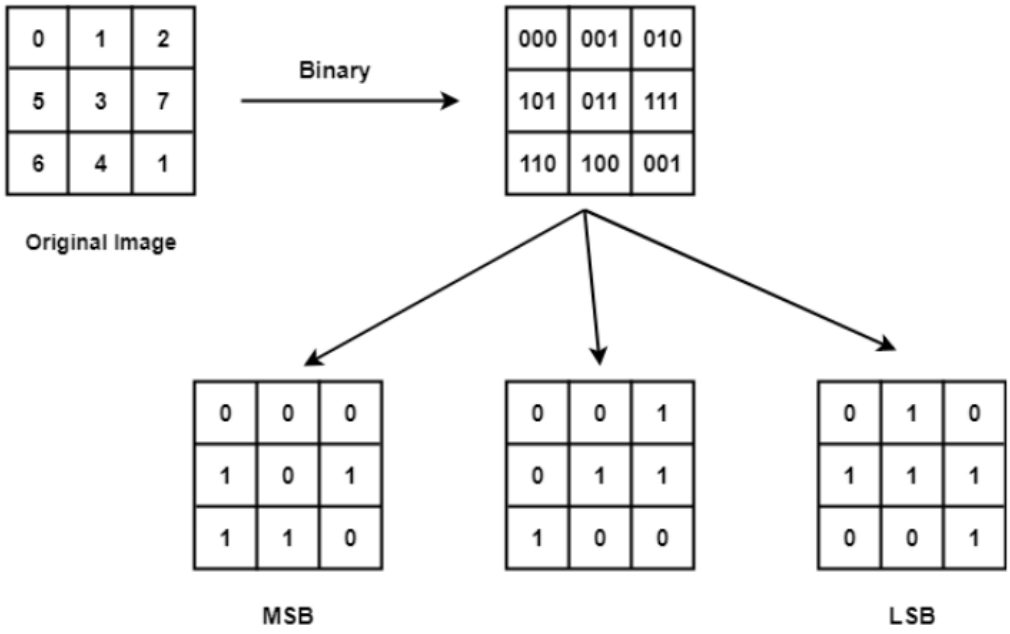
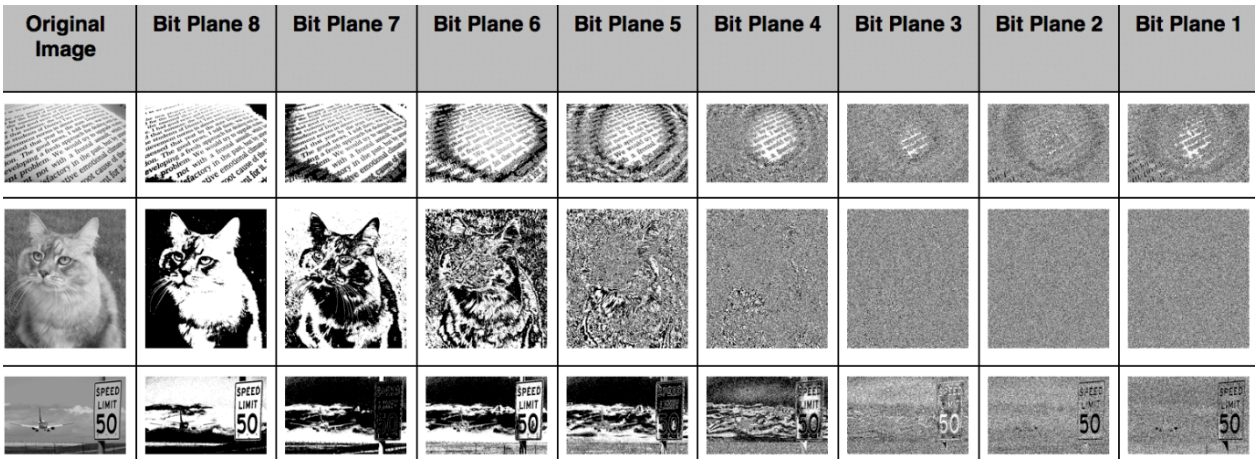
Steganography (Hiding Information in Images)

We are used to the term cryptography (secret writing) where we aim to encrypt or otherwise disguise a message or information so that it can only be read by the intended recipient. Steganography (hidden writing) on the other hand aims to make information undetectable. Basic digital image processing techniques can be employed as a form of Stenography. The underlying principle is based on the concept of Bit Planes, which follows directly from the pixel intensities making up a grayscale image being held in the computer as a binary number. The basic concept can be understood by considering the 3 x 3 image consisting of 3-bit (0 to 7) pixel intensities. Instead of representing the pixel intensities in the decimal number system, we instead look at the binary equivalent. Bit Plane slicing then takes only a single significant bit position from each pixel (0 or 1) and represents each resulting 'Bit Plane' as a binary image (i.e. with pixels only containing 0 or 1). The interpretation of the pixels in each bit plane is then that a '0' represents 'Black' and a '1' represents 'White'.



