Hiding secret data into a carrier image

Ovidiu COSMA[*]

[*] North University / Department of Mathematics and Computer Science, Baia Mare, Roumania

Abstract—**The object of steganography is embedding hidden information in an appropriate multimedia carrier, e.g., image, audio, or video. There are several known methods of solving this problem, which operate either in the space domain or in the frequency domain, and are distinguished by the following characteristics: payload, robustness and strength. The payload is the amount of secret data that can be embedded in the carrier without inducing suspicious artefacts, robustness indicates the degree in which the secret data is affected by the normal processing of the carrier e.g., compression, and the strength indicate how easy the presence of hidden data can be detected by steganalysis techniques. This paper presents a new method of hiding secret data into a digital image compressed by a technique based on the Discrete Wavelet Transform (DWT) [2] and the Set Partitioning In Hierarchical Trees (SPIHT) subband coding algorithm [6]. The proposed method admits huge payloads and has considerable strength.**

I. Introduction

The word steganography comes from Greek, and means "covered writing". Steganography is a method of security complementary to cryptography, whose purpose is to hide information into an innocent looking container (digital image, audio file, video file or printed image). After embedding the data, it may be sent to the destination using insecure communication lines or even posted in public places. The container will not reveal its true content to anyone but an advised observer. There are many applications of steganography, such as: copyright control, increasing the robustness of images, search engines and smart ID cards.

The currently known steganography techniques operate in the space domain or in the frequency domain, and most of them can be detected by methods of steganalysis.

II. Spatial domain steganography

Any digital image is made up of pixels, each being represented by an integer value, on a number of bits. The first modern methods of steganography for digital images were based on the replacement of the last significant bit (LSB), or a number of insignificant bits of the image pixel values with the secret data. If more bits of each pixel are used in the process, the distortions can be easily observed with the naked eye.

The main disadvantage of this method is the fact that it places uniformly the secret data over the whole surface of the image, without taking into account the characteristics of different regions. The regions lacking of details are unsuitable for secret data embedding, because in such regions the slightest distortions are easily observed.

Bit Plane Complexity Segmentation (BPCS) [5] is a more sophisticated method of steganography that operates in the spatial domain, but selects the regions in which the secret data is embedded by their complexity. Complex regions are better because their frequent variations in luminosity make distortions harder to spot. For selecting the best places to embed the secret information, the image data is divided into bit planes. A bit plane is a data structure composed by all the bits of the image pixel values, located at a certain position. The first bit plane is composed by the LSB of all the pixel values, and the last bit plane contains all the most significant bits. Each bit plane is divided into 8 x 8 bit segments, and then the complexity of each of the segments is determined. The complexity of a segment is defined as the number of non edge transitions from 0 to 1 and from 1 to 0 both horizontally and vertically. Generally for a square of 2n x 2n pixels, the maximum complexity is 4n (2n-1) and the minimum complexity is 0. Therefore for the 8 x 8 square segments the complexity is situated between 0 and 112. An example of bitplane segments and their complexities is presented in figure 1.

The most complex segments of the bit planes are used to store the secret data. This choice is justified by the fact that usually high complexity segments correspond to complicated regions in the image, and not to uniform areas of homogenous color or simple shapes.

The BPCS technique works well with natural images, because they usually have lots of high complexity areas that can hold the secret data. Images with complex textures and well shaded objects usually support large payloads. On the opposite side are the computer generated images, because they have large areas covered with uniform colors, and sharp contours. In this type of images, there are no places to hide the secret data, and the slightest modifications of the pixel values tend to create obvious artefacts.

BPCS is not a robust embedding technique. The secret data is affected by the usual processing of the image container. Any lossy compression and most of the transformations and filters will corrupt the secret data inside the container. The embedding rate depends on the complexity of the image container. For visually complex images, embedding rates of 30% to 50% are possible, with low degradation. The artefacts that are inevitable at high embedding rates are often overlooked because they are disguised in regions of high complexity.

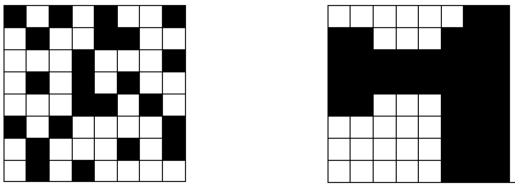

Figure 1. Sample bit plane regions of complexities 66 and 19

The ABCDE (A Block Complexity based Data Embedding) [3] algorithm presents an improvement over BPCS. The suitability of the blocks to hold secret data is evaluated by two complexity measures: run-length irregularity and border noisiness. Both BPCS and ABCDE embedding techniques show little strength. The presence of hidden data can be revealed by current steganalysis techniques.

