DICTA2002: Digital Image Computing Techniques and Applications, 21—22 January 2002, Melbourne, Australia

BPCS Steganography Using EZW Encoded Images

Jeremiah Spaulding, Hideki Noda
Kyushu Institute of Technology, Kitakyushu, Japan 804-8550

Mahdad N. Shirazi
Communications Research Lab, Kobe, 651-2401 Japan

Michiharu Niimi, Eiji Kawaguchi
Kyushu Institute of Technology, Kitakyushu, Japan 804-8550

Abstract

Bit Plane Complexity Segmentation (BPCS) digital picture steganography is a technique to hide data inside an image file. BPCS achieves high embedding rates with low distortion based on the theory that noise-like regions in an image's bit-planes can be replaced with noise-like secret data without significant loss in image quality. BPCS is not a robust embedding scheme, and any lossy compression after the embedding of secret data usually destroys it.

Wavelet image compression using the Discrete Wavelet Transform (DWT) is the basis of many modern compression schemes. The coefficients generated by certain wavelet transforms have many image-like qualities. These qualities can be exploited to allow BPCS to be performed on the coefficients. The results can then be losslessly encoded, combining the good compression of the DWT with the high embedding rates of BPCS. Expermients on test images have yielded embedding rates of 30-40%, with low distortion.

1. Introduction

With the Internet boom of the last ten years and the even more recent boom in e-commerce, data security has become a very important field of study. There are many good security protocols in use, such as public key encryption and SSL. These overt methods cause the data, if intercepted by an unauthorized party, to be unintelligible and useless. However, as it has been proven again and again in the past, today's best encryption may be broken tomorrow by constant advancement of the art. There have been examples of encrypted messages being stored for over a decade until technology had progressed enough to break the encoding.

The topic of this paper is a form of security through obscurity. If nobody knows the encrypted data is there then they can't try to break its protection. Steganography is the practise of hiding or camouflaging secret data in an innocent looking dummy container. This container may be a digital still image, audio file, movie file, or even a printed image. Once the data has been embedded, it may be transferred across insecure lines or posted in public places. The dummy container will seem innocent under all but the most detailed of examinations.

One form of steganography, Bit Plane Complexity Segmentation[1][2] steganography, has proven to be very effective in embedding data into many classes of dummy files including 24bpp[4], 8bpp, and indexed color[3] images. BPCS has also been successfully applied to stereo and mono digital audio files. The benefits of this technique over traditional steganography are the relatively large percentage of the dummy file that can be replaced with secret data and the much lower occurrence of visual artifacts in the post-embedding image.

In this paper, we discuss the use of BPCS on natural 8bpp greyscale images compressed with the DWT and encoded with Shapiro's Embedded Zerotree Wavelet (EZW) encoder. Necessary background on BPCS is given in the Section 2, followed by information on the DWT and EZW algorithms in Section 3. Section 4 covers the techniques used to join the processes in EZW BPCS. The paper continues with a summary of experiments done with an implimentation of the algorithm. Conclustions from the experiments are followed by references and some results gathered.

2. BPCS steganography

Bit Plane Complexity Segmentation steganography is a modern method of data hiding. Earlier methods of image steganography simply replaced the least significant bits of each pixel with hidden data. This practise had very low embedding rates because visual defects rapidly develop as more significant bits are used. These defects are most noticeable in areas of homogenous color, where they usually appear as noise-like static. As more data is added, the noise becomes more pronounced, until the image fades to unintelligible static. The degradation can become obvious and severe with only 10 to 15 percent of the image replaced with secret data.

On computer and television screens, the smallest division of color data is a pixel. In computer memory, each pixel is represented by a binary value. The more bits that are used to represent each value, the wider the range of colors is for each pixel. Typical amounts of bits per pixel (bpp) are 8, 24, and 32. With these binary pixel values, and knowledge of which part of the picture each one represents, we can construct bit planes. A bit plane is a data structure made from all the bits of a certain significant position from each binary digit, with the spacial location preserved. In Figure 1, position (0, 0) from bit plane 2 is bit 2 from pixel (0, 0) in the image.

