

# Self recovering capabilities in images using the BPCS Steganography algorithm

Hendrik J Kolver

October 6, 2014

## Abstract

Abstract of this paper and what I wrote in it. This will give the reader a short overview of what they can expect to find in this paper.

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Introduction . . . . .	2
1.2	Problem statement . . . . .	2
1.3	Research methodology . . . . .	3
1.4	Terminology . . . . .	3
1.5	Conclusion . . . . .	4
<b>2</b>	<b>Self-recovering images</b>	<b>4</b>
2.1	Introduction . . . . .	4
2.2	Steganography . . . . .	4
2.3	Fragile Watermarking . . . . .	4
2.4	Self embedding and recovery . . . . .	5
2.5	Quality Trade-off . . . . .	5
2.5.1	Increasing quality with capacity . . . . .	5
2.5.2	High capacity embedding algorithms . . . . .	6
2.6	Conclusion . . . . .	6
<b>3</b>	<b>Current self recovering image schemes and algorithms</b>	<b>6</b>
3.1	Introduction . . . . .	6
3.2	Overview . . . . .	7
3.2.1	Erasure channel model utilizing the remaining authentic content . . .	7
3.2.2	Method using difference expansion . . . . .	7
3.2.3	Dynamic block allocation for self embedding . . . . .	8
3.3	Comparison . . . . .	8
3.4	Conclusion . . . . .	9

<b>4 Implementation</b>	<b>9</b>
4.1 Introduction . . . . .	9
4.2 Embedding . . . . .	9
4.2.1 BPCS Steg . . . . .	9
4.2.2 Fragile Watermark . . . . .	11
4.3 Extraction and Reconstruction . . . . .	13
4.3.1 Image authentication . . . . .	13
4.3.2 Image reconstruction . . . . .	13
4.4 Conclusion . . . . .	14
<b>5 Results</b>	<b>14</b>
5.1 Introduction . . . . .	14
5.2 Conclusion . . . . .	17
<b>6 Interpretation of results</b>	<b>18</b>
6.1 Introduction . . . . .	18
6.2 Conclusion . . . . .	19
<b>7 Conclusion</b>	<b>19</b>

# 1 Introduction

## 1.1 Introduction

This paper proposes a novel method for self-recovering images.

**Self-recovering images:** Self-recovering images are images that have their own image content embedded into themselves using image steganography techniques. This content can then be extracted from the image at a later stage in order to reconstruct the image to an approximate original state in the case that the image was damaged.

Current self-recovering image techniques generally have poor quality restoration capabilities, damage the original image greatly or does not allow a large amount of the original image to be damaged before recovery is no longer possible. The proposed method will focus on improving the quality of self-recovering images.

## 1.2 Problem statement

The proposed method attempts to achieve the following

- The proposed method should achieve a high quality image after the self-embedding process as well as after the restoration process with a reasonable tamper rate(compared to currently available methods)

- There should not be clear visual evidence that the image has been modified after embedding as well as after restoration and the image should still carry its original semantic meaning

### 1.3 Research methodology

Firstly a literature review was conducted to determine the currently available methods for self-recovering images. After a shortcoming was identified the proposed method was designed and then implemented in the form of a Java software application in order to attempt to improve on the identified shortcoming. The results of the proposed method were then compared to the results of other relevant self-recovering image methods.

### 1.4 Terminology

This paper makes use of several terms and acronyms. This section serves to make the meaning of those terms clear.

- Original Image: This refers to the original image without any modifications.
- Embedding: Embedding in the context of this paper refers to the process of storing digital information in an image's content in such a way that the original image is still distinguishable.
- Image after embedding: This refers to the original image after the embedding process has been applied.
- Image after reconstruction: This refers to the reconstructed version of the original image that was obtained by applying the reconstruction process to the image after embedding.
- LSB: This refers to the least significant bit in a binary string.
- MSB: This refers to the most significant bit/bits in a binary string.
- Section, Image section, Pixel section: This refers to a section of 8x8 pixels in an image.
- Bit plane: A bit plane is a 8x8 grid representation of a specific bit value's for a specific pixel section. Each pixel has 24 bits which in total represents each colour, 8 bits per colour. These bits are the 8 binary bits that represent the numerical value of the colour. These 8 bits are separated from each other and are each stored in a 8x8 grid in the same location as that of the pixel in the pixel section. This process is repeated for each pixel in the pixel section and the result is 8, 8x8 grids of binary numbers. Each one of these 8 grids represent a single bit plane.
- Block: This refers to an individual bit plane of a specific section of the image. There are 8 such blocks in each section since each pixel is represented by 8 bits for each colour.
- BPCS Steganography: Bit-Plane Complexity Segmentation Steganography

