9-12-2005

High-capacity steganography using a shared colour palette

G. Brisbane
University of Wollongong, uow@brisbane.edu.au

R. Safavi-Naini
University of Wollongong, rei@uow.edu.au

P. Ogunbona
University of Wollongong, philipo@uow.edu.au

High-capacity steganography using a shared colour palette

Abstract

Seppanen, Makela and Keskinarkaus (SMK) have proposed a high-capacity steganographic technique to conceal information within a colour image. The technique is significant because of the high volume of data that is embedded into pixels but it results in a high level of noise and so the quality of the resulting image is not acceptable. A new type of coding structure is proposed, which maintains a high capacity but lowers the level of noise. Secondly, an adaptive algorithm is used to identify pixel values that have a high capacity to distortion ratio. Also the maximum size of the coding structures is limited to improve the capacity/distortion tradeoff. For the tested images, an average capacity of nearly 6 bits/pixel was achieved with a peak signal to noise ratio of 40 dB.

Disciplines

Physical Sciences and Mathematics

Publication Details

Brisbane, G., Safavi-Naini, R. & Ogunbona, P. (2005). High-capacity steganography using a shared colour palette. Vision, Image and Signal Processing, IEE Proceedings, 152 (6), 787-792.

High-capacity steganography using a shared colour palette

G. Brisbane, R. Safavi-Naini and P. Ogunbona

Abstract: Seppanen, Makela and Keskinarkaus (SMK) have proposed a high-capacity steganographic technique to conceal information within a colour image. The technique is significant because of the high volume of data that is embedded into pixels but it results in a high level of noise and so the quality of the resulting image is not acceptable. A new type of coding structure is proposed, which maintains a high capacity but lowers the level of noise. Secondly, an adaptive algorithm is used to identify pixel values that have a high capacity to distortion ratio. Also the maximum size of the coding structures is limited to improve the capacity/distortion tradeoff. For the tested images, an average capacity of nearly 6 bits/pixel was achieved with a peak signal to noise ratio of 40 dB.

1 Introduction

Information hiding (IH) is the embedding of a secret message (stego-data, $\mathcal{M}$) within an ordinary item of communication (cover-data, I). An IH system uses two algorithms to communicate: an embedding algorithm to produce the modified cover-data (I') from $\mathcal{M}$ and I; and an extraction algorithm to recover $\mathcal{M}$ from I'. Steganography is 'secret communication between two parties' and is one application area within the IH domain, along with fragile and robust watermarking ([1]). In this paper we limit ourselves to using image data as the cover-data.

Simmons allegorises the steganography problem in [2] by posing it in terms of three participants, two imprisoned accomplices and a warden. The accomplices are known as Alice and Bob, while the warden is named Wendy. Alice and Bob communicate to each other in the hope of developing an escape plan. They have also shared a short secret prior to their incarceration.

Wendy permits their communications so long as she is satisfied that the content is legible. Her purpose for doing this is to catch them planning and so prove their conspiracy. The primary objective of the prisoners is to conceal the stego-data from detection.

The three primary attributes of an IH scheme are imperceptibility, capacity and robustness [3]:

- The imperceptibility is the level of concealment, which prevents the warden from being able to distinguish between a modified cover-data and an unmodified cover-data. It incorporates both minimising the visible effect of changes to pixel values (distortion), as well as the level of detectability of the stego-data by the warden ([4]). The main measure of the distortion in this paper is the peak signal to noise ratio (PSNR), defined formally in Section 2.5.2. It is a widely used measure defined as the ratio of the peak signal to the average root mean square of the difference between the stego-data and the modified cover-data.
- The capacity of a scheme is the quantity of stego-data that can be embedded. For images this is measured in bits per pixel (bpp).
- The robustness of a scheme is the ability to recover the stego-data in spite of modifications to the modified cover-data.

For the problem of steganography, the most important requirement is high imperceptibility followed by high capacity. This is because the warden must not be able to discern, or suspect, the presence of any stego-data in I'. Robustness is of importance in scenarios where the warden can alter I'.