If we look at a more complex images, we can see the relative information content held in each of the Bit Planes. Some are shown below, where the primary observation of interest to us is that Bit Planes of the Least significant bits become increasingly random and contribute relatively little to the overall image content, whereas the most significant bit contribute the greatest amount.



Stenography using images is then a process by which we replace the least significant Bit Plane Slice of an image with some other information of our choice, and then reconstitute the resulting image with the substituted Bit Plane Slice containing the information that we want 'hide'.

The process can be readily shown by example. Firstly, we are going to load two images, Hugo.jpg and IamHere.png, into Matlab and change them to grayscale.

```
%read in image
Pic1Colour=imread("Hugo.jpg");
Pic2Colour=imread("IamHere.png");

%changeto grayscale
Pic1 = im2gray(Pic1Colour);
Pic2 = im2gray(Pic2Colour);
%show grayscale image
figure;
imshowpair(Pic1,Pic2,"montage");
```

I am going to pretend I want to tell someone where I am by hiding my location in the grayscale version of the original Hugo.jpg. To do this I want to Bit Plane Slice both images. Matlab allows us to easily do this:

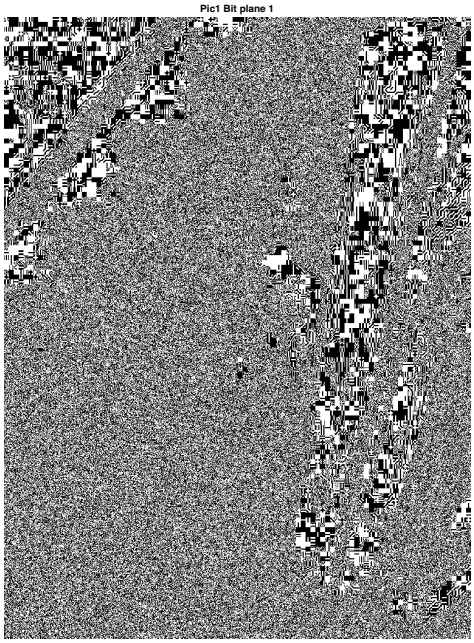
```
%Extract Bit Planes of Pic 1
BitImage1Pic1=bitget(Pic1,1); figure,
imshow(logical(BitImage1Pic1));title('Pic1 Bit plane 1');
BitImage2Pic1=bitget(Pic1,2); figure,
imshow(logical(BitImage2Pic1));title('Pic1 Bit plane 2');
BitImage3Pic1=bitget(Pic1,3); figure,
imshow(logical(BitImage3Pic1));title('Pic1 Bit plane 3');
BitImage4Pic1=bitget(Pic1,4); figure,
imshow(logical(BitImage4Pic1));title('Pic1 Bit plane 4');
BitImage5Pic1=bitget(Pic1,5); figure,
imshow(logical(BitImage5Pic1));title('Pic1 Bit plane 5');
BitImage6Pic1=bitget(Pic1,6); figure,
imshow(logical(BitImage6Pic1));title('Pic1 Bit plane 6');
BitImage7Pic1=bitget(Pic1,7); figure,
imshow(logical(BitImage7Pic1));title('Pic1 Bit plane 7');
BitImage8Pic1=bitget(Pic1,8); figure,
imshow(logical(BitImage8Pic1));title('Pic1 Bit plane 8');

%Extract Bit Planes of Pic 2
BitImage1Pic2=bitget(Pic2,1); figure,
imshow(logical(BitImage1Pic2));title('Pic2 Bit plane 1');
BitImage2Pic2=bitget(Pic2,2); figure,
imshow(logical(BitImage2Pic2));title('Pic2 Bit plane 2');
BitImage3Pic2=bitget(Pic2,3); figure,
imshow(logical(BitImage3Pic2));title('Pic2 Bit plane 3');
BitImage4Pic2=bitget(Pic2,4); figure,
imshow(logical(BitImage4Pic2));title('Pic2 Bit plane 4');
BitImage5Pic2=bitget(Pic2,5); figure,
imshow(logical(BitImage5Pic2));title('Pic2 Bit plane 5');
BitImage6Pic2=bitget(Pic2,6); figure,
imshow(logical(BitImage6Pic2));title('Pic2 Bit plane 6');
BitImage7Pic2=bitget(Pic2,7); figure,
imshow(logical(BitImage7Pic2));title('Pic2 Bit plane 7');
BitImage8Pic2=bitget(Pic2,8); figure,
imshow(logical(BitImage8Pic2));title('Pic2 Bit plane 8');
```