## 1.5 Conclusion

This chapter discussed the purpose behind self-recovering images as well as the aim of this research and the proposed method. The research methodology followed was described. A list of terminologies used in this paper was also given. The next chapter will give a more detailed overview of all the components that make up self-recovering images.

# 2 Self-recovering images

## 2.1 Introduction

//TODO Self-recovering images are images that have their own image content embedded into themselves using image steganography techniques. This content can then be extracted from the image at a later stage in order to reconstruct the image to an approximate original state in the case that the image was damaged.

In order to understand the self-recovering system proposed in this dissertation, one has to understand the underlying technologies. This chapter will look at image steganography in more detail in section 2.2 and at watermarking in section 2.3 Self-embedding and recovery will be discussed in section 2.4 as well as the inherent quality trade off in section 2.5 with a conclusion in section 2.6

Self recovering images poses an interesting problem.

//TODO insert formal definition here

There is an inherent trade-off between the quality of the image after embedding and the quality of the image after recovery.

## 2.2 Steganography

Steganography is the process of hiding information in such a way that it does not get detected [5]. Steganography, derived from Greek means "covered writing". This means it differs from cryptography which does not attempt to hide information but merely scrambles it so it cannot be understood. Steganography does not scramble the information but rather relies on the information to remain hidden.

## 2.3 Fragile Watermarking

Fragile watermarking is a variation of traditional watermarking where a watermark is embedded into an image such that subsequent alterations to the watermarked image can be detected with high probability [7]. This means that if modifications are made to the watermarked image it would be possible to detect that modifications were made as well as what parts of the image were modified. This is a necessary component in self recovering images because in order to be able to recover image data it first needs to be determined if the image after embedding has been modified and if so what parts of the image was modified. The proposed method will use this information to determine which parts of the image should be recovered from the image reference content. A fragile watermark is used for this purpose.

## 2.4 Self embedding and recovery

Self embedding is a scheme where image content is embedded into the image itself. This embedded content can then later be used to recover damaged or modified parts of the image without accessing the original image itself [3]. A self embedding and recovery scheme (Further referred to as reconstruction) such as the proposed method contains a couple of high level steps. Firstly the authentic image content is embedded into the original image along with a fragile watermark. The resulting image is referred to as the image after embedding. This watermark will help determine later on which parts of the image after embedding was modified if any (see 2.3). When the authentic image needs to be reconstructed, after modification, the fragile watermark is used to determine which content was modified. An attempt is then made to extract the original content for the areas that were modified in the image after embedding. If this process is successful the user is left with a high quality reconstruction of the original image.

## 2.5 Quality Trade-off

There is an inherent quality trade-off present within self embedding schemes. The trade-off exists between the image after embedding and the image after reconstruction. If an embedding algorithm such as LSB(Least significant bit) embedding is used (as in 3.2.1) the actual bit values of the original image are overwritten. Naturally the more bit values are overwritten the more noise is introduced into the image. This effectively reduces the quality of the image after embedding.

The solution to this would be to simply embed a small amount of information into the image to keep the noise low. However this presents another problem. The quality of the image after reconstruction is dependent on the amount of authentic image data (further referred to as the image reference content) embedded into the image.

To achieve maximum quality after embedding the embedded information must be as little as possible. To achieve maximum quality after reconstruction the embedded image reference content must be as large as possible to be as close as possible to the original image. There exists then a trade-off between the two.

