An Optimal Least Significant Bit Based Image Steganography Algorithm

Shuting Xu
Georgia Gwinnett College
1000 University Center Lane
Lawrenceville, GA, 30043, USA
sxu@ggc.edu

Shuhua Lai
Georgia Gwinnett College
1000 University Center Lane
Lawrenceville, GA, 30043, USA
slai@ggc.edu

ABSTRACT
In this paper, we propose an effective Least Significant Bit (LSB) based steganography algorithm. The new algorithm is based on the classic K-means algorithm. We split bits of a secret message into clusters so that clusters of bits can be assigned to replace the LSB of each pixel of a cover image. To successfully use K-means we define a function to calculate the distance between the bits and the clusters. Bits can be moved among neighboring clusters based on the distance to the centroids of clusters. Since the classic K-means algorithm converges to an optimum, our approach leads to an optimized stego-image, compared to results of other LSB based approaches. Real test cases show that this approach can hide 60% of the size of the cover image without any noticeable visual artifacts.

Categories and Subject Descriptors
I.3.3 [**Computer Graphics**]: Picture/Image Generation

General Terms
Algorithms, Performance, Security.

Keywords
Image, steganography, capacity, K-means.

1. INTRODUCTION
Huge amount of information is transferred via Internet every day. A large part of it requires a secured and private communicational environment. To protect the security and privacy of information, encryption and steganography methods are often applied to the sensitive information. Encryption is the process of encoding messages or information in such a way that only authorized parties can read it [1]. The encrypted message usually looks abnormal which may arise suspicion, and may be incriminating in the countries where encryption is banned [2]. Steganography, on the other hand, maintain the existence of the communication secret through the concealment of the information within other information [3]. The stego-file (file with hidden message) is no difference from the original file in the point of view of a regular humane observer. Steganography has a variety of applications such as authentication, watermarking, bank transaction, health information systems, etc. It is also used by criminal organizations to exchange secret messages [4]. There are various types of carriers that had been used to hide information, such as text, image, audio, and video files. Digital images are the most popular type of carrier used in steganography.

ICIMCS'14, July 10–12, 2014, Xiamen, Fujian, China.

The most important features of an image steganography algorithm are undetectability, hiding capacity, and robustness [2]. Undetectability is the main requirement of steganography, which means that it is impossible to determine whether there is a hidden message in the image or not [4]. The maximal length of the secret message that can be hidden in an image without causing perceptually or statistically detectable artifacts is known as hiding capacity (or steganographic payload). Robustness means that the hidden information can be reliably extracted after the stego‐image (image with hidden message) has been modified by common image processing operations, such as scaling, rotation, adding random noise, and lossy compression. An ideal steganographic method should satisfy all three features. However, there is no such method proposed yet. People use trade-offs in designing steganographic methods that are suitable for specific applications. For example, some applications may require the largest amount of information to be hidden, whereas other applications may require the stego-image to be robust. In this paper, we propose an image steganography algorithm that can significantly improve undetectability and hiding capacity.

The rest of the paper is organized as follows. We briefly discuss the related work in literature in Section 2. Our algorithm is described in detail in Section 3. The experimental results are shown in Section 4. Finally, we conclude the paper in Section 5.

2. LITERATURE REVIEW
Steganography methods can be categorized into spatial domain methods and frequency domain methods [5]. In spatial domain methods such as LSB, the processing is applied on the image pixel values directly. These methods are widely used because of their simplicity. However, they are vulnerable to small modifications. In frequency domain methods such as DCT (Discrete Cosine Transforms), DWT (Discrete Wavelet Transforms), and DFT (Discrete Fourier Transforms), the cover image is firstly transformed into a different domain. Then the transformed coefficients are processed to hide the secret information. The changed coefficients are transformed back into spatial domain to get stego-image. The frequency domain methods are considered more robust, but they are computationally more complex and provide less embedding

capacity. The steganography method proposed in this paper falls in spatial domain category.

