针对信息隐藏技术进行举例说明。

1. 空间域信息隐藏方法
2. 变换域信息隐藏方法
3. 二值图像信息隐藏方法
4. 音频信息隐藏方法
5. 统计特征信息隐藏方法
6. 载体生成信息隐藏方法

# 二值图像信息隐藏方法

## Koch-Zhao 方法

### 基本方法

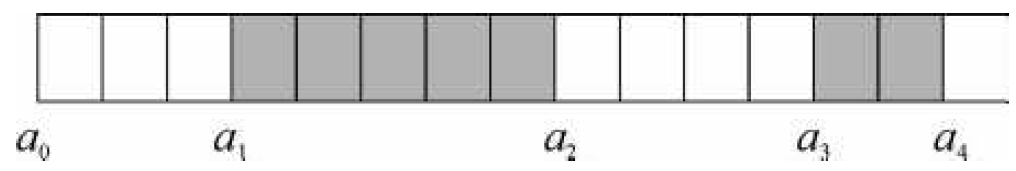
* 首先，把一个二值图像分成 个矩形图像区域
* 如果某个区域中黑色像素的个数大于一半，则嵌入一个 0；如果白色像素的个数大于一半，则嵌入一个 1
* 当需要嵌入的比特与所选区域的黑白像素的比例不一致时，则需要修改一些像素的颜色。修改应该遵循一定的规则，而这些规则都应该以不引起感观察觉为目的。通常，应该对黑白像素的边缘进行修改

### 一些细节

* **为了提高隐藏系统对传输错误和图像修改的鲁棒性﹐选择图像块时，应该考虑有一定的冗余度。**如果由于传输的误差导致某些像素改变了颜色，使得原来大于 的像素颜色降到 以下，那么所提取出的隐藏信息正好相反，也就是发生了错误。因此，要避免这样的情况发生，保证系统具有一定的稳健性，需要选择有效的图像块。要确定两个阈值 和 ，以及一个鲁棒性参数 ，是传输过程中可能被改变的像素百分比。发送者在嵌入过程中，应确保隐藏 0 时，该块的黑色像素的个数在修改后应属于 的范围;要嵌入数据 1，该块的黑色像素的个数在修改后应属于 的范围。
* **如果为了适应所嵌入的比特，目标块必须修改太多的像素，就把该块设为无效，再选择下一块进行嵌入。**比如，当前块有 的黑色像素，要隐藏比特 1，需要把将 以上的黑色像素改为白色像素，对图像的影响太大，遇到这样的情况，就放弃这块，设为无效。但是，发送端设为无效的块，如何让接收端知道呢?发送端将无效块中的像素进行少量的修改，使得其中黑色像素的百分比大于 ，或者小于 ，用这样的方法标识无效块。
* 这里 ， 和 需要根据实验给出经验值，对于不同类型的二值图像，有不同的选择。

## 游程编码方法

在二值图像中，连续像素具有同种颜色的概率很高。因此，对每一行，不再直接地对每一个位置具有什么像素进行编码，**而是对颜色变化的位置和从该位置开始的连续同种颜色的个数进行编码**。因此，对于图中的例子，可以得到编码如下: 。



在这样的编码中仍然考虑在最低比特位的隐藏。现在修改二值图像的游程长度，如果第 个秘密信息位 是 0，则令该游程长度为偶数；如果 是 1，则修改游程长度为奇数。

* 如果 是 0，而对应的游程长度是奇数，则把长度个数加 如
* 如果 并且长度是偶数,则把长度减
* 如果 的取值与长度个数的奇偶性相匹配,则不改变游程长度

# 音频信息隐藏

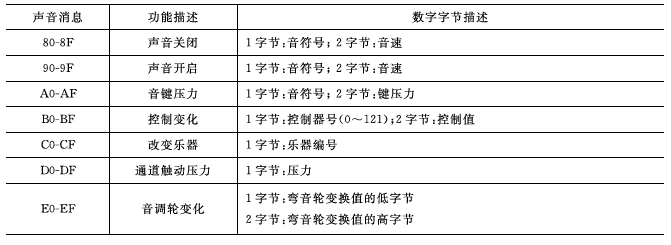
## 算法

* LSB算法
* 回声隐藏算法
* 变换域算法
* ……

## 例子

### MIDI

声音消息:



根据大量实验，改变**声音开启**的最低位比特、**乐器编号**的最低位比特和**通道触动压力**的低4比特位，都不会引起听觉差异，因此可在这三种声音消息中嵌入水印信息。

隐藏：

clear;   
clc;   
hidefid = fopen('隐藏.txt', 'r');   
[msg, count] = fread(hidefid, inf, 'ubit1');   
fid = fopen('待隐藏.mid', 'r');   
[wa, length] = fread(fid, inf, 'uint8');   
fclose(hidefid);   
fclose(fid);   
modcount = 1;   
p = 1;   
   
for i = 1:length   
 if bitand(wa(i, 1), 240) == 144 && p <= count %声音开启   
 wa(i + 2, 1) = bitset(wa(i + 2, 1), 1, msg(p, 1));   
 p = p + 1;   
 i = i + 3;   
 modcount = modcount + 1;   
 elseif bitand(wa(i, 1), 240) == 192 && p <= count %乐器编号   
 wa(i + 1, 1) = bitset(wa(i + 1, 1), 1, msg(p, 1));   
 p = p + 1;   
 i = i + 2;   
 modcount = modcount + 1;   
 elseif bitand(wa(i, 1), 240) == 208 && p <= count - 3 %通道触动压力   
 wa(i + 1, 1) = bitset(wa(i + 1, 1), 1, msg(p, 1));   
 wa(i + 1, 1) = bitset(wa(i + 1, 1), 2, msg(p + 1, 1));   
 wa(i + 1, 1) = bitset(wa(i + 1, 1), 3, msg(p + 2, 1));   
 wa(i + 1, 1) = bitset(wa(i + 1, 1), 4, msg(p + 3, 1));   
 p = p + 4;   
 i = i + 2;   
 modcount = modcount + 1;   
 else   
 i = i + 1;   
 end   
   
end   
   
fid = fopen('隐藏.mid', 'a');   
fwrite(fid, wa);   
fclose(fid);

提取：

clear;   
clc;   
fid=fopen('隐藏.mid','r');   
[wa,length]=fread(fid,inf,'uint8');   
getfid=fopen('提取.txt','a');   
fclose(fid);   
count=72;   
p=1;   
for i=1:length   
 if(bitand(wa(i,1),240)==144)&&(p<=count)   
 fwrite(getfid,bitget(wa(i+2,1),1),'ubit1');   
 i=i+3;   
 if p==count   
 break;   
 end   
 p=p+1;   
 elseif(bitand(wa(i,1),240)==192)&&(p<=count)   
 fwrite(getfid,bitget(wa(i+1,1),1),'ubit1');   
 i=i+2;   
 if p==count   
 break;   
 end   
 p=p+1;   
 elseif(bitand(wa(i,1),240)==208)&&(p<=count-3)   
 fwrite(getfid,bitget(wa(i+1,1),1),'ubit1');   
 fwrite(getfid,bitget(wa(i+1,1),2),'ubit1');   
 fwrite(getfid,bitget(wa(i+1,1),3),'ubit1');   
 fwrite(getfid,bitget(wa(i+1,1),4),'ubit1');   
 i=i+2;   
 if p==count   
 break;   
 end   
 p=p+4;   
 end   
end   
fclose(getfid);