III. Frequency domain steganography

Frequency domain steganography represents a step of evolution over the spatial domain methods. The Discrete Cosine Transform is used extensively in lossy video and image compression e.g. the Joint Photographic Experts Group (JPEG) [9] and Moving Picture Experts Group (MPEG) [10] standards.

A JPEG compressor starts by dividing the image into 8 x 8 bit blocks, and then computes the DCT for each of the blocks. The resulting 64 coefficients are not of equal importance for the quality of the reconstructed image, because the sensitivity of the human eye decreases with the spatial frequency. In consequence the coefficients corresponding to the high frequency harmonics are less important then those corresponding to the low spectrum of the image.

The next step in the JPEG compression is quantization of the DCT coefficients, which has the purpose to eliminate the information in the transformed image that is less relevant for the visual quality of the reconstructed image. In order to achieve that, a coarser approximation is applied to the coefficients corresponding to the high spectrum of the image. But the high harmonics of natural images are usually of low amplitudes, and as a result of the quantization step many of them will be erased i.e., the corresponding DCT coefficients will be zeroed. There is a trade off between the number of zeroed DCT coefficients and the quality of the reconstructed image. The more zeroes in the DCT blocks, the more efficient will perform the next compression steps in the JPEG standard, that have the task to reduce the redundancy, achieving greater compression, but great compression means also great distortions.

Redundancy reduction uses perfect reversible transformations and as a consequence the DCT blocks offer a perfect hiding place to store secret data, which will not be affected by the following transforms of the JPEG compression scheme.

A steganographic method that uses the middle frequency DCT coefficients to store secret data into a JPEG image is presented in [4]. A number of 36 coefficients are used to store the secret data, from a total of 64 in a DCT block, which yields to a reasonable payload. Altering a single coefficient would affect the whole 64 pixel block, but because the change operates in the frequency domain instead of the spatial domain, there will be no visual artefacts in the reconstructed image if those coefficients are handled with care.

JSteg was among the first steganographic algorithms to use JPEG images. The algorithm does not leave visual traces in the container image, but the existence of hidden data can be revealed by examining the statistical distribution of the DCT coefficients. The detection is possible because the JSteg algorithm disturbs the usual Gaussian distribution of the coefficients in a JPEG image.

IV. Discrete Wavelet Transform Steganography

The most important disadvantage of the image compression schemes that are based on the DCT is the fact that the transform does not reveal any information about the space localization of the frequency components. Because of that, images must be partitioned in blocks that are transformed separately. At high compression ratios, the approximations performed in the quantization step can create important differences between the neighboring pixels, at the edge of the blocks. Those differences correspond to horizontal and vertical steps in the luminance and chrominance components. Such distortions are highly visible especially in the long smooth areas with continuous tones, and because of them images loose their natural aspect at high embedding rates. The embedding rate at which this effect appears depends on the characteristics of the image, but when it becomes noticeable, if the embedding rate is increased any further, the visual quality worsens very quickly.

A much better transform for image compression is the Discrete Wavelet Transform, (DWT) because it localizes the frequency components in space and does not require the partitioning of images. This type of transform is used in the JPEG 2000 compression standard [11].

The Discrete Wavelet Transform (DWT) decomposes an image into bands that vary in spatial frequency and orientation. The next two steps in an image compression application are quantization and redundancy reduction. The two steps are usually integrated in a subband coding algorithm. The coding algorithm performs a progressive approximation of the subbands, processing a bit plane at every pass, starting with the most significant one.

The Embedded image coding using Zerotrees of Wavelet coefficients (EZW) [7] algorithm successfully speculates the correlations between the DWT coefficients in successive subbands, with a data structure called zerotree. A zerotree is a quad-tree with all the nodes smaller or equal with the root. A zerotree that contains only insignificant coefficients, can be coded with a single symbol, and will be completed with zeros by the decoder. The EZW algorithm progressively codes an image in several passes. Each of the passes has an associated threshold p. For the first pass, p is initialized with the largest power of two, which is smaller or equal with the largest DWT coefficient. At each of the passes, the coefficients whose absolute values reach or exceed the threshold become significant. They are moved in a List of Significant Coefficients (LSC), and then the threshold is halved. The process is ended when the bit rate reaches a target value. At each pass the next bit from all the significant coefficients is placed in the output stream of the coder.