As hiding capacity is one of the important features of image steganography methods, there are many papers in literature addressing this issue. In 2003, a method named Bit Plane Complex Steganography was proposed in [6] which claimed the amount of hidden data reached 20% of the size of the image without noticeable distortion. Another approach using triway pixel-value differencing (T-PVD) [7] achieved a hiding capacity of 50%. [8] applied image compression technique and artificial neural work algorithm to steganography and increased the hiding capacity to 88.89% of the image size. The hiding capacity is not only affected by the design of the steganographic methods, but also limited by the given hidden message or image. In this paper, we propose a K-means based method that can optimize the hiding performance for any cover image. In other words, for the same amount of hiding data, our approach can achieve the best steganography result when compared to other Least Significant Bit (LSB) based steganography methods.

K-means as a machine learning method [9] has been applied in steganography. Some of the palette based methods using K-means to train palette by quantization and classification of similar colors into clusters [10]. A palette is a small set of colors used to represent a true-color image so that the amount of data can be reduced. In [11], the cover image is parted into R, G, and B and each part is clustered to k clusters by K-means algorithm. Then the encrypted secret message is embedded in the clusters randomly by a Least Significant Bit (LSB) based method. In [12], the entire secret image is divided into many non-overlapping blocks. For each block of the secret image, a block-matching procedure is used to search for the best similar block from a series of numbered candidate blocks. For the blocks that are not well matched in the block-matching step, a K-means clustering method is applied to determine some representative blocks. To the best of our knowledge, none of the Least Significant Bit (LSB) based steganography methods used K-means to dynamically divide the secret message into clusters.

3. OPTIMAL K-MEANS BASED STEGANOGRAPHY ALGORITHM

In this section, we describe the proposed optimal K-means based steganography algorithm. We briefly introduce K-means algorithm first.

3.1 K-means algorithm

The K-means [9] clustering method is one of the unsupervised learning algorithms. The goal is to divide the data points in a data set into *k* clusters such that the distance of data points to the centroids of the clusters is minimized. The algorithm consists of two stages: an initial stage and an iterative stage. In the initial stage the *k* initial centroids of clusters are set. One policy for selecting the initial centroids is to place them as far as possible from each other. In the second stage, each data point is assigned to the nearest centroid, and the *k* new centroids are calculated according to the new assignments. This process repeats until a certain criterion is met, for example, when there is no further change in the assignment of the data points.

As K-means is a heuristic algorithm, there is no guarantee that it will converge to the global optimum, and the result may depend on the initial clusters. However, as tested in many cases, the algorithm converges to near global optimum very fast.

3.2 Basic Idea

For simplicity, we illustrate our idea using gray scale images. It can be easily extended to color images, just by regarding a color image as three gray scale images, with each corresponding to its Red, Green and Blue components respectively.

For an **M**×**N** gray scale cover image, and for a secret message with **L** bits, our goal is to hide the secret message into the gray scale image using Least Significant Bit (LSB) approach so that the resulting stego-image looks very close to the original cover image. To achieve the best result using LSB approach, we must assign the bits of the secret message non-uniformly to each pixel of the cover image. In other words, some pixels receive more bits in the process. So the key issue is to determine how many bits each pixel of the cover image needs to take so that

1. All the **L** bits of the secret message are assigned to the pixels of the cover image, and
2. The assignment of the bits to the pixels is optimal, i.e., the resulting stego-image is optimally similar to the original cover image.

Both of the above can be achieved by using a variant of K-means algorithm with the redefinition of distance function.

3.3 Optimal K-means Based Steganography Algorithm

For a secret message with **L** bits, we first uniformly distribute them to **M**×**N** pixels. So each pixel's $\frac{L}{M\times N}$ least significant bit is replaced by the secret message. Now we have **M**×**N** clusters, which can be used as the initial cluster assignment of our K-means algorithm. The next step is to define the distant between each bit to each cluster so that we can reassign bits of the secret message to different pixels.