BPCS addresses the embedding limit by working to disguise the visual artifacts that are produced by the steganographic process. Optometric studies have shown that the human visual system is very good at spotting anomalies in areas of homogenous color, but less adept at seeing them in visually complex areas. When an image is deconstructed into bit planes, the complexity of each region can be measured. Areas of low complexity such as homogenous color or simple shapes appear as uniform areas with very few changes between one and zero. Complex areas such as a picture of a forest would appear as noise-like regions with many changes between one and zero. These random-seeming regions in each bit plane can then be replaced with hidden data, which is ideally also noise-like. Because it is difficult for the human eye to distinguish differences between the two noise-like areas, we are able to disguise the changes to the image. Additionally, since complex areas of an image tend to be complex through many of their bit planes, much more data can be embedded with this technique than with those that are limited to only the lowest planes.

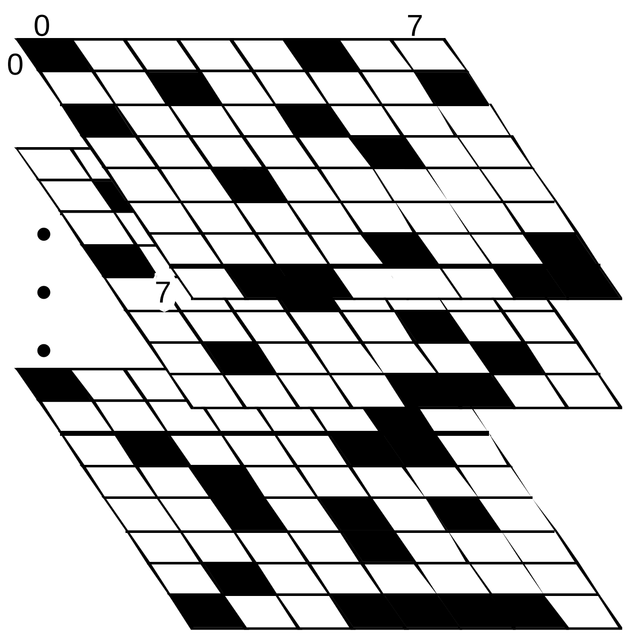

Figure 1. Image pixel location (0,0) has the binary value 01001110. In these bit planes, black is a 0 and white is a 1. In the first bit plane in the figure, position (0,0), there is a black zero. In the second bit plane, there is a white one, and so on down to the last bit plane.

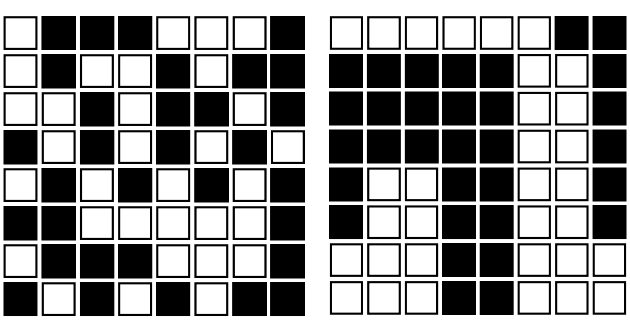

Figure 2. Noise-like patch (a) and informative patch (b): (a) complexity 69, (b) complexity 29.

In BPCS, the complexity of each subsection of a bit plane is defined as the number of non edge transitions from 1 to 0 and 0 to 1, both horizontally and vertically. Thus the complexity of each section is not determined only by the number of ones or zeros it contains. Generally, for any square of 2nx2n pixels, the maximum complexity is 2x2nx(2n-1) and the minimum is of course 0. Most versions of image BPCS use an 8 pixel square, where the maximum complexity is 112. In Figure 2, white represents a one and black a zero. Both squares, or 'patches', have the same number of ones and zeros, but very different complexities. This shows that one contains much more visual information than the other. The complex patch (A) has very little visually informative information, therefore it can be replaced with secret date and with a very low effect on the image's quality. However, if the more visually informative patch (B) was replaced, it would cause noise-like distortion of the definite edges and shapes.