A method of embedding secret data into a DWT transformed image is presented in [8]. The method is based on the EZW and BPCS algorithms. First the image is transformed using the DWT, and then it is encoded with the EZW algorithm, at the desired bit rate. Next the EZW decoding algorithm is applied in order to obtain the quantized image subbands. Then as in the BPCS algorithm each bit plane is divided in segments, which are classified by their complexity, and the secret data is inserted in the most complex segments. The main difference from BPCS is the fact that the operation is performed on the transformed image, i.e., in the frequency domain. Because the largest of the subbands is only ¼ the size of the original image [1], the embedding process uses only 4 x 4 bit patches. Because the smaller subbands respond differently to changes than the larger ones, a weighting scheme was proposed, for increasing the complexity level required for embedding in the more significant subbands. This reduces visual distortion, but reduces the embedding capacity of the subbands.

The algorithm offers an embedding capacity of 25% for usual images, without inducing large, obvious distortions in the reconstructed image. At higher embedding rates, the distortions tend to show as a blurred mottling of the image, in areas that are rich in details.

V. The Stego SPIHT algorithm

The Set Partitioning in Hierarchical Trees (SPIHT) algorithm is an improved variant of EZW that also uses zerotrees for an efficient representation of data, but achieves better image quality at a given bit rate.

The proposed method is called Stego SPIHT, because it attaches secret data to the DWT coefficients, during the SPIHT coding process.

The secret data is inserted at the end of the image code, only after a certain level of quality is achieved. The visual quality of the reconstructed image can be estimated with the Peak Signal-to-Noise Ratio (PSNR). No tampering of the DWT coefficients is performed before the desired level of quality is reached. Like EZW, the SPIHT algorithm codes the DWT coefficients bit planes in a progressive manner. When a certain level of quality is reached, the first bits of the significant coefficients have been already coded, down to the position of the current bit plane. If the code would end at this point, the decoder will replace all the remaining bits and all the insignificant coefficients with zeroes. But the following bits will be used in the embedding process.

The DWT coefficients are not integers, and there is no limitation on the number of iterations of the SPIHT algorithm. Therefore theoretically there is no limitation on the payload, because the container can be extended as needed. However, there is a common sense limit, because if the bit rate increases too much, the method looses its strength because the existence of hidden data in an oversized image container would be obvious.

Embedding the secret data does not affect the image quality, because only the least significant bits are altered. Those bits would be considered to be null by the decoder in the absence of hidden data, but in fact they are not all zeroes. As a consequence, the embedding process could actually increase the approximation accuracy for some of the DWT coefficients.

The insertion of the secret bits into the DWT coefficients is performed in a manner that minimizes the risks of detection by steganalysis methods. As a difference from the previous techniques, the selection of the coefficients that will hold secret data, does not involve the segmentation of the bit planes. All that matters are the actual values of the coefficients, and not some bits from their binary representation. Only the coefficients that became significant at the previous passes of the SPIHT algorithm are used for storing the secret data. The null coefficients are left unchanged, because otherwise the natural distribution of null coefficients in the image code will be affected. Those coefficients are normally processed by the SPIHT algorithm, depending on the actual image data, and will be used in the embedding process, only in the next pass after the one in which they became significant.

Each coefficient in a subband has four descendents in the next subband, and each of the descendents has other for descendents in the following subband, and so on. Therefore all of the DWT coefficients, except the general average can be placed in three quad trees. One of them corresponds to vertical oscillations, the second to horizontal and the third to diagonal oscillations. The structure of the image subbands is presented in figure 2.

There is a well known correlation between the descendents of a coefficient in several subbands, which must not be affected by the embedding process. There is a very small probability for this correlation to be disturbed, because only the last negligible bits of the coefficients are modified.

The first subbands contain important data, and very few coefficients. That is why the secret data is embedded only in the high subbands. For example the total number of coefficients in the first 5 subbands is 1024, and the next subband contains alone three times as many. The total number of subbands depends on the size of the digital image.

The block scheme of the steganographic application is presented in figure 3. The secret data is passed through the encryption and redundancy reduction blocks, in order to have a random nature. The two blocks have the main role of eliminating any correlation between the symbols, which would ease the task of the steganalysis methods.

The bit rate is an input to the Stego SPIHT coding block, because it determines the position where the embedding process begins. The embedding process increases the bit rate, and the result is a stego image, encoded at a new bit rate. Because the bit rate is a poor measure of the reconstructed image quality, the PSNR could be used instead, but the algorithm is more time consuming in this case, because in order to determine the PSNR, the inverse transforms have to be performed, for generating the reconstructed image. As a compromise, the PSNR is evaluated only when the SPIHT algorithm begins to process a new bitplane.

