Self recovering capabilities in images using the BPCS Steganography algorithm

Hendrik J Kolver

October 23, 2014

Abstract

Many current self-recovering image methods exist, however few of the current methods provide good quality images after embedding as well as after reconstruction. This is due to an inherent trade off between the amount of information that can be embedded into an image and the quality of the image after embedding. This dissertation proposes a novel method for increasing the quality of the image after embedding as well as the quality of the image after reconstruction in the self-recovering image problem using the BPCS Steganography algorithm for content embedding as well as employing a fragile watermark for detecting image modifications. The proposed method achieves a good quality image after embedding as well as a good quality image after reconstruction when compared to some of the best currently available self-recovering image methods.

Contents

1 Introduction

1.1 Introduction

During communication images can by modified either intentionally or unintentionally. These modifications can be small changes that does not change the meaning of the image, or bigger changes, for example the removal of key aspects from the image. Various techniques exist for authenticating images of which self-recovering is one technique. Self-recovering techniques attempt to preserve image content by embedding the image within itself to enable the reconstruction of the image if the image was tampered with. Self-recovering techniques use steganography, a technology used to embed information in digital media [8], to hide the image content within the image. Self-recovering techniques also use fragile watermarks which is a technology commonly used in image authentication, to determine which areas of an image was tampered with.

Most self-recovering techniques have a trade-off concerning the quality of the image after embedding and the quality of the image after recovery. If the one's quality is increased the other's quality will be reduced.

This dissertation proposes using a high capacity steganography algorithm called Bit-plane Complexity Segmentation (BPCS) steganography to increase the amount of data that can be embedded in an image to attempt to increase the quality of the embedded image.

1.2 Problem statement

The proposed method aims to answer the following questions

- Can a method be proposed that will achieve a high quality image after the self-embedding and restoration process as well as a reasonable tamper rate? How does such a system compare to currently available methods?
- Determining the quality of the resulting image should include, but are not limited to:
  - There should not be clear visual evidence that the image has been modified after embedding as well as after restoration
  - The image should still carry it's 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 dissertation 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 dissertation 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 the original image after the embedding process has been applied.

- Reconstruction: This refers to the process of using the content that was embedded into the image to repair the image after embedding if it has been modified.
- 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 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.

1.5 Chapter layout

- Chapter 1 serves as an introduction to the concept of self-recovering images
- Chapter 2 discusses self-recovering images and their different components
- Chapter 3 discusses currently available self-recovering image schemes, comparing their strengths and weaknesses
- Chapter 4 provides an overview of the system proposed in this dissertation

- Chapter 5 discusses the results achieved during experimentation with the proposed system.
- Chapter 6 Provides a conclusion, determining questions stated in section 1.2 have been answered

1.6 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 dissertation 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

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 parts of the image was damaged without the need to access the original image [6].

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 with a conclusion in section 2.5

2.2 Steganography

Steganography is the science that involves communicating secret data in an appropriate multimedia carrier eg images. It comes under the assumption that if the feature is visible,

the point of attack is evident, thus the goal is always to conceal the very existence of the embedded data [2]. 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.

The proposed method uses steganography to embed the information into the image in a way that is not easily visibly detectable by a human observer. A successful steganographic method has certain characteristics according to literature:

- Invisibility: The invisibility of a steganographic algorithm is very important. The strength of steganography lies in its ability to be unnoticed by the human eye. If one can see that an image has been tampered with then invisibility has been lost and the algorithm has been compromised [14].
- Robustness: The resistance to various image processing methods, compression and other image modifications. Thus how much of the image can be modified without destroying the embedded content [2].
- Capacity: The maximal number of bits that can be embedded without introducing statistically of visually detectable artifacts [5].

Least significant bit embedding (LSB) is a steganographic method that can be used to embed information. LSB works by replacing the least significant bit (or n least significant bits) of an image colour value with a binary bit from a message. The least significant bit of a colour value is the binary bit/s that carry the least value and thus the change in colour is not perceptible to the human eye. LSB embedding is the most common and easiest method for embedding messages in an image [11], and many methods exist for detecting the hidden information

Bit plane complexity steganography (BPCS) is another steganographic embedding method that was developed in 1999 by E. Kawaguchi and R.O. Eason. [9]. The goal of the BPCS al-

gorithm is to embed as much data as possible into a cover image while maintaining invisibility [1].

