

CSCI 8810: Project #4

Due on Tuesday, April 21, 2015

Prof. Arabnia 08:00am

Sina Solaimanpour

Contents

I. Introduction	3
II. Make Lower N bits White or Black	3
III. Steganography	3
IV. Experiments	4
V. Suggestions	6

I. Introduction

In this iteration of the ColonD (:D) Image Processor project, I am going to implement a Steganography feature to hide a subject image in a cover image. The following sections will be talking about this feature and the experiments I did on the Steganographer I implemented. An screenshot of the program can be seen in **Figure 1**.

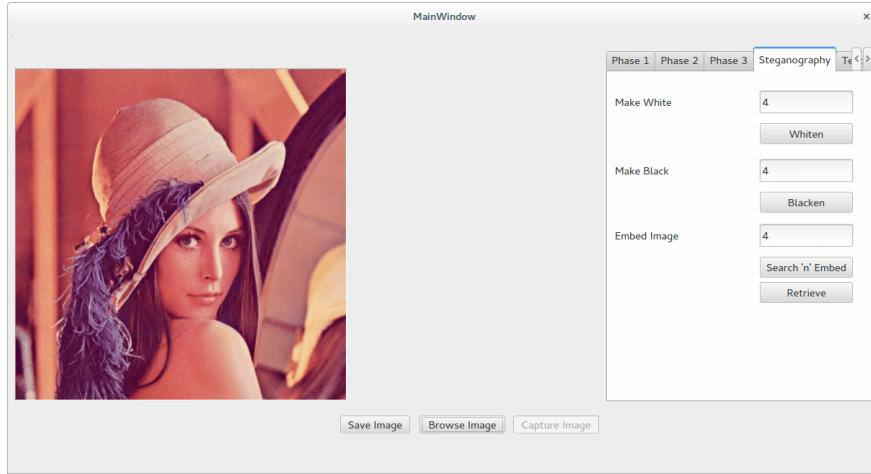


Figure 1: Screenshot of the program with the new features implemented in this iteration of the project.

II. Make Lower N bits White or Black

As an extra feature, to illustrate how the number of bits chosen to hide the subject image impacts the quality of the cover image I implemented this. This feature will whiten out or blacken out the lower N bits of the image in all three RGB channels. In **Figure 2** you can see results for different N values when whitening the bits and in **Figure 3** you can see the results for the exact N values but with blacken bits.

This shows that in this particular image we can hide $512 \times 513 \times 3$ bits of information without any change in the appearance of the cover image. This number is calculated by multiplying the width by the height of the image and the number of bits whitened or blackened without seeing any effects in the image which was 3 bits. This value can be very big for large pictures taken even by old phone cameras which are typically 5 mega pixels. The amount of data that can be hidden in a 5 mega pixel image is about **15 mega bits = 1875 Kilobytes** when using only 3 lower bits of the pixel values. This amount can hide 1875 characters in it, which is more than enough for transmitting a secret message.

III. Steganography

After implementing the above features and doing some experiments with those two features, I decided to implement the main feature which is the steganographer. To do the steganography, I assume that the image sizes are equal and I consider both of them as color images. The user has an option to choose how many bits he/she wants to use to hide the subject image. After deciding about the N value, I take the N higher bits of the subject image and replace the N lower bits of the cover image. In **Figure 4** you can see the resulting image from hiding the bike image in the lena's image using 3 bits. You can also see the original images and the recovered image after the process. The same images just with the N values equal to 4 is shown in **Figure 5**. As you can see, there exists a shadow of the bike image in the resulting image with N = 4. The experiment is repeated with the same images but with N = 5 in **Figure 6**. The resulting cover



Figure 2: Results of applying different N values to the whiten feature. As you can see the effects of the whitening starts to appear from the 4th image and the 7th image is completely whitened which is removed in this report to save some space.

image is obviously showing that there is something going around in it but the recovered image is as if no data is damaged in it.

The above experiments show that $N = 3$ is an optimal value for N to be used with the lena's picture as the cover image. At this point, I wanted to try different images to see if this optimal value differs for different images or not. I started with an image with high change frequencies.

