

```

%{ /*符号常量定义
    LT, LE, EQ, NE, GT, GE,
    IF, THEN, ELSE, ID, NUMBER, RELOP */
}%

/* 正规定义 */
delim      [ \t\n]
ws         {delim}+
letter     [A-Za-z]
digit      [0-9]
id         {letter}({letter}|{digit})*
number     ({digit})+({digit})?(E(+|-)?({digit})+)?

%% //定义相应的动作和返回值
{ws}       {}
if         {return(IF);}
then       {return(THEN);}
{id}       {return(ID);}
{number}   {return(NUMBER);return(TYPE);}
"<="      {return(LE);}
">"      {return(GT);}
.....

```

图 6-75 一个 Lex 输入实例

#### (4) 实现提示

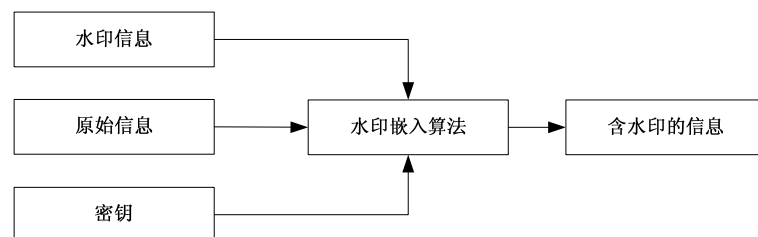
关键在于思考对状态转化图的处理。

### 47. 二值图像数字水印技术的实践

#### (1) 问题描述

随着计算机通信技术的迅速发展，多媒体存储和传输技术的进步使存储和传输数字化信息成为可能，然而，这也使盗版者能以低廉的成本复制及传播未经授权的数字产品内容。数字水印就是近年来为实现数字产权保护而产生的技术。数字水印是永久镶嵌在其它数据（宿主数据）中具有可鉴别性的数字信号或模式，而且并不影响宿主数据的可用性。作为数字水印技术基本上应当满足下面几个方面的要求：（1）安全性：数字水印的信息应是安全的，难以篡改或伪造；（2）隐蔽性：数字水印应是不可知觉的，而且应不影响被保护数据的正常使用；（3）稳健性：数字水印必须难以被除去，如果只知道部分数字水印信息，那么试图除去或破坏数字水印将导致严重降质或不可用。

数字水印技术是通过一定的算法将一些标志性信息直接嵌到多媒体内容中，水印的嵌入和提取方法如图 6-76 所示：



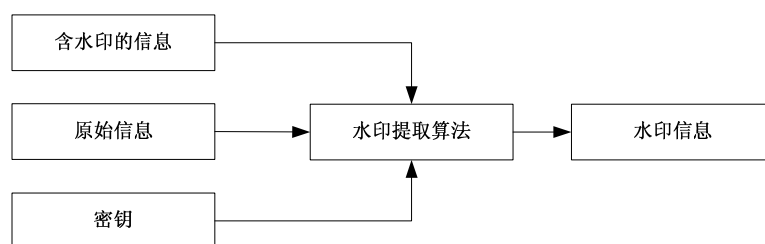


图 6-76 数字水印的嵌入和提取

数字水印的实现可以分为空间域数字水印和变换域数字水印两大类，较早的数字水印算法都是空间域上的，通过改变某些像素的灰度将要隐蔽的信息嵌入其中，将数字水印直接加载在数据上。空间域方法具有算法简单、速度快、容易实现的优点。特别是它几乎可以无损的恢复载体图象和水印信息，可细分为如下几种方法：最低有效位法，该方法就是利用原始数据的最低几位来隐蔽信息的，具体取多少位以人的听觉或视觉系统无法察觉为原则。Patchwork 方法及纹理映射编码方法，该方法是通过任意选择  $N$  对图象点，增加一点亮度的同时，降低相应另一点的亮度值来加载数字水印。文档结构微调方法，在通用文档图象（postscript）中隐藏特定二进制信息的技术，主要是通过垂直移动行距，水平调整字距，调整文字特性等来完成编码。