The BPCS algorithm consists of 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 message to be embedded into similar binary 8x8 blocks
7. Calculate the complexity for each message and conjugate if not complex enough
8. For each image block that is complex enough, replace that block with the next message block
9. Convert the blocks back to binary representation

The BPCS algorithm is a high capacity embedding algorithm [1]. This refers to detection by human perception as well as statistical analysis although the latter is not the focus of this dissertation. The BPCS algorithm uses an adaptive scheme to classify blocks that are suitable for embedding. and blocks can be conjugated in a checkerboard pattern to increase complexity and decrease detectability if needs be. 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.

2.3 Fragile Watermarking

Fragile watermarks are used to determine if a piece of watermarked digital content has been tampered with. A fragile watermark system should be able to distinguish tampered areas from non-tampered areas without referring to the original digital content [13].

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 [12]. This means that if modifications are made to the watermarked image it would be possible to detect that modifications were made as well as which 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 which parts of the image was modified.

2.4 Self embedding and recovery

Self embedding is a scheme where image content is embedded into the image itself [6]. The embedded content can then later be used to recover damaged or modified parts of the image without accessing the original image itself. 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 The 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, if any. An attempt is then made to extract the original content for the areas that were modified in the image. If this process is successful the user is left with a high quality reconstruction of the original image.

There is an inherent quality trade-off present within self embedding schemes. The trade-off exists between the quality of the image after embedding and the quality of the image after

reconstruction. If an embedding algorith such as LSB embedding is used 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 but increases the amount of data that can be stored in the image thus increasing the quality of the reconstructed image.

On one hand one could 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 (referred to as the image reference content) embedded into the image.

To achieve maximum quality after embedding the embedded information must be as small 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 thus exists then a trade-off between the two.

The proposed method attempts to apply the algorithm for BPCS steganography to the problem of self-recovering images, specifically to increase overall quality. The BPCS algorithm can achieve a very high capacity while still remaining undetectable to human perception. This makes the BPCS algorithm a good choice to increase the capacity of a self-recovering image scheme and possibly increase the quality through this process.

2.5 Conclusion

This chapter gave important background information regarding steganography in general as well as two steganographic embedding methods, LSB embedding and BPCS steganography. Fragile watermarking was discussed in section 2.3 as well as self embedding and recovery in section 2.4. The next chapter will discuss related work in existing self recovery techniques.

3 Current self-recovering image schemes and algorithms

3.1 Introduction

This chapter compares current content reconstruction algorithms using self recovery techniques serving as 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 analysed were chosen because they all provide good quality images after embedding as well as good quality images after reconstruction. They also each offer an unique approach or aspect to the self-recovering image problem.

3.2 Overview

There are 3 current methods discussed in this section. An erasure channel model utilizing the remaining authentic content is discussed in section 3.2.1, a method using difference expansion in section 3.2.2 and a method using dynamic block allocation in 3.2.3. A comparison between these 3 methods are provided in section 3.3. These 3 methods will be discussed based on their PSNR, tapering rate, security, payload size and embedding capacity as follows.

- Peak signal to noise ratio (PSNR): PSNR is a commonly used metric for objective signal quality measurement. PSNR provides a quality measurement based on the squared error between the original and processed signal [20]. PSNR is represented as a logarithmic decibel scale. In terms of images PSNR can objectively measure the amount of noised introduced into an image when compared with the original image. As the mean squared error between the two images approaches 0 the PSNR approaches infinity. A higher PSNR value thus provides a higher quality image. A small PSNR value implies a high numerical difference between the two images [7]. The quality measures mentioned in the 3 methods discussed were measured using the PSNR method.

- Tampering rate: Tampering rate is the percentage of image content that has been modified in the image after embedding [16]. If the tampering rate is too high the

embedded image content would be lost and reconstruction would not be possible.

- Security: Security refers to how easy it is to detect that information has been embedded into the image using either visual or statistical methods. Good security implies that it is difficult to detect and poor security implies that it is to simple detect [5].

- Payload size: Payload size is defined as the amount of bits that is embedded into an image using steganographic embedding techniques [3]. Embedding capacity is defined as the **maximum** amount of bits that can be embedded into an image using steganographic embedding techniques before visual distortion occurs.

