

一种简单的图像隐写

简单介绍一种将信息隐藏在图像像素中的方法。主要步骤如下

1、隐藏信息：

首先找到一张图片，将其某一通道的像素值[如 R]全部取偶数；

将想要隐藏的信息写在一张小于等于这张图片的图片上；

通过读取写有隐藏信息的图片的像素值，将载体图片同等坐标的 R 像素值改成奇数。

2、信息提取：

创建一张空白图片：遍历有隐藏信息的图片，寻找 R 值为奇数的坐标；

在空白图片对应的坐标设置一种颜色；

输出创建的图片。

其实看完摘要，基本上原理就已经很明白了，就是通过一个全偶数的“刻板”，将刻字内容刻在上面就是把对应的地方改成奇数。提取的时候把奇数部分“翻印”出来就行了。下面看一下实现。

step1:找到一张图片[BMP 格式]，并将其所有的 R 通道值改成偶数

为什么非得是 bmp？我昨天也被图片格式给整了一把，简单来说，在我们比较熟知的集中图片格式中，如 png、jpeg、bmp，只有 bmp 是无压缩的，我们可以精准的控制像素值。jpg 图像由于受到的压缩的影响，像素值有 1-3 的波动，这还怎么搞啊。。。昨天我用 PIL 搞了一下午都在奇怪为什么图片保存后，像素值不是我设定好的。怒了就用 C 读取 bmp 图片试了一下成功了（纯

c 就能很简单的解析 bmp), 发现可以正常实现 , 就想到了图片格式这个梗。

对于图片格式的细节我不想去细细研究 , 我就想得到图片的数据矩阵 , 改变矩阵值。所以就用 bmp 图片了。

伪代码如下 :

```
for i in range(w):
    for j in range(h):
        if img[i,j,R] %2 != 0:
            img[i,j,R] = img[i,j,R]+1 if img[i,j,R] <255
            else img[i,j,R]-1
```

step2:在意个白色的图片上写上黑色的文字[同样是 bmp]

step3 : 通过读取写有隐藏信息的图片的像素值 , 将载体图片同等坐标的 R 像素值改成奇数

```
for i in range(w):
    for j in range(h):
        if code[i,j,R] == 0 and code[i,j,G] == 0 and code[i,j,B] == 0:
            img[i,j,R] += 1
```

解码 :

遍历有隐藏信息的图片 , 寻找 R 值为奇数的坐标 ; 在空白图片对应的坐标设置一种颜色

```
for i in range(w):
    for j in range(h):
        if img[i,j,R] %2 != 0:
```

```
out[i,j,R]=255  
out[i,j,G]=255  
out[i,j,B]=255
```

完整的代码 (Python+OpenCV2 , 其实就是用了一下读取、更改像素的功能)

隐藏信息 :

```
#coding:utf-8  
import cv2  
import numpy as np  
# import random  
img = cv2.imread("lena.bmp")  
code = cv2.imread("code.bmp")  
w,h = img.shape[:2]  
# print w,h  
for i in range(w):  
    for j in range(h):  
        if img[i,j,2] %2 != 0:  
            img[i,j,2] = img[i,j,2]+1 if img[i,j,2] <2 else img[i,j,2]-1  
        # cv2.imwrite("quanou.bmp",img)  
    for i in range(w):  
        for j in range(h):  
            if code[i,j,0] == 0 and code[i,j,1] == 0 and code[i,j,2] == 0:  
                img[i,j,2] += 1  
cv2.imwrite("final.bmp",img)
```

提取信息：

```
#coding:utf-8

import cv2

import numpy as np

img = cv2.imread("final.bmp")

out = np.zeros(img.shape,np.uint8)

w,h = img.shape[:2]

for i in range(w):
    for j in range(h):
        if img[i,j,2] %2 != 0:
            out[i,j,0]=255
            out[i,j,1]=255
            out[i,j,2]=255

cv2.imwrite("out.bmp",out)
```