IV. Experiments

I suspected that the $N = 3$ is a very small number and there might be some images that can be used as cover image and can handle more than $N = 3$ bits to be removed from their pixel values. I tried to find a picture with high frequency in changes or better say with a chaotic pattern. The **Figure 7** shows this image which is a picture of some birds (probably sea gulls) flying with the blue sky in the background. I also created a test image to use as the subject image which simply has three sentences in it saying "MSG TO TRANSFER!!!". I used $N = 5$ to hide this image in the cover image and the resulting image and the recovered image is shown in the **Figure 7**. As you can see, there's not much difference between the original image and the resulting cover image as the change frequency is very high in the original image and the subject image is a simple one. Even though I used 5 lower bits of the original image, still, there is very little effects of the steganography visible in the resulting image. This experiment shows that, the more complex and sophisticated pattern the cover image contains, the more number of bits one might be able to use to hide other information. Also, if the data which needs to be hidden in the cover image is a simple image might have some impacts on the N value.



Figure 3: Results of applying different N values to the blacken feature. As you can see the effects of the blackening starts to appear from the 4th image and the 7th image is completely blackened.



Figure 4: The images in this figure are respectively, the original lena's picture, original bike picture, the resulting cover image with the bike image hidden in it with N = 3 and the last image is the retrieved image.



Figure 5: The images in this figure are respectively, the original lena's picture, original bike picture, the resulting cover image with the bike image hidden in it with $N = 4$ and the last image is the retrieved image.



Figure 6: The images in this figure are respectively, the original lena's picture, original bike picture, the resulting cover image with the bike image hidden in it with $N = 5$ and the last image is the retrieved image.

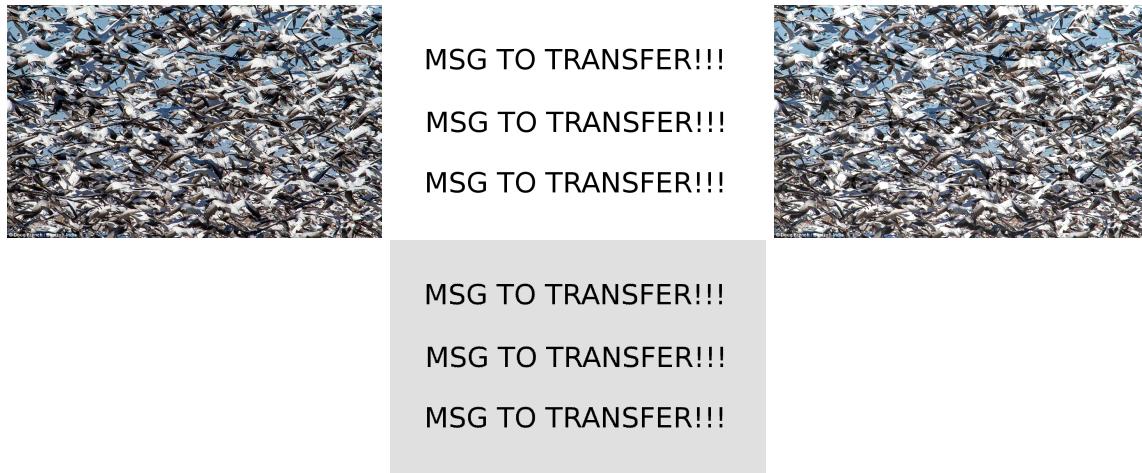


Figure 7: The experiment showing that we can use higher values for N and without showing any visible effect in the resulting image. From left to right: The original cover image, The subject image to be hidden, The resulting cover image, The retrieved image.

V. Suggestions

As for suggestions to make it more difficult for the cracker to recover the hidden image, I can say that we can use a pattern to put the subject data inside the original image instead of just storing the data in the same order as it was in the original image. This can scramble the recovered data and make a little more difficult for the cracker to figure out the pattern. Also, we can use variable bits to store the data such that, by having a pattern, we can change the number of bits that we use in the pixel value in the cover image to

hide the subject data. This might also help in hiding the data more effectively. Breaking up the data into multiple images, encrypting the data and then storing them and many other ways of data hiding techniques can be used in this application.