3.2.1 Erasure channel model utilizing the remaining authentic content

The method proposed in [10] uses an erasure channel as a model for the content reconstruction problem. An erasure channel is a channel that can transmit a bit of information and the receiver can either receive the bit or it receives a message that the bit was not received, ie the bit was erased [10]. The method uses LSB embedding to embed the image reference content into the image itself and uses a global spreading technique to spread the image reference content across the image. The method also proposes 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 [10] achieves good quality images after embedding with a PSNR >35dB. The method also achieves an image after reconstruction quality of 35dB <PSNR >40dB and 40dB <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%. The authors do however note that they can only achieve a minimal increase in reconstruction performance by decreasing the amount of information in the image reference content. By using only 50% of the available capacity the maximal tampering rate increases form 50% to 59%.

This method [10] is robust since 50% of the image may be tampered with before reconstruc-

tion ability starts to deteriorate. The security for this method is also very good since the quality of the cover image is not susceptible to visual checks. The authors did not do any statistical analysis on the image. The embedding capacity of this method is acceptable 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 thus good even though the method only has acceptable embedding capacity.

3.2.2 Method using difference expansion

Another method [19] uses difference expansion and generalized LSB embedding. Difference expansion works by exploiting the high redundancy that are present in images [17]. With difference expansion the payload is embedded in the difference of pixel values [17]. It should be noted that image quality reduces rapidly as the payload size is increased when using difference expansion due to the large amount of change that is introduced into the image. The method achieves a high embedding capacity while keeping distortion relatively low. The method achieves a PSNR >35dB after embedding. The method achieves an embedding capacity of 1.78 bits per pixel (bbp) when using up to the 4th LSB on a 512x512 8bit grayscale version of the Lena image. Lena is the name given to a standard test image widely used in the field of image processing since 1973 [15].

The method's [19] 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.

The authors did not mention the image tamper rate that this method [19] achieves.

3.2.3 Dynamic block allocation for self embedding

Qian et al [16] proposed 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 [16].

The advantage of using this dynamic scheme is that there is very little visual distortion introduced into the image after embedding. This is because if the specific block is already very rough (very busy visually) the human eye does 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 [16].

The method proposed by Qian et al [16] uses 3 LSB bit planes to embed the necessary 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 did not analyze the security of the algorithm.

3.3 Comparison

Each of the methods described in 3.2.1, 3.2.2 and 3.2.3 provide good quality images after embedding as well as good quality images after reconstruction. There are however important differences. The erasure channel method described in 3.2.1 offers a very good tamper rate of 50% compared to the difference expansion 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 difference expansion method described in 3.2.2 achieves an image after embedding with a PSNR >35dB and the erasure channel method achieves similar results, however the dynamic block allocation method achieves an image after embedding PSNR >37dB. This means that the dynamic block allocation method 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 erasure channel method achieves an image after reconstruction PSNR >40dB whereas the dynamic block allocation method only achieves an image after reconstruction PSNR = 35dB. The difference expansion method only achieves an image after reconstruction quality of about 50% the original image. This means that the erasure channel method would generally produce better quality reconstructed images than the other two methods.

The erasure channel method thus provides the best tamper rate and the best reconstructed image quality, and the dynamic block allocation method provides the best image after embedding quality. The difference expansion method 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 dissertation.

3.4 Conclusion

This chapter looked at currently available methods for self-recovering images. An erasure channel model utilizing the remaining authentic content in section 3.2.1, a method using difference expansion in section 3.2.2 and a method using dynamic block allocation in 3.2.3. A comparison between these methods was provided in section 3.3. This information will be used to determine the success of the self-recovering image method proposed in this dissertation by comparing the results from the proposed method with the results achieved by the methods

discussed in this chapter. The next chapter will discuss the implementation of the of the self-recovering image method proposed in this dissertation.

4 Overview of the proposed system

4.1 Introduction

This chapter discusses the details of the implementation for the self-recovering image method proposed in this dissertation. Embedding is discussed in section 4.3 and BPCS steganography and fragile watermarking will also be discussed again. The extraction and restoration process is discussed in section 4.4 under which authentication and restoration will also be discussed. Diagrams are provided to visualize the embedding as well as the extraction process.