This technique works very well with natural images, as they tend to have many areas of high complexity. Images with many complex textures and well shaded objects are usually have a high embedded data capacity. BPCS works much less well with computer generated images and line art, as those classes of images tend to have large areas of uniformity and sharply defined border areas. With these types of images, there is very little complexity to exploit and any changes tend to generate very obvious artifacts. This is one flaw BPCS shares with traditional steganography, though for slightly different reasons. Traditional steganography works poorly with computer generated pictures because the static distortion effect produced by embedding is very obvious in areas of homogenous color.

Another shared flaw is fragility of the secret data with respect to changes in the post-embedding image. Any lossy compression will corrupt the hidden data, as will most transformations and filters. Since this makes the hidden data very vulnerable to any destructive attack, BPCS is almost useless for watermarking purposes.

Despite these drawbacks, BPCS is very effective. With visually complex images, embedding rates of 30% to 50% are possible with low degradation. Even at high embedding rates, the artifacts generated are often overlooked because they are disguised in complex visual areas. This research proposes a way to combine BPCS with wavelet image compression and EZW encoding to create a system ideal for Internet use.

3. Wavelet compression and the EZW encoder

The original JPEG standard made use of the block Discrete Cosine Transform (DCT) for its compression. This standard has been in wide use for some time, having gained popularity in part because of the demand for a good standard compression scheme to speed the download of images from the newly popular World Wide Web. At the same time, wavelet image compression was in the early stages of research and beginning to gain acceptance in the academic community. After being refined, wavelet techniques achieved even better compression than the DCT, with fewer artifacts and distortions. There have been great advances in the field of wavelet compression within recent years and many of today's best image, audio, and video COmpressor/ DEcompressors (CODECs) are based on wavelets.

The Discrete Wavelet Transform, when used on images, generally creates a lossy representation of that image. The image can then be reconstructed from the transform coefficients by using the inverse DWT. The coefficients produced have some image-like properties, which are exploited in many encoders, and which are used by EZW BPCS as explained in the next section. Figure 3 describes how one property of the DWT coefficients is exploited to improve encoding efficiency.

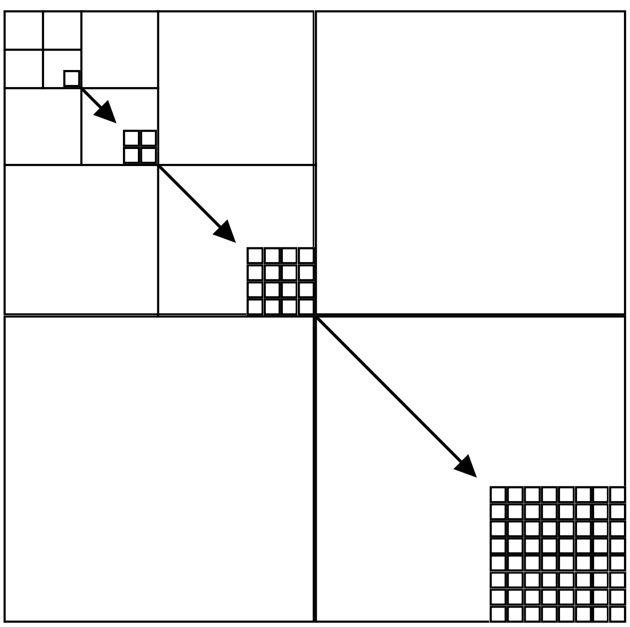

Figure 3. This figure shows the correlation between subbands in a wavelet coefficient set, and how EZW exploits it. A Zerotree the first highlighted pixel in the upper left corner would mean that all the other highlighted pixels could be represented with only one symbol.

