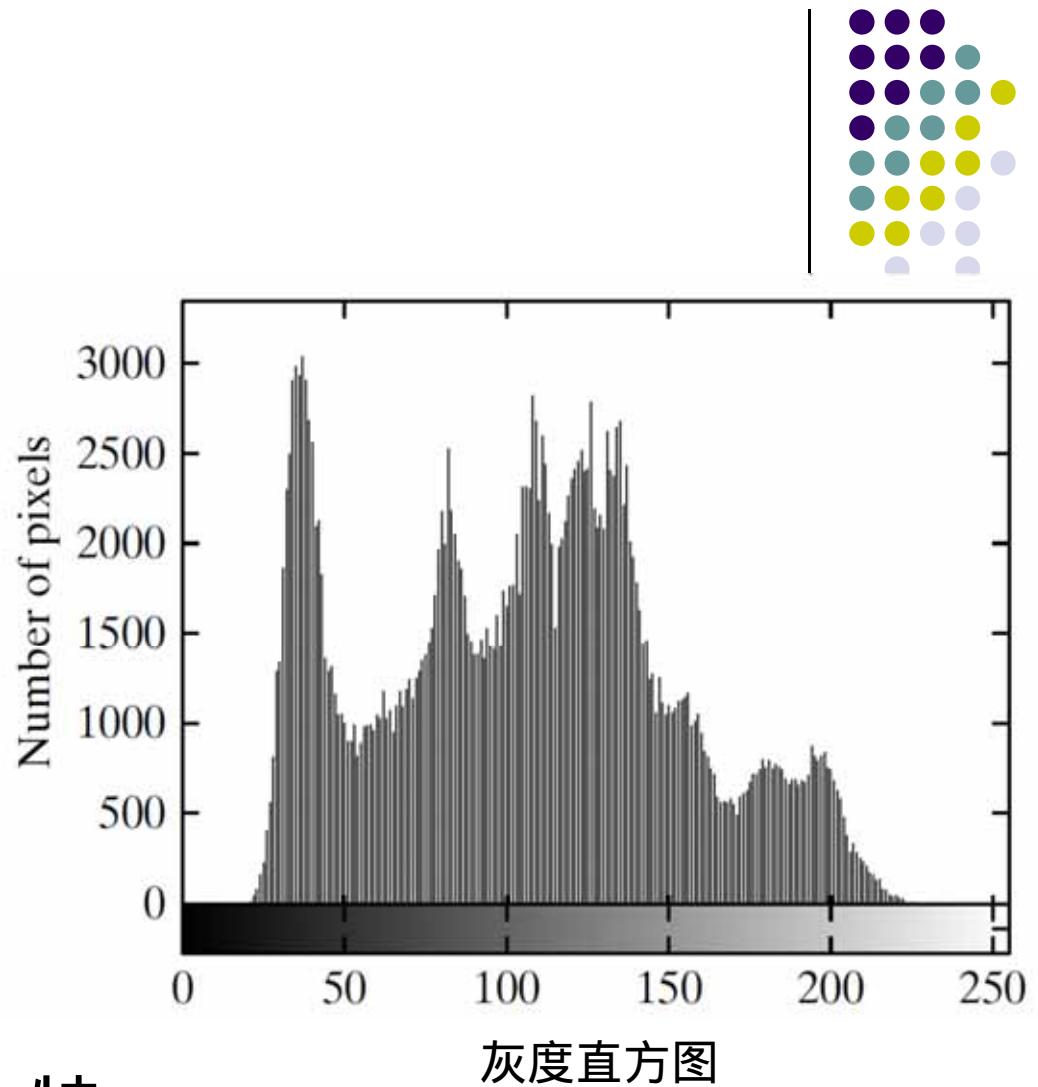
1. 构造霍夫曼编码
2. 查表法
 - 编码、解码
 - 瞬时性
 - 每个编码独立解码
 - 唯一可解码
 - 序列的解码方式唯一

Original source		
Symbol	Probability	Code
a_2	0.4	1
a_6	0.3	00
a_1	0.1	011
a_4	0.1	0100
a_3	0.06	01010
a_5	0.04	01011

