Chapter 2 - Bit Planes

By: Himanshu Sharma

Introduction

In this chapter, we will be focusing on bit planes which will help us in improving our understanding of steganography. Since we have already discussed *LSB Matching* in previous chapter, it will make our tasks easier. The reader must note that this chapter does not implement any IEEE paper and therefore no issue and volume numbers are specified.

Bit plane is the set of bits of any digital image using an n bit per pixel binary number representation. **By default, I am considering grayscale images in this chapter**. So if an image being used here is in RGB format, it will be converted suitably to grayscale. For example, consider an image using 8 bit per pixel format to represent its pixel values, i.e., each pixel value is made up of 8 bit binary number, let's say, a pixel is 11100101, since its a grayscale image. Then, the 2^{nd} bit plane of such an image would contain the 2^{nd} bit from left of all the pixels in the image. For this pixel, 2^{nd} bit would be 1 (1$\underline{1}$100101). Now, if we want to save this bit plane as an image, then 1 in the bit plane array represents white color and 0 represents black color.

Figure 1: Image

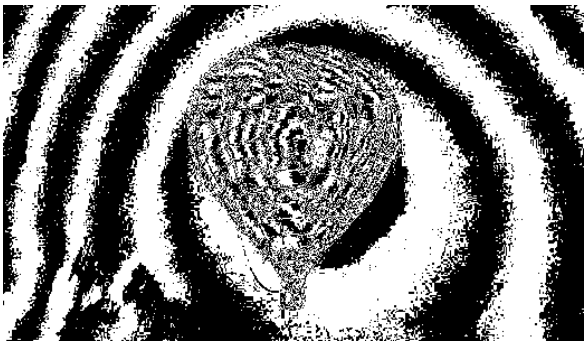

Figure 2: 5^{th} Bit Plane of Image

Figure 3: 1^{st} Bit Plane of Image

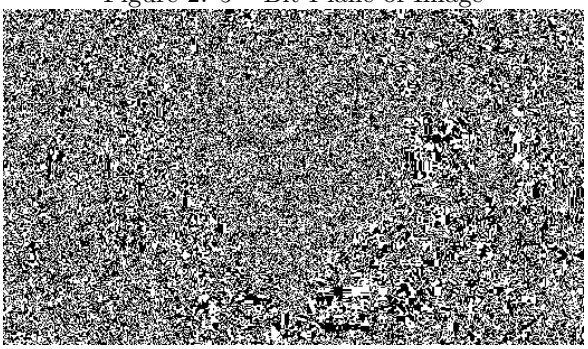

Figure 4: 8^{th} Bit Plane of Image

Least Significant Bit and Most Significant Bit

Since we are dealing with steganography, for us, the LSB plane and the MSB plane are of most importance. Let's see why. In LSB Matching, we saw that the LSB is the most efficient bit to store the data. Let us prove this by simple example considering a grayscale image. We pick up a pixel and lets say it has a value 245, which in binary is 11110101. If by standard steganography techniques, the LSB is changed to 0, then we have 11110100. This number in decimal is 244. It is clearly obvious from this value that the level of *grayness* of this pixel has not changed much because the pixel value has only changed by 1 decimal value. This is the reason why LSB is more efficient in storing data, because it does not change the pixel value much.

Let us now consider the effect of changing MSB, if that is what we would have used in steganography. So, let's say we have the same grayscale pixel with value 245 (= 11110101). If by standard steganography techniques, the MSB changes from 1 to 0, we get 01110101 as the new pixel value, which in decimal is 117. This is a huge change. It will significantly increase the *grayness* of the image. The brightness of the grayscale image will decrease or increase depending on the changes in each pixel's MSB and therefore anyone could recognize that the image has been manipulated. And this is the reason, that for steganography MSB is never preferred.

Since we are dealing with bit planes here, I will show you MSB and LSB planes of the same balloon image shown above.