### 2.5.1 Increasing quality with capacity

It might be possible to increase the quality of the image after reconstruction while still keeping the quality after embedding high. It is proposed that applying an embedding algorithm designed for high capacity while retaining quality (as discussed in 2.5.2) to the problem of self recovering could possibly increase the quality of the image after embedding as well as after reconstruction. This is because such an algorithm should provide higher embedding capacity while producing less noise in the image.

### 2.5.2 High capacity embedding algorithms

The proposed method in this paper attempts to apply the algorithm for BPCS Steganography (Bit-Plane Complexity Segmentation Steganography) [1] to the problem of self recovering images, specifically to increase overall quality.

The goal of the BPCS algorithm is to embed as much data as possible into a cover image without detection [1]. This refers to detection by human perception as well as statistical analysis although the latter is not the focus of this paper. The BPCS algorithm uses an adaptive scheme to classify blocks that are suitable for embedding (similar to 3.2.3). BPCS steg differs however in the fact that the blocks can be conjugated in a checker board pattern to increase complexity and decrease detectability if needs be. BPCS steg also differs in the way in which embedding takes place. Instead of always embedding into the 3 LSB planes as in 3.2.3. Bit planes are dynamically allocated for embedding. One section may thus only embed into one bit plane while another may use all 7 available bit planes.

The BPCS algorithm can achieve a very high capacity while still remaining undetectable through human perception. This makes the BPCS algorithm a possible good choice to increase the capacity of a self recovering image scheme and possible increase the quality through this process.

## 2.6 Conclusion

//TODO

# 3 Current self recovering image schemes and algorithms

## 3.1 Introduction

//TODO

This section compares current Content reconstruction algorithms using self embedding. This serves to give an overview of what is currently available and to possibly highlight shortcomings of the current algorithms. This would be useful to determine if the proposed method does indeed improve on existing methods. It would also help highlight strengths and shortcomings in the proposed method. The methods analyzed were chosen because they all provide good quality images after embedding as well as good quality images after reconstructed. They also each offer an unique approach or aspect to the self recovering image problem and would thus provide a good overview of the different approaches that have already been tested.

## 3.2 Overview

### 3.2.1 Erasure channel model utilizing the remaining authentic content

The method proposed in [6] uses an erasure channel as a model for the content reconstruction problem. The method uses LSB embedding to embed the image reference content into the image itself. The method also uses a global spreading technique to spread the image reference content across the image. They also propose that by using the remaining authentic content in the image it is possible to have a high tamper rate while at the same time achieving good quality images before and after reconstruction.

The method proposed in [6] achieves good quality images after embedding with a PSNR  $>35\text{dB}$ . The method also achieves an image after reconstruction quality of  $35\text{dB} < \text{PSNR} > 40\text{dB}$  and  $40\text{dB} < \text{PSNR}$  with tamper rates of 50% and 33% respectively. The method does not let the quality of the image after reconstruction deteriorate much if the tampering rate is increased up to a value of 50%. They do however note that they can only achieve minimal reconstruction performance increases by decreasing the amount of information in the image reference content. By using only 50% of the available capacity the maximal tampering rate increases from 50% to 59%.

This method [6] is thus quite robust since 50% of the image may be tampered with before reconstruction ability starts to deteriorate. The security for this method is also very good since the quality of the cover image is not very susceptible to visual checks. The authors did not do any statistical analysis on the image. The embedding capacity of this method is also acceptable, but because the method uses some of the authentic image data to aid in the reconstruction process the quality of the image, before and after reconstruction, is still very good even without a very high embedding capacity.

### 3.2.2 Method using difference expansion

[11] Proposed a method that uses difference expansion and generalized LSB embedding. Difference expansion works by exploiting the high redundancy that are present in images. With difference expansion the payload is embedded in the difference of pixel values. For a pair of pixel values  $(x,y)$ . [9] It should however be noted that image quality reduces rapidly as the payload size is increased when using difference expansion. The method uses these two techniques in combination to achieve a high embedding capacity while keeping distortion relatively low. The method achieves a PSNR  $>35\text{dB}$  after embedding. The method achieves an embedding capacity of 1.78bpp when using up to the 4th LSB on a 512x512 8bit gray-scale version of the Lena image.

