STEGANALYSIS OF ±1 EMBEDDING USING LOSSLESS IMAGE COMPRESSION

Charles Boncelet[*]

University of Delaware
Newark DE 19716
boncelet@udel.edu

Lisa Marvel

US Army Research Laboratory
APG, MD 21005
marvel@arl.army.mil

ABSTRACT

We present a method for detecting steganography using ±1 embedding. The method uses a lossless compression technique to compress the last two bitplanes in an effort to model the image structure where the data may be hidden. A small number of statistics are then computed using the model and fed into a support vector machine to classify detection results. Results presented are obtained using k-fold cross-validation method using a large set of never compressed grayscale images. Detection results show improvement over current ±1 steganalysis methods.

1. INTRODUCTION

Steganography is the art and science of communicating by embedding hidden information in a *cover* object. The embedded data should be invisible to anyone not participating in the communication. Steganography has multiple uses such as covert communication and hiding information to maintain backward compatibility. Steganography has similarities to digital watermarking in the desire to keep information hidden but watermarking has the additional facet of requiring that the hidden message be robust to removal.

There are many techniques for steganography. Some are simple substitution methods where the hidden data is directly substituted for the information in the cover. Others can be quite complex and use communication techniques such as spread spectrum [1]. Various types of data can be embedded with hidden information. Digital images are popular choices for covers. Other choices include audio, video and even computer code and network traffic. Because steganography can be easily performed—in fact least significant bit (LSB) replacement in digital images can be accomplished with a two-line Perl script—it is desirable to be able to reliable detect the presence of steganography. Such an effort is called steganalysis.

In this paper, we present a steganalysis method that targets a type of LSB steganography for digital images called ±1 embedding. ±1 embedding is as simple to accomplish as other LSB replacement methods but much more difficult to detect. Our steganalysis method makes novel use of lossless compression to model an image and detect steganography. On a test suite of 1200 images, half with 0.5 bits per pixel of embedded information, it classifies 97% correctly. This performance is excellent.

In the following sections we will provide background on ±1 embedding and give a brief overview of the performance of current detection methods. Then in Section 3 our method will be described in detail. Experimental results will follow. Lastly, conclusion and recommendations for future work will be presented.

2. BACKGROUND

In this section, we describe ±1 embedding, and discuss existing software and techniques that attempt to detect its presence.

Both LSB replacement and ±1 embedding select a subset of the pixels pseudorandomly using a secret key known to both sender and receiver. In LSB replacement, the least significant bit of each selected pixel is replaced by a bit from the hidden message. Note, on average only half these bits will actually be changed; for the other half, the message bit is the same as the image bit already there. In ±1 embedding, if the bit must change, ±1 is added to the pixel value. Whether to use "+" or "−" is chosen randomly and has no effect on the hidden message. The detectors for both LSB replacement and ±1 embedding work the same way: the LSB for each selected pixel is the hidden bit.

Since LSB techniques are fairly easy to implement and have a potentially large payload capacity, there is a large selection of steganography software available for purchase and via shareware (e.g., www.stegoarchive.com).

There are also reliable methods used to detect LSB replacement such as sample pairs analysis [2, 3]. These methods exploit the imbalance in pairs of adjacent pixels. When a change is made in LSB replacement, odd pixels can only be decremented and even pixels can only be incremented. Such an imbalance does not exist in ±1 steganography and therefore pairs analysis does not work very well at all.

[*] Supported by NSF Grant AST-0442071.

1-4244-1437-7/07/$20.00 ©2007 IEEE

Detecting ± 1 steganography is much harder than detecting LSB replacement. One technique uses a method akin to pairs analysis to detect ± 1 embedding [4] and good performance was obtained when the cover image has been previously compressed and for $\pm k$ embedding when $k = 3$ [5]. Goljan, et al. presented a method that uses wavelet decomposition to calculate the high frequency components from which features are extracted [6]. They also incorporate the use of side information (e.g, specific camera make and model) if available.

Detection of ± 1 embedding is most challenging when the cover image is a never-compressed image. With previously compressed images much high frequency information has been removed by the compression. If ± 1 embedding is then applied to a compressed image, the added high frequency information in the image is more easily detected.

