信息隐藏与数字水印实验：图片类隐写（MATLAB）

原创

Hardworking666 于 2021-12-03 22:29:18 发布 3302 收藏 6

分类专栏： MATLAB隐写 文章标签： 图像处理 matlab 计算机视觉 bmp 安全

本文链接：https://blog.csdn.net/Hardworking666/article/details/121705292

MATLAB隐写 专栏收录该内容

3 篇文章 0 订阅

订阅专栏

文章目录

一、在BMP图像中隐藏数据

1、在图像文件**尾部**添加任意长度的数据，仅需修改文件头中**文件长度的值**即可。
2、如果将秘密数据放在**文件头与图像数据**之间，则至少需要修改文件头中**文件长度、数据起始偏移地址**这两个域的值。
3、修改文件头和信息头中**保留字段**隐藏信息。
4、在图像像素区利用图像宽度字节必须是4的倍数额特点，在**补足位**处隐藏数据。

表 7-1-1　bmp 图像文件头和图像信息表

BMP 文件头（14 字节）	0000—0001（2 字节）	文件标识，为字母 ASCII 码“BM”
	0002—0005（4 字节）	文件大小，高位高字节
	0006—0009（4 字节）	保留字，每字节以“00”填写
	000A—000D（4 字节）	记录图像数据区的起始位置，为 36H
图像信息头（共 40 字节）	000E—0011（4 字节）	图像描述信息块大小，常为 28H
	0012—0015（4 字节）	图像宽度
	0016—0019（4 字节）	图像高度
	001A—001B（2 字节）	图像的位面数，该值总为 1
	001C—001D（2 字节）	记录像素的位数，图像的颜色数由该值决定
	001E—0021（4 字节）	数据压缩方式（0：不压缩；1：8 位压缩；2：4 位压缩）
	0022—0025（4 字节）	用字节数表示的图像数据的大小，该数必须是 4 的倍数，数值上等于图像宽度×图像高度×每个像素位数
	0026—0029（4 字节）	水平每米有多少像素，在设备无关位图（.DIB）中，每字节以 00H 填写
	002A—002D（4 字节）	垂直每米有多少像素，在设备无关位置（.DIB）中，每字节以 00H 填写
	002E—0031（4 字节）	图像所用的颜色数
	0032—0035（4 字节）	对图像显示有重要影响的颜色索引的数目。如果是 0，表示都很重要
颜色表（非必有）	颜色表的大小根据所使用的颜色模式而定：2 色图像为 8 字节；16 色图像位 64 字节；256 色图像为 1024 字节；24 位真彩色图像则没有颜色表这一块。其中，每 4 字节表示一种颜色，并以 B(蓝色)、G(绿色)、R(红色)、alpha(32 位位图的透明度值，一般不需要)。即首先 4 字节表示颜色号 1 的颜色，接下来表示颜色号 2 的颜色，依此类推。	

实验一：在BMP图像数据后隐藏数据

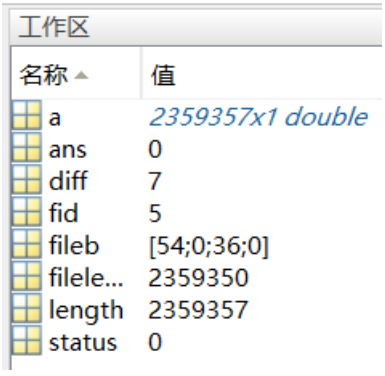
待隐藏的密码信息文件名称为：hidden.txt（里面有6个a），pic1.bmp 为载体图像，将载体和秘密信息放置在同一目录下，在Windows的CMD中执行命令：

```
Copy pic1.bmp /b + hidden.txt /a pic11.bmp
```