The method's [11] quality regarding the image after reconstruction is acceptable at roughly 50%. The difference expansion this method uses provides extra space for embedding. The authors did thus not implement compression on the image data because of the extra space the difference expansion provides. The embedding capacity of this method could thus be further improved by compressing the embedded data. This could possibly lead to better quality than what their experimental results achieved at the expense of complexity. Difference

expansion thus seems a good solution to increase the embedding capacity while still keeping the distortion low.

The authors do not mention the image tamper rate that this method [11] achieves.

### 3.2.3 Dynamic block allocation for self embedding

(Qian et al) [8] proposes a method for fragile watermarking with good image restoration capabilities. The proposed method differs from the other methods mentioned in 3.2.1 and 3.2.2 due to the fact that the method does not embed the information into all blocks uniformly. The method classifies different blocks of the image according to the block smoothness. The less smooth the block is the more information will be embedded into it. The authors argue that the current methods that use a fixed embedding size for each block are inadequate since less information should be embedded into the very smooth blocks and more should be embedded into the rough blocks [8].

The advantage of using this dynamic scheme is that there would be less visual distortion to the image after embedding. This is because if the specific block is already very rough (very busy visually) the human eye would not notice small changes. If however the specific block is very smooth (very uniform visually) the human eye is more likely to notice small changes. The method thus creates less distortion in the image after embedding than fixed size methods while still retaining good image after reconstruction quality [8].

The method proposed by (Qian et al) [8] uses the 3 LSB bit planes to embed the needed information in. The method achieves good results in experimentation with a image after embedding PSNR >37dB as well as a image after reconstruction PSNR = 35dB. The method also allows for a tamper rate <35%. The authors do not analyze the security of the algorithm.

## 3.3 Comparison

Each of the methods described in 3.2.3, 3.2.1 and 3.2.2 provide good quality images after embedding as well as good quality images after reconstruction. There are however important differences. The method described in 3.2.1 offers a very good tamper rate of 50% compared to the method described in 3.2.3 which has a tamper rate of 35%. This means that a larger part of the image after embedding can be tampered with while still being able to reconstruct the image.

The method described in 3.2.2 achieves an image after embedding with a PSNR >35dB, the method described in 3.2.1 achieves similar results, however the method described in 3.2.3 achieves an image after embedding PSNR >37dB. This means that the method described in 3.2.3 would have less noise in the image after embedding and would thus have better overall quality of the image after embedding.

At almost equal tamper rates of around 33%-35% the method described in 3.2.1 achieves an image after reconstruction PSNR >40dB whereas the method described in 3.2.3 only

achieves an image after reconstruction  $\text{PSNR} = 35\text{dB}$ . The method described in 3.2.2 only achieves an image after reconstruction quality of about 50% the original image. This means that the method described in 3.2.1 would generally produce better quality reconstructed images than the methods described in 3.2.3 and 3.2.2.

This means that the method described in 3.2.1 provides the best tamper rate and the best reconstructed image quality, and the method described in 3.2.3 provides the best image after embedding quality. The method described in 3.2.2 provides decent quality of the image after embedding as well as the image after reconstruction.

All three methods thus provide good results and would serve as a good benchmark for comparing the method proposed in this paper.

### 3.4 Conclusion

//TODO

## 4 Implementation

### 4.1 Introduction

//TODO

### 4.2 Embedding

#### 4.2.1 BPCS Steg

In order to be able to reconstruct the image after it has been tampered with the image needs to be embedded into itself. For this embedding the proposed method used BPCS Steganography [1].

This method works with the following steps:

1. Get the binary representation of the image
2. Convert the binary representation into Gray code
3. Divide the image into bit planes
4. Divide the bit planes into 8x8 pixel blocks
5. Calculate the complexity for each block
6. Divide the image reference content to be embedded into similar binary 8x8 blocks
7. Calculate the complexity for each image reference block and conjugate if not complex enough

8. for each image block that is complex enough, replace that block with the next image reference content block
9. Convert the blocks back to binary representation
10. reconstruct the image from the processed blocks

The proposed method works with colour images which means that for each pixel there is a red, green and blue value. This means that each block effectively stores 3 bits for each pixel.