In this paper we consider a steganography scheme proposed by Seppanen, Makela and Keskinarkaus in [5] (the SMK algorithm for short). They proposed a method for communicating information in colour images with a capacity of between 6.9 and 13.3 bpp. The algorithm is conceptually interesting and because of its high embedding capacity could be of high interest for subliminal communication. However, our preliminary experiments with the algorithm demonstrated that the quality of the resulting image is very low and so not acceptable. The aim of this paper is to address this problem. Of course it is always possible to increase quality by embedding in a smaller number of pixels. The challenge is to increase imperceptibility and maintain a high capacity.

There is no agreed definition for a high-capacity steganographic scheme. There are many algorithms (e.g. [6, 7]) that can be used for hiding data in images, which are lossy compression tolerant, with capacity ranging up to 0.17 bpp. In addition, a number of high capacity algorithms with less emphasis on imperceptibility and robustness have been proposed.

An algorithm by Lee and Chen specifically designed for high capacity embedding is given in [4]. Their scheme modifies the least significant bits (LSBs) of pixels in an image and then attenuates the result to control the imperceptibility. They were able to embed and extract 4.06 bpp from a monochrome image which can be

IEE Proceedings online no. 20045047

doi: 10.1049/ip-vis:20045047

Paper received 7th June 2004

The authors are with the School of Information Technology and Computer Science, The University of Wollongong, NSW, 2522, Australia

E-mail: gazol@optusnet.com.au

extrapolated to 12.18 bpp for a three-colour image. The imperceptibility is moderate with a PSNR of 34.03 dB.

Kawaguchi and Eason proposed a high-capacity steganography algorithm in [8]. They divided an image into bit planes following a colour transform from the red/green/-blue domain to 'canonical Gray coding'. By statistical analysis they determine regions within each plane that contain data that appears random, which they substitute for meaningful information. They claim a capacity of up to 12 bpp for a 24 bit colour image but make no quantitative statement on the imperceptibility of their technique.

Both the above techniques do not require the original image to be accessible to the receiver nor to provide any robustness to modification by a third party. In [9], Brisbane *et al.* calculated the maximum capacity for a steganographic method with no consideration of the robustness. They considered the model where both parties possess I prior to communication. Their result gives an upper bound for steganographic techniques. It encodes the stego-data using an entropy decoder to construct a sequence of symbols with a Gaussian distribution. This shows that, for a PSNR of 40 dB, at most 3.40 bpp per colour can be communicated. The other results above were able to embed at a higher bit rate only with a lower PSNR.

2 SMK algorithm

The SMK algorithm is a steganography method for colour images [5]. A colour palette is a subset of colour points in red/green/blue (RGB) space. It can be used to achieve image compression when pixels in an image are quantised to the nearest point in the palette. The basic idea of the SMK algorithm is to construct a set of coding structures surrounding the palette elements. Each coding structure is a set of points in RGB space with each point representing a unique binary string for embedding message bits.

There are four main sub-algorithms comprising the SMK method:

- Key generation constructs a secret key and a palette (Section 2.1).
- Constructing a coding structure where a subset of size 2^k points in RGB space is labelled by k-bit strings and used for embedding message bits (Section 2.2).
- The embedding and extraction algorithms, which allow Alice and Bob to communicate (Section 2.3).

2.1 Key generation

There are two pieces of information that are required to be shared between Alice and Bob: a key K, and a palette of colours C. The key generates a randomly chosen binary string that is used to mask the stego-data, $\mathcal{M}$. This is described in more detail in Section 2.3.

The colour palette is obtained by using the k-means algorithm trained with the pixels in I. The k-means algorithm derives a set of N key feature vectors (centroids) grouped as $C = \{C_0, \ldots, C_{N-1}\}$. A feature vector is then classified by identifying the 'closest' centroid. Each centroid, C_j, is the average of all vectors in the training set that are classified to it. A Voronoi region, $\mathcal{R}_j$, is the set of all points in the feature space that are classified by C_j. For a more complete discussion of the algorithm, refer to [10].

To apply the algorithm to images, each pixel p in I is represented by the feature vector (p_r, p_g, p_b), where p_r, p_g and p_b are the values of the red, green and blue components for the pixel. This produces a set of training data for k-means. The distance in the feature space between two pixels p and q is:

$$D(p,q) = \sqrt{(p_r - q_r)^2 + (p_g - q_g)^2 + (p_b - q_b)^2} \quad (1)$$

The initial locations of the centroids and N are chosen by Alice. Using the training data, the k-means algorithm produces the set of centroids, C. This set C can now be considered as a palette for I. For consistency, we denote each item within C as a palette element, rather than as a centroid. We define $C(p)$ as the function that returns the index of the closest palette element to the pixel p.