For the *i*th cluster, $0 \le i < M \times N$, assume the cluster assigned to it has bits $B_i = \{B_i^1, B_i^2, B_i^3, ...\}$. We define the distance D_{ij}^k between bit B_i^k and cluster B_j as follows.

$$D_{ij}^k = \begin{cases} f(i,j,k), & k = 1 \; and \; j = i-1 \\ g(i), & k = 1 \; and \; j = i \\ f(i,j,k), & k = |B_i| \; and \; j = i+1 \\ g(i), & k = |B_i| \; and \; j = i \\ 0, & i = j \\ \infty, & |j-i| > 1 \end{cases}$$

where $|B_i|$ is the number of bits of B_i. Function $g(i)$ is defined as the Gaussian curvature of the current stego-image at the pixel *i*, $0 \le i < M \times N$. And $f(i,j,k)$ is defined as the Gaussian curvature of the current stego-image at the pixel i after the bit B_i^k is moved from *i*th cluster to *j*th cluster (the two clusters of bits are actually neighbors). Also we define $g(i)$ as the centroid of the *i*th cluster because $g(i)$ provides the overall contribution of each bit of cluster B_i to the Gaussian curvature at pixel *i*.

Now with an initial assignment of clusters and a new definition of distance and centroid, we can use K-means to refine the clusters as follows.

Initialization: Assign the L bits to $M \times N$ pixels uniformly to get an initial set of $M \times N$ clusters

Reassignment: Assign each bit to the cluster whose distance to that cluster is the shortest. Note here the bits only move among neighboring clusters according to the definition of distance D_{ij}^k.

Update: Calculate the centroid for each cluster after reassignment. The updated centroid for cluster B_i is the new curvature at the pixel *i* after the bit reassignment. After the curvature is updated, call **Reassignment** step again until there is no more bit moved to a different cluster.

Since in most cases, K-means converges to an optimum, we believe the resulting stego-image after the above process is the optimal Least Significant Bit assignment for a given secret message and a cover image.

4. IMAGE STEGANOGRAPHY METRICS

In this paper, we use three metrics PSNR, MSE, and CORR to measure the quality of the stego-images.

PSNR is Peak Signal to Noise Ratio [13]. It is utilized as a performance measurement for the distortion of the image. It measures the image quality through a comparison between the cover image *C* and the stego-image *S*. It is defined as:

$$PSNR(C,S) = 10\log_{10}\frac{(2^d - 1)^2}{MSE}, \quad (1)$$

where *d* denotes the bit depth of the cover image, and is equal to 8 for gray-scale images.

MSE represents the cumulative mean square error between the cover image and the stego-image. It is defined as:

$$MSE(C,S) = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(S_{ij} - C_{ij}), \quad (2)$$

where S_{ij} and C_{ij} denote the pixel values of the cover image and the stego-image, respectively. *M* and *N* represent the dimensions of the cover image.

The larger the PSNR, the better is the stego-image quality. The values of the PSNR that are lower than 30 dB imply a low quality, i.e., the embedding distortion can be obvious, while 40 dB and above imply a high-quality stego-image [5].

CORR represent correlation coefficient, which measures the differences between the cover image and the stego-image. It is defined as:

$$CORR(C,S) = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}(C_{ij} - \bar{C})(S_{ij} - \bar{S})}{\sqrt{(\sum_{i=1}^{M}\sum_{j=1}^{N}(C_{ij} - \bar{C})^2)(\sum_{i=1}^{M}\sum_{j=1}^{N}(S_{ij} - \bar{S})^2)}}, \quad (3)$$

where $\bar{C}$ is the average of all pixel values in cover image, and $\bar{S}$ is the average of all pixel values in stego-image. CORR value is between [-1, 1]. If CORR is 1 it means the cover image and the stego-image are so similar that there is no big difference between them. If CORR is -1 it means the two images are far from each other and there is huge difference between them.