BPCS Steg has an inherent property that if any part of the image is modified the embedded message will be lost. This would obviously be a problem for the proposed method since the image will definitely be modified.

As a solution to this problem each bit plane block used for embedding is treated as a separate embedding entity. There is thus no overlap between two different blocks used for embedding. The image reference content is first processed using the method explained later in this section. Each block has 12 pixels (with a bit depth of 16 bits) of the image reference content embedded into it. An embedding map containing the location of the first pixel to be embedded in the block is also created. Since the individual image reference content pixels are embedded in order it is only necessary to know the location of the first pixel. This embedding map is then embedded into the LSB (Least significant bit) layer of that specific 8x8 block of pixels. This allows the extractor to be able to identify where all the extracted pixels belong in the image even if the image has been modified. This serves to give random access to the embedded content which is an essential property of self embedding as defined by (Korus et al. [6]). This is essential in order for the reconstruction process to function properly.

Total space available for embedding per block =  $(8*8)*3 = 64*3 = 192$

Total embedded bits =  $12*16 = 192$

**Image resizing** The image reference content that is embedded into the original image cannot be embedded with perfect quality due to space restrictions caused by the embedding algorithm. The proposed method however does not utilize compression as it is beyond the scope of this paper. The proposed method rather reduces the bit rate of the image from 24 bits to 16 which reduces the size of each pixel by 6 bits. The image's size is also reduced according to the space available for embedding. When extracting the image reference content again the assembled image reference content is scaled up to its original size of 512x512 pixels before completing the reconstruction. This is done to ensure that the pixels contained in the image reference content matches up (location wise) with the pixels in the original image.

**Block Spread** It is very important how the image reference content is spread across the image because this affects the tamper rate

[6] Has found that it is important that the image reference content should be uniformly distributed across the image during embedding and that each part of the image after should contain equal information about the image reference content. Due to the fact that the

proposed method uses an adaptive embedding algorithm it is not possible for each part of the image after embedding to contain equal information about the image reference content. The proposed method does however distribute the embedded image reference content evenly among the blocks that were identified as suitable for embedding. The proposed method uses a simple method for spreading the image reference content across the image. The image reference content to be embedded is first divided into blocks as per the BPCS steg algorithm. These blocks are subsequently stored in a list. The created list is then reversed before embedding. Reversing the list means that each corner of the image in the image reference content is embedded into the opposing corner diagonally in the original image. For example image reference content that contains the bottom right part of the original image will now be embedded into the top left part of the original image during embedding.

This creates an effect where the embedded image reference content is far away from the content that it is referencing. This is important because if a piece of the image after embedding is destroyed it would not be able to reconstruct itself if that destroyed piece's reference content was destroyed with it.

An attempt was initially made to embed the image reference content blocks randomly into the original image. This process however produced a much lower tamper rate due to the fact that the image reference content was often embedded very close to the original image content that it was referencing. The result was a lowered tamper rate. The proposed method thus rather used the more successful method described above.

**Conjugation** If an image reference content block is found to not be complex enough that block needs to be conjugated before embedding. This is important because if a block is not complex enough it cannot be detected when extracting and it may cause visual distortions to the image since it is replacing a complex block in the original image.

The block complexity is calculated by using a border complexity method. [1] That is for each binary bit that has an adjacent binary bit that is different than itself the border count is increased. The final border count is then divided with the theoretical maximum border count, which would be the border count for a grid of the same size with a checkerboard pattern.

To conjugate a block each value in the grid needs to be changed to the XOR value of the original grid value and the corresponding value in a checkerboard grid (starting with a 1 in the top left hand corner). To reverse the conjugation this process simply needs to be repeated.

This method of conjugation is a simple method that makes a block more complex while still allowing the original values of the block to be retrieved.

#### 4.2.2 Fragile Watermark

In order to be able to detect which parts of the image have been tampered with a fragile watermark can be used.