执行该命令后，生成一个新的pic11.bmp文件，使用图像浏览工具浏览该文件会发现它与原始载体图像几乎完全相同。
信息隐藏在pic11.bmp文件的**尾部**。
从BMP图像的结构中可知，图像的3、4、5、6四个字节表示整个BMP图像的长度（大小）。使用该方式隐藏信息时，未修改图像文件的文件长度字节，通过比较文件的实际长度和文件中保存的文件长度，就可发现该图像是否隐藏秘密信息。

```
clc;
clear;
fid=fopen('pic11.bmp','r');
[a,length]=fread(fid,inf,'uint8');
status=fseek(fid,2,'bof');
fileb=fread(fid,4,'uint8');
filelength=fileb(1)*1+fileb(2)*256+fileb(3)*256^2+fileb(4)*256^3;
diff=length-filelength;
fclose(fid);
```

```
clc;
clear;
fid=fopen('pic11.bmp','r');
 % 以只读方式打开pic11.bmp图像文件，该文件必须存在，fid用于存储文件句柄值
[a,length]=fread(fid,inf,'uint8');
 % 创造存放读取数据的矩阵，length中返回所读取的数据元素个数
 % inf读出fid指向的打开文件的全部数据
% fclose(fid);
 % 关闭fid所表示的文件，文件在进行完读、写等操作后，应及时关闭以免数据丢失
% fid=fopen('pic11.bmp','r');
status=fseek(fid,2,'bof');
 % fseek函数用于定位文件位置指针，fid为文件句柄，
 % offset(第2个参数)表示位置指针相对移动的字节数，“2”指的是跳过前两个文件标识字节
 % origin(第3个参数)表示位置指针移动的参照，'bof'，表示文件头；'cof'，表示当前位置 ；'eof'，表示文件尾
fileb=fread(fid,4,'uint8');
 % 读取4个元素（就是文件大小），精度为uint8
filelength=fileb(1)*1+fileb(2)*256+fileb(3)*256^2+fileb(4)*256^3;
 % 把这4个字节转换成大小
diff=length-filelength;
 % 有差值，可发现该图像隐藏秘密信息
fclose(fid);
```



实验二：在文件头与图像数据之间隐藏信息

```
clear;
clc;
fid=fopen('pic1.bmp','r');
[a,length]=fread(fid,inf,'uint8');
fclose(fid);
msgfid=fopen('hidden.txt','r');
[msg,count]=fread(msgfid,inf,'uint8');
fclose(msgfid);
wa=a;
wa(3)=wa(3)+count;
 % 图像头（从3开始）+ 隐藏文件内容的字节
wa(11)=wa(11)+count;
 % 000A-000D:从文件开始到图像数据之间的偏移量
 % 偏移量 + 隐藏文件内容的字节
j=1;
for i=55:55+count-1
    wa(i)=msg(j,1); % msg(行,列)
    j=j+1;
end
 % 文件头有54个字节（从0开始算），从55个字节开始藏
for i=55:length
    wa(i+count)=a(i);
end
 % 把原来的图像信息往后面移动要隐藏的信息长度
new=fopen('newpic11.bmp','wb'); % wb:只写打开或新建一个二进制文件
fwrite(new,wa);
 % 数组wa的元素按 列顺序 以 8 位无符号整数的形式写入二进制文件new
figure;
imshow('newpic11.bmp');
```

名称	值
a	*2359350x1 double*
ans	2359356
count	6
fid	3
i	2359350
j	7
length	2359350
msg	[97;97;97;97;97;97]
msgfid	3
new	3
wa	*2359356x1 double*

pic1.bmp | newpic11.bmp

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	ANSI ASCII
00000000	42	4D	36	00	24	00	00	00	00	00	36	00	00	00	28	00	BM6 $ 6 (
00000016	00	00	00	04	00	00	00	03	00	00	01	00	18	00	00	00	
00000032	00	00	00	00	24	00	00	00	00	00	00	00	00	00	00	00	$
00000048	00	00	00	00	00	00	00	21	09	01	23	0B	04	26	0E	08	! # &
00000064	27	12	08	27	12	09	25	11	08	23	0F	08	22	10	09	1F	' ' % # "
00000080	0D	08	1E	0C	08	1A	0D	05	16	0B	04	12	0C	01	0E	0C	
00000096	00	0A	0A	00	07	07	00	03	06	00	03	06	00	03	06	00	
00000112	02	05	00	01	05	00	01	05	00	00	04	00	00	04	00	00	

pic1.bmp | newpic11.bmp

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	ANSI ASCII
00000000	42	4D	3C	00	24	00	00	00	00	00	3C	00	00	00	28	00	BM< $ < (
00000016	00	00	00	04	00	00	00	03	00	00	01	00	18	00	00	00	
00000032	00	00	00	00	24	00	00	00	00	00	00	00	00	00	00	00	$
00000048	00	00	00	00	00	00	61	61	61	61	61	61	00	21	09	01	aaaaaa !
00000064	23	0B	04	26	0E	08	27	12	08	27	12	09	25	11	08	23	# & ' ' % #
00000080	0F	08	22	10	09	1F	0D	08	1E	0C	08	1A	0D	05	16	0B	"