The seminal paper on wavelet image compression is the 1993 paper Embedded Image Coding Using Zerotrees of Wavelet Coefficients[5] by J.M. Shapiro. The EZW encoder is simple, fast and provides very good compression rates. It takes advantage of the correlations between subbands in a wavelet coefficient set to lower the amount of bits needed to represent them. The successive approximation method used by EZW encodes the wavelet coefficients one bit plane at a time, starting with the most significant bits. The encoding is lossless as long as all bit planes are processed. However, significant compression can be achieved by not encoding all the bit planes. In fact, with many images, a very good representation can be achieved by using only half of the available planes. A custom fast recursive indexed EZW encoder was written for this project which trades extra memory usage for an increase in speed.

4. EZW based BPCS

This research proposes a method of embedding secret data into a DWT transformed image using the previously described BPCS. The coefficients of the DWT have many image-like properties, and BPCS is ideal for exploiting them. The main properties leveraged for BPCS are:

•Correspondence: Spacial areas in each section of the coefficients' subbands correspond directly to areas in the original image.
•Complexity: The bit planes at corresponding significance levels of the wavelet coefficients and the original image are usually proportionally complex.
•Resilience: Changes in the values of the wavelet coefficients do not create disproportionately large changes in the reconstructed image.

The property of correspondence states that in each subband of the wavelet coefficients, any sub section of that subband directly corresponds to a section of the original image. This correspondence is of course proportional, as the subbands decrease in size by a factor of two with each pass of the DWT. For example, an 8x8 'patch' of pixels in the original image corresponds to a 4x4 patch of pixels in the largest subband. This allows the same complexity metrics to be used on the wavelet coefficients as are used on the original image.

In the wavelet coefficients, the complexity of any sub section is related to the complexity of the corresponding sub section of the original image. While the amount of complexity in the wavelet coefficients is very important, the distribution of the complexity is also important. In the wavelet coefficients, the bits are ordered in decreasing significance, just as in the original image. Because of this, bit planes tend to become more complex towards the least significant bits. This is good for BPCS because this is where changes will have the smallest impact.

The capacity of a container image is limited not only by its complexity, but by the decoder's resilience to changes made in the coefficients. Resilience indicates the ability of the wavelet coefficients to absorb changes in value without changing the final image. The more resilient they are, the more changes that can be made and thus the more data that can be embedded. The inverse DWT is quite resilient to small changes in the coefficient values, and large changes experience a blending and blurring effect from the 'smoothing' nature of the wavelet transform. This property is extremely useful for BPCS, as many slight changes in the coefficients are blended out and result in very little visual impact on the reconstructed image.

However, there are many differences between the wavelet coefficients and the original image that they are generated from. The main difference that has to be accounted for is the subband structure. Normally in BPCS, an 8x8 square block of pixels called a 'patch' is used for complexity measuring and embedding. In wavelet coefficients, the largest subband sections are only 1/4 the size of the original image. Embedding into an 8x8 block of pixels in this subband would be like embedding into a 16x16 block in the original image. A smaller block size of 4x4 can be used to compensate for this. Unfortunately, a 4x4 patch has a much smaller complexity range, 24 compared to the 8x8 maximum of 112. The smaller range results in a much coarser change gradient in the amount of both distortion artifacts and embedding capacity. However, at proportional complexity values, the 4x4 patches seem to provide better overall results for both distortion and capacity than the 8x8 patches.

Another difference has to do with non-uniform significance accross the sub bands within each bit plane. The smaller subbands respond differently to changes than the larger subbands. A subband based weighting scheme was devised to increase the complexity level required for embedding in the more significant subbands. This decreased the embedding potential for each bit plane, but resulted in vastly reduced visual distortion.

An important but easily overlooked permutation is the quality of the secret data to be embedded. If the data is not random-seeming, the distortion of the output image may be greatly increased. Replacing a complex, noise-like area in an image with data that is all zeros or all ones would result in a high amount of distortion. Encryption and compression of the data before embedding solves this problem, and also allows for much greater amounts of data to be embedded. If the data is random-seeming, the post-embedding encoding will not be able to compress it much. Because of this, the final output file size usually increases by about the same amount as the size of the data that is embedded. However, the larger the file, the better the arithmetic encoder performs, so this increase varies depending on how much data is embedded. Below is the embedding algorithm (see Figure 4):

1: The image is converted into raw pixel values.
2: The DWT is applied to the image.
3: The Wavelet coefficients are encoded to the desired resolution by the EZW encoder.
4: The Wavelet coefficient bit planes are decoded and reconstructed to the encoded resolution.
5: BPCS is performed on the Wavelet coefficients.
6: The Wavelet coefficients are re-encoded to the previous resolution by the EZW encoder.
7: The EZW file is arithmetically encoded.