Alice provides Bob with C and K by some secure means. The authors of the SMK scheme propose that N be 'small', although a precise definition is not given. If we assume I is a 24-bit colour image and $N = 1024$ (the maximum size discussed), then C is approximately 3 kbyte in size.

2.2 Coding structures

Let S_j be a set of size $2^{h_j} (h_j \in \mathbb{Z})$ contained within $\mathcal{R}_j$. We use v_j to denote the number of points in S_j, that is $v_j = 2^{h_j}$. Each point in S_j is assigned a unique bit label of length h_j and so there is a one-to-one correspondence between binary strings of length h_j and points within S_j.

In the SMK algorithm, each S_j is a 'cube' centred at C_j, with length l_j and refers to the set of points $\{(p_r, p_g, p_b) : p_r \in [C_{j_r} - l_j, C_{j_r} + l_j - 1], [C_{j_g} - l_j, C_{j_g} + l_j - 1], [C_{j_b} - l_j, C_{j_b} + l_j - 1]\}$, which we denote as $\mathtt{cube}(C_j, l_j)$. S_j can be constructed using algorithm 1:

Algorithm 1: Constructing a coding structure, S_j
Input: j, C_j
Output: S_j, h_j

1 $l_j = 1$
2 Do
3 $\quad l_j = 2l_j$
4 $\quad S_j = \mathtt{cube}(C_j, l_j)$
5 While S_j is entirely contained within $\mathcal{R}_j$
6 $l_j = \frac{l_j}{2}$
7 $h_j = 3\log_2 l_j$
8 Each point in S_j is assigned a unique binary label of length h_j

2.3 Embedding and extracting the SMK algorithm

The embedding algorithm is given as algorithm 2. The main requirement of this algorithm is that $C(p) = C(p')$, where p' is the result of embedding in p. $\mathcal{M}$ is converted to a uniformly distributed message to remove the potential for distortion that could occur when non-uniformly distributed messages are embedded. When Bob receives I' from Alice, he can extract $\mathcal{M}$ using algorithm 3:

Algorithm 2: Embedding $\mathcal{M}$ into I using $\{C, K\}$
Input: I, C, $\mathcal{M}$, K
Output: I'

1 Generate a uniformly distributed bit-sequence $\mathcal{M}_K$ using K (e.g. m-sequence)
2 $\mathcal{M}' = \mathcal{M} \oplus \mathcal{M}_K$, $\oplus$ is the bit-wise XOR operation
3 Use C to generate the coding structures, $S_0, \ldots, S_{N-1}$ (algorithm 1)
4 For each pixel, $p \in I$
5 $\quad j = C(p)$ (j is the index of the Voronoi region containing p)
6 Let m' be the next block of h_j bits from $\mathcal{M}'$
7 Move p to $p' \in S_j$ such that $m' = \text{label}(p')$
8 Next p

Algorithm 3: Extracting $\mathcal{M}$ from I' using $\{C, K\}$
Input: I', C, K
Output: $\mathcal{M}$

1 $\mathcal{M}' = \emptyset$
2 Generate the same bit-sequence $\mathcal{M}_K$ using K (e.g. *m*-sequence)
3 Use C to generate the same coding structures, $S_0, \ldots, S_{N-1}$ (algorithm 1)
4 For each pixel, $p' \in I'$
5 $\quad \mathcal{M}' = \mathcal{M}' \| \text{label}(p')$ (where $\|$ is appending)
6 Next p
7 $\mathcal{M} = M' \oplus M_K, \oplus$ is the bit-wise XOR operation

2.4 Experimental framework

All experiments described in this paper are the average of ten experiments, using ten predetermined pseudorandomly chosen values of the secret key K. The images used for testing are 'Airplane', 'Mandrill' and 'Peppers', found at [11]. 'Airplane' is of size 512×512 pixels, whereas the others are 256×256 pixels. The 'Airplane' image contains the least amount of detail, followed by 'Peppers' and then 'Mandrill'.

