

【语音隐写】基于 LSB语音隐藏matlab源码

原创

Matlab科研辅导帮  于 2021-08-31 00:15:16 发布  123  收藏 1

分类专栏: [信号处理](#) 文章标签: [matlab](#)

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/m0_60703264/article/details/119994884

版权



[信号处理](#) 专栏收录该内容

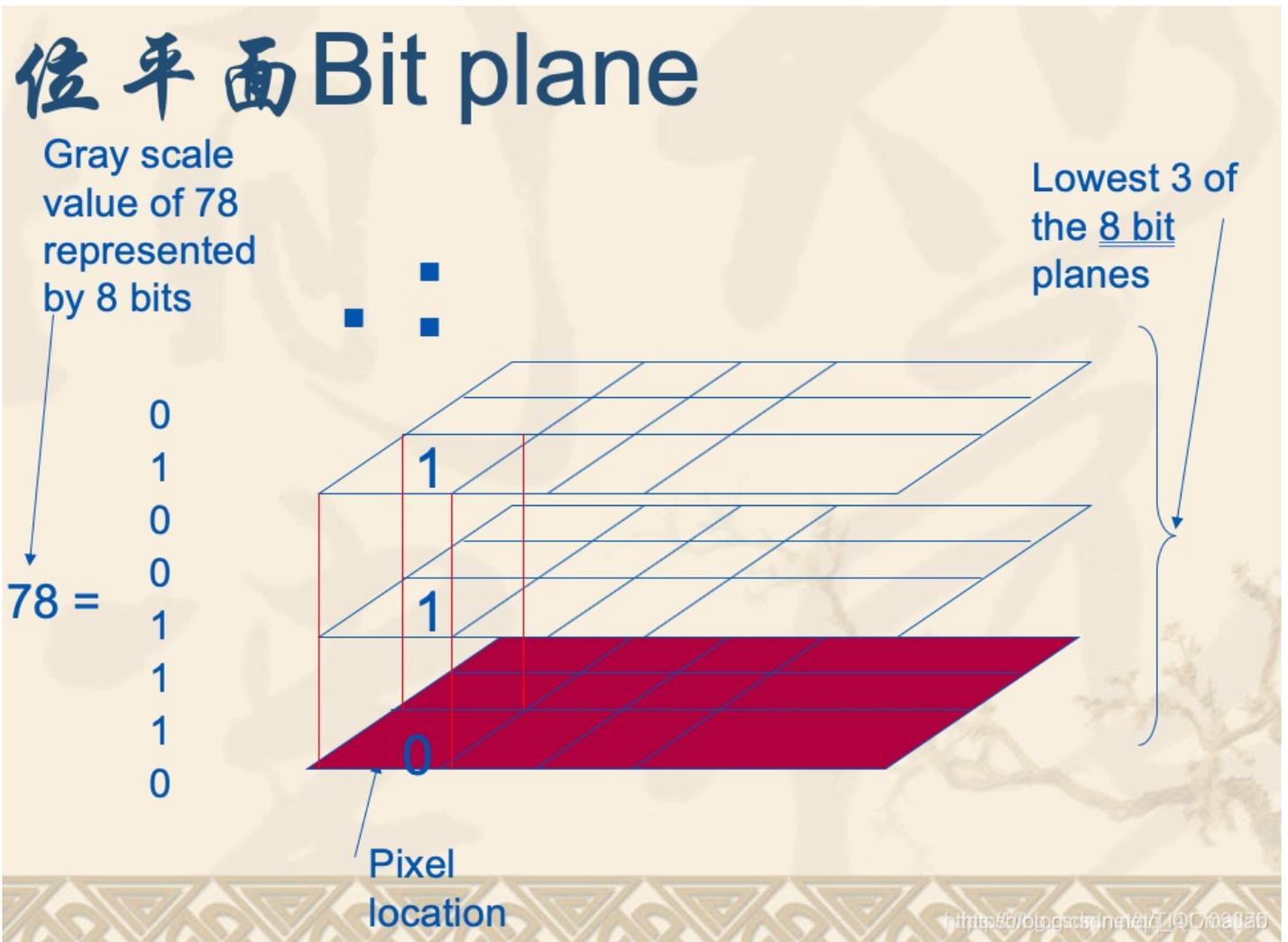
40 篇文章 6 订阅

订阅专栏

一、简介

1 LSB算法简介

LSB全称为 Least Significant Bit（最低有效位），是一种简单而有效的数据隐藏技术。LSB隐写的基本方法是用欲嵌入的秘密信息取载体图像的最低比特位，原来的图像的高位平面与代表秘密信息的最低平面组成含隐蔽信息的新图像。