An initial attempt was made to apply the method proposed in [10]. This method is a reversible fragile watermark which means that the watermark can be removed, during the extraction and reconstruction process, to restore the original content bit by bit to what it was before watermarking, if the image was found to be authentic.

The method needed to be adapted to authenticate the image on a block level, because the image is divided into 8x8 blocks. The method requires compression of bit streams in order to embed the watermark and to be able to extract the watermark again. These bit streams are used to store the embedding locations of watermark. The bit streams are rather small in the proposed method because they are only applicable to a specific 8x8 pixel block. This is because each block is treated as a separate embedding entity. This results in the fact that there is not always enough redundancy to be able to compress the bit streams for embedding.

Another problem with this method was the large amount of noise that this method of watermarking introduces into the cover image since it also uses difference expansion as mentioned in [9].

The fragile watermark used in the proposed method embeds a calculated hash into the image that is used to authenticate the content when the image is reconstructed on the receiving end.

For each of the 8x8 pixel section in the image after embedding a hash is calculated by combining the binary of each of the 7 Most significant bit planes of that section. This binary hash stream is then used as the input to the SHA-256 hash function. The output is a 256 bit hash string for that image section. This hash string is then truncated to a length of 57 bits. This is done to enable the conjugation map as well as the embedding map that was generated when embedding the image content (4.2.1) to be stored at the end of the stream.

The SHA-256 hash used is truncated to a 57 bit hash. This means that the hash has  $y$  possible unique values

$$y = 57^2 = 3249$$

The image has 512x512 pixels that are divided into 8x8 sections. This means that there are  $k$  amount of unique sections in the image

$$x = \frac{512}{8} = 64$$

$$k = x^2 = 64^2 = 4096$$

Since  $k > y$  the proposed method is subject to the pigeonhole principle [2]. The pigeonhole principle states that since the input for the hash function ( $k$ ) is greater than the amount of unique outputs of the hash function ( $y$ ) there will be duplicate hash values produced. However since the generated hash values are calculated from the 7 MSB planes of each section and then compared to the bits stored in the LSB plane of that section it is highly

unlikely, but not impossible, that a false positive for tampering will occur. Extensive testing was conducted on a number of images and no situation was found where such a collision occurred within normal operation. However should this case arise, the proposed method would simply restore that section as well. Thus not having a major impact on the final result of the reconstruction.

The conjugation map is appended to the truncated hash string to create a binary string of 192 bits. These 192 bits are finally embedded into the LSB plane of that specific section of the image after embedding effectively replacing all the bits in that bit plane with the generated bit stream.

## 4.3 Extraction and Reconstruction

### 4.3.1 Image authentication

The first step for the image extraction and reconstruction process is to determine which, if any, parts of the image after embedding has been tampered with. This is accomplished by first completing steps 1-4 of the BPCS steg algorithm (4.2.1). for each section (section refers here to 8x8 pixel section, thus a section here contains all 8 bit planes) the authentication hash string is calculated as described in 4.2.2. The embedded hash is then extracted from the LSB plane of that block. The extracted hash is then compared with the calculated hash and if they match the section is authentic. If they do not match then that section has been tampered with.

### 4.3.2 Image reconstruction

For each authentic block the 7 MSB planes have their complexity calculated. If a bit plane block is found to be complex enough it is assumed that there is image information embedded into it. The embedded conjugation map is extracted at this point and if it is found that any of the complex enough bit plane blocks were conjugated the conjugation algorithm is applied again to reverse the conjugation.

The embedded image content is extracted at this point from the processed bit plane blocks and the embedded image is reconstructed. In order to know the location of the extracted pixels the embedding map that was embedded into the LSB bit plane of that image section is consulted 4.2.1. The reconstructed image is then scaled up back to it's original size of 512x512 pixels. The reconstructed image is then also split into bit plane blocks. This is accomplished by first completing steps 1-4 of the BPCS steg algorithm (4.2.1). The bit plane blocks generated from the image after embedding now get iterated and each block that was found not to be authentic is replaced by the corresponding reconstructed bit plane block.