Unless otherwise stated, all experiments use a benchmark imperceptibility of a PSNR of 40 dB. At this threshold, I' is considered to be visually indistinguishable from I.

2.5 Performance of SMK algorithm

2.5.1 Capacity: Figure 1 shows the results of our implementation of the SMK algorithm. The capacity for $N = 32$ ranges between 7.9 and 11.9 bpp for the test images, which matches the authors' reported values.

2.5.2 Distortion: As mentioned previously, the imperceptibility includes both the goals of reducing distortion and reducing the detectability of the stego-data. In this paper, we focus on reducing the distortion, as measured by the peak signal to noise ratio (PSNR):

$$\text{PSNR} = 20 \log_{10} \frac{255}{\sqrt{\text{MSE}}}$$

where MSE is the mean squared error for each pixel in I and I'. Although the PSNR is not a perfect measure of visual distortion (see e.g. [12]) it is sufficient to describe the average difference.

Figure 1 demonstrates that, in nearly all images and values of N, the level of imperceptibility is lower than our accepted threshold. For $N = 32$, the PSNR ranges between 31.5 and 27.4 dB.

2.5.3 Detectability: The detectability of a steganographic scheme is measured by the success of steganalysis to prove the existence of some stego-data in I'. The embedding algorithm transforms the set of pixel values contained in a Voronoi region, $\mathcal{R}_j$, to be uniformly distributed throughout its coding structure, S_j. As this distribution is non-typical, it may be possible to identify the use of this scheme.

A broad outline of such an attack is now given. As there is a uniform distribution surrounding the palette elements, training k-means using the pixel values of I' will yield a palette that approximates to that of C. The value of N may be obtained by the use of 'splitting' in k-means ([10]). The set of coding structures, $\mathcal{S}$, is then derived using algorithm 1. Now if the pixels in each coding structure, S_j, are distributed approximately uniformly throughout then one can conclude that the SMK algorithm was used to form I'.

Our modifications to this algorithm reduce the distortion of the image to control imperceptibility. The main supplementary benefit to reducing detectability is that we propose the use of only a subset of pixels for embedding. This results in fewer pixels being contained in S_j and hence reducing the 'non-typicality' of the distribution. However a steganalytic technique may still be effective as uniformly distributed pixel values will still surround palette elements.

2.5.4 Robustness: There is no provision in this algorithm for any robustness as even the modification of a single pixel could cause synchronisation loss. That is, any modification to I' will result in information loss. In addition, the embedding algorithm hides information pixel by pixel. This is equivalent to adding high-frequency noise, which is targeted for removal by lossy compression algorithms such as JPEG. Although it is possible to introduce measures that trade capacity for robustness, they are not addressed in this paper.

3 Improving imperceptibility by pixel selection

The aim of this paper is to improve both the imperceptibility and capacity of the algorithm. In this Section it is shown that the imperceptibility can be increased by embedding in a subset of pixels of the cover-data. However, this higher imperceptibility is obtained at the cost of capacity.

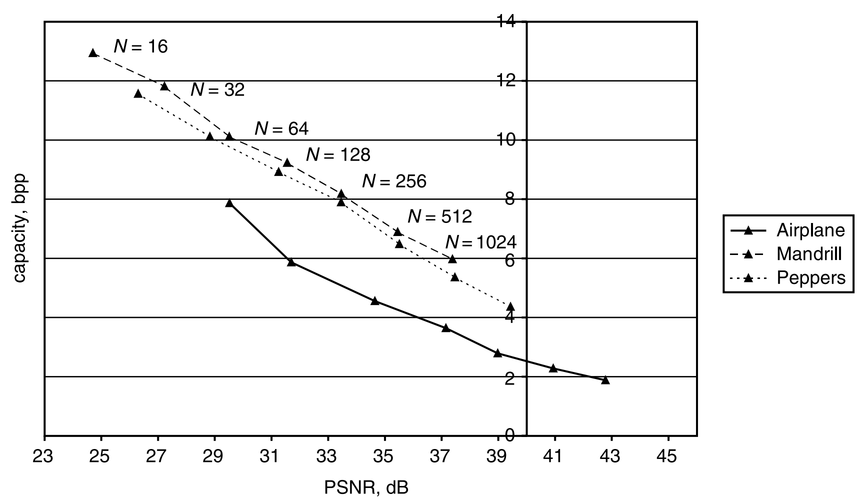

Fig. 1 *Capacity and imperceptibility of SMK algorithm*

Figure 1 shows that the SMK algorithm results in an excessive amount of distortion. Let $\mathcal{I}$ denote the two-dimensional array of pixel positions in I. An obvious remedy is to use a set $P \subseteq \mathcal{I}$ for embedding and trade capacity for imperceptibility.