灰度化的图像为单通道格式存储像素，每个像素值在0~255内，而像素的位平面则是对应二进制的像素的各个位。以上图为例，某个像素的值为78，其二进制01001110，从左到右位权依次降低，最左边为最高有效位（MSB，其位权为 2^7 ）

7），最右边位最低有效位（LSB，位权为 2^0 ）

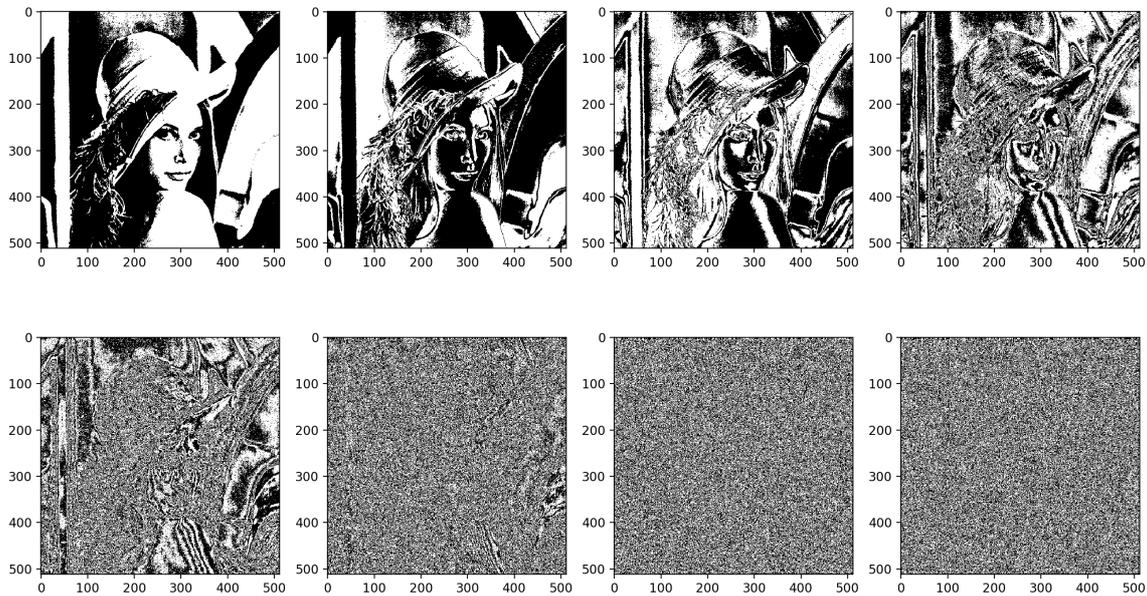
0）。把每个像素的相同位抽取出来组成一个新的平面，就是所谓的图的位平面。而LSB隐写算法，如其名字，是在LSB也就是最低位平面进行信息嵌入/隐藏。

需要注意的一点是，LSB嵌入的时候，载体图像格式应该为灰度图格式

以著名的Lena图为例，一下是灰度图Lena原图：



下面是其各个位平面图，从左到右、从上到下位平面依次降低：



可以看到，位平面越高包含的原图像信息越多，对图像的灰度值贡献越大，并且相邻比特的相关性也越强，反之则相反。LSB最低位平面基本上不包含图像信息了，类似随机的噪点/噪声，因此，可以在此处填入水印/秘密信息。