3. THE MODEL

Succesful steganalysis requires an understanding of images without steganography and of images with steganography. The steganalyst looks for differences in the two types of images.

Our method uses lossless image compression to model the image and looks for discrepancies between the model for original images and for those containing steganography. In a typical lossless compressor, a model generates predictions. These predictions are fed to an entropy coder, which performs the actual compression. The compressor generates various statistics and those statistics are fed to a support vector machine (SVM). The SVM classifies the input into two classes: not containing steganography, or containing steganography.

3.1. Lossless Compression

The lossless compression we use is called BCTW [7], for Bitplane-CTW, where CTW is the Context Tree Weighting algorithm [8, 9]. BCTW compresses an image bitplane by bitplane, from the most significant to the least significant. BCTW uses two different contexts, one for the most significant bitplane and one for all other bitplanes. The pixels within each context are quantized with a novel quantization scheme.

Since the steganography in which we are interested mostly affects the least significant bitplanes, we only consider the compression of the last two bitplanes.

The context used for the last two bitplanes is given in Figure 1. CTW weights the bits in the context in order given. Pixel 1 is the most important, followed by pixel 2, etc., until pixel 6, the least important. For the 7th and 8th bitplanes, information from previous bitplanes is used in the quantization. Let $\hat{X}$ be the unknown pixel whose e'th bit is being compressed. Since $e - 1$ bits of $\hat{X}$ have already been determined, $\hat{X}$ can be any value in a range from X_l to $X_u - 1 = X_l + 2^{8-e} - 1$. For X_l, the remaining bits are all 0's; for $X_u - 1$, all 1's. For example if the first 6 bits are all 0, then X can be any value from 0 to 3; if the first 7 bits are all 0, then X can be 0 or 1.

3	2	4
1	?	5
	6	

Figure 1: C_1, the context used for the second through last bitplanes. The "non-causal" pixels, 5 and 6, use information only from previous bitplanes.

In Figure 1, pixels 1–4 are quantized to four values: below the range, in the bottom half of the range, in the top half of the range, and above the range. Let X_i ($i = 1, 2, 3,$ or 4) be a neighboring pixel and let $Q(X_i, X)$ be the quantization of X_i. Then,

$$Q(X_i, X) = \begin{cases} 0 & X_i < X_l \\ 1 & X_l \geq X < (X_u + X_l)/2 \\ 2 & (X_u + X_l)/2 \leq X < X_u \\ 3 & X_u \leq X \end{cases} \tag{1}$$

Pixels 5 and 6 use information in previous bitplanes only (their values in the current bitplane are unknown). They are quantized to three values: below the range, in the range, and above the range. There are $4^4 * 3^2 = 2304$ entries in the CTW lookup table.

3.2. Statistics and Training

From the context and quantization model above, the following twelve statistics are kept and given to a support vector machine (SVM) for classification:

- The average bits per pixel for the 7th and for the 8th bitplanes. The idea is that these give a measure of the complexity of the image.
- The probability of the 8th bit being a 1 given two specific contexts: the first is where all neighbors are equal to X_l and the second is where all neighbors are equal to X_u.
- The probability of the 8th bit being a 1 for eight contexts chosen as described below.

First consider the case where the image does not contain any steganography. If we consider a particular context,

c, then $\Pr(B = 1|c) = p_c$, where B is the LSB being compressed. If the compression algorithm is compressing the 8th bitplane–even a little!–then some contexts must be estimating the pixels with p_c close to 0 or to 1. (If close to 1, then replace p_c by $1-p_c$.) Thus, there must be some context with

$$\Pr(B = 1|c) \approx 0. \tag{2}$$

Now, consider the same context, but for an image containing steganography. For an embedding rate of ϵ, about half the embedded bits will be different from the original bits. If we assume that context has not changed, then

$$\Pr(B = 1|c) \approx \epsilon/2 \tag{3}$$

Thus, the idea behind this steganalysis technique is to look at the best contexts and see whether the probability estimate is zero or is greater than zero.

After running the compression algorithm on the suspect image, we sort the probability estimates. To compensate for noise and statistical effects, we select eight statistics, numbering from 0: 0, 1, 3, 7, 15, 31, 63, 127. These twelve statistics are input to a SVM classifier. We use the freely available *libsvm*[10]. In particular, we use a Gaussian radial basis function kernel and five-fold cross-validation to determine the best model.