36（16）=54，3C（16）=60，61（16）=97=a（ASCII）

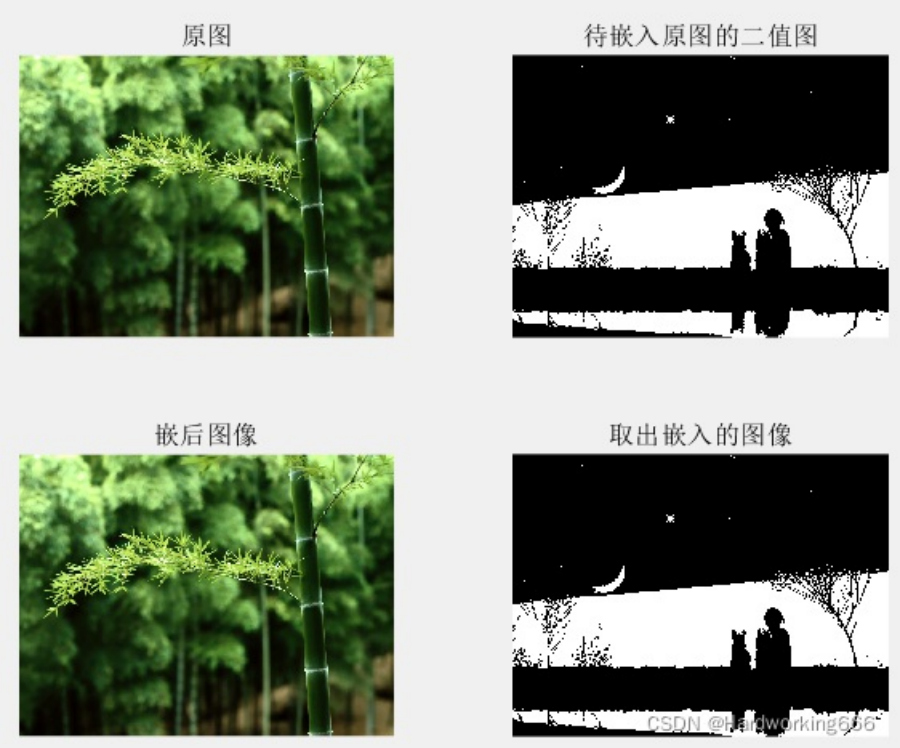
二、图像LSB隐写

最低有效位（ least significant bit，LSB）指的是一个二进制数字中的第0位（即最低位）。

实验一：在图像中嵌入相同分辨率的二值图像

将BMP图像pic2.bmp转换为二值图像，并将其嵌入**相同分辨率**的RGB图像pic1.bmp的任一层(R层、G层或B层灰度图像)的最低位平面。

```
clear;
clc;
I=imread('pic1.bmp');
J=I;
JR=J(:,:,1);
L=imread('pic2.bmp');
BW=im2bw(L);
set1=bitset(JR,1,BW);
 % 将JR转换成二进制，并将其右边第一个位置（即最低位）换成BW的值
J(:,:,1)=set1;
get1=bitget(J(:,:,1),1);
 % 输出J(:,:,1)最低位的数值
figure;
subplot(2,2,1);imshow(I);title('原图');
subplot(2,2,2);imshow(BW);title('待嵌入原图的二值图');
subplot(2,2,3);imshow(J);title('嵌后图像');
subplot(2,2,4);imshow(get1,[]);title('取出嵌入的图像'); % 加[]让图像自适应显示
```

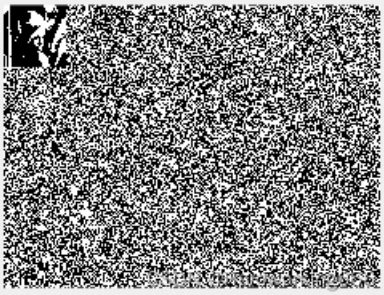


实验二：在图像中嵌入不同分辨率的二值图像

将lena.jpg图像嵌入到不同分辨率RGB图像pic1.bmp任一层(R层、G层或B层灰度图像)的最低位平面并提取。

```
clear;
clc;
I=imread('pic1.bmp');
L=imread('lena.jpg');
BW=im2bw(L);
J=double(I);
[m,n]=size(BW);
[l,w]=size(I);
if or(m>l,n>w/3)
    error('嵌入信息量过大，请重新选择图像');
end
for i=1:m
    for j=1:n
        J(i,j,1)=J(i,j,1)-mod(J(i,j,1),2)+BW(i,j);
        % 把J的红色图层最低位清空换成二值图像BW相应的行列值
    end
end
M=bitget(J(:,:,1),1);
J=uint8(J);
imwrite(J,'embedded.bmp'); % embed vt.嵌入
figure;
subplot(221);imshow(I);
subplot(222);imshow(BW);
subplot(223);imshow('embedded.bmp');
subplot(224);imshow(M);
```