基于变换域的技术可以嵌入大量比特的数据而不会导致不可察觉的缺陷，往往通过改变频域的一些系数的值，采用类似扩频图象的技术来隐藏数字水印信息。这类技术一般基于常用的图象变换，如离散余弦变换（DCT）、小波变换（WT）、付氏变换（FT 或 FFT）以及哈达马变换（Hadamard Transform）等。频域方法具有如下优点：（1）在频域中嵌入的水印的信号能量可以分布到所有的像素上，有利于保证水印的不可见性；（2）在频域中可以利用人类视觉系统的某些特性，可以更方便、更有效的进行水印的编码。不过，频域变换和反变换过程中是有损的，同时其运算量也很大，对一些精确或快速应用的场合不太适合。目前常用的方法有平面隐藏法和基于 DCT 或 DFT 的系数隐藏法。其中基于分块的 DCT 是最常用的变换之一，现在所采用的静止图像压缩标准 JPEG 也是基于分块 DCT 的。

下面概要描述一种简单的二值图像的数字水印算法，本设计的基本目标就是该水印算法的编程实现。

二值图像又称为单色图像或黑白图像，一般用 1 或 0 表示黑色或白色像素的颜色值。二值图像数字水印的常见方法是：一、使用一个特定图像区域中的黑色像素个数来编码水印信息。把一个二值图像分解成一系列图像块  $B_i$ ，分别令  $P_0(B_i)$  和  $P_1(B_i)$  代表黑白像素在该块图像中所占的百分比。基本做法是判断如果  $P_1(B_i) > 50\%$ ，则在该块中隐藏一个 1，如果  $P_0(B_i) > 50\%$ ，则在该块中隐藏一个 0。隐藏是需要修改图像块中的一些像素的颜色。该修改是在那些邻近像素有相反的颜色像素中进行的。二、将二值图像进行分块，使用一个与图像块同样大小的密钥二值图像块，该密钥图像块和每一格图像块按像素进行“与”运算，根据“与”的结果确定是否在该块中隐藏数据，并隐藏怎样的数据。但这两种简单处理方法会导致图像质量下降，如会在一个全白的图像块中插入一个黑点，如也应避免在图像中的细线（一个像素宽）、直线边的中间像素、孤立像素等区域隐藏像素。所以二值图像的数字水印嵌入算法的关键是隐藏点的选取。

二值图像的数据隐藏应该着眼于图像黑(或白)色区域的边界上。下面给出一种筛选隐藏点的方法，该方法的基本思想是在满足下面条件的位置上不隐藏数据：

该像素既是其所在区域的左边界，又是其右边界（实际上是一条直线）；

该像素既是其所在区域的上边界，又是其下边界；

该像素只是左边界、右边界、上边界、下边界 4 种边界情况中的一种；

该像素的周围 8 个像素中与该像素同色的所有像素都是既是左边界或右边界，同时又

是其上边界和下边界(实际上是一个小点)。此处给出一个算法, 设二值图像为  $I$ , 其像素矩阵为  $A_{M \times N}$ , 数字水印为  $W$ , 是一个长度为  $L$  的二进制比特流  $W=\{w_1, w_2, \dots, w_L\}$ 。设  $A(i:j, s:t)$  为矩阵  $A$  的从第  $i$  行到第  $j$  行, 从第  $s$  列到第  $t$  列的子矩阵。数据隐藏算法如下:

第 1 步: 计算图像边界。

$$A_L = A(0: (M-1), 0: (N-2)) - A(0: (M-1), 1: (N-1))$$

$$A_R = A(0: (M-1), 1: (N-1)) - A(0: (M-1), 0: (N-2))$$

$$A_T = A(0: (M-2), 0: (N-1)) - A(1: (M-1), 0: (N-1))$$

$$A_B = A(1: (M-1), 0: (N-1)) - A(0: (M-2), 0: (N-1))$$

将  $A_L, A_R, A_T, A_B$  中值为 -1 的元素置为 0, 得到  $B_L, B_R, B_T, B_B$ 。则  $B_L, B_R, B_T, B_B$  为图像  $I$  的左、右、上、下边界矩阵。