4. EXPERIMENTAL RESULTS

For testing and evaluation, we used 1200 images from the van Hateran database[11]. These images are generally outdoor, nature images. They are greyscale and have never been compressed. The images are 1536 × 1024 pixels and were converted from 16 bits per pixel to 8 bits per pixel. Generally speaking, greyscale, never compressed, images are considered to be a difficult dataset for ±1 embedding steganalysis.

In testing, we used five-fold cross-validation, meaning that the dataset is (randomly) divided into five groups. Each group is tested against the other four. I.e., the four groups are used to train the SVM classifier and the fifth group is used for evaluation. In this way, all images are used for both training and testing.

Half the 1200 images contained ±1 embedded steganography at various rates and half did not. For each rate, we trained a SVM to classify the images into two groups: containing steganography or not. The results are shown below in Figure 2

As a basis of comparison, we also show the results of the same algorithm on (the much easier to detect) LSB replacement. In this case, the classifier was trained using LSB replacement steganography in the manner described above. Clearly, LSB replacement is easier to detect, but ±1 embedding is well detected also. At a 20% embedding rate (only 10% of the pixels changed), our algorithm correctly identifies 86% of the images; at 30% embedding, 91%; and at 50% embedding rate, 97% are correctly identified. We regard this as an excellent result for ±1 steganalysis.

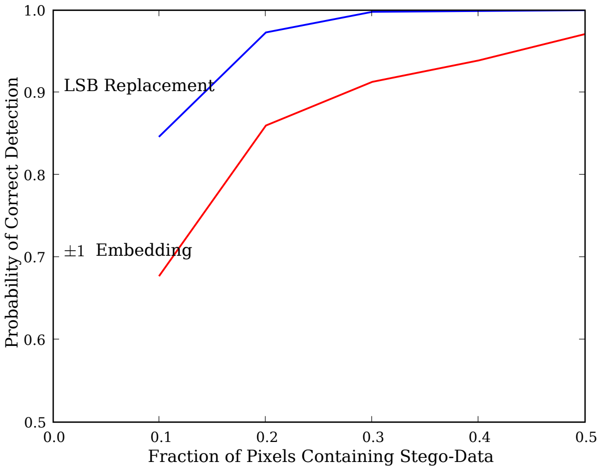

Figure 2: Probability of Correct Classification Versus Embedding Rate. Success probabilities for both LSB replacement and ±1 embedding are presented.

In Figure 3, we show Receiver Operating Characteristic (ROC) curves for two different embedding rates: 20% embedding and 50% embedding. Obviously, 50% embedding is more detectable than 20% embedding. Another common way to present False Positive rates as in Table 1.

Table 1: False Positive (FP) Rates

Embedding Rate	50% Detection	80% Detection
50%	0.00	0.003
20%	0.06	0.18

5. CONCLUSIONS AND FUTURE WORK

We have presented a new technique for detecting ±1 steganography in digital images. The technique makes use of a lossless image compressor to generate statistics and uses a machine classifier to determine whether or not a suspect image contains steganography. We presented the results of a study of 1200 never compressed, greyscale, images.

While we believe the results are excellent, much further work remains. Firstly, we need to test other methods on this same database. Then we can make a precise comparison to other methods. However, in steganalysis which method works best is not too important. Two or more methods that individually work well can be combined into a single fused method that should outperform any of them. Clearly, this method works at least well enough to be a candidate for fusion with other methods.

We also need to optimize the selection of statistics (they were somewhat arbitrarily chosen here). The execution time of our method is almost completely dominated by time to compress the image. Some study is needed to see if the compression time can be reduced without significantly impairing the overall detection performance. Also, the method should be extended to estimate the embedding fraction, ϵ.

The obvious next step is to extend this work to steganalysis of JPEG compressed images (where the hidden information is embedded in JPEG coefficients). Finally, the method presented here should also be extensible to other covers, such as audio and video.

6. REFERENCES

[1] Lisa M. Marvel, Charles G. Boncelet Jr., and Charles T. Retter, “Spread spectrum image steganography,” *IEEE Trans. on Image Processing*, Aug. 1999.