5. EXPERIMENTAL RESULTS

We conducted experiments to test the performance of the proposed algorithm. Here we illustrate the results of four groups of experiments in Figure 1-4. In each group, we show the cover image, the hidden secret image, the stego-image, and the bits assignment image. The size of the bits assignment image is exactly 60% of the stego-image. It replaces the different number of lease significant bits in each pixel of the cover image with the bits of the secret image. The bits assignment image is generated when our steganography algorithm is applied, and will be used to restore the secret image from the stego-image.

Figure 1 (a). Cover image.

Figure 1 (b). Secret image.

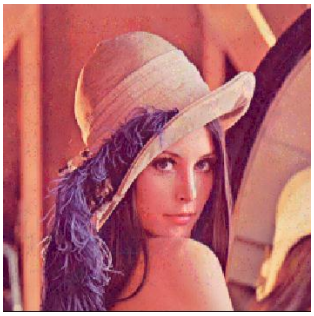

Figure 1 (c). Stego-image

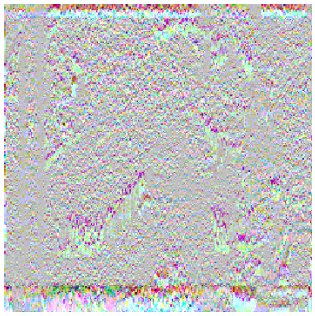

Figure 1 (d). Bits assignment.

Figure 1. Lena

Figure 2 (a). Cover image.

Figure 2 (b). Secret image.

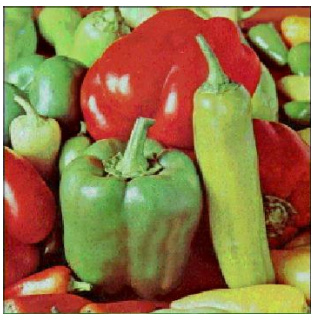

Figure 2 (c). Stego-image

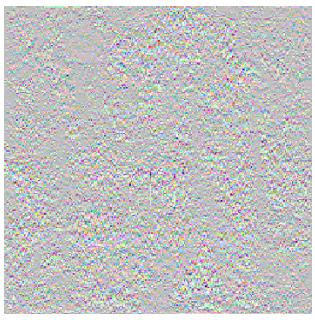

Figure 2 (d). Bits assignment.

Figure 2. Peppers

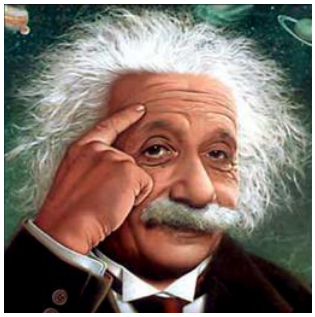

Figure 3 (a). Cover image.

Figure 3 (b). Secret image.

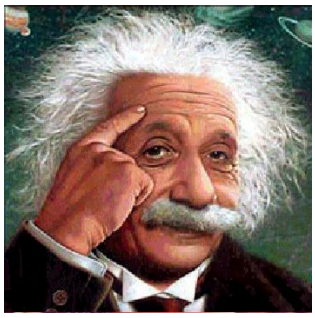

Figure 3 (c). Stego-image

Figure 3 (d). Bits assignment.

Figure 3. Einstein

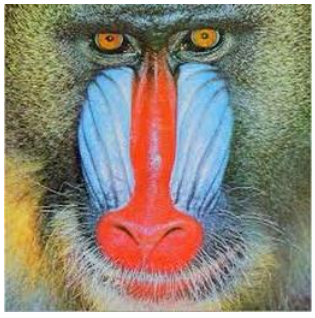

Figure 4 (a). Cover image.

Figure 4 (b). Secret image.