三、直接4bit替换法

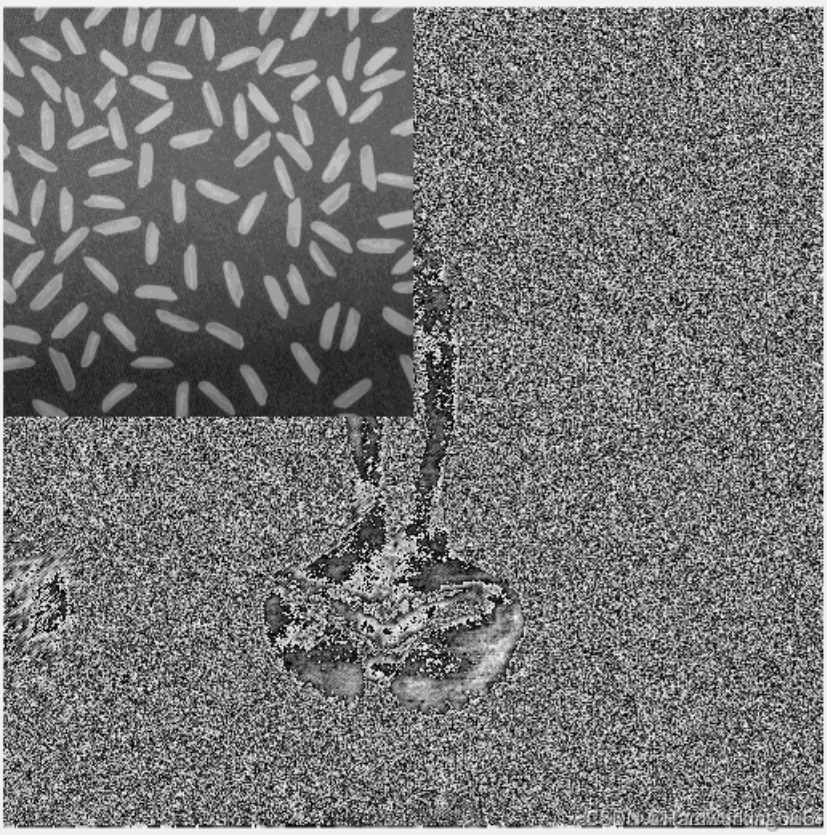
图像的低4bit，通常可以做隐藏信息的空间，也就是载体的冗余空间。
直接4bit替换法，就是直接用**秘密图像像素值的高4bit**去替换**载体图像像素值的低4bit**。

目的：把rice.png（256×256）藏到baboon.bmp（512×512）
不同分辨率直接4位替换法：

```
clear;
clc;
I=imread('baboon.bmp');
msg=imread('rice.png');
[row,col]=size(msg);
IR=I(:,:,1);
[m,n]=size(IR);
if or(row>m,col>n)
    error('嵌入信息量过大，请重新选择图像');
end
for i=1:row
    for j=1:col
        IR(i,j)=bitand(IR(i,j),240); % 240转二进制为11110000
    end
end
 % 置载体图像R层的低4bit为0
figure;imshow(IR);
figure;imshow(msg);
takemsg4=bitand(msg,240);
 % 置秘密图像的低4bit为0
figure;imshow(takemsg4);
shiftmsg4=bitshift(takemsg4,-4);
 % 将秘密图像的高4bit右移4位
figure;imshow(shiftmsg4,[]);
for i=1:row
    for j=1:col
        IR(i,j)=bitor(IR(i,j),shiftmsg4(i,j));
    end
end
 % 图像隐藏
I(:,:,1)=IR;
 % 写回并保存
figure;imshow(I);
imwrite(I,'bit4hide.bmp');
```

不同分辨率直接4位替换法**提取**：

```
clear;
clc;
I=imread('bit4hide.bmp');
[row,col]=size(I)
A=I(:,:,1);
for i=1:row/2 % 这里除以2是为了完整显示rice.png，下图展示的是不除2的
    for j=1:col/3/2
        M(i,j)=bitand(A(i,j),15); % 1111
    end
end
M=bitshift(M,4);
imshow(M);
```