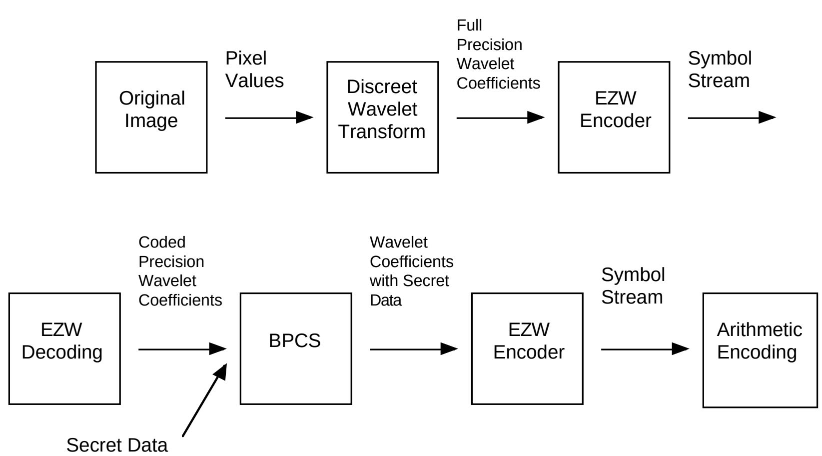

Figure 4. The algorithm in graphical form. The arrows represent input/output data streams.

EZW is a progressive encoder, so it encodes one bit plane at a time. Steps 3 and 4 are performed so that portions of the final bit plane can be omitted, so as to meet bpp requirements. It is also very easy to construct bit planes from the encoded coefficients during the decoding process. Any transform like the DWT that results in coefficients with progressively significant bits can be used, as long as a suitable complexity metric can be found. The arithmetic encoding in step 7 is useful for further compressing the final symbolic output.

Theoretically, the image can be of any color depth or aspect ratio. A trivial way to extend this process to 24bpp RGB color would be to separate the pixel values into three 8bit color spaces. The same algorithm as above could then applied from step 2. An additional step would be needed after step 7, to join the three sperate streams into a final output file. Using an RGB image would yield a much higher embedding capacity as well as a much larger post embedding file than 8bpp greyscale.

Applying this algorithm to 32bpp RGBA color would be more difficult. The changes in the Alpha values could cause very obvious transparency problems, with point distortions being very easy to see. Also, Alpha values tend not to be very complex. Most times they define large regions of the image to be either completely transparent or completely opaque. The second most common use is to define a smooth gradient of transparency. In either case, the Alpha channel would likely have a low embedding capacity, and be prone to producing large distortions. It would likely be best to embed in only the RGB color space and to ignore the Alpha channel completely.

5. Experimental results

The process described in the previous sections was implemented and tested on several standard images including “Lena”, “Goldhill”, and “Barbara”. The initial results are very good, and further refinements to the technique should be able to boost embedding rates even more. At embedding levels up to 25% of the final compressed image size, there is an absence of large, obvious distortions in the post embedding image. At higher levels, artifacts that do appear tend to show as a blurred mottling of the image in the areas of high complexity. When more significant bit planes are used, there are sometimes point irregularities where a section with a low amplitude trend has a high amplitude spike caused by embedding. This spike shows as a light spot on the darker background, or vice versus. The spot fades into the surroundings, but is sometimes noticeable. As in regular steganography, for this reason as well as others, use of the higher bit planes should be limited.

