隐写学原理与技术 整理

原创

岁月漫长_ 已于 2022-03-03 21:01:59 修改 53 收藏 1

分类专栏： 图像隐写 文章标签： 计算机视觉

于 2022-03-03 21:01:32 首次发布

版权声明：本文为博主原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接和本声明。

本文链接：https://blog.csdn.net/qq_40859587/article/details/123263504

版权

图像隐写 专栏收录该内容

13 篇文章 2 订阅

订阅专栏

以前的OneNote笔记搬运过来

前言

加密：保护保密通信或存储的内容不受非授权的浏览。

隐写：保护保密通信或存储的事实不被发现，是一种信息隐藏技术，目的是隐蔽保密的事实。

隐写分析：识别隐写载体的技术。

隐写的**主要设计目标**：提高含密载体的隐蔽性，提高数据的隐藏量。

第1章 绪言

机密性：确保敏感或机密数据的传输和存储不遭受未授权的浏览，甚至不暴露保密通信。

因为密码数据存在伪随机特性，与非密码数据在统计特征上不同，直接发送或存储密文难以掩盖保密事实。

1.1 从密码到信息隐藏与隐写

1.1.1 密码方法的局限

密码：消息保密传输、数据来源认证、完整性认证。

不能解决的问题：

1）保密通信的行为隐蔽性。密码加密数据有伪随机性（自然数据不具有）。

2）松散环境下的内容保护与内容认证问题。信息安全中，内容指媒体表达的信息，和编码形式无关；数据指信息的具体表现形式。数字签名和验证码保护数据，实际中也要对内容的所有权或者真实性进行认证或保护。

1.1.2 信息隐藏基本概念

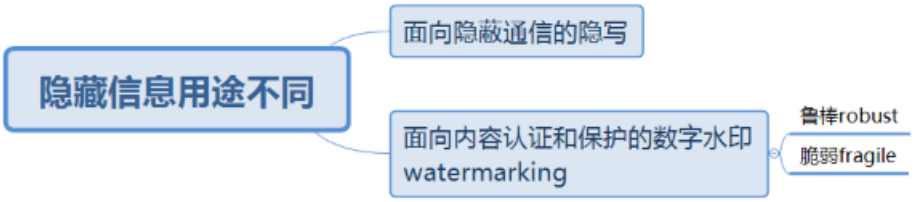
定义：将信息隐蔽的存储在其他载体（cover）中，使其难以发现或消除。

载体：可公开的数字内容，多媒体或网络包。

隐藏的信息：保密通信的加密消息、内容所有权标识、内容用户标识等验证信息，可以被授权用户提取或验证。

前提：载体内容存在冗余性。

CSDN @岁月漫长_



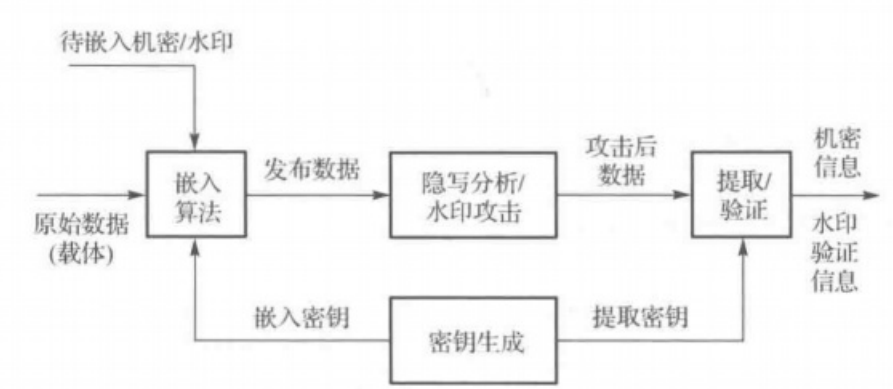


图 1.1　信息隐藏基本研究模型

CSDN @岁月漫长_

1. 隐写

   隐写后的载体：隐文（stego-text）/隐写媒体（stego-media）/含密载体/隐写载体。

   现代隐写：以数字媒体为载体的隐写方法

   主要性能：

   i. **安全性**：隐写后媒体特征变化的隐蔽性。

   ii. **隐写容量**：隐写传输的信息量。负载率(payload)/嵌入率(embedding rare):平均每个嵌入位置所承载的隐蔽消息量：m表示传输的消息量，n为嵌入位置的数量，负载率α=m/n

