

【matlab】一篇文章看懂图像隐写的简易实操

原创

[RouTineD](#) 于 2020-03-15 15:15:46 发布 1353 收藏 27

分类专栏: [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/weixin_44012551/article/details/104877624

版权



[matlab](#) 专栏收录该内容

7 篇文章 1 订阅

订阅专栏

目录

一、图像隐写介绍

二、使用到的函数介绍

1、读取图像函数

2、展示图像函数

3、彩色图片变成灰度图片

4、bitget()和bitset()函数

5、logical()函数

三、黑白图像隐写示例

1、使用图片

2、代码

3、结果演示

四、彩色图像隐写示例

1、使用图片

2、代码

3、结果演示

一、图像隐写介绍

图像隐写, 即把一张图片隐藏写入在另外一张图片中。

作用: 隐藏信息, 这里不做过多赘述, 详情见这个连接

二、使用到的函数介绍

1、读取图像函数

```
Img1 = imread('yxj.jpg')
```

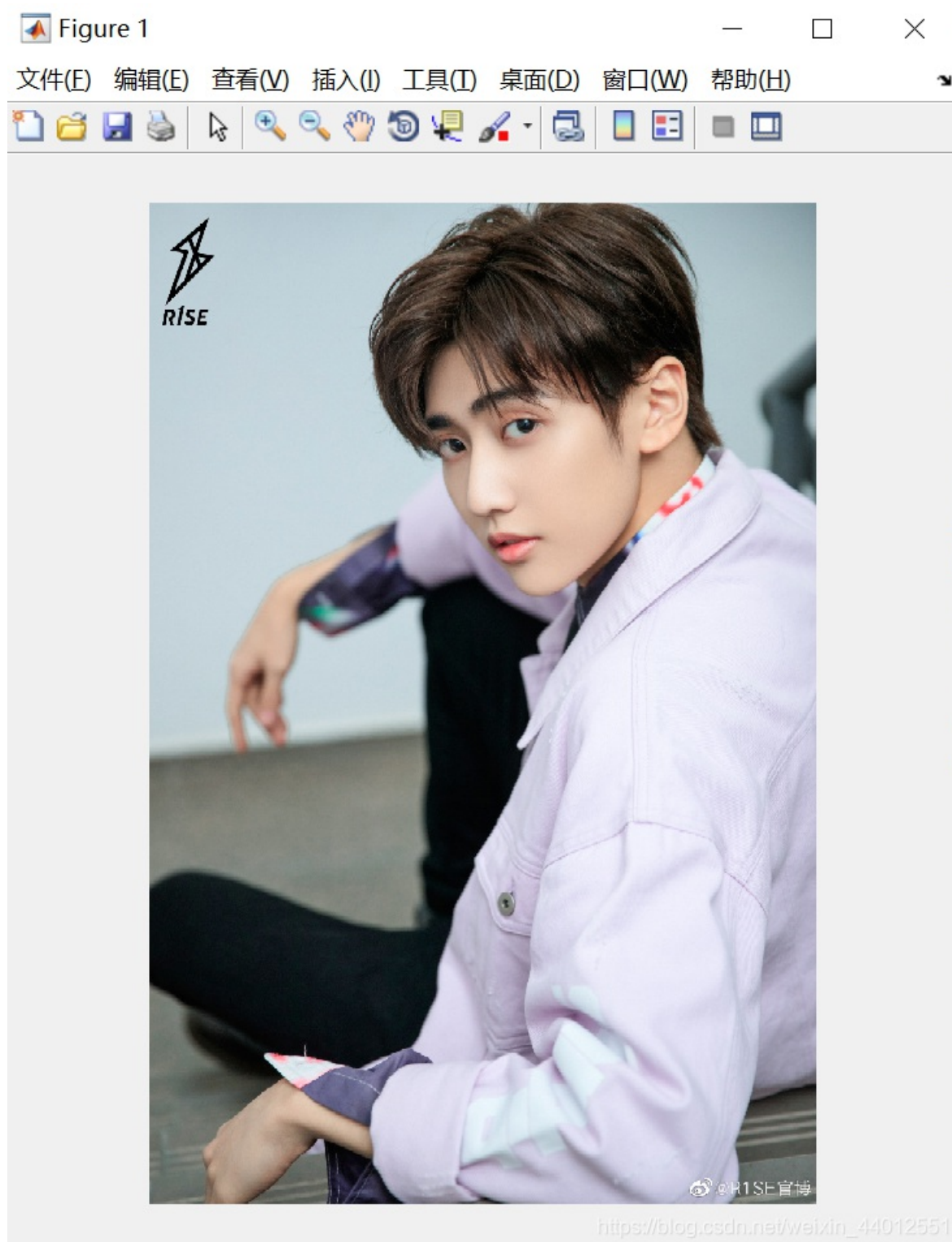
图像被读取后会以像素矩阵的形式存储在变量中

 `img1 1536x1024x3 uint8`