Alice will only use pixels in P for embedding. For Bob to extract $\mathcal{M}$ correctly it is required that he is able to determine P. We consider two algorithms for selecting P:

1 randomly selecting a proportion ρ of elements from $\mathcal{I}$
2 an adaptive method, which selects elements from $\mathcal{I}$ where the corresponding pixels provide the best tradeoff between capacity and distortion.

3.1 Random pixel selection

In this approach we assume that all pixels are equivalent in the amount of distortion that the embedding process introduces. A trivial algorithm for reducing imperceptibility is only to embed using a proportion ρ of pixels in $\mathcal{I}$. Let $N(X)$ denote the number of pixels in a set X. So $N(P) = \rho N(\mathcal{I})$.

Let c be the MSE caused by the SMK embedding algorithm. Therefore in the SMK algorithm,

$$c = \frac{1}{N(\mathcal{I})} \sum_{\forall p \in I} D(p, p')^2$$

So by using the set P, the new MSE will be

$$c' = \frac{1}{N(\mathcal{I})} \sum_{\forall p \in P} D(p, p')^2 = \rho\, c$$

Therefore there will be a linear decrease of the MSE. Using a similar argument, the embedding capacity will also be reduced linearly on average.

The selection of P can be performed by using K as the seed for a pseudorandom number generator, shared by Alice and Bob. Of the pixels in $\mathcal{I}, \rho$ are pseudorandomly selected for inclusion in P. Alice shares ρ with Bob, allowing him to repeat the selection of P from $\mathcal{I}'$.

3.1.1 Results:

The capacity for the SMK algorithm using random pixel selection is shown in Fig. 2 for each image. There are no figures for 'Airplane' for $N > 256$ because the PSNR exceeds 40 dB even when $P = \mathcal{I}$.

This algorithm does not take into account the variation among pixels by evaluating their 'suitability' for embedding.

3.2 Calculating benefit of a pixel

An alternative pixel selection algorithm is to construct $P \subseteq \mathcal{I}$ by considering the pixel values to find an 'optimal' subset, that is, the subset of pixels with the least distortion for a given capacity. The benefit of a pixel is defined as the ratio of embedding capacity to the estimated distortion caused by embedding the pixel. For a pixel, p, the embedding capacity is $h_{C(p)}$. The average distortion when p is used for embedding is calculated by:

$$\Delta(p) = \frac{1}{v_{C(p)}} \sum_{\forall p \in S_{C(p)}} D(p, q)$$

where $v_{C(p)}$ is the number of points in the coding structure $S_{C(p)}$. Because $\mathcal{M}'$ is uniformly distributed, the estimated distortion resulting from quantising p to p' is the average distortion from p to each location in $S_{C(p)}$.

Now we can define the benefit of p as:

$$B(p) = \frac{h_{C(p)}}{\Delta(p)} \qquad (2)$$

Pixels with a higher benefit are more suitable for embedding.

3.3 Pixel-adaptive algorithm for deriving P

As Bob does not know I, he identifies P as the set of all pixel positions in I' where a pixel $p \in S_{C(p)}$. As a result, Alice's first step is to include all pixel positions in I, where the pixel $p \in S_{C(p)}$. This is referred to as the minimum set of P and is shown in algorithm 4. Although this is slightly inefficient most pixels in the minimum set would constitute the pixels with the least values of $\Delta(p)$. After this stage, she then increases P by including pixel positions with the greatest benefit.

3.3.1 Algorithm for determining P:

Algorithm 4 describes how P is selected for the embedding process. The algorithm may not succeed in meeting the required imperceptibility for a given number of segments if the distortion caused by the minimum set exceeds it. This can be overcome by decreasing N or using a method such as random pixel selection, given in Section 3.1.