   单位bpp（bits per pixel）和bpnac（bits per nonzero alternating-current coefficient)

   iii. **嵌入效率e**（embedding efficiency）：平均没修改一个位置单元所能传输的消息量

CSDN @岁月漫长_

$$e=\frac{\text{平均每个载体样点承载的消息比特}}{\text{平均每个载体样点被修改量}}=\frac{\alpha}{d}=\frac{\frac{m}{n}}{\frac{E(K)}{n}}=\frac{m}{E(K)}\ \text{bit/次}\qquad(1.2)$$

d为平均每个载体样点被修改量；E（K）表示总修改次数的期望。提高e有助于减少修改次数，增加安全性。

应用安全性：对手难以从隐写应用协议与实现上发现检测隐写媒体的方法。

i. 计算效率：隐写算法执行效率

ii. 鲁棒性：信道有损情况下的抗干扰能力（目前普遍假设不存在干扰）

1. 水印

数字产权管理（digital rights management,DRM）技术

鲁棒水印：将版权或者购买者有关的信息嵌入到数字媒体中，使得攻击者难以在载体不被破坏情况下消除水印，授权者可以通过检测水印实现对版权所有者或内容购买者的认定（**指纹化**fingerprinting）。

嵌入方法：将水印信息进行信号调制，嵌入载体相对稳定成分中

性能：

1) **鲁棒性**（robustness）：对含水印媒体进行改动（信号处理，添加噪声，有损压缩编码，**尺寸缩放和裁剪**等不显著破坏内容），授权用户仍能够提取水印。（难度大）
2) **水印容量**（capacity）：水印能够可靠传输的信息量。
3) **安全性**（security）：攻击者难以从水印算法，应用协议或者实现上获得有益于攻击的信息。

   盲性（blindness）：水印检测不依赖于原始媒体或其相关信息的存在（公有的public）

脆弱水印（fragile watermarking）：将防伪信息隐藏在数字内容中，通过水印检测发现篡改和位置（信息随着内容改动而变化），用于内容认证和篡改定位。（比鲁棒水印技术更成熟）**完整性**

基本性质：

1) 脆弱性（fragileness）：水印随内容改动而变化
2) 定位精度：反映篡改位置
3) 可逆性（reversibility）：水印可被授权者消除，原始媒体得到还原（可逆水印）
4) 安全性：同上
5) 盲性：同上

CSDN @岁月漫长_

1.1.3 隐写与隐写分析对抗模型

隐写分析：针对隐写的攻击，通过检测载体特征变化判定隐写；对隐写算法、参数的估计或者对隐藏信息的非授权提取。

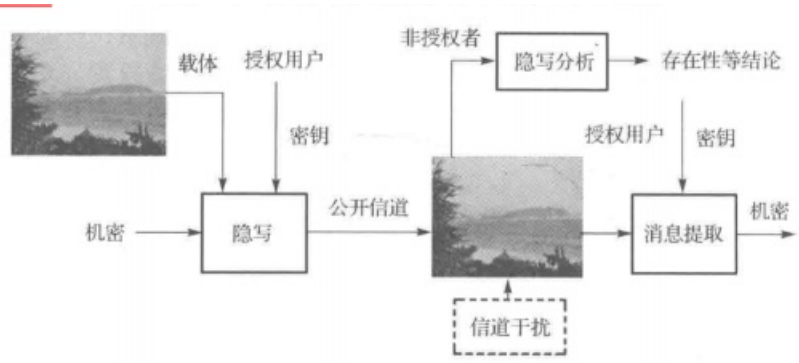


图 1.2　隐写与隐写分析对抗模型示意CSDN @岁月漫长_

隐写分析（被动攻击）的主要性能指标：

a. **漏检率/假阴性率FN**：将隐写媒体判断为自然媒体比率；**真阳性率TP/检测率**=1-FN，隐写媒体判断正确
b. **虚警率/假阳性率FP**：将自然媒体判断为隐写媒体；**真阴性率TN**=1-FP，自然媒体判断正确
c. 正确率/精度：(TP+TN)/2 = 1-(FN+FP)/2;错误率=(FN+FP)/2=1-(TP+TN)/2