For increasing the strength of the method, the application doesn't use all of the significant DWT coefficients in the embedding process. Instead, the used ones are selected at a random step. Thus a tradeoff has to be done between the total length of the image container and the strength of the embedded data.

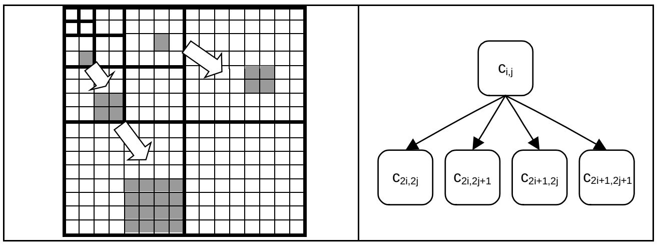

Figure 2. The structure of the image subbands

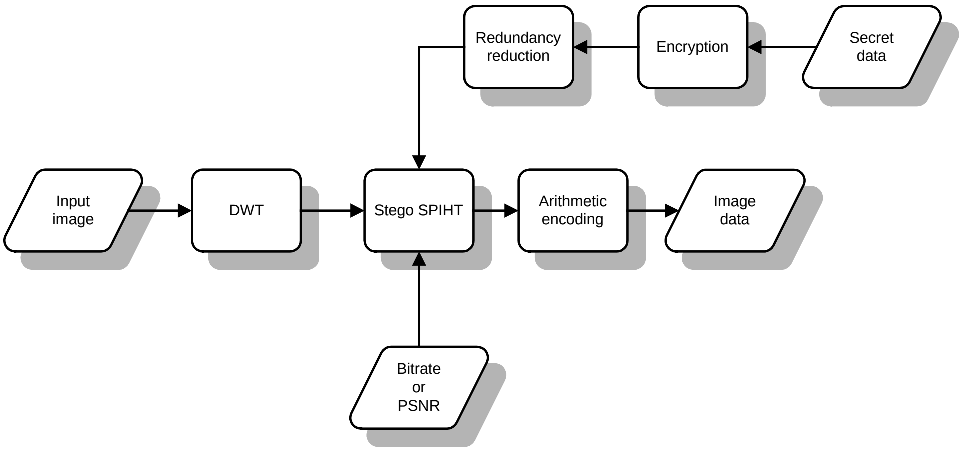

Figure 3. The block scheme of the Stego SPIHT application

In order to restore the secret data, the decoding algorithm needs the following information: the position in the stego image code where the embedding process begins, the seed for the random number generator and the length of the secret data. These could be embedded in the stego image, but in order not to weaken the method, it is better to provide them separately, as in the case of a secret key.

Stego SPIHT is not a robust embedding algorithm. The hidden data will be corrupted if the image file is further processed by irreversible techniques. But that is not a real problem, because the algorithm is proposed for steganographic and not for watermarking purposes.

References

[1] O. Cosma, Contributions to the coding of image subbands, Ph D Thesis (in Romanian), Politehnica University Bucharest, 2003.

[2] Ingrid Daubechies, Ten lectures on wavelets, SIAM, 1992

[3] H. Hioki, A data embedding method using BPCS principle with new complexity measures, Proceedings of Pacific Rim Workshop on Digital Steganography, 2002.

[4] X. Li, J. Wang, A steganographic method based upon JPEG and particle swarm optimization algorithm, Information Sciences 177 (15) (2007).

[5] M. Niimi, H. Noda, E. Kawaguchi, A Steganography Based on Region Segmentation by Using Complexity Measure. Trans. of IEICE, Vol. J81-D-II, No. 6, 1998.

[6] Amir Said, William A. Pearlman, A New Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees, IEEE Transactions on Circuits and Systems for Video Technology, vol. 6, 1996.

[7] J. M. Shapiro, Embedded Image Coding Using Zerotrees of Wavelet Coefficients, IEEE Transactions on Signal Processing, vol. 41 no. 12, 1993.

[8] Jeremiah Spaulding et al., BPCS Steganography Using EZW Encoded Images, DICTA2002: Digital Image Computing Techniques and Applications, 2002.

[9] Gregory K. Wallace, The JPEG Still Picture Compression Standard, Communications of the ACM, Volume 34, Issue 4,1991

[10] http://mpeg.chiariglione.org/standards.htm

[11] ISO/IEC, JPEG 2000 part I, final committee draft version 1.0, ISO/IEC JTC 1/SC 29/WG 1, 03.2000