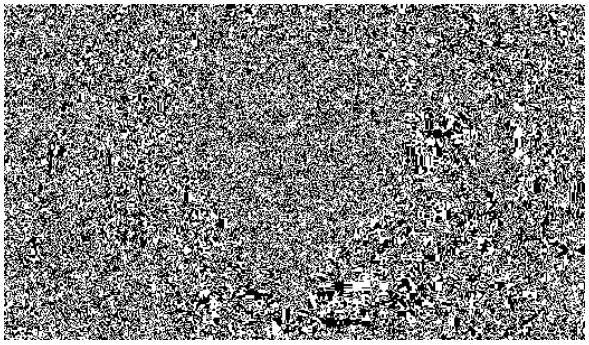

Figure 5: LSB Plane

Figure 6: MSB Plane

Clearly, changing any pixel from black to white or from white to black in the left figure will not affect the plane much because it has random black and white dots. However, if one tries to change pixels of right figure, the changes can be identified without any difficulty. Images below show the LSB planes of cover image and stego image.

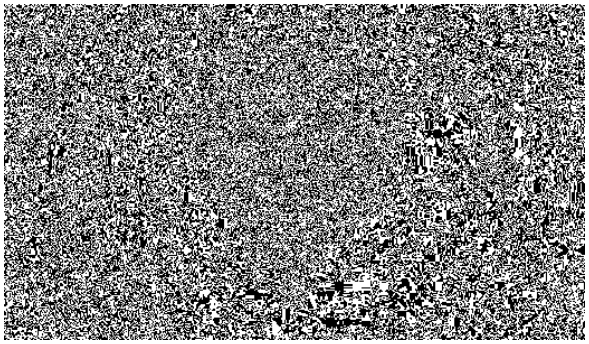

Figure 7: LSB Plane of Cover Image

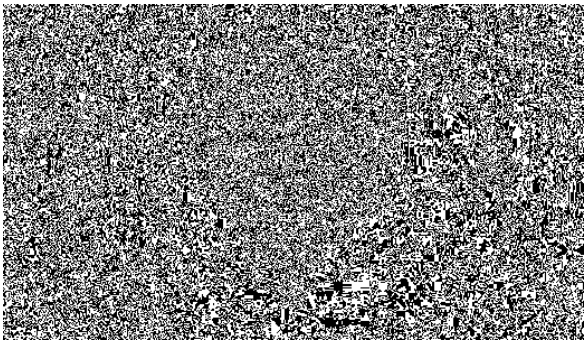

Figure 8: LSB Plane of Stego Image

From the figures, it is clear that LSB planes of cover image and stego image are indistinguishable to

a human eye. However, if one really wants to detect steganography in any image, then only these planes can help. A simple procedure for that could be to quickly move between LSB planes of cover and stego images and observe the changes in black and white dots. If there is any change, then one can quickly detect steganography. This is one of the methods I propose to detect steganography. However, it is little inefficient because you need to have the cover image for detection.

Demo

In this section we will see a demo on how to create bit planes of an image. I have made a Python script to build these planes. Open the python interpreter and type this.

```
>>> for i in range(1, 9):
        generateBitPlane('balloon.png',i)

>>>
```

Where generateBitPlane() is a function that I have created in my script. Full code is available on my github page.

This will generate these 8 bit planes.

Figure 9: MSB Plane

Figure 10: 2^{nd} Bit Plane

Figure 11: 3^{rd} Bit Plane

Figure 12: 4^{th} Bit Plane

Figure 13: 5^{th} Bit Plane

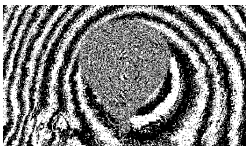

Figure 14: 6^{th} Bit Plane

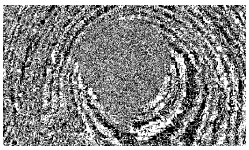

Figure 15: 7^{th} Bit Plane

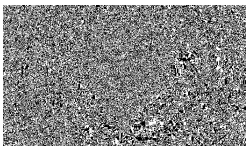

Figure 16: LSB Plane

Conclusion

In this chapter, we investigated the need of bit planes and their significance in the field of steganography. We found that changing the MSB has a serious effect on the images as compared to the LSB. This was proved both by binary mathematics and by bit planes. From bit planes it was clearly visible that changing a pixel value in 8^{th} bit plane (or LSB plane) does not affect the plane from the human eye point of view but in the 1^{st} bit plane (or MSB plane), a change in pixel value is easily dectectable from human eye. Then we concluded that LSB is most efficient bit for steganography with decreasing efficiency as we go from 8^{th} bit to the 1^{st} bit. We also concluded that steganography can be detected by quickly moving between cover image's LSB plane and that of stego image's LSB plane. Hence, our purpose of discussing bit planes has been fulfilled.

✧✧✧