2、展示图像函数

`imshow(img1)`

其中为图像像素矩阵



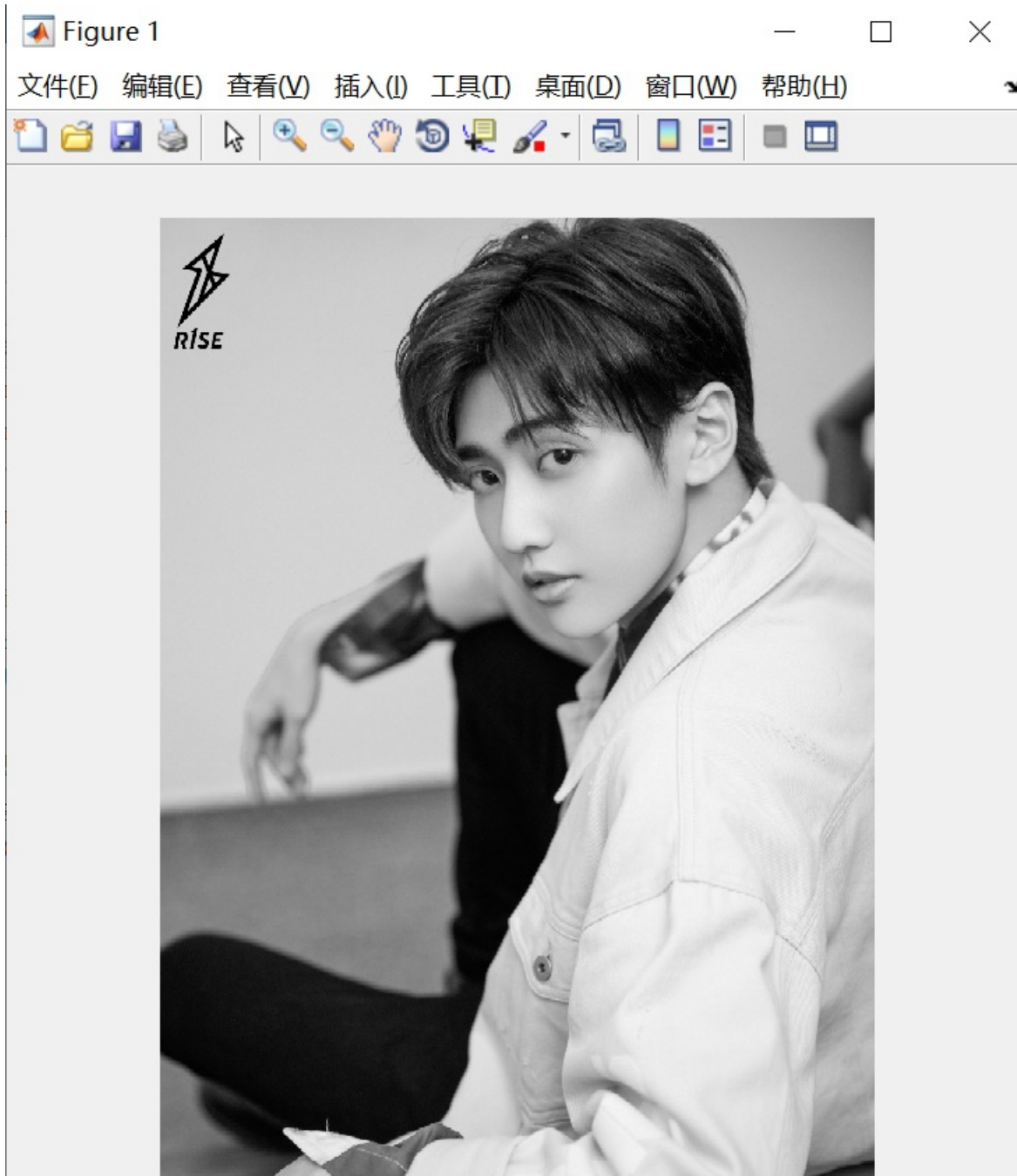
3、彩色图片变成灰度图片

img1是一个有三个通道的彩色图片

彩色图片具有RGB三个通道，gray是灰度图，rgb2gray==rgb to gray
转换后的img1变成了一个二维像素矩阵

	1	2	3	4
1	194	193	193	192
2	193	193	193	193
3	193	193	193	193
4	194	194	193	193
5	195	195	195	194

使用imshow函数可以发现图像已经变成黑白



4、bitget()和bitset()函数

bitget获取二进制位

```
% 八位下数字8的二进制码是00001000
A=bitget(8,4);
运行结果: A=1;
```

```
c=8;
A=bitget(c,8: -1:1);
运行结果: A=[0 0 0 0 1 0 0 0]
```

bitset 设置指定bit位

```
% 八位下数字9的二进制码为00001001
A=bitset(uint8(9), 4,0)
运行结果: A = 1,将9的第4位设置为0.
```

```
A=bitset(uint8(9), 5,1)
运行结果: A = 25,将9的第5位设置为1.
```

5、logical()函数

logical()函数能把数据类型改成true or false类型

```
% d为矩阵
logical(d)
```

三、黑白图像隐写示例

1、使用图片

如下图两张图，我们要把第二张图的信息存在第一张图中。



Routine

https://log.cdn.net/weixin_44012551

2、代码


```

% 这里的Img1是主图，Img2是隐藏信息
Img1 = rgb2gray(imread('yxj1.jpg'));
Img2 = rgb2gray(imread('RouTine.jpg'));
% 获取Img1图像大小
sizeImg1 = size(Img1);
% 我们需要隐藏的信息可能和主图的大小不一致，使用imresize进行大小的缩放
% 图片需要插入到bit中，bit中的数据是0 1，所以把Img2的类型改成logical类型
Img2 = logical(imresize(Img2,[sizeImg1(1),sizeImg1(2)]));
% n为我们要插入的位数
for n = 1:8
    % 写入隐藏信息到第n位
    d = bitset(Img1,n,Img2);
    subplot(2,4,n);
    imshow(d)
end