4.2 Overview of proposed system

The self-recovering method proposed in this dissertation uses BPCS steganography to increase the payload capacity of the cover image in order to be able to embed more image reference content into the image without introducing visual distortion. In addition a fragile watermark was used to detect which parts of the image is damaged and thus needs to be reconstructed.

4.3 Embedding

In the embedding phase BPCS steganography and fragile watermarks were used. How these two technologies were implemented are discussed in sections 4.3.1 and 4.3.2

4.3.1 BPCS steganography

BPCS steganography 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

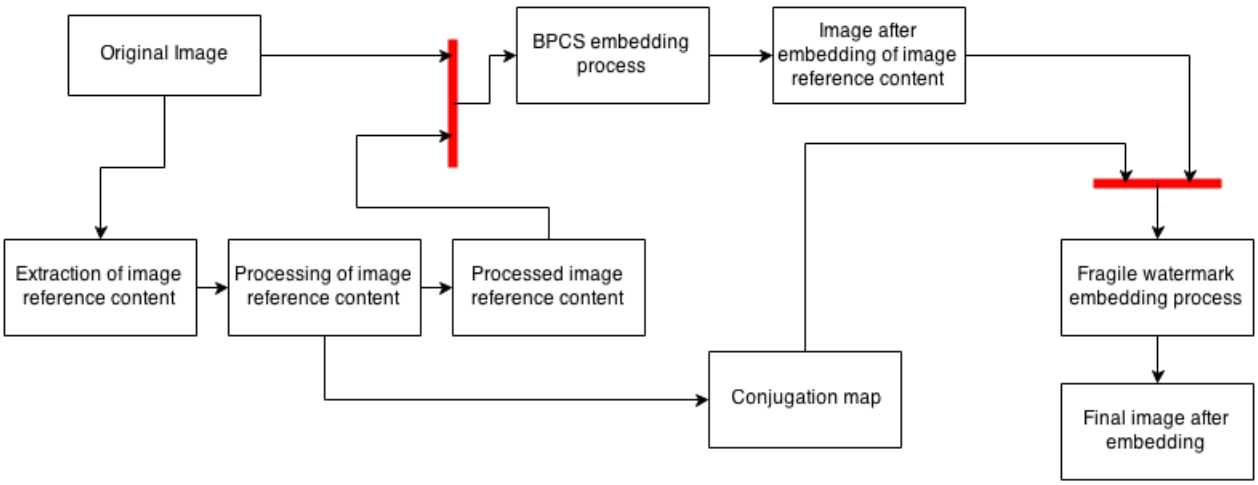

Figure 1: The embedding process of the proposed self-recovering method

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. [10]. This is essential in order for the reconstruction process to function properly.

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.

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 dissertation. 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 it's 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. [10] 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 steganography 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 proposed method thus rather used the more successful method describe 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]. 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. A conjugation map is also created to indicate which blocks have been conjugated. This is needed since the receiver of the image would not be able to determine if a block has been conjugated or no without such a map. The conjugation map is also embedded into the image.

4.3.2 Fragile Watermark

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

An initial attempt was made to apply the method proposed in [18]. 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 resulted in the fact that there was 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 [17].

The fragile watermark used in the proposed method thus 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 sections 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 (4.3.1) 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 [4]. 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 it does not have 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.

4.4 Extraction and Reconstruction

In the extraction and reconstruction process of the proposed method the image is first authenticated, this process is discussed in section 4.4.1. The image reference content is then extracted and the image is restored to an approximation of the original image before embedding, this process is discussed in section 4.4.2.