010100111100 \rightarrow $a_3a_1a_2a_2a_6$



举例



- 霍夫曼编码 : 7.428比特
- 图像的熵 : 7.3838比特
- 压缩比 : $C = 8/7.428 = 1.077$





提纲

- 引言
- 基础知识
 - 图像冗余
 - 图像信息的度量
 - 保真度准则
 - 图像压缩模型
- **基本的压缩方法**
 - 霍夫曼编码
 - **行程编码**
 - 基于符号的编码
- 数字图像水印





行程编码

- 行程对 (run-length pairs)
 - 灰度值，该灰度连续出现的次数
- 适用于存在连续灰度值的图像
 - 去除简单的空间冗余（连续的相同灰度）
 - 如果行程短或没有，导致数据增多
- 广泛应用于各种压缩标准中
 - CCITT
 - JBIG2
 - JPEG
 - M-JPEG
 - MPEG-1,2,4
 - BMP





BMP文件格式

- 编码模式 (encoded mode)
 - 两个字节 : 连续的像素数量、灰度值
- 绝对模式 (absolute mode)
 - 第一个字节 : 0
 - 第二个字节 : 4种情况

Second Byte Value	Condition
0	End of line
1	End of image
2	Move to a new position
3-255	Specify pixels individually

后面的两个
字节指明位
置偏移

后面是连续
的未压缩灰
度值





举例

- 未压缩的BMP文件
 - 263,244字节
- 压缩的BMP文件
 - 267,706字节
- 压缩率 $C = 0.98$





二值图像

- 行程编码特别有效
 - 容易出现连续的灰度值
- 只需要指明长度
 - 指明每一行开始的灰度值
 - 假设每行的初始灰度值总是1
- 对长度本身进行压缩
 - 霍夫曼编码





提纲

- 引言
- 基础知识
 - 图像冗余
 - 图像信息的度量
 - 保真度准则
 - 图像压缩模型
- **基本的压缩方法**
 - 霍夫曼编码
 - 行程编码
 - 基于符号的编码
- 数字图像水印





基于符号的编码

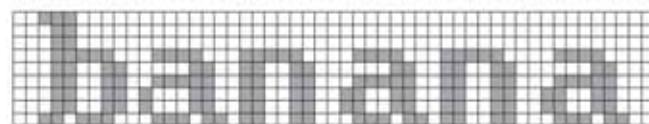
- 符号
 - 图像中频繁出现的子图像
 - 方法
 - 将图像表示为符号的集合
1. 符号字典
 - 存储符号的位图
 2. 三元组集合
 - $\{(x_1, y_1, t_1), (x_2, y_2, t_2), \dots\}$
 - (x_i, y_i) 指明符号位置、 t_i 指明具体符号



基于符号的编码

● 压缩原理

- 频繁出现的符号只存储了1次
- 适用于压缩扫描的文档



Token	Symbol	Triplet
0		(0, 2, 0) (3, 10, 1)
1		(3, 18, 2) (3, 26, 1)
2		(3, 34, 2) (3, 42, 1)

$$9 \times 51 \times 1 = 459 \text{ 比特}$$

压缩比 : $C = 1.61$

$$\begin{aligned} & 9 \times 7 + 6 \times 7 + 6 \times 6 \\ & + 6 \times 3 \times 8 = 285 \text{ 比特} \end{aligned}$$





JBIG2 压缩

- 压缩二值图像的国际标准
- 把图像分成3种类型的区域
 1. 字符组成的**文本区域**
 - 使用基于符号的编码
 - 每个大小写字母对应一个符号
 - a. 有损压缩（感知无损）
 - 忽略字典内符号和图像的差异
 - b. 无损压缩
 - 额外存储差异





JBIG2 压缩

2. 模式组成的半色调区域

- 类似于文本区域
- 符号：周期性的模式

3. 普通区域

- 用其他方式压缩
- 定义解码器特性
- 没有明确指明具体的编码器



举例

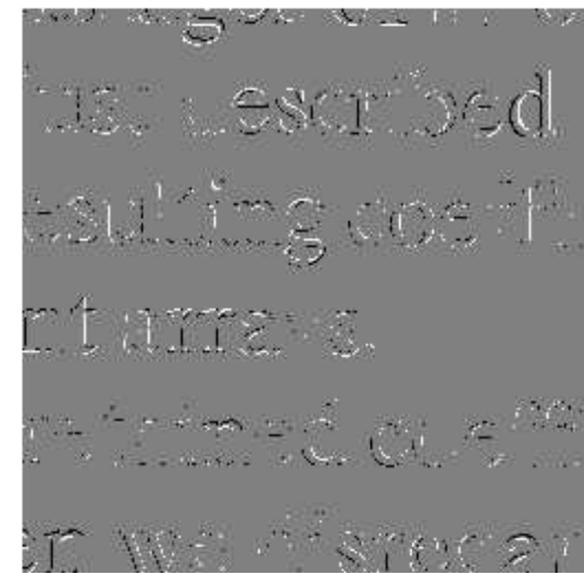
• JBIG2 压缩

just described resulting coefficient arrays.
retained coefficients in we disregard