嵌入示意图如下：

| Original Image Bytes | Message to hide | Embedded Image Bytes |
|----------------------|-----------------|----------------------|
| 10010010 | 0 | 10010010 |
| 01010011 | 1 | 01010011 |

| | | | | |
|----------|---|---|---|----------|
| 10011011 | | 0 | | 10011010 |
| 11010011 | → | 0 | → | 11010010 |
| 10001010 | | 0 | | 10001010 |
| 00000010 | | 1 | | 00000011 |
| 01110010 | | 0 | | 01110010 |
| 00101011 | | 0 | | 00101010 |

选取不同位平面嵌入时，LSB算法的保真度：

LSB算法的保真度

$$PSNR = -10 \log_{10} \left(\frac{1}{255^2 MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} [d(m,n)]^2 \right)$$

- 最低1位

1/2, MN/2, PSNR=-10log₁₀(1/255/255/2)=51.14dB

- 最低2位

3MN/2, PSNR=46.37

- 最低3位

7MN/2, PSNR=42.69

- 最低4位, PSNR=39.38; 最低5位, PSNR=36.23.

2 算法原理

通俗来讲我们看到的图片都是由一个个小的像素点来构成的，所有像素点摆在一起，构成一个大方块，这个大方块就是我们所见的图像。灰度图像（也就是我们平时所说的黑白图像）是由一层像素点组成的，而彩色图像是由三层这样的灰度图像组成的。这里拿灰度图像举例，我们之所以能在图像上看到黑色和白色，是因为每个像素点的像素值不同。0表示纯黑，255表示纯白，灰色就是由这两个数字之间的值构成。越靠近0越黑，越靠近255越白。那为什么是0和255呢？因为计算机是二进制，它会用8个比特来表示一个像素点（也可以用更多的比特，这样图像的颜色分级就越多，同时图像也会占用更大的空间，但是普通人的眼睛并不能辨认这么多的颜色，除非你异于常人），所以最大值是255，最小是0。lsb就是基于2进制这一特点来隐藏信息的，因为人眼并不是很精密的颜色或亮度的感知器，所以把像素灰度上下微调1是不会被人眼察觉的，也就是修改8位二进制码中最小的一位。当我们把图片每个像素的最后一位按照我们的想法改变，使他表现为我们想要的信息，但用户却不能看出，也不会影响图片的内容。这就是lsb数字水印。

3 LSB算法的基本特点：

LSB是一种大容量的数据隐藏算法

LSB的鲁棒性相对较差（当stego图像遇到信号处理，比如：加噪声，有损压缩等，在提取嵌入信息时会丢失）

4 常见LSB算法的嵌入方法：

秘密信息在最低位平面连续嵌入至结束，余下部分不作任何处理（典型软件MandelSteg）

秘密信息在最低位平面连续嵌入至结束，余下部分随机化处理（也称沙化处理，典型软件PGMStealth）

秘密信息在最低位平面和次低位平面连续嵌入，并且是同时嵌入最低位平面和次低位平面

秘密信息在最低位平面嵌入，等最低位平面嵌入完全嵌入之后，再嵌入次低位平面

秘密信息在最低位平面嵌入，夺取最低位平面嵌入元素嵌入之后，再嵌入次最低位平面

秘密信息在最低位平面随机嵌入

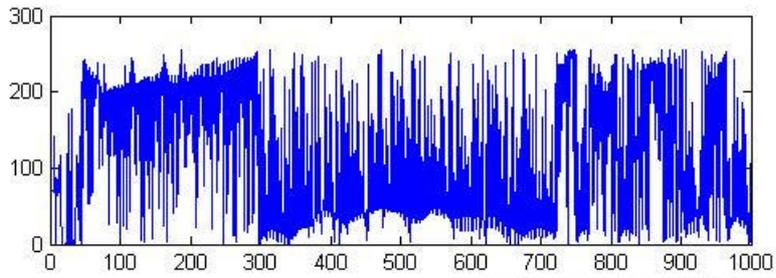
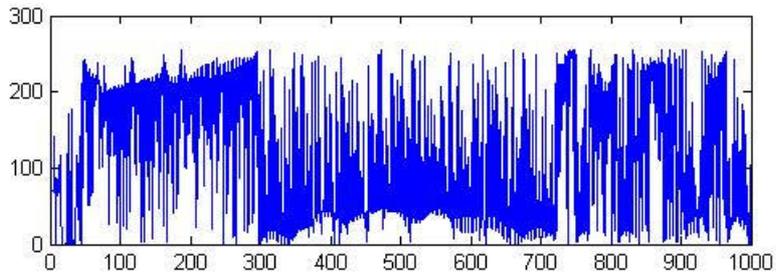
以上五种方式，当嵌入容量不同时，鲁棒性不同