隐写分析实验：对原始载体进行检测得到TN或FP，对隐写载体进行检测得到FN或TP，进而得出正确率。ROC曲线描述隐写分析分类器的综合分析性能

CSDN @岁月漫长_

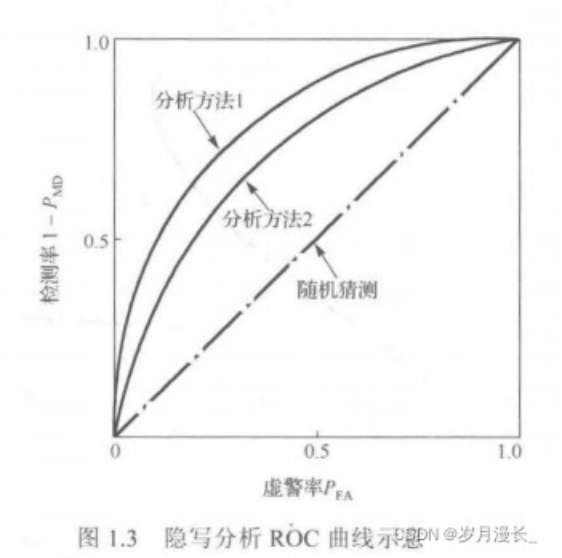


图 1.3　隐写分析 ROC 曲线示意

CSDN @岁月漫长_

错误率=min 1/2(虚警率+漏检率)

“囚犯”问题：隐写和隐写分析的主要对抗手段

1.**隐写**对抗隐写分析的主要方法

降低对特征的扰动，干扰各类检测方法

i. 特征保持（困难）
ii. 降低修改次数
iii. 降低修改扰动：降低修改信号幅度或能量
iv. 降低被检测代价
v. 提高应用方式安全：难以找到漏洞和合适的检测对象

2.**隐写分析**对抗隐写的主要方法

发现、识别隐写对特征的扰动

i. 提取隐写分析特征
ii. 构造隐写特征识别系统
iii. 获得先验知识

通用隐写分析一般基于监督学习，需要基于已知隐写算法制作训练样本；专用分析基于已知隐写算法进行专门设计

1.2 隐写的应用发展

多媒体冗余信息多，互联网提供了传输平台

1.3 隐写安全指标

1.3.1 基于分布偏差的指标

载体分布的变化程度

CSDN @岁月漫长_

针对某种隐写方法，设 $\boldsymbol{x} \in \mathcal{X}$ 为样本空间，P_C 为自然载体分布，P_S 为含密载体分布，

相对熵或KL散度：（理论描述，难以计算）

相对熵称KL散度：（理论描述，难以计算）

$$D(P_C \| P_S)=\sum_{x\in\mathcal{X}} P_C(x)\log\frac{P_C(x)}{P_S(x)}=\sum_{x\in\mathcal{X}} P_C(x)(\log(P_C(x))-\log(P_S(x)))$$

总的K散度等于各维度上的和（用于估计）

用最大平均偏差MMD来计算

设 $\mathcal{F}$ 为一组函数集合，$x\sim p$，$y\sim q$，则 MMD 的定义是

$$\mathrm{MMD}(\mathcal{F},p,q)=\sup_{f\in\mathcal{F}}(E_{x\sim p}(f(x))-E_{y\sim q}(f(y)))$$

如果将函数 $f\in\mathcal{F}$ 视为不同的投影或者变换，则 MMD 的含义是，对依两个分布抽样得到的数据，取它们在不同投影下期望值差值的最大者。在以上计算中，考虑投影的作用是，可能存在一个投影使分布偏差加大或减小，当前的分类器普遍基于投影距离进行判决，因此，合理构造 $\mathcal{F}$ 有助于确保有效评估分布之间的各方向差异。在用 MMD 衡量载体与含密载体特征的偏差中，需要对载体集生成载体特征集 $\mathcal{X}=\{x_1,\cdots,x_D\}$ 与相应的含密载体特征集 $\mathcal{Y}=\{y_1,\cdots,y_D\}$，如果可以假设每个载体的出现可能相同，则 MMD 为

$$\mathrm{MMD}(\mathcal{F},p,q)=\sup_{f\in\mathcal{F}}\left(\frac{1}{D}\sum_{i=1}^{D}f(x_i)-\frac{1}{D}\sum_{i=1}^{D}f(y_i)\right) \tag{1.8}$$

显然，计算 MMD 的关键是确定合适的 $\mathcal{F}$，具体可参见文献[28]。

1.3.2 基于抗隐写分析性能的指标

直接用隐写分析的正确率等

1.4 内容安排

1. 基本信息嵌入方法
2. 提高隐写安全的方法：分布特性保持，隐写编码，最优嵌入，自适应隐写
3. 检测特定隐写的专用隐写分析
4. 检测多种隐写的通用隐写分析和盲隐写分析
5. 自适应/深度学习隐写分析