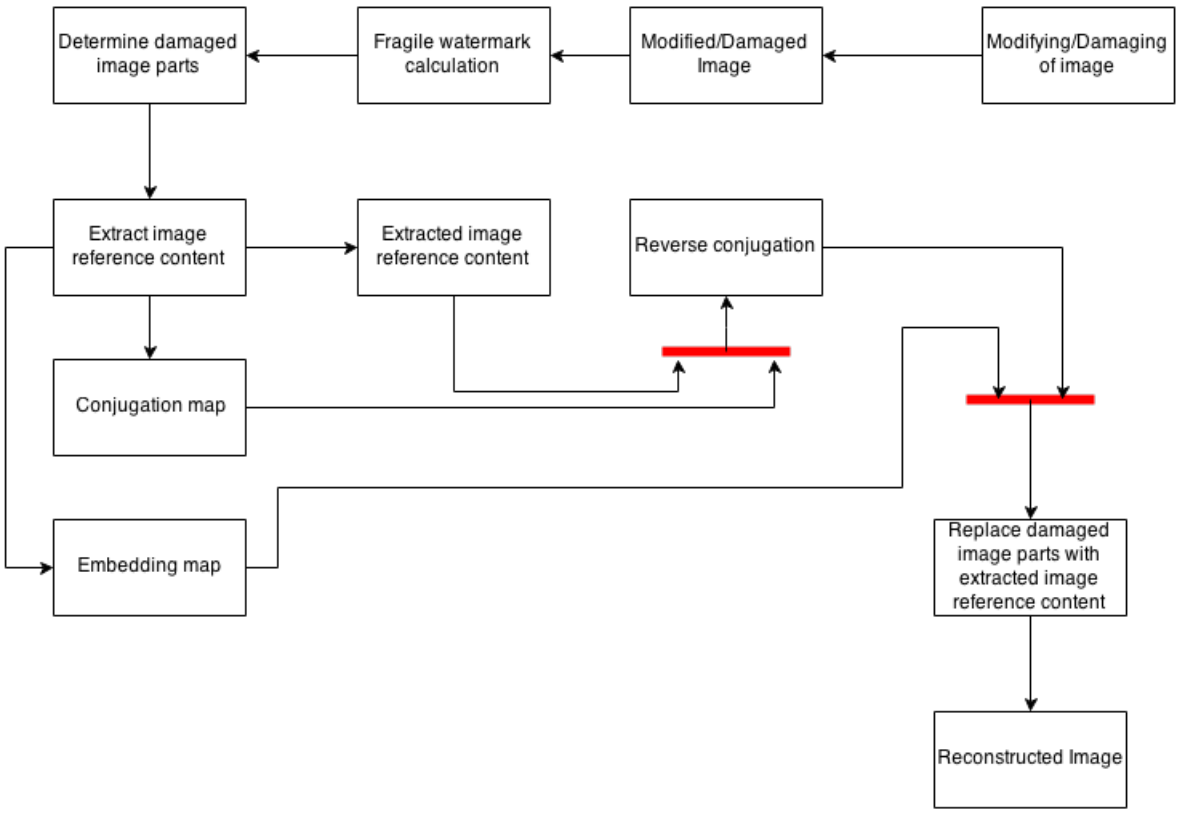

Figure 2: The extraction process of the proposed self-recovering method

4.4.1 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 (4.3.1). 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 4.3.2. 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 and needs to be reconstructed.

4.4.2 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. 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 steganography algorithm (4.3.1). The bit plane blocks generated from the image after embedding are traversed 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.

4.5 Conclusion

This chapter discussed the proposed self-recovering system. An overview was given of the proposed method in section 4.2. Embedding was discussed in section 4.3 under which BPCS steganography as well as fragile watermarking was also discussed in sections 4.3.1 and 4.3.2. The extraction process was discussed in section 4.4 under which authentication and reconstruction was also discussed in sections 4.4.1 and 4.4.2. Diagrams were also provided to visualize the embedding process in figure 1 and the extraction process in figure 2. The next chapter will discuss the experimental results achieved using the proposed method.

5 Experimental Results

5.1 Introduction

This chapter discusses the experiments conducted to test the proposed method. The results of the conducted experiments are also interpreted and evaluated.

5.2 Experiments conducted

In testing the proposed method a simple experiment was set up in which the original image was processed using the embedding process described in 4.3. The results of that process can be seen in figure 3, 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 3: 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. [21] 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 3). After the image was modified it was reconstructed using the method described in 4.4. The results of this reconstruction can be seen in figure 4, showing the image after tampering and the image after reconstruction.

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

After the image (shown in figure 4) was reconstructed the PSNR of the image after reconstruction (figure 4) was calculated in relation to the original Image (shown in figure 3). 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 5 and figure 6

In order to embed the image reference content twice the image reference content needs to 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 $<35\%$

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

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

This experiment was also completed on the following image. First by embedding the image reference content once (figure 7 and figure 8):

The PSNR of the image after embedding (show in figure 7) was 27.94dB.