二、源代码

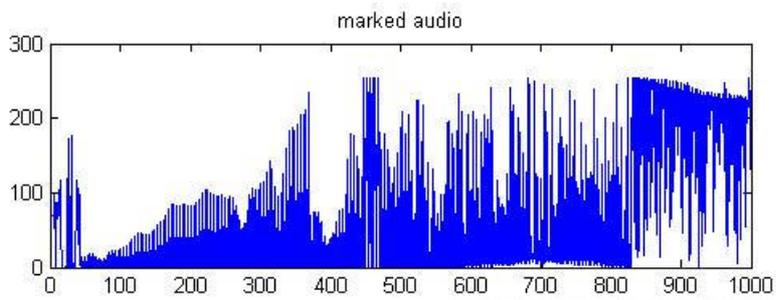
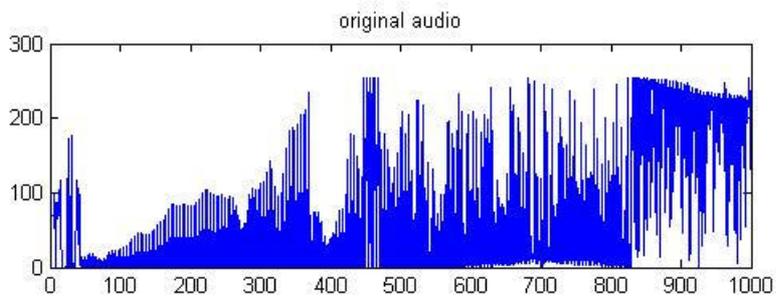
```
clc;
clear all;
fid=fopen('1.wav','r');%读入音频文件
oa=fread(fid,inf,'uint8');
fclose(fid);
n=10;%n为a的长度，为10
d=randsrc(1,n,[0,1]);%产生随机数列即水印
M=oa;
for i=45:45+n-1
    M(i)=bitset(M(i),1,d(i-44));%进行LSB嵌入水印
end
function A=Vector2Matrix(oi,row)
%将数组转换成二维数组,row为行数
cl=length(oi)/row;
A=zeros(row,cl);
for i=1:row
    for j=1:cl

        end
    end
end
marklength=5394;%提取的水印图像大小，已知
row=58;           %row是水印图像行数，已知
fid=fopen('marked.wav','r');
oa=fread(fid,inf,'uint8');
status=fseek(fid,44,'bof');
a=fread(fid,marklength,'uint8');
for i=1:marklength
    w1(i)=bitget(a(i),1);
end
w1=w1';
m=Vector2Matrix(w1,row);
imwrite(m,'markedbupt.bmp','bmp');
figure;
imshow('markedbupt.bmp');title('extracted watermark');
fid=fopen('2.wav','rb');
oa=fread(fid,inf,'uchar');
n=length(oa)-44;%wav文件从文件头到数据前面共有44字节的格式说明
fclose(fid);
io=imread('bupt.bmp');
[row,col]=size(io);
io=io';
wi=io(:);
if row*col>n
    error('载体太小，请更换载体');
end
markedaudio=oa;
marklength=row*col;
fprintf('嵌入图像的长度为: %.0d\n',marklength);
for k=1:row*col
    markedaudio(44+k)=bitset(markedaudio(44+k),1,wi(k));
end
```

三、运行结果



<https://blog.csdn.net/TIQCmatlab>



<https://blog.csdn.net/TIQCmatlab>



Matlab科研助手

擅长路径规划、车间调度、优化求解、神经网络预测、信号处理、元胞自动机、图像处理等多种领域的Matlab仿真



扫一扫二维码，加我QQ好友。

©2016-2020 matlab-helper.com 02711472