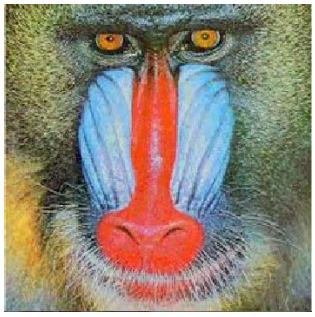

Figure 4 (c). Stego-image

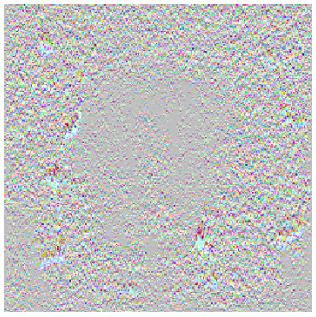

Figure 4 (d). Bits assignment.

Figure 4. Baboon

We measure the performance of the proposed algorithm using the metrics mentioned in Section 4. The values of MSE, PSNR, and CORR for each group of images are listed in Table 1. All the stego-images have the PSRN value greater than 50, which means the hidden image is undetectable. All the CORR values are close to 1, which means there is very tiny difference between the cover images and the corresponding stego-images.

Table 1. Performance metrics for each group of images.

Cover image	Stego-image	MSE	PSNR (dB)	CORR
Figure 1 (a)	Figure 1 (b)	354.20	52.13	0.97
Figure 2 (a)	Figure 2 (b)	221.12	56.84	0.99
Figure 3 (a)	Figure 3 (b)	366.84	51.78	0.99
Figure 4 (a)	Figure 4 (b)	273.63	54.71	0.98

6. CONCLUSION

In this paper we propose a LSB based steganography algorithm. The algorithm borrows the iterative procedure of K-means algorithm and could optimize the location of the secret message hidden in the cover image while keeping the sequence of the content of the secret message. The experimental results show that this algorithm can generate high quality stego-images with PSNR values higher than 50 and CORR close to 1.

7. REFERENCES

[1] Wikipedia, DOI= http://en.wikipedia.org/wiki/Encryption.

[2] Atawneh, S., Almomani, A., and Sumari, P. 2013. Steganography in Degital Images: Common Approaches and Tools. *IETE Technical Review*, Vol. 30, Issue 4, 344-358.

[3] Johnson, N.F., Jajodia, S. 1998. Exploring steganography: Seeing the unseen. *Computer*, IEEE, Vol. 31, 26-34.

[4] Ingemar, J.C. et. al. 2008. *Digital watermarking and steganography*, Morgan Kaufmann.

[5] Cheddad, A. et. al. 2010. Digital image steganography: Survey and analysis of current methods. *Signal Processing*, Vol. 90, 727-752.

[6] Srinivasan, Y. 2003. High capacity data hiding system using BPCS steganography. Texas Tech University.

[7] Chang, H. et. al. 2008. A Novel Image Steganographic Method Using Tri-way Pixel-Value Differencing. *Journal of Multimedia*, Vol. 3, No. 2, 37-44.

[8] Al-Jbara H. 2012. Increased Capacity of Image Based Steganography Using Artificial Neural Network. *AIP Conf. Proc.* 1482, 20-25.

[9] MacQueen, J. B. 1967. Some Methods for classification and Analysis of Multivariate Observations. *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, 281–297.

[10] Chen, Y. et. al. 2009. True Color Image Steganography Using Palette and Minimum Spanning Tree. *Proc. of the 3rd WSEAS International Conference on Computing Engineering and Applications.*

[11] Khashandarag, A. et. al. 2011. A Hybrid Method for Color Image Steganography in Spatial and Frequency Domain. *IJCSI International Journal of Computer Science Issues*, Vol. 8, Issue 3, No. 2, 113-120.

[12] Wang, R. and Tsai, Y. 2007. An image-hiding method with high hiding capacity based on best-block matching and k-means clustering. *Pattern Recognition*, 40, 398 – 409.

[13] Cheddad, A. et. al. 2008. Enhancing Steganography in digital images, *Proc. Canadian Conference on Computer and Robot Vision*, 326-33.