原图
无损的压缩和解压缩

有损的压缩
感知无损

差异



压缩比
20.3

字母d

压缩比
27.7





提纲

- 引言
- 基础知识
 - 图像冗余
 - 图像信息的度量
 - 保真度准则
 - 图像压缩模型
- 基本的压缩方法
 - 霍夫曼编码
 - 行程编码
 - 基于符号的编码
- 数字图像水印





数字图像水印

- 图像压缩使得图像便于传播
 - 可以被频繁地复制
- 如何保护图像所有者的版权？
- 水印
 - 在图像中插入的信息
 - 不能从图像中分离出来
 - 增加了图像中的信息（数据）
 - 不可见水印
 - 可见水印（不透明、半透明）





图像水印的作用

1. 版权识别
 - 证明图像的拥有者
2. 用户识别
 - 在水印中编码合法用户的身份，识别非法复制来源
3. 证明真实性
 - 保证一幅图像未被篡改
4. 自动监控
 - 通过监控水印来检测图像的使用情况
5. 复制保护
 - 指明图像的使用规则





举例

- $f_w = (1 - a)f + aw$



$a = 0.3$



差异





不可见水印

- 无法用视觉感知
 - 插入视觉上冗余的信息
 - 视觉系统忽略或无法感知的信息
- 解码算法可以恢复该水印
- 利用8比特图像的低阶比特
 - 利用2个最低阶比特