四、灰度图8位分别藏于RGB层

将灰度图rice.png的8位像素数据分别藏于载体图像baboon.bmp的R、G、B层像素值的低几位：

```
clear;
clc;
I=imread('baboon.bmp');
J=imread('rice.png');
[row,col]=size(J);
for h=1:3
for i=1:row
    for j=1:col
        I(i,j,h)=bitand(I(i,j,h),248); % 1111 1000
    end
end
end
figure;
subplot(221);imshow(I);
subplot(222);imshow(I(:,:,1));
subplot(223);imshow(I(:,:,2));
subplot(224);imshow(I(:,:,3));
Y=bin2dec('11100000'); % 224
y1=bitshift(Y,-3);
y2=bitshift(Y,-6);
for h=1:3
    M=bitand(J(1:row,1:col),bitshift(Y,-3*(h-1)));
    shiftM=bitshift(M,h*3-8);
    I(1:row,1:col,h)=bitor(I(1:row,1:col,h),shiftM);
end
figure;imshow(I);
imwrite(I,'fullhide.bmp');
```

将上题中隐藏的灰度图提取出来并保存：

```
clear;
clc;
I=imread('fullhide.bmp');
for h=1:3
for i=1:256
    for j=1:256
        I(i,j,h)=bitand(I(i,j,h),7); % 0111
    end
end
end
figure;
subplot(221);imshow(I,[]);
subplot(222);imshow(I(:,:,1));
subplot(223);imshow(I(:,:,2));
subplot(224);imshow(I(:,:,3));
K=uint8(zeros(256,256));
for h=1:3
    M=bitshift(I(1:256,1:256,h),8-h*3); % 向左移动8-h*3位，5、2、-1
    K=bitor(K,M);
end
imwrite(K,'riceget.bmp');
figure;imshow('riceget.bmp');
```

五、图片中隐藏文本信息

目的：实现文本信息在载体图像中的伪随机位置的信息隐藏并提取文本信息。

1、编写伪随机序列发生函数randinterval.m

```
function [row,col]=randinterval(map,count,key)
[m,n]=size(map);
interval1=floor(m*n/count)+1; % 向下取整，即取不大于m*n/count的最大整数
interval2=floor(m*n/count)-1;
if interval2==0
    error('载体太小，请重新选择载体图像');
end
rand('seed',key);   % 生成随机序
a=rand(1,count);
row=zeros(1,count);  % 初始化
col=zeros(1,count);
r=1;
c=1;
row(1,1)=r;
col(1,1)=c;
for i=2:count
    if a(i)>=0.5
        c=c+interval1;
    else
        c=c+interval2;
    end
    if c>n
        r=r+1;
        if r>m
            error('载体太小,无法隐藏密秘信息');
        end
        c=mod(c,n);
        if c==0
            c=1;
        end
    end
    row(1,i)=r;
    col(1,i)=c;
end
```

2、编写程序binhide.m，完成功能：将文本信息hidden.txt嵌入pic1.bmp图像中的某一层的伪随机位置

```
clear;
clc;
I=imread('pic1.bmp');
IR=I(:,:,1);
cover=double(IR);
fid=fopen('hidden.txt','r');
[msg,len_total]=fread(fid,inf,'ubit1');
[m,n]=size(cover);
if len_total>m*n
    error('嵌入信息量过大，请重新选择图像');
end
[row,col]=randinterval(cover,len_total,2021);
p=1;
for i=1:len_total
    cover(row(i),col(i))=cover(row(i),col(i))-mod(cover(row(i),col(i)),2)+msg(p,1);
    if p==len_total
        break;
    end
    p=p+1;
end
cover=uint8(cover);
I(:,:,1)=cover;
imwrite(I,'randhide.bmp');
M=cover-IR;
figure;imshow(M,[]);
figure;imshow('randhide.bmp');
fclose(fid);
```

名称	值
ans	0
col	*1x11520 double*
cover	*768x1024 uint8*
fid	5
i	11520
I	*768x1024x3 uint8*
IR	*768x1024 uint8*
len_total	11520
m	768
M	*768x1024 uint8*
msg	*11520x1 double*
n	1024
p	11520
row	*1x11520 double*

3、编写程序**binget.m**,将上题中隐藏的文本信息提取出来，并保存在**binget.txt**中

```
clear;
clc;
I=imread('randhide.bmp');
IR=double(I(:,:,1));
count=11520;
[row_1,col_1]=randinterval(IR,count,2021);
fid=fopen('binget.txt','a');
for i=1:count
    fwrite(fid,mod(IR(row_1(i),col_1(i)),2),'ubit1');
end
fclose(fid);
```