[2] A. Ker, “A general framework for structural steganalysis of lsb replacement,” in *Proc. of the 7th International Workshop on Information Hiding*, Berlin, Germany, 2005, LNCS, Springer-Verlag.

[3] S. Dumitrescu, X. Wu, and Z. Wang, “Detection of lsb steganography via sample pair analysis,” in *IH ’02: Revised Papers from the 5th International Workshop on Information Hiding*, London, UK, 2003, pp. 355–372, Springer-Verlag.

[4] P. W. Wong, H. Chen, and Z. Tang, “On steganalysis of plus-minus one embedding of continuous tone images,” in *Security, Steganography, and Watermarking of Multimedia Contents VII*, E. Delp and P. W. Wong, Eds., January 2005, vol. 5861 of *Proc. of SPIE-IS&T*.

[5] Taras Holotyak, Jessica Fridrich, and David Soukal, “Stochastic approach to secret message length estimation in $\pm k$ embedding steganography,” in *Security, Steganography, and Watermarking of Multimedia Contents VII*, E. Delp and P. W. Wong, Eds., January 2005, vol. 5681 of *Proc. of SPIE-IS&T*.

[6] Miroslav Goljan, Jessica Fridrich, and Taras Holotyak, “New blind steganalysis and its implications,” in *Security, Steganography, and Watermarking of Multimedia Contents VIII*, E. Delp and P. W. Wong, Eds., January 2006, vol. 6072 of *Proc. of SPIE-IS&T*.

[7] C. G. Boncelet Jr., “Lossless image compression with BCTW,” in *Proc. of the IEEE ICIP 2006*, Atlanta, GA, October 2006.

[8] Frans Willems, Yuri Shtarkov, and Tjalling Tjalkens, “The context-tree weighting method: basic properties,” *IEEE Trans. on Info. Theory*, vol. 41, no. 3, pp. 653–64, May 1995.

[9] S. Xiao and C. G. Boncelet Jr., “On the use of context-weighting in lossless bilevel image compression,” *IEEE Trans. on Image Proc.*, 2004, Submitted for publication.

[10] Chih-Chung Chang and Chih-Jen Lin, *LIBSVM: a library for support vector machines*, 2001.

[11] J. H. van Hateran and A. van der Schaaf, “Independent component filters of natural images compared with simple cells in primary visual cortex.,” in *Proc. Royal Society of London*, 1998, pp. 359–366.

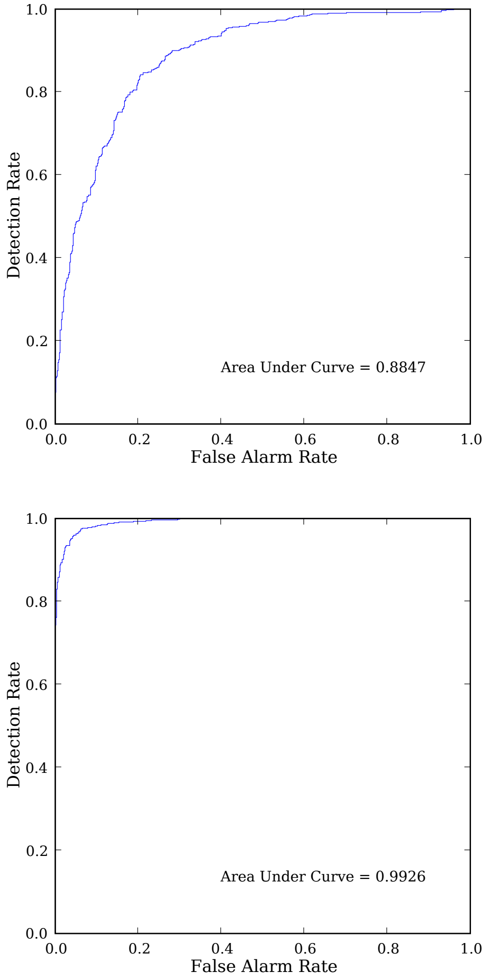

Figure 3: ROC curves for 20% and 50% embedding rates. The areas under the curves are 0.88 and 0.99.