

LSB隐写算法的实现（matlab源码和界面）

原创

[Diammmm](#) 于 2021-03-19 16:23:53 发布 2956 收藏 51

分类专栏: [信息隐藏](#) 文章标签: [算法](#) [matlab](#) [信息安全](#) [图像处理](#)

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/diamonds_cz/article/details/115009414

版权



[信息隐藏](#) 专栏收录该内容

1 篇文章 0 订阅

订阅专栏

LSB隐写算法的实现（matlab源码和界面）

[Matlab的GUI界面效果展示](#)

[放matlab源代码](#)

[GUI界面的源码](#)

[LSB算法原理分析](#)

Matlab的GUI界面效果展示

嵌入水印后的宿主图像

宿主图像



加载宿主图片

水印图像



加载水印图片

二值化水印图片

嵌入水印后的宿主图像



嵌入水印图片

提取水印图片

https://blog.csdn.net/diamonds_cz

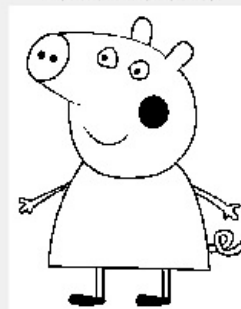
提取到的水印图像

宿主图像



加载宿主图片

二值化后的水印图像



加载水印图片

二值化水印图片

提取到的水印图像



嵌入水印图片

提取水印图片

https://blog.csdn.net/diamonds_cz

放matlab源代码

主函数部分 `main.m`.

```

// author:Lucifer diamond
clc;%清除命令行窗口
close all;%关闭所有打开的窗口
clear all;%清除工作空间

%%获取载体图像并将其转化为灰度图
CarrierImg = rgb2gray(imread('lena.bmp'));
% 获取载体图像的行M和列N
[M, N] = size(CarrierImg);
% 显示载体图像
figure, imshow(CarrierImg);
title('载体图像');

%%获取要隐藏的水印图像并将其转化为灰度图
BinImg = rgb2gray(imread('SmallPig.bmp'));
% 获取水印图像的行m和列m
[m, n] = size(BinImg);
% 比较载体图像和水印图像, 若水印图像的行和列均小于载体图像, 不做处理
% 反之, 若有水印图像的行或列大于载体图像, 则将水印图像调整和载体图像一样大来嵌入
if m>M || n>N
    BinImg = imresize(BinImg,[M,N]);
end
% 将水印图像进行二值化处理, 先转化为double, 阈值根据水印图片自行调节
% imbinarize(BinImg,0.90)以0.90为阈值将BinImg转化为二值图像
% 大于0.90的变成逻辑1(白色), 小于0.90的变成逻辑0(黑色)
BinImg = im2double(BinImg);
BinImg = imbinarize(BinImg,0.90);
% 显示处理后的水印图像
figure, imshow(BinImg);
title('需要被隐藏二值化图像');

%% 嵌入水印
WatermarketImg = LSB_Encode(CarrierImg,BinImg,M,N,m,n);
figure,imshow(WatermarketImg);
imwrite(WatermarketImg,'WatermarketImg.bmp');

%% 提取水印
% Image = imread('WatermarketImg.bmp');
% BinImage = LSB_Decode(Image,m,n);
% imshow(BinImage);

```

LSB嵌入算法 `LSB_Encode.m` .

```

// author:Lucifer diamond
%% LSB算法实现嵌入图片水印
function CarrierImg = LSB_Encode(CarrierImg,BinImg,M,N,m,n)
% CarrierImg为载体图像，BinImg为二值化后的要隐藏的水印图像
% M为载体图像的行,N为载体图像的列,m为水印图像的行，n为水印图像的列

% bitget(CarrierImg(i,j),1)获取图像CarrierImg中一个像素点的亮度值
% 将该亮度值用二进制表示，bitget(CarrierImg(i,j),1)中的1表示获取最低位的值

% 注意!!! 水印图像的m*n 要小于等于载体图像的 M*N
% 即m<=M,n<=N
if (m <= M && n <= N)
    for i = 1:m
        for j = 1:n
            if BinImg(i,j) == bitget(CarrierImg(i,j),1)
                continue;
            elseif BinImg(i,j) == 0 && bitget(CarrierImg(i,j),1) ==1
                CarrierImg(i,j) = CarrierImg(i,j)-1;
            elseif BinImg(i,j) == 1 && bitget(CarrierImg(i,j),1) ==0
                CarrierImg(i,j) = CarrierImg(i,j)+1;
            end
        end
    end
else
    fprintf('BinImg is too big than CarrierImg!!!')
end
end

```

提取水印算法 [LSB_Dncode.m](#) .

```

// author:Lucifer diamond
%% 提取LSB算法嵌入的水印
function OutputImage = LSB_Decode(InputImage,m,n)
% InputImage为含有水印的图像，m为要提取水印的行，n为要提取水印的列
% zeros(m,n)生成一个m*n的全0矩阵
% bitget(InputImage(i,j),1)获取图像InputImage中一个像素点的亮度值
% 将该亮度值用二进制表示，bitget(InputImage(i,j),1)中的1表示获取最低位的值
OutputImage = zeros(m,n);
for i = 1:m
    for j = 1:n
        if bitget(InputImage(i,j),1) == 1
            OutputImage(i,j) = 255;
        else
            continue;
        end
    end
end
end
end

```

GUI界面的源码

点我头像查看上传的资源

LSB算法原理分析

LSB (Least Significant Bit) 是信息隐藏领域中实现隐写术的一种最基础的隐写算法，LSB属于空域算法的一个分支，主要原理是利用人眼的视觉界限，无法识别出一张图片细微改变，从而将信息隐藏在图像的最低位，达到信息隐藏的目的。优缺点就不细说了，可以自行百度了解。本文源代码是在灰度图层面上实现的信息隐藏，即只用了一个维度隐藏信息。对于一张RGB图片，有三个维度可以隐藏信息，实现原理相同；