CSDN @岁月漫长_

第2章 图像编码与基本嵌入方法

数字照相机接受物体的反光或发光，光信号→电信号，采样、量化得到以数字阵列形式表达的自然图像，也称初始图象raw image，阵列中每个元素称为像素pixel，是 对自然图像电信号的采样点，再经过图像编码（隐写消息嵌入域）进行实际使用。

2.1 空域图像编码

在图像空间域进行编码，直接对 像素 进行编码

空域编码图像：光栅格式（位图）和调色板格式

图像编码标准常包括多类编码方法，BMP TIFF PNG等标准都支持光栅和调色板格式，每个编码又有多种具体编码方法。

2.1.1 光栅格式

直接用数字阵列的形式存储图像像素，每个像素有位置和颜色属性。

RGB模型，三通道，BMP PNG TIFF用8bit表示一个颜色分量，色彩总数为2的3*8次方，色深为3*8bit，PNG和TIFF用16bit颜色分量，色彩总数为2的3*16次方

灰度图和二值图采用单通道，仅表达亮度信息，三基色

Y=0.299R+0.587G+0.114B

对亮度设置阈值，得到二值图像

YUV模型：同时存在亮度和色度分量，可直接显示为灰度信号，亮度分量是RGB色彩分量的线性组合

CSDN @岁月漫长_

另外一种常用的彩色图像色彩模型是 YUV 模型，其主要特点是同时存在亮度与色度分量，有利于直接显示为灰度信号。由式(2.1)可知，亮度分量(luminance component)是 RGB 色彩分量的线性组合，因此，若令 $U=0.492(B-Y)$、$V=0.877(R-Y)$ 代表色度分量，则也可以用 (Y,U,V) 表示像素[29]，此时有线性变换关系

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{2.2}$$

常见的情况是，RGB 每个分量用 8bit 表示，则取值范围是[0,255]，但是，以上 YUV 中除了 Y 分量在此范围取值外，U 与 V 均可能取负整数和正整数。为了统一取值范围为[0,255]，使得可以用 1 个 8bit 表示，以上 YUV 模型通过以下方法变换为 YC_bC_r 模型

$$\begin{pmatrix} Y \\ C_b \\ C_r \end{pmatrix} = \begin{pmatrix} 0 \\ 128 \\ 128 \end{pmatrix} + \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{2.3}$$

例如，常用的 TIFF、JPEG 等图像标准采用了以上 YC_bC_r 模型[30]。需要指出，其他编码标准[29]也定义了类似的 YUV 或者 YC_bC_r 模型。

CSDN @岁月漫长_

量化值也叫步长

2.1.2 调色板格式

构造一个调色板palette，包括全部色彩RGB向量的一张表，每个色彩按照排列次序用一个索引值标定，像素仅包含色彩索引值。

常用GIF，（PNG TIFF BMP也支持调色板，但不支持动画）

2.2 变换域编码图像

JPEG：在变换域进行有损压缩编码，用无损压缩编码进行数据存储

流程 https://www.cnblogs.com/buaaxhzh/p/9138307.html

1. YCbCr格式化。

   图片由 RGB 色彩空间转换到 YUV 色彩空间，转换关系如下

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

$$U = -0.169R - 0.331G + 0.500B + 128$$

$$V = 0.500R - 0.419G - 0.081B + 128$$

   一般来说U,V是有符号的数字，但这里通过加上128，使其变为无符号数，方便存储和计算

2. 图像分块。

   8*8分块，不够就扩充，对Y Cb Cr按4：1：1比例采样

3. DCT离散余弦变换。

   先将输入值从[0，255]平移至[-128，127]，再用DCT变换矩阵实现，

CSDN @岁月漫长_

DCT变换矩阵计算公式为

$$T_{ij} = \begin{cases} \frac{1}{\sqrt{M}} & i = 0, 0 \le j \le M-1 \\ \sqrt{\frac{2}{M}} \cos \frac{\pi(2j+1)i}{2M} & 1 \le i \le M-1, 0 \le j \le M-1 \end{cases}$$