This process restores the tampered areas of the image if their matching bit plane blocks could be extracted from the image. That is if the embedded blocks were not destroyed in the

image modification. Obviously if too large a part of the image has been modified it would not be possible to accurately reconstruct the image because too much of the embedded image content would be lost.

## 4.4 Conclusion

//TODO

# 5 Results

## 5.1 Introduction

//TODO

In testing the proposed method a simple experiment was set up in which the original image was processed using the embedding process described in 4.2. The results of that process can be seen in figure 1, showing the original image and the image after the embedding process. Note that in this case the Image Reference content was only embedded once



Figure 1: Embedding Once. Left: Original Image, Right: Image After embedding

All images will be evaluated using Peak Signal to Noise Ratio (PSNR) because it is the most widely used measurement for objective image quality and is easily comparable with previous research. [12] After the image after embedding was generated the PSNR of the image after embedding in relation to the original image was calculated. In this instance the PSNR after embedding was: 34.92dB

The next step in the experiment conducted was to modify the image after embedding (shown in figure 1). After the image was modified it was reconstructed using the method described in 4.3. The results of this reconstruction can be seen in figure 2, showing the image after tampering and the image after reconstruction.



Figure 2: Embedding Once. Left: Image After Tampering, Right: Image After Restoration

After the image (shown in figure 2) was reconstructed the PSNR of the image after reconstruction (figure 2) was calculated in relation to the Original Image (shown in figure 1). In this instance the PSNR after restoration was: 34.89dB.

It was found in the experimentation process that if the image reference content is only embedded once into the original image that the quality of the Image after embedding as well as the quality of the image after reconstruction was relatively high. It was however found that the tamper rate possible was quite low at <3%. To attempt to remedy this problem another instance of the experiment was run with the only independent variable being the fact that the Image reference content was now being embedded twice. The results can be seen in figure 3 and figure 4



Figure 3: Embedding Twice. Left: Original Image, Right: Image After embedding

In order to embed the image reference content twice the image reference content needs to



Figure 4: Embedding Twice. Left: Image After Tampering, Right: Image After Restoration

be half the size that is used when embedding once. This reduces the quality of the image after reconstruction from the previous 34.89dB to 32.52 dB. The tamper rate however is now increased from <3% to <35%

This experiment was also completed on the following image. First Embedding the Image reference content once (figure 5 and figure 6):



Figure 5: Embedding Once. Left: Original Image, Right: Image After embedding

The PSNR of the image after embedding (show in figure 5) was 27.94dB.

The PSNR of the image after reconstruction (show in figure 6) was 27.72dB. The tamper rate, as was the case with the first image (figure 2), was also <3%. The experiment was also conducted by embedding the Image reference content twice (figure 7 and figure 8).

The PSNR of the image after embedding (show in figure 7) was 27.93dB.



Figure 6: Embedding Once. Left: Image After Tampering, Right: Image After Restoration



Figure 7: Embedding Twice. Left: Original Image, Right: Image After embedding

The PSNR of the image after reconstruction (show in figure 6) was 26.05dB. The tamper rate, as was the case with the first image image (figure 4), was increased from <3% to <35% with a similar reduction in quality experienced.

The second image used thus behaved in a similar manner to the first image with the only significant difference being the much lower PSNR values produced.

As an experimental control the results from the experiment in the proposed method will be compared to other current methods for images with self recovering capabilities. This is discussed further in section 6.1.

## 5.2 Conclusion

//TODO



Figure 8: Embedding Twice. Left: Image After Tampering, Right: Image After Restoration

## 6 Interpretation of results

### 6.1 Introduction

//TODO

The proposed method for self-recovering images using the BPCS Steg embedding algorithm described in the preceding sections was shown to be effective. According to Appendix B in [6] the proposed method achieved Average to High quality reconstruction in the case of figure 4 and figure 2 respectively. The proposed method achieved Low quality reconstruction in the case of figure 6 and figure 8 according to [6].

The quality measures mentioned were measured using the PSNR method. [4] However determined that the validity of PSNR is limited to modifications within the same image. It is thus not reliable to compare PSNR values between two different images. With that in mind the image from figures 1 - 4 was specifically chosen because it is well used in the literature and can serve to compare to proposed method to other methods available in the literature. Such comparisons follow below.