```

3、结果演示



从左上到右下分别为隐写在1-8位的情况

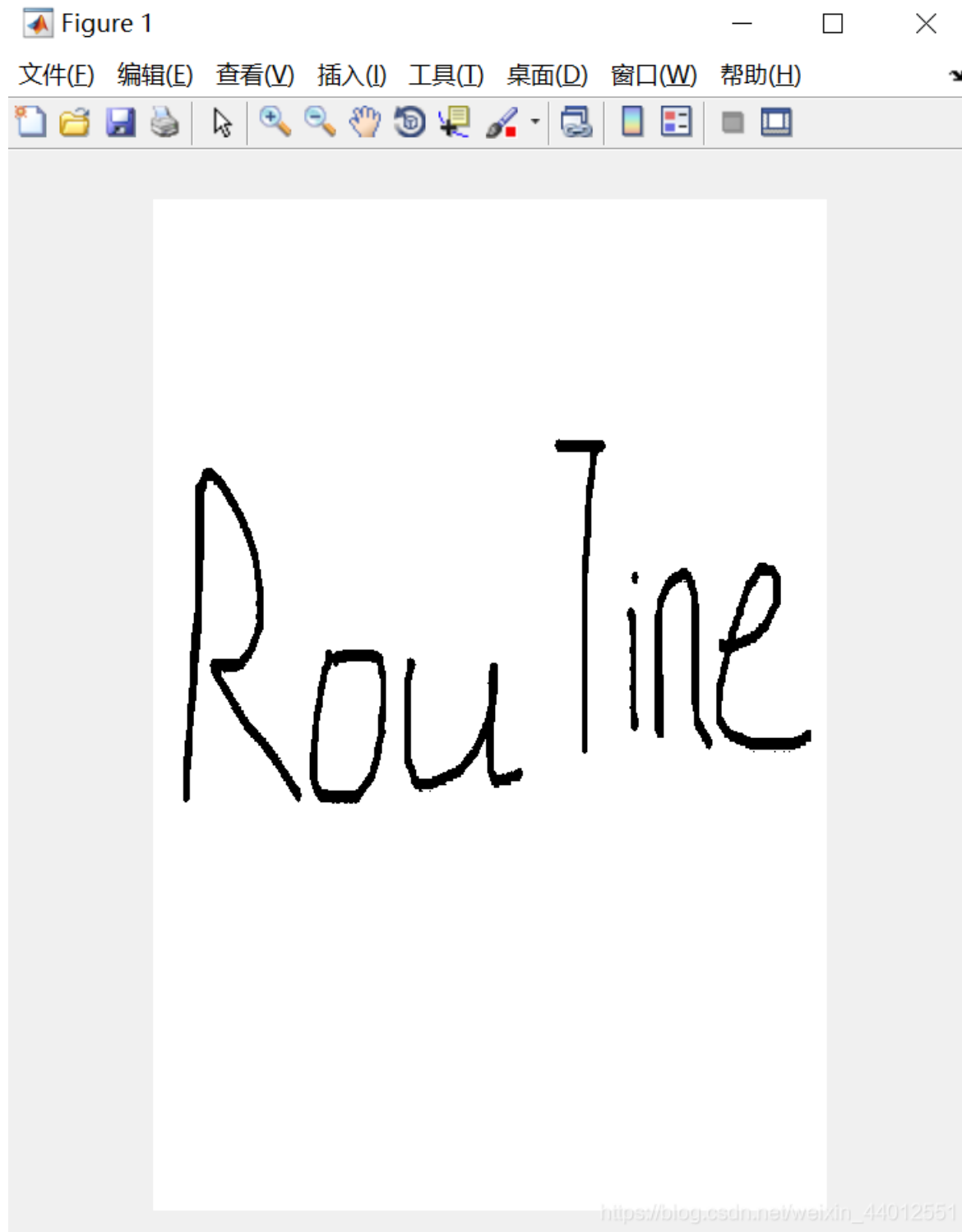
事实上，现在截下上面八张图的任何一张，提取对应位数矩阵，可得到RouTine.jpg原图

```

% d为图像像素矩阵，n为位数
>> imshow(logical(bitget(d,n)))