故8*8的DCT变换矩阵

$$T = \begin{bmatrix} 0.35355 & 0.35355 & 0.35355 & 0.35355 & 0.35355 & 0.35355 & 0.35355 & 0.35355 \\ 0.49039 & 0.41573 & 0.27779 & 0.09755 & -0.09755 & -0.27779 & -0.41573 & -0.49039 \\ 0.46194 & 0.19134 & -0.19134 & -0.46194 & -0.46194 & -0.19134 & 0.19134 & 0.46194 \\ 0.41573 & -0.09755 & -0.49039 & -0.27779 & 0.27779 & 0.49039 & 0.09755 & -0.41573 \end{bmatrix}$$

$$\begin{bmatrix} 0.35355 & -0.35355 & -0.35355 & 0.35355 & 0.35355 & -0.35355 & -0.35355 & 0.35355 \\ 0.27779 & -0.49039 & 0.09755 & 0.41573 & -0.41573 & -0.09755 & 0.49039 & -0.27779 \\ 0.19134 & -0.46194 & 0.46194 & -0.19134 & -0.19134 & 0.46194 & -0.46194 & 0.19134 \\ 0.09755 & -0.27779 & 0.41573 & -0.49039 & 0.49039 & -0.41573 & 0.27779 & -0.09755 \end{bmatrix}$$

其转置矩阵

$$T' = \begin{bmatrix} 0.35355 & 0.49039 & 0.46194 & 0.41573 & 0.35355 & 0.27779 & 0.19134 & 0.09755 \\ 0.35355 & 0.41573 & 0.19134 & -0.09755 & -0.35355 & -0.49039 & -0.46194 & -0.27779 \\ 0.35355 & 0.27779 & -0.19134 & -0.49039 & -0.35355 & 0.09755 & 0.46194 & 0.41573 \\ 0.35355 & 0.09755 & -0.46194 & -0.27779 & 0.35355 & 0.41573 & -0.19134 & -0.49039 \\ 0.35355 & -0.09755 & -0.46194 & 0.27779 & 0.35355 & -0.41573 & -0.19134 & 0.49039 \\ 0.35355 & -0.27779 & -0.19134 & 0.49039 & -0.35355 & -0.09755 & 0.46194 & -0.41573 \\ 0.35355 & -0.41573 & 0.19134 & 0.09755 & -0.35355 & 0.49039 & -0.46194 & 0.27779 \\ 0.35355 & -0.49039 & 0.46194 & -0.41573 & 0.35355 & -0.27779 & 0.19134 & -0.09755 \end{bmatrix}$$

DCT可简化为T * B * T'，其中B为8*8的原矩阵。

实施二维DCT可将图像的能量集中在极少的几个系数之上，其他系数相比于这些系数，绝对值要小很多。这些系数大都集中在左上角，即低频分量区。

CSDN @岁月漫长_

左上角为直流direct current DC系数，即低频 平坦区域，其他系数为alternating current AC系数，即高频 边界纹理区域

这里是4.量化

1. 量化（破坏性的、不可逆的）
   把频率领域上每个成分，除以一个对于该成分的常量，接着舍位取最接近的整数。这是整个过程中的主要有损运算。
   这经常会把很多更高频率的成分舍位成0，且剩下很多会变成小的正或负数。

JPEG提供的量化算法如下

$$B_{i,j} = round(\frac{G_{i,j}}{Q_{i,j}})j, j = 0, 1, 2, \ldots, 7$$

其中G是我们需要处理的图像矩阵，Q称作量化系数矩阵，round函数是取整函数。JPEG算法提供了两张标准的量化系数矩阵，分别用于处理亮度数据Y和色差数据U以及V。

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

标准亮度量化表

$$Q_C = \begin{bmatrix} 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{bmatrix}$$

标准色差量化表

$$\begin{bmatrix} 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{bmatrix}$$

DCT系数矩阵中的不同位置的值代表了图像数据中不同频率的分量，这两张表中的数据时人们根据人眼对不不同频率的敏感程度的差别所积累下的经验测定的，

矩阵右下角值大（分母大），量化后得到的值小。

低频分量包含了图像的主要信息，而高频与之相比就不那么重要了，高频分量主要表示的是图像的细节信息，
所以可对高频分量进行处理，以此来减小数据量，达到数据压缩的目的。

5.无损压缩

量化后矩阵左上角(0,0)位置的值被称为直流分量DC，其他63个值被称为交流分量AC。
其中DC不参与Z字形扫描，而是与前一矩阵的DC系数进行差分编码(DPCM)；
AC分量则采用Z字形扫描排列并进行游程长度编码(RLE)。将其拼成一块后用Huffman编码，加上头文件

