1. Introduction — 1.1 Give a short paragraph on what this paper is about

1.2 Give your problem statement: what are you trying to solve?

1.3 ~~Terms used~~ 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: ...
- Image after embedding: This refers to the 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:
- 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

2 ~~Introduction~~ Self recovering images.

Self recovering images are...

This section gives a brief description of the concepts used in this report. The reasoning behind the proposed method will also be explained in 2.5.1 and 2.5.2. This will serve to to show the applicability of the proposed method as well as the shortcomings it attempts to address.

2.1 ~~Overview~~ Defin Formal definition of self recovering images

Self recovering images ~~(described in 2.4)~~ poses an interesting problem. There is an inherent trade-off ~~(described in 2.5)~~ between the quality of the image after embedding and the quality of the image after recovery. This report proposes a method in which a high capacity image steganography embedding algorithm (described in 2.5.2) is applied to the problem of self-recovering images to attempt to increase the quality of the images (described in 2.5.1) both after embedding and after recovery.

In each chapter:

For example: This ~~section~~ chapter ~~2.1~~ gives a definition of steganography in section 2.1, defines watermarking in section 2.2 and gives a comparison of these two technologies in section 2.3.

* steganography differs from cryptography in the sense that where cryptography scrambles information so that it cannot be understood, steganography relies on the information being hidden

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.

A- Try to refrain from using "it" since it can be confusing to understand what you are referring to.

Here I think you can add a bit more about how stego is done and explain perhaps LSB & BPCS.

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~~ 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 ~~what~~ which parts of the image ~~was~~ were 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.

Leave this part for the chapter where you describe the proposed method.

2.4 Self embedding and recovery

Self embedding is a scheme where image content is embedded into the image itself. ~~This~~ The embedded content can then later be used to recover damaged or modified parts of the image without accessing the original image itself [3] for what purpose? 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~~

Either sub-paragraph (2.4.1) or part of 2.4 or perhaps move to a later chapter?

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 3.1.1) 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.

define

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 3.1.3)~~. 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 3.1.3. 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.

3 Current self recovering image schemes and algorithms

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.

3.1 Overview

3.1.1 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 >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%. 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 form 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.

3.1.2 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 >35dB 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.

3.1.3 Dynamic block allocation for self embedding

stick to one tense – either past tense as in 3.1.1 or present tense like here

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 3.1.1 and 3.1.2 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].

How is smoothness measured? define

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].

define

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.

3.2 Comparison

Each of the methods described in 3.1.3, 3.1.1 and 3.1.2 provide good quality images after embedding as well as good quality images after reconstruction. There are however important differences. The method described in 3.1.1 offers a very good tamper rate of 50% compared to the method described in 3.1.3 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 3.1.2 achieves an image after embedding with a PSNR >35dB, the method described in 3.1.1 achieves similar results, however the method described in 3.1.3 achieves an image after embedding PSNR >37dB. This means that the method described in

3.1.3 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 3.1.1 achieves an image after reconstruction PSNR >40dB whereas the method described in 3.1.3 only achieves an image after reconstruction PSNR = 35dB. The method described in 3.1.2 only achieves an image after reconstruction quality of about 50% the original image. This means that the method described in 3.1.1 would generally produce better quality reconstructed images than the methods described in 3.1.3 and 3.1.2.

This means that the method described in 3.1.1 provides the best tamper rate and the best reconstructed image quality, and the method described in 3.1.3 provides the best image after embedding quality. The method described in 3.1.2 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.

3.3 Conclusion – Short summary of the chapter and remind the reader about what you are trying to do/solve in this research and why this chapter is important/relevant.

4 Implementation

4.1 Embedding

4.1.1 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].

I think this should move to 2.1

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