$$f_w = 4\left(\frac{f}{4}\right) + \frac{w}{64}$$





举例

- LSB水印图像



去掉6位高
阶比特恢
复的水印

Digital Image
Processing



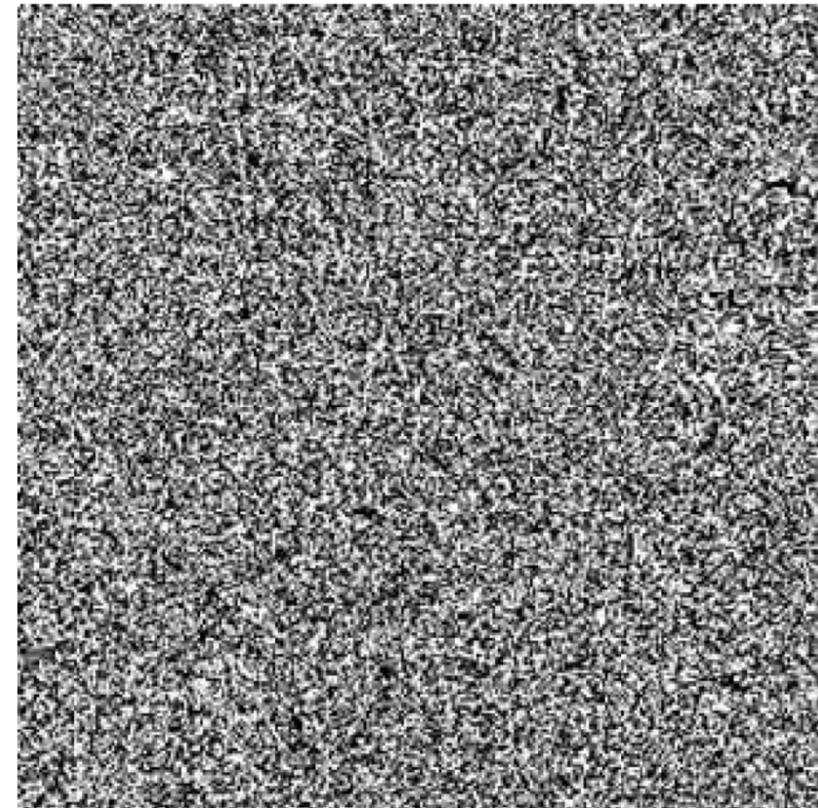


不可见水印

- 抵抗有意或无意的删除企图/修改



用有损JPEG压缩、解压缩



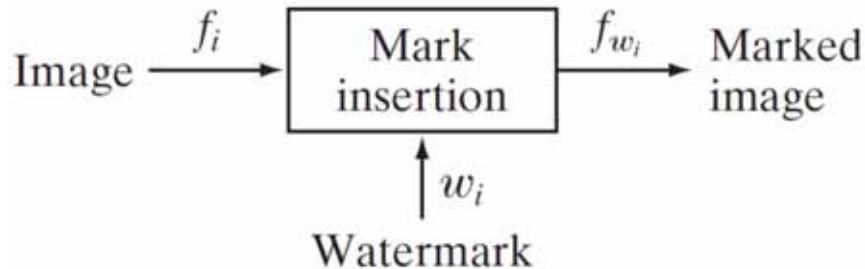
恢复出的水印



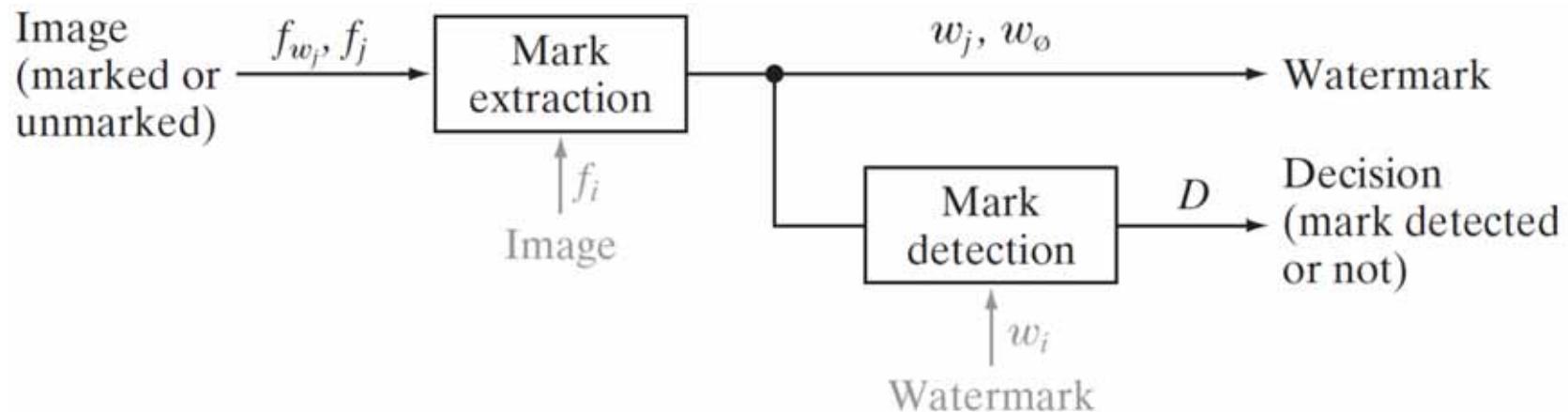


图像水印系统

- 编码器
 - 插入水印



- 解码器
 - 提取、验证水印



- 输入可能有图像和水印





在变换域加水印

- 水印的插入和提取也可以在变换域
- 基于DCT的不可见水印
 - DCT：离散余弦变换
 1. 计算图像的2维DCT变换
 2. 找到 K 个最大的系数 c_1, c_2, \dots, c_K
 3. 从标准高斯分布生成 K 个伪随机数 $\omega_1, \dots, \omega_K$
 4. 将上述伪随机数嵌入到 K 个最大的系数中

$$c'_i = c_i \cdot (1 + \alpha\omega_i) \quad 1 \leq i \leq K$$

5. 基于 c'_i 计算反DCT变换





在变换域加水印

- 水印的插入和提取也可以在变换域
- 基于DCT的不可见水印
 - DCT：离散余弦变换
- 安全性强
 - 随机数没有明显的结构
 - 多个频域分量影响了整幅图像
 - 难以定位水印的位置
 - 攻击水印会影响图像质量
 - 水印和重要频率分量耦合在一起