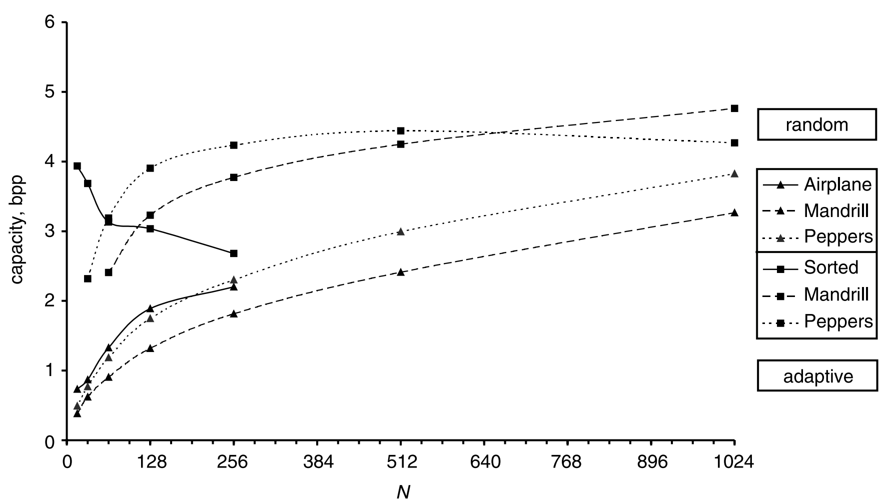

Fig. 2 *Capacity achieved by using pixel selection to limit distortion to 40 dB*

Algorithm 4: Determining the set of pixels P for embedding
Input: I, C
Output: P

```
CurrentDistortion = 0
L = ∅
For each p ∈ I
   If p ∈ S_{C(p)} Then
      Insert the location of p in 𝓘 into P
      CurrentDistortion = CurrentDistortion + Δ(p, S_{C(p)})
   Else
      L = L||{p, Δ(p), B(p)}
   End If
Next p
(The minimal set is the current set P)
If CurrentDistortion > c Then
   P could not be selected to meet the required level of distortion
Else
   L' = Sort(L) by B(p) in decreasing order
   While CurrentDistortion < c
    Insert the location of L'[0] in 𝓘 into P
    CurrentDistortion = CurrentDistortion + Δ(L'[0])
    Remove L'[0] from L'
   End While
End If
```

3.3.2 Results:

The results of using this algorithm are shown in Fig. 2. The capacities for $N < 64$ with 'Mandrill' and $N < 32$ with 'Peppers' are absent as the distortion from the minimum set exceeds the imperceptibility threshold. The pixel adaptive algorithm for constructing P provides a consistent improvement over the random selection algorithm of between 1 and 1.5 bpp for most values of N. As N increases the proportion of pixels used also increases to compensate for the reduced average distortion. Ultimately the two pixel selection methods converge when all pixels in $\mathcal{I}$ are used in P.

The pixel selection algorithms provide a framework for comparison of the embedding capacity between the three test images for an equivalent imperceptibility. 'Airplane' performs better for low values of N while the other two images perform better as N increases. It is not possible to make a general conclusion about the best sort of image to use for data hiding from these figures because the final result depends on:

1 the degree of colour variation that occurs within the image
2 the degree to which clustering can be represented by a palette of size N.

For a constant imperceptibility the second factor is of greater importance than the first because this limits the level of distortion that occurs.

By using these techniques the imperceptibility can be lowered to a level determined acceptable by Alice. The capacity of the algorithm at this level is in the range 4.4–4.7 bpp for the test images.

4 Limiting maximum capacity

Different coding structures possess different properties in trading capacity for imperceptibility. We estimate this tradeoff by calculating the benefit for a palette element C_j as $B(C_j)$, using (2). The figures for cubic coding structures of varying capacities h_j are given in Table 1.