Huffman编码

JEPG压缩编码时，通过查表实现霍夫曼编码器，本次实验中我使用了ISO/IEC International Standard 10918-1中JPEG推荐的典型霍夫曼表(Typical Huffman tables)，我已经将4张表格发表在个人博客，篇幅所限这里就不再给出了。

之所以需要四张Huffman 编码表，是因为编码时每个矩阵数据的1个DC值与63个AC值分别使用不同的Huffman 编码表，而且亮度Y与色度U,V也要使用不同的Huffman 编码表。

为提高储存效率，JEPEG里并不直接保存数值，而是将数值按实际值所需要的位数分成16组，如下表所示

Value	Size	Bits
0	0	-
-1, 1	1	0, 1
-3, -2, 2, 3	2	00, 01, 10, 11
-7, -6, -5, -4, 4, 5, 6, 7	3	000, 001, 010, 011, 100, 101, 110, 111
-15, ..., -8, 8, ..., 15	4	0000, ..., 0111, 1000, ..., 1111
-31, ..., -16, 16, ..., 31	5	0 0000, ..., 0 1111, 1 0000, ..., 1 1111
-63, ..., -32, 32, ..., 63	6	00 0000, ..., ..., 11 1111
-127, ..., -64, 64, ..., 127	7	000 0000, ..., ..., 111 1111
-255, ..., -128, 128, ..., 255	8	0000 0000, ..., ..., 1111 1111
-511, ..., -256, 256, ..., 511	9	0 0000 0000, ..., ..., 1 1111 1111
-1023, ..., -512, 512, ..., 1023	10	00 0000 0000, ..., ..., 11 1111 1111
-2047, ..., -1024, 1024, ..., 2047	11	000 0000 0000, ..., ..., 111 1111 1111
-4095, ..., -2048, 2048, ..., 4095	12	0000 0000 0000, ..., ..., 1111 1111 1111
-8191, ..., -4096, 4096, ..., 8191	13	0 0000 0000 0000, ..., ..., 1 1111 1111 1111
-16383, ..., -8192, 8192, ..., 16383	14	00 0000 0000 0000, ..., ..., 11 1111 1111 1111
-32767, ..., -16348, 16348, ..., 32767	15	000 0000 0000 0000, ..., ..., 111 1111 1111 1111

CSDN @岁月漫长

设一个一维化后的亮度的数据块为

$$data = 5, 31, 45, 0, 0, 0, 0, 23, 0, -30, -8, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, \ldots, 0$$

其中第一个数字代表本数据块DC值与前一数据块DC值之差为5。

那么RLE压缩后的data变为

$$data' = (5); (0, 31); (0, 45); (4, 23); (1, -30); (0, -8); (2, 1); EOB$$

其中EOB = (0,0)。

$$变换函数 s = f(v), 其中 s \in Size, v \in Value, 且 s, v 位于表格的同一行$$

对data'中每个数对的第二个数v求对应的s值，并将s置于数对中v之前，得到data''

$$data'' = (3, 5); (0/5, 31); (0/6, 45); (4/5, 23); (1/5, -30); (0/4, -8); (2/1, 1); (0/0, 0)$$

其中第一个数对为DC分量运算得到的结果。

由于假设该以为数据由亮度数据块变换得来，因此对于DC和AC分量，分别对data''数对中的第一个数字查Huffman DC亮度表和Huffman AC亮度表，再对数对中的第二个数字查上表，即可得到一维数据块data编码后的序列。data''查表的结果为：

$$100, 101; 11010, 11111; 111000, 101101; 1111111110011000, 10111; 11111110110, 00001; 1011, 0111; 11100, 1; 1010$$

其中的标点是为了方便对照，实际编码中没有标点。

CSDN @岁月漫长_

一般将JPEG系数的Y分量作为隐写嵌入域（非0系数较多），DC系数是分块系数的平均值，一般只修改AC系数

2.3 基本嵌入方法

确定嵌入载体的类型和嵌入域。像素、JPEG系数等数域的样点存在最低意义比特位least significant bit，LSB

一般先对可用样点进行位置置乱，将消息扩散在嵌入域中，使数据局部性质趋于一致。

2.3.1 LSB替换

LSB replacement，LSBR：将样点的LSB用秘密消息替换

1 读取图像，将每个十进制的像素值转换成二进制

2 秘密消息转换成二进制

3 图像的LSB位替换成二进制的秘密消息

CSDN @岁月漫长_

创作打卡挑战赛
赢取流量/现金/CSDN周边激励大奖