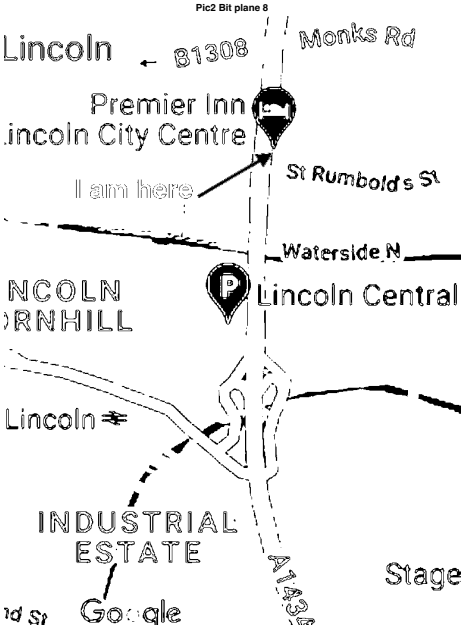
We won't show all the resulting Binary Bit Plane Slices here (you can look at them yourself), but the MSB and LSB of each image are shown below. I've also included Bit Plane Slice 6 of the 2nd image because in this instance it appears to show where I am ("I am Here") better than Bit Plane Slice 8.

Pic1 Bit plane 8

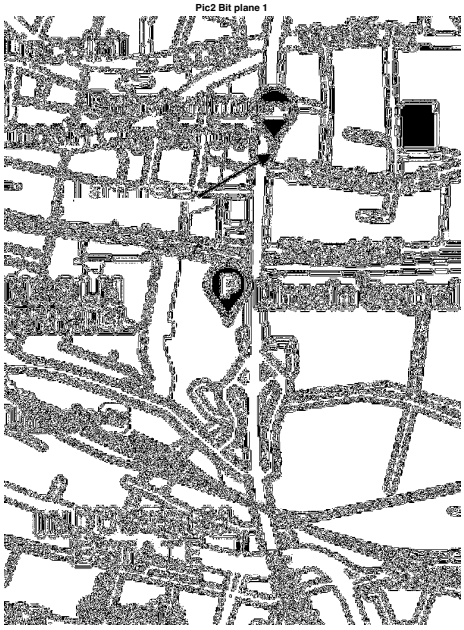
Pic1 Bit plane 1

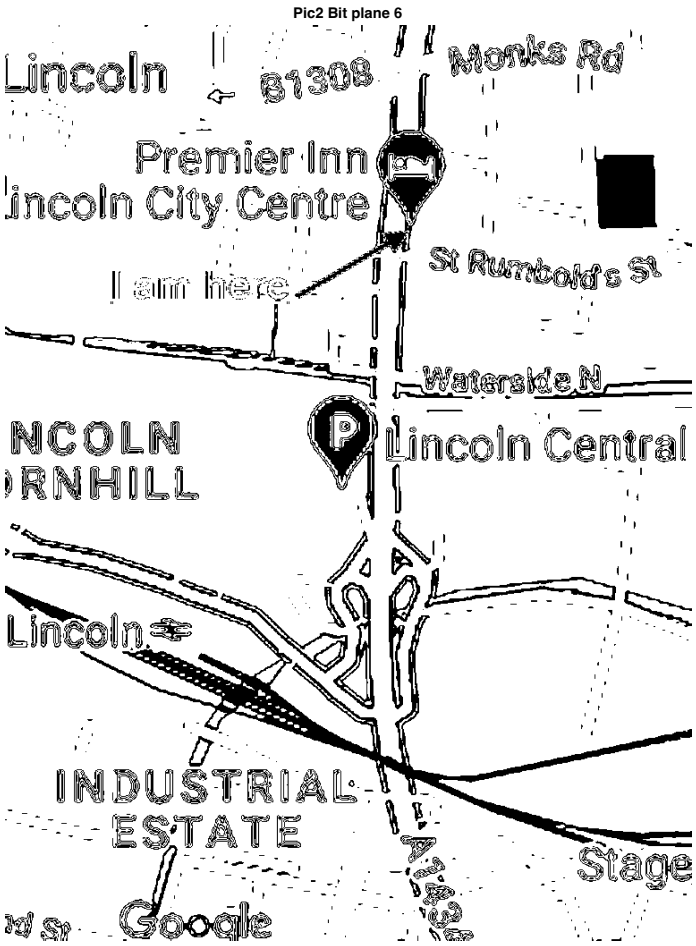


Pic2 Bit plane 8



Pic2 Bit plane 1





I am now going to replace Bit Plane Slice 1 of "Hugo" with Bit Plane Slice 6 of "IamHere". Again Matlab can do this for us with the command *bitset*.

NewPic1 = bitset(Pic1,1, BitImage6Pic2);

Here, we replace Bit Plane Slice 1 of Pic 1 (Hugo) with Bit Place Slice 6 of Pic2 (IamHere).

A comparison of the original image with the one with Bit Plane Slice 1 substituted is given below. Visually it can be seen that there is no perceived difference between the images—we have effectively hidden the information in the image.

Pic1 vs Pic 1 with bit layer 1 replaced by bit layer 6 of Pic2

Nevertheless, if we look at Bit Plane Slice 1 of the right-hand image, we obtain the map as expected.

NewPic1 Bit plane 1