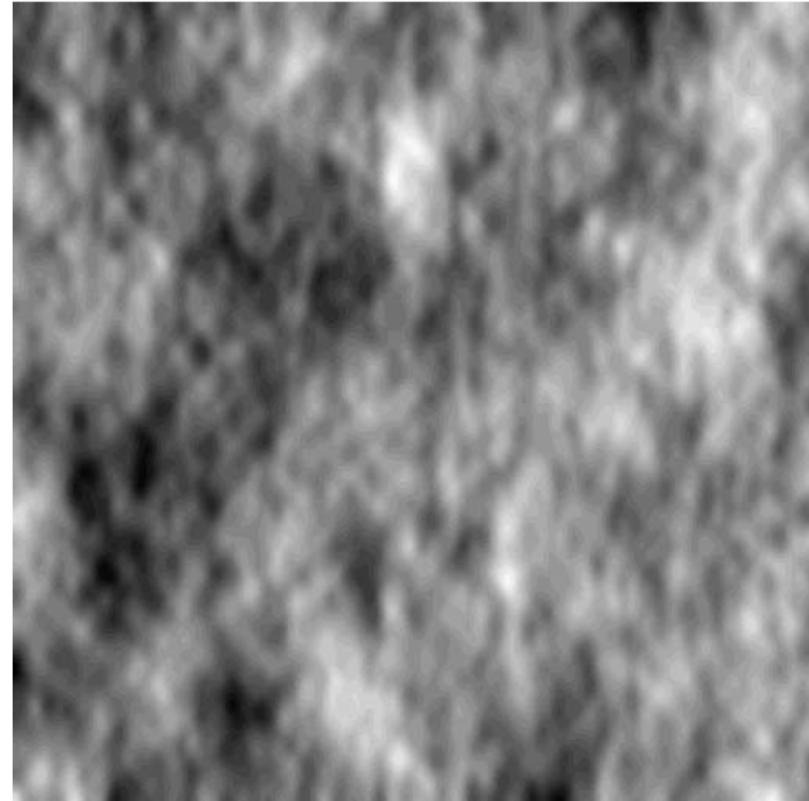




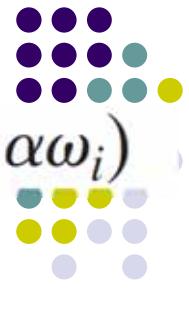
举例

- 基于DCT的不可见水印
 - $a = 0.1, K = 1000$

缩放后
的差异



检测变换域中的水印

$$c'_i = c_i \cdot (1 + \alpha \omega_i)$$


● 检测DCT水印

- 给定水印 $\omega_1, \omega_2, \dots, \omega_K$
- 给定DCT系数 c_1, c_2, \dots, c_K
 1. 计算图像的2维DCT变换
 2. 提取 K 个最大的DCT系数，记为 $\hat{c}_1, \hat{c}_2, \dots, \hat{c}_K$
 - 判断 c'_i 和 \hat{c}_i 的关系
 3. 计算水印 $\hat{\omega}_1, \hat{\omega}_2, \dots, \hat{\omega}_K$

$$\hat{\omega}_i = \frac{\hat{c}_i - c_i}{a c_i}, \quad \text{for } 1 \leq i \leq K$$





检测变换域中的水印

- 检测DCT水印

4. 计算水印之间的相似度

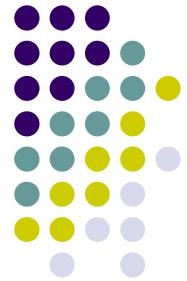
- 相关性系数

$$\gamma = \frac{\sum_{i=1}^K (\hat{\omega}_i - \bar{\hat{\omega}})(\omega_i - \bar{\omega})}{\sqrt{\sum_{i=1}^K (\hat{\omega}_i - \bar{\hat{\omega}})^2 \cdot \sum_{i=1}^K (\omega_i - \bar{\omega})^2}} \quad 1 \leq i \leq K$$

5. 与阈值 T 进行比较

$$D = \begin{cases} 1 & \text{if } \gamma \geq T \\ 0 & \text{otherwise} \end{cases}$$





抵抗水印攻击

JPEG有损压缩 (RMS=7)



$\gamma = 0.9945$

JPEG有损压缩 (RMS=10)



$\gamma = 0.7395$





抵抗水印攻击

空间滤波平滑



$\gamma = 0.8390$

添加高斯噪声



$\gamma = 0.8230$





抵抗水印攻击

直方图均衡



$\gamma = 0.5210$

轻微旋转



$\gamma = 0.3113$