The PSNR of the image after reconstruction (show in figure 8) was 27.72dB. The tamper rate, as was the case with the first image (figure 4), was also <3%. The experiment was also conducted by embedding the image reference content twice (figure 9 and figure 10).

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

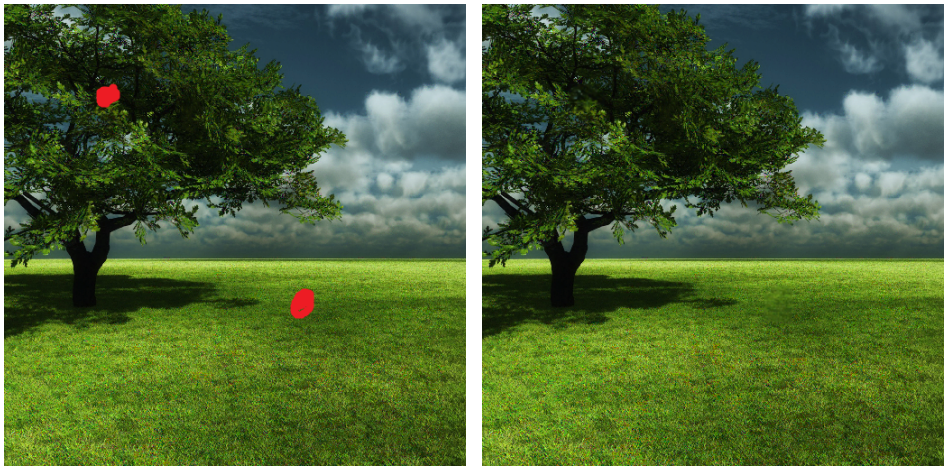

Figure 8: Embedding Once. Left: Image After Tampering, Right: Image After Restoration

The PSNR of the image after embedding (show in figure 9) was 27.93dB.

The PSNR of the image after reconstruction (show in figure 8) was 26.05dB. The tamper rate, as was the case with the first image image (figure 6), was increased from <3% to <35% with a similar reduction in quality experienced. //TODO <Summarize all findings in an additional table>

Figure 9: Embedding Twice. Left: Original Image, Right: Image After embedding

Figure 10: Embedding Twice. Left: Image After Tampering, Right: Image After Restoration

Experimental results summary			
Image	PSNR after embedding	PSNR after restoration	Tampering rate
Figure 3 - 4 (Embed once)	34.92dB	34.89dB	<3%
Figure 5 - 6 (Embed twice)	34.92dB	32.52 dB	<35%
Figure 7 - 8 (Embed once)	27.94dB	27.72dB	<3%
Figure 9 - 10 (Embed twice)	27.93dB	26.05dB	<35%

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 5.3.

5.3 Interpretation of results

The proposed method for self-recovering images using the BPCS steganography embedding algorithm described in the preceding sections was shown to be effective. According to Appendix B in [10] the proposed method achieved average to high quality reconstruction in the case of figure 6 and figure 4 respectively. The proposed method achieved low quality reconstruction in the case of figure 8 and figure 10 according to [10].

The image from figures 3 - 6 was specifically chosen for the experiment because it is well used in the literature and can serve to compare the proposed method to other methods available in the literature. Such comparisons follow below.

The proposed method does not perform as well as the erasure channel method and achieves PSNR values that are about 5dB worse than the erasure channel method. Compared to the difference expansion method the proposed method performs in a similar manner. The PSNR values achieved are worse than that of the difference expansion method 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 difference expansion method. This method thus serves as a more equal comparison.

Research in [21] 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 images 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 dynamic block allocation method. 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 dynamic block allocation method.

5.4 Conclusion

This chapter discussed the experiments conducted to test the proposed method in section 5.2. Two implementations of the proposed method were tested, embedding the image reference content once and twice into the original image. The results of the experiments were evaluated in section 5.3. The results achieved using the proposed method were compared with the results achieved by other currently available self-recovering image methods discussed in section 3.

6 Conclusion

6.1 Introduction

This chapter summarizes the findings of the proposed method and relates those findings back to the problem statement in section 1.2. Possible future work with regards to the proposed method is also discussed.

6.2 Summary

The proposed method described used the BPCS steganaography algorithm, which is an adaptive embedding algorithm described in 4.3.1, and applied the algorithm to the self-embedding problem. 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 as well as the image quality after reconstruction differs depending on the cover image chosen. This was attributed to the adaptive embedding algorithm used due to the fact that the embedding differs depending on the cover image. 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.