Table 1: Measuring the average benefit for a palette element

h_j	$B(C_j)$ for cubic coding structures	$B(C_j)$ for spherical coding structures
3	2.68	3.23
6	2.97	3.21
9	2.31	2.42
12	1.56	1.61
15	0.98	1.01
18	0.59	0.60

For $h_j > 6$, the benefit decreases as h_j increases, that is, the tradeoff between capacity and imperceptibility is altered so that the capacity is reduced against a fixed imperceptibility. Therefore a capacity increase can be achieved by altering algorithm 1 to restrict the maximum capacity of the coding structures to some value $\max(h)$. This modification requires Alice to share $\max(h)$ with Bob. The following Section implements this modification for spherical coding structures to demonstrate its effectiveness.

5 Spherical coding structures

In this Section we consider a 'sphere' as the coding structure. A sphere of radius r with centre C_j is the set of points (in feature space) that have a maximum Euclidean distance of r from C_j, that is, $S_j \subseteq \{(p_r, p_g, p_b) : (C_{j_r} - p_r)^2 + (C_{j_g} - p_g)^2 + (C_{j_b} - p_b)^2 \le r^2\}$, where $v_j = 2^{h_j}, h_j \in \mathbb{Z}$. As distortion is measured using Euclidean distance, this set of points has the property that the average embedding distortion from the palette element is minimised.

The comparison of this measure against the cubic coding structures is shown in Table 1. Based on this measure it outperforms the cubic coding structures for all values of h_j. The algorithm for the construction of the sphere is given as algorithm 5:

Algorithm 5: Generating a spherical coding structure, S_j
Input: j, C_j, $\max(h)$
Output: S_j

```
S_j = ∅
p = C_j
While p ∈ 𝓡_j and v_j ≤ 2^{max(h)}
   S_j = S_j||p
   Let p be the next closest point by distance to C_j
   where p ∉ S_j
End While
Remove the most recently appended points from S_j until v_j
is of the form 2^{h_j}, h_j ∈ ℤ.
Each point in S_j is assigned a unique binary label of length h_j
```

5.1 Results

The results for the spherical encoding algorithm for the 'Peppers' image are given in Fig. 3. The 'Peppers' image is chosen as the image with moderate detail of the test images. The capacity of the spherical coding structures exceeds that of the cubic coding structures by approximately 1.5 bpp. In addition, the limitation of the capacity of the coding structures is also shown to be effective in increasing capacity. The other images have similar results and the highest capacities under these conditions are given in Table 2.

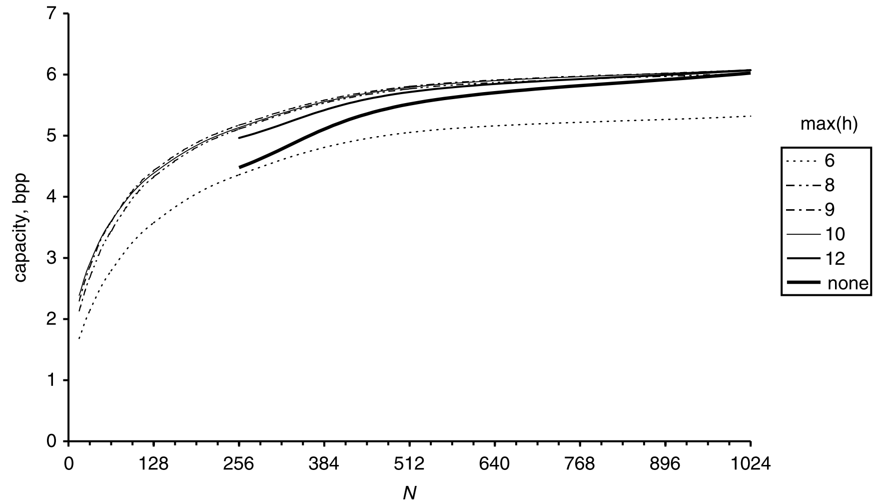

Fig. 3 *Embedding capacity with spheres using 'Peppers' and a PSNR of 40 dB*

Table 2: Best results for each coding structure with a PSNR of 40 dB

Image	Structure	N	max(h)	Capacity (bpp)
Airplane	cube	16	9	4.68
	sphere	16	10	5.75
Mandrill	cube	1024	9	4.76
	sphere	1024	9	5.85
Peppers	cube	512	9	4.48
	sphere	1024	10	6.08

Table 2 provides a summary of the results for both coding structures. Each entry used the adaptive pixel selection algorithm to provide a PSNR of 40 dB and selected the value of max(h), which gave the maximum capacity. The capacity is in the lower end of the range of the claims of the SMK algorithm, but it has been improved to an acceptable level of imperceptibility.