Of the four algorithms used (DWT, EZW, BPCS, and Arithmetic Encoding) BPCS is the most computationally intensive. Even so, on a middle range 333MHZ Pentium running Red Hat Linux 7, the average time for the entire process from step 1 to step 7 in section 5 takes less than 10 seconds. On a 400MHZ G3 Apple Powerbook, the process takes less than 5 seconds. As with many techniques, in BPCS steganography memory usage can be traded for speed. With minor optimizations for memory, peak memory usage was cut in half, while the time taken was less than doubled.

The results listed are from an implementation of the EZW BPCS technique. The Moffat[6] adaptive arithmetic encoder was used as the final step in Figure 4. The system was tested on 8bpp greyscale images, which were 512x512 pixels in size. The 5 most significant bit planes were not used, as experiments show that these planes tend to have both low embedding capacity and low resilliance to change. Also in these results, the 4x4 patch size was used as it gave better performance. Table 1 is of “Lena”, encoded with 8 and 9 bit planes, respectivly. Figures 5 and 6 show the post embedding file for two of the results listed in the table. Generally, the system was able to achieve embedding rates of 20% to 25% with little or no obvious degradation in image quality. Rates of over 50% were attained but some distortion was observed.

6. Conclusions

The proposed system of BPCS steganography for wavelet encoded images is still being refined. There are many optimizations that can be made to increase embedded image quality and embedding capacity. Variations can be made in the size of the patches that the bit planes are divided into, from a 4x4 pixel area to an 8x8 pixel area. The weighting scheme for determining the importance of each sub band can be adjusted as well. These permutations, as well as other ones, are still being explored.

The low compressed file size produced by the EZW encoder means that the embedding rates must be proportionally smaller than those of larger dummy files. For instance, 30% of an 8bpp uncompressed image is a much greater quantity than 30% of a .6bpp compressed image. While it is much smaller, the capacity is still more than adequate for many purposes, such as secret emails and embedded annotations.

Since this system embeds into the wavelet coefficients, the subsequent encoder is not important, as long as it is lossless. EZW was chosen for its ease of implementation and relatively good encoding efficiency. This opens the way for application of this system to many of today’s best wavelet-based image CODECs, as well as future ones. This will create a system that will allow many more people access to the benefits of BPCS steganography without sacrificing the use of a cutting edge image compression system.

Figure 5. Test image 'Lena', 3 of 8 bit planes used, with a complexity threshold of 6.

Figure 6. Test image 'Lena', 3 of 8 bit planes used, with a complexity threshold of 2.

Table 1. Experimental results for "Lena"

image	#planes for embedding	#planes used	complexity threshold	embedded data (bytes)	compressed (bytes)	PSNR (dB)
(a)	8	-			19333	35.9
(b)	8	3	6	5104	24219	31.6
(c)	8	3	4	7216	26599	30.3
(d)	8	3	2	9934	30482	29.1
(e)	9	—			37630	39.1
(f)	9	4	6	12442	47791	33.0
(g)	9	4	4	16750	52073	31.5
(h)	9	4	2	22908	60032	30.0

[1] E. Kawaguchi, M. Niimi. "Modeling Digital image into informative and Noise-Like Regions by a Complexity Measure", *7th European-Japanese Conference on Information Modeling and knowledge Bases*, 1997.
[2] 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, pp. 1132-1140*, 1998.
[3] R. Ouellette, H. Noda, M. Niimi, E. Kawaguchi. "Topological Ordered Color Table for BPCS Steganograpy Using Indexed Color Images", *IPSJ Journal*, 2000.
[4] E. Kawaguchi, R.O. Eason. "Principal and Applications of BPCS Steganography", *Proc of SPIE, Vol. 3528, pp. 464-473*, 1998.
[5] J. M. Shapiro. "Embedded Image Coding Using Zerotrees of Wavelet Coefficients", *IEEE Transactions on Signal Processing*, pp. 3445-62, 1993.
[6] A. Moffat, R. Neal, I. H. Witten. "Arithmetic Coding Revisited", *ACM Transactions on Information Systems*, 16(3):256-294, 1998.