第 2 步: 选择隐藏点。

$$B = B_L + B_R + B_T + B_B$$

$$B_{LR} = B_L + B_R$$

$$B_{TB} = B_T + B_B$$

矩阵  $B_{LR}$  和  $B_{TB}$  的元素值可能为 0, 1, 2。值为 2 表示该像素同时为左右或上下边界。矩阵  $B$  中元素的可能值为 0, 1, 2, 3, 4, 值为 3 和 4 的一定是左右或上下边界, 值为 2 的像素可能会同时为左右边界或上下边界, 此时需根据  $B_{LR}$  和  $B_{TB}$  的值加以区分。

计算  $B' = (b_{ij}')$ , 其中  $b_{ij}' = 0$ , 如果  $b_{ij}=3, 4$ ;  $b_{ij}' = 0$ , 如果  $b_{ij}=2$  且  $b_{ij}^{LR}=2$  或  $b_{ij}^{TB}=2$ ;  $b_{ij}' = 0$ , 如果  $i=0$  或  $j=0$ ;  $b_{ij}' = 1$ , 如果  $b_{ij}=2$  且  $b_{ij}^{LR} \neq 2$  或  $b_{ij}^{TB} \neq 2$ 。

再计算  $B'' = (b_{ij}'')$ , 其中  $b_{ij}''=0$ , 如果  $b_{ij}'=0$ ;  $b_{ij}''=0$ , 如果  $b_{ij}'=1$  且  $\sum_{u=i-1}^{i+1} \sum_{v=j-1}^{j+1} a_{uv} = \sum_{u=i-1}^{i+1} \sum_{v=j-1}^{j+1} b'_{uv}$ ;  $b_{ij}''=1$ , 其他情况。

第 3 步: 水印数据处理。为确保水印数据的安全, 输入一个密钥  $K$ , 根据该密钥产生一个长度为  $L$  的伪随机比特流  $R=\{r_1, r_2, \dots, r_L\}$  ( $R$  的产生方法是应用一个高级语言提供的伪随机函数, 如 C 语言的 `rand` 函数, 将  $K$  作为该函数的种子产生随机数, 可产生 0-1 的随机数, 如果该数小于等于 0.5 就输出 0, 否则输出 1。也可以用其他方法获得。) 将  $W$  与  $R$  按位异或就得到了  $W'=\{r'_1, r'_2, \dots, r'_L\}$ 。

第 4 步: 数据隐藏。设计一个遍历图像矩阵  $A$  诸元素的遍历算法, 根据  $B''$  和  $W'$  计算获得  $A'$ 。  $A' = (a_{ij}')$ ,  $a_{ij}' = a_{ij}$ , 如果  $b_{ij}'' = 0$ ;  $a_{ij}' = w'_l$ , 如果  $b_{ij}'' = 1$ , 其中  $l=M \times i+j$ 。  $A'$  对应的就是含有水印信息的新图像  $I_W$ 。

第 5 步: 水印数据的提取。第 1 步, 第 2 步同上, 在  $A$ (图像  $I$ ) 上可以计算出矩阵  $B''$ , 此时可以获得隐藏在  $A'$ (图像  $I_W$ ) 中的数据  $S$ , 注意需按照同样的遍历方法, 然后根据  $K$  获得同样的伪随机序列  $R$ ,  $R$  和  $S$  按位异或后就是数字水印。

## (2) 课程设计目的

对数字水印技术建立一定的认识, 能建立位矩阵、位向量等 ADT, 并能用这些 ADT 完成上述二值图像数字水印的嵌入和抽取。

## (3) 基本要求

- ① 设计并实现位矩阵、位向量等 ADT, 要求支持 +、-、与、或、异或等基本操作。
- ② 针对二值图像实现上述水印嵌入和提取算法。
- ③ 针对二值图像实现另一个水印基本做法: 判断如果  $P_1(B_i) > 50\%$ , 则在该块中隐藏一个 1, 如果  $P_0(B_i) > 50\%$ , 则在该块中隐藏一个 0。也可以自行设计一个嵌入方式。
- ④ 针对一幅二值图像(要求是 BMP 文件格式), 分别用上面的两种方法嵌入水印, 查看嵌入水印后的图像  $I_W$  的区别, 水印信息可以设定为一段文字。

## (4) 实现提示

可以查阅关于数字水印方面的相关资料。