```

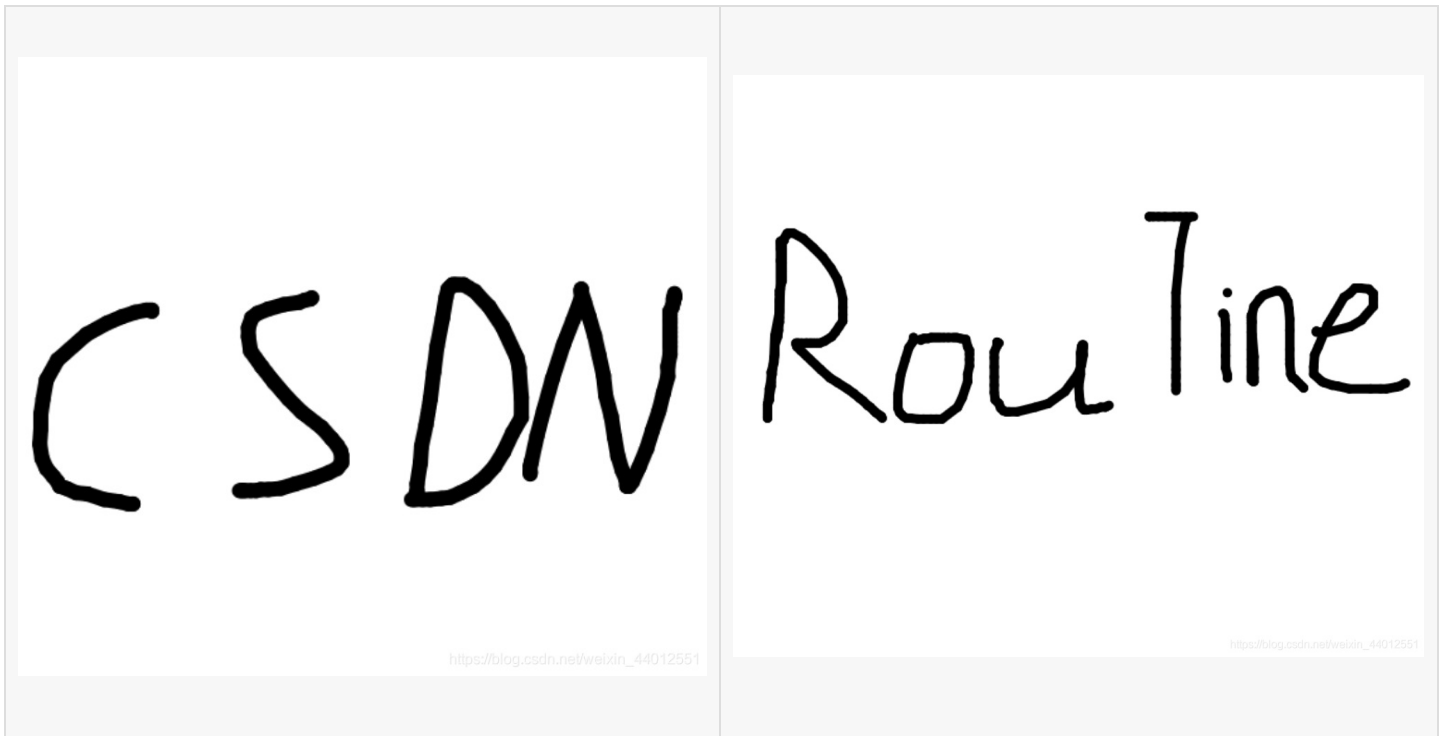
注意，这里使用logical函数，因为得到的位均为0 1，使用bitget得到的矩阵默认格式为uint8，即显示范围为[0 255]。若不改成logical格式格式，输出的图片会是全黑。



四、彩色图像隐写示例

1、使用图片

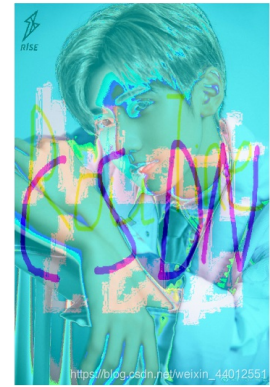




2、代码

```
% 这里的Img1是主图，Img234是隐藏信息
Img1 = imread('yxj2.jpg');
Img2 = rgb2gray(imread('MATLAB.jpg'));
Img3 = rgb2gray(imread('CSDN.jpg'));
Img4 = rgb2gray(imread('RouTine.jpg'));
% 获取Img1图像大小
sizeImg1 = size(Img1);
% 我们需要隐藏的信息可能和主图的大小不一致，使用imresize进行大小的缩放
% 图片需要插入到bit中，bit中的数据是0 1，所以把Img2的类型改成logical类型
Img2 = imresize(Img2,[sizeImg1(1),sizeImg1(2)]);
Img3 = imresize(Img3,[sizeImg1(1),sizeImg1(2)]);
Img4 = imresize(Img4,[sizeImg1(1),sizeImg1(2)]);
% n为我们插入的位数
for n = 1:8
    % 写入隐藏信息到第n位
    % R通道隐藏MATLAB.jpg，G通道隐藏CSDN.jpg，B通道隐藏RouTine.jpg
    d(:,:,1) = bitset(Img1(:,:,1),n,logical(Img2));
    d(:,:,2) = bitset(Img1(:,:,2),n,logical(Img3));
    d(:,:,3) = bitset(Img1(:,:,3),n,logical(Img4));
    subplot(2,4,n);
    imshow(d)
end
```

3、结果演示



从左上到右下分别为隐写在1-8位的情况

事实上，现在截下上面八张图的任何一张，提取对应位数对应通道，可得到隐写原图

```
% d为图像像素矩阵, n为位数, cnt为通道数  
>> imshow(logical(bitget(d(:,:,cnt),n)))
```

注意，这里使用logical函数，因为得到的位均为0 1，使用bitget得到的矩阵默认格式为uint8，即显示范围为[0 255]。若不改成logical格式格式，输出的图片会是全黑。