6 Conclusions

The SMK algorithm provides a high-capacity steganography technique without robustness or imperceptibility. We have proposed a new algorithm based on the SMK algorithm, which increases the imperceptibility and embedding capacity.

First, the embedding algorithm is modified only to embed in the pixels in the cover data, which provide the best tradeoff of capacity and imperceptibility. As a result, the algorithm can be used in situations where the imperceptibility is required to be at a visually lossless level.

The second improvement has been in recognising that the generation of coding structures to their maximum volume can unnecessarily trade away imperceptibility for capacity. By limiting the maximum capacity of a coding structure it has been shown that the capacity can be increased relative to a fixed level of imperceptibility.

The final modification was the introduction of spherical coding structures, which increased the capacity by between 1.5 and 2.4 bpp for our test images. Further tests of ours on a 'greedy' version of the spherical coding structures have shown that capacity can be further increased by between 0.2 and 0.8 bpp.

Although it is possible to increase capacity without reducing the incurred distortion, there are two other areas in which this technique would benefit from further development. First, robustness can be increased by altering the embedding and extraction algorithms to tolerate variations in pixel values caused by attacks. Secondly, as mentioned briefly, steganalysis could reveal the use of this scheme in an image. By adjusting the embedding and extraction algorithms it might be possible to reduce the incidence of uniformly distributed 'patches' in the RGB space. One tactic could be to alter the probability of the symbols in the coding structures to present a less obvious transformation in I'.

From tests performed on the three test images, we claim that it is possible to embed at up to 6 bpp with a PSNR of 40 dB having shared up to approximately 3 kbyte in data. This compares well with the maximum amount of 10.2 bpp using the model where I is shared ([9]). This algorithm can therefore be considered as a high-capacity steganographic technique where a limited amount of information is able to be shared.

7 References

1 Cachin, C.: 'An information-theoretic model for steganography', *Inf. Hiding*, 1998, pp. 306–318

2 Simmons, G.J.: 'The prisoner's problem and the subliminal channel'. Proc. of CRYPTO '83, Advances in Cryptology (Plenum Press, 1984, pp. 51–67

3 Smith, J.R., and Comiskey, B.O.: 'Modulation and information hiding in images'. Workshop on Information Hiding, Isaac Newton Institute, University of Cambridge, UK, May 1996 (*Lect. Notes Comput. Sci.*, **1174**)

4 Lee, Y.K., and Chen, L.H.: 'High capacity image steganographic model', *IEE Proc., Vis. Image Signal Process.*, 2000, **147**, pp. 288–294

5 Seppannen, T., Makela, K., and Keskinarkaus, A.: 'Hiding information in color images using small color palettes'. Proc. Information Security, Third Int. Workshop, ISW, December 2000, pp. 69–81

6 Eggers, J.J., Bauml, R., and Girod, B.: 'A communications approach to image steganography'. Security and Watermarking of Multimedia Contents IV, SPIE, 2002, pp. 26–37

7 Marvel, L., Boncelet, C., and Retter, J.: 'Spread spectrum image steganography', *IEEE Trans. Image Process.*, 1999, **8**, pp. 1075–1083

8 Kawaguchi, E., and Eason, R.O.: 'Principle and applications of BPCS-steganography'. Int. Symp. on Voice, Video and Data Communications, SPIE, 1998

9 Brisbane, G., Safavi-Naini, R., and Ogunbona, P.: 'Evaluating the optimal probability distribution for steganography under zero error conditions'. Proc. Conf. on Mathematics of Data/Image Coding, Compression, and Encryption V, with Applications, SPIE, 2002, pp. 145–155

10 Gray, R.M.: 'Vector quantization', *IEEE ASSP Mag.*, 1984, **1**, pp. 4–29

11 SIPI Image Database - Miscellaneous: http://sipi.usc.edu/services/database/database.cgi?volume=misc

12 Girod, B.: 'What's wrong with mean-squared error?', *Vis. Factors Electron. Image Commun.*, 1993, pp. 207–220