The proposed method achieved a high quality image after the self-embedding process as well as after the restoration process with reasonable tamper rate in one set of the images tested. The quality though is dependent on the image used. It would thus not achieve high quality with all images. The proposed method did not introduce clear visual evidence that the image has been modified after embedding as well as after restoration and the image did still carry it's original meaning. The proposed method thus achieved most of what it attempted to achieve with the only problem being that the quality is dependent on the image used.

6.3 Future work

The method proposed in this dissertation focused on the implementation of the BPCS steganography algorithm as well as the implementation of the fragile watermark. The focus was to combine these methods in order to create a high quality self-recovering image scheme. It could however be possible to increase the quality of the proposed method even further if compression is used on the image reference content that is embedded into the original image. By compressing the image reference content it would be possible to embed a higher quality reference image into the original image without increasing the payload size. It would also

be possible to reduce the amount embedding capacity used of the original image, due to a smaller amount of bits that need to be embedded, which would result in less of the original image being changed and thus less noise being introduced into the original image. This could possibly lead to increasing the overall quality of the proposed self-recovering image scheme.

References

[1] Steve Beaullieu, Jon Crissey, and Ian Smith. Bpcs steganography. *University of Texas at San Antonio*.

[2] Abbas Cheddad, Joan Condell, Kevin Curran, and Paul Mc Kevitt. Digital image steganography: Survey and analysis of current methods. *Signal processing*, 90(3):727–752, 2010.

[3] Wen-Jan Chen, Chin-Chen Chang, and T Le. High payload steganography mechanism using hybrid edge detector. *Expert Systems with applications*, 37(4):3292–3301, 2010.

[4] Stephen A Cook. A short proof of the pigeon hole principle using extended resolution. *Acm Sigact News*, 8(4):28–32, 1976.

[5] Jessica Fridrich. *Steganography in digital media: principles, algorithms, and applications*. Cambridge University Press, 2009.

[6] 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.

[7] Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 2366–2369. IEEE, 2010.

[8] Neil F Johnson and Sushil Jajodia. Exploring steganography: Seeing the unseen. *Computer*, 31(2):26–34, 1998.

[9] Eiji Kawaguchi and Richard O Eason. Principles and applications of bpcs steganography. In *Photonics East (ISAM, VVDC, IEMB)*, pages 464–473. International Society for Optics and Photonics, 1999.

[10] Pawel Korus and Andrzej Dziech. Efficient method for content reconstruction with self-embedding. *Image Processing, IEEE Transactions on*, 22(3):1134–1147, 2013.

[11] Yeuan-Kuen Lee and Ling-Hwei Chen. High capacity image steganographic model. *IEE Proceedings-Vision, Image and Signal Processing*, 147(3):288–294, 2000.

[12] 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.

[13] Shao-Hui Liu, Hong-Xun Yao, Wen Gao, and Yong-Liang Liu. An image fragile watermark scheme based on chaotic image pattern and pixel-pairs. *Applied Mathematics and Computation*, 185(2):869–882, 2007.

[14] Tayana Morkel, Jan HP Eloff, and Martin S Olivier. An overview of image steganography. In *ISSA*, pages 1–11, 2005.

[15] BBC news online. Playboy centrefold photo shrunk to width of human hair, August 2012. [Online; posted 14-August-2012; URL: http://www.bbc.com/news/technology-19260550].

[16] Zhenxing Qian, Guorui Feng, Xinpeng Zhang, and Shuozhong Wang. Image self-embedding with high-quality restoration capability. *Digital Signal Processing*, 21(2):278–286, 2011.

[17] Jun Tian. Reversible watermarking by difference expansion. In *Proceedings of workshop on multimedia and security*, pages 19–22, 2002.

[18] Jun Tian. Wavelet-based reversible watermarking for authentication. In *Electronic Imaging 2002*, pages 679–690. International Society for Optics and Photonics, 2002.

[19] 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.

[20] Deepak S Turaga, Yingwei Chen, and Jorge Caviedes. No reference psnr estimation for compressed pictures. *Signal Processing: Image Communication*, 19(2):173–184, 2004.

[21] 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.