The proposed method does not perform as well as the method described in 3.2.1. The proposed method achieves PSNR values that are about 5dB worse than the method described in 3.2.1. Compared to the method described in 3.2.2 the proposed method performs in a similar manner. The PSNR values achieved are worse than that of the method in 3.2.2 but the difference is around 1-2dB and is thus almost not noticeable to the human eye. It should also be mentioned that the proposed method uses no compression on the image reference content, similar to the method in 3.2.2. This method thus serves as a more equal comparison.

[12] Points out several shortcomings of methods of measuring and comparing image quality using methods such as PSNR. PSNR is a useful measure for objective image quality assessment to a certain degree however there are other important factors to consider. The one important factor is how a person would perceive the image quality. Another important factor

is if the meaning of the image is still clear even after reconstruction. This would typically mean being able to identify key features of the image easily. The proposed method performs well under this form a visual analysis.

It was found during experimentation that image quality can vary greatly among different image using the proposed method. This can be attributed to the fact that an adaptive embedding method is used and where the image reference content is embedded thus differs depending on the chosen original image.

The embedding rate and effects of the embedding rate seemed consistent across the different images used. The embedding rate was low where the image reference content was only embedded once into the original image and compared poorly to methods such as the one described in 3.2.3. However it was found that if the image reference content is embedded twice into the original image that the tamper rate compares equally to that of methods such as the one described in 3.2.3.

## 6.2 Conclusion

//TODO

## 7 Conclusion

The proposed method described used an adaptive embedding algorithm described in 4.2.1. The proposed method was shown to provide good quality as well as a good tamper rate. It was also shown that the quality of the image after embedding the image reference content and the image quality after reconstruction differs depending on the cover image chosen. This was attributed to the adaptive embedding algorithm used. It was also mentioned that no compression was used on the image reference content that was embedded into the original image. If compression is used it may result in higher quality of the image after reconstruction as well as the image after embedding due to fewer modifications that will be made during embedding.

## References

- [1] Steve Beaulieu, Jon Crissey, and Ian Smith. Bpcs steganography. *University of Texas at San Antonio*.
- [2] Stephen A Cook. A short proof of the pigeon hole principle using extended resolution. *Acm Sigact News*, 8(4):28–32, 1976.
- [3] Jiri Fridrich and Miroslav Goljan. Images with self-correcting capabilities. In *Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on*, volume 3, pages 792–796. IEEE, 1999.

- [4] Quan Huynh-Thu and Mohammed Ghanbari. Scope of validity of psnr in image/video quality assessment. *Electronics letters*, 44(13):800–801, 2008.
- [5] Neil F Johnson and Sushil Jajodia. Exploring steganography: Seeing the unseen. *Computer*, 31(2):26–34, 1998.
- [6] Paweł Korus and Andrzej Dziech. Efficient method for content reconstruction with self-embedding. *Image Processing, IEEE Transactions on*, 22(3):1134–1147, 2013.
- [7] Eugene T Lin and Edward J Delp. A review of fragile image watermarks. In *Proceedings of the Multimedia and Security Workshop (ACM Multimedia'99) Multimedia Contents*, pages 25–29. Citeseer, 1999.
- [8] Zhenxing Qian, Guorui Feng, Xinpeng Zhang, and Shuzhong Wang. Image self-embedding with high-quality restoration capability. *Digital Signal Processing*, 21(2):278–286, 2011.
- [9] Jun Tian. Reversible watermarking by difference expansion. In *Proceedings of workshop on multimedia and security*, pages 19–22, 2002.
- [10] Jun Tian. Wavelet-based reversible watermarking for authentication. In *Electronic Imaging 2002*, pages 679–690. International Society for Optics and Photonics, 2002.
- [11] Jun Tian. High capacity reversible data embedding and content authentication. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 3, pages III–517. IEEE, 2003.
- [12] Zhou Wang, Alan C Bovik, and Ligang Lu. Why is image quality assessment so difficult? In *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, volume 4, pages IV–3313. IEEE, 2002.