Improving LSB Steganalysis using marginal and joint probabilistic distributions

Benoit Roue, Patrick Bas, Jean-Marc Chassery

Improving LSB Steganalysis using marginal and joint probabilistic distributions

Benoit Roue
Laboratoires des Images et des Signaux
Grenoble, France
benoit.roue@lis.inpg.fr

Patrick Bas
CNRS, Laboratoires des Images et des Signaux
Grenoble, France
patrick.bas@lis.inpg.fr

Jean-Marc Chassery
CNRS, Laboratoires des Images et des Signaux
Grenoble, France
Jean-Marc.Chassery@lis.inpg.fr

ABSTRACT
The goal of steganalysis is to search for the presence of hidden information in numerical contents. This paper is based on a previous LSB steganalysis scheme for digital images that estimates the size of the hidden message.
The accuracy of this algorithm is first outlined, then the limits are presented in order to introduce some solutions based on texture processing: analysis of histograms and coocurrence matrices are presented and their uses, to improve steganalysis using segmentation, is proposed[1].

Categories and Subject Descriptors
H.2.0 [**Software Arcitectures**]: Security

General Terms
Steganalysis

Keywords
Steganography, Steganalysis, Texture Analysis, LSB

1. INTRODUCTION
Since centuries, steganography has been used to share information between *friends* while keeping it secret to other people. In the first time techniques were simple, for example Romeo could have used invisible ink to write his love to Juliet...
Steganographic techniques have to embed a message in such a way that it could not be found, neither with analytic techniques, nor visually (c.f. Figure 1). On this point steganography differs with watermarking which is more devoted to embed robust watermarks.
Nowadays with the appearance of numerical contents, like digital images or sharewares, steganography is more and more used via internet. The hidden information may represent particular key points and may be used for strategic and politic purposes [5].

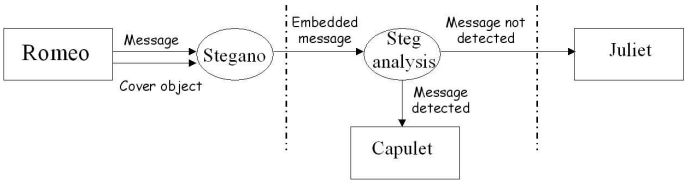


Figure 1: Steganography's problem.

The goal of steganographic analysis, also called steganalysis, is in a first step to bring out drawbacks of steganographic schemes by proving that an hidden information is embedded in a content. The estimation of the size of the hidden information, which, for example, is proportional with the probability p that a pixel has been modified by LSB embedding, can also be performed in a second step. Such a schemes may be used by organisms of security to verify that internet sites does not keep suspect (e.g. hidden) information. Moreover steganalysis techniques can help a customer to protect his privacy by proving that a numerical product such that an Internet software is not a *spyware* and does not sent private information without the knowledge of the customer.
A lot of steganographic techniques have been developed in the past years, they can be divided into two classes: *ad hoc* schemes (schemes that are devoted to a specific steganographic scheme) [4][7] and schemes that are generic and that use classifiers to differenciate original and stego images[3]. The last one work in two steps, you have first to extract the feature vectors (high pass components, prediction of error...) and then to train the classifier to separate steganalysed images from original images. The first category is much more efficient than the second and consequently we have focused our paper on Low Significant Bit *ad hoc* techniques.

2. LSB EMBEDDING IN THE SPATIAL DOMAIN
The LSB insertion scheme (c.f. Figure 2) is obviously the simplest way to hide information in cover data, although a

[1]This work has been done in the context of the national project ACI-SI Fabriano.

great amount of bits can be ranged without causing perceptible degradation of the cover object.
Indeed, digital images are generally coded with eight bits by colour channel, so an insertion in the low significant bitplane is not visible.
It is obvious that such a scheme is not interesting in water-

Figure 2: Message embedding in the Low Significant Bit of an image.

marking, because the watermark would not be robust and could easily be removed, but , again, in steganography the robustness is not the main point.
In this paper we restrict the scope of our studies to LSB embedding in the spatial domain, but other LSB schemes exist in transformed domain (*e.g.* insertion in the DCT domain [2]).

3. DUMITRESCU'S SCHEME

The goal of this section is to explain the scheme introduced by Dumitrescu *et al.* in [7] which can be seen as a mathematical formulation of the RS method, one of the first LSB steganalysis scheme presented by Fridrich [4].

3.1 Definitions

This scheme is based on the statistical analysis of adjacent sample pairs of a numerical image. The authors define P as the multiset of values of sample pairs, P is partitioned in two submultisets D_n and C_m :

- D_n is defined such as the values of the sample pair are $(u, u+n)$ or $(u+n, u)$ with $0 \leq n \leq 2^b - 1$.
- C_m is defined such as the values of the sample pair after right shifting are $(u', u'+m)$ or $(u'+m, u')$ with $0 \leq m \leq 2^{b-1} - 1$.

In those two definitions, b is an integer equal to the number of bits which are used to code a sample (*e.g.* $b = 8$ in a 256 gray levels image).
This leads to notice that the multisets C_m are invariant under the LSB embedding.
Finally D_{2m+1} is fractioned into two sub-multisets X_{2m+1} and Y_{2m+1}, defined such as :

$$X_{2m+1} = D_{2m+1} \cap C_{m+1} \text{ et } Y_{2m+1} = D_{2m+1} \cap C_m.$$

An other way to define X_{2m+1} and Y_{2m+1} is to say that the sample pair having the largest component which is odd is in Y_{2m+1} and the sample pair having the largest component which is even is in X_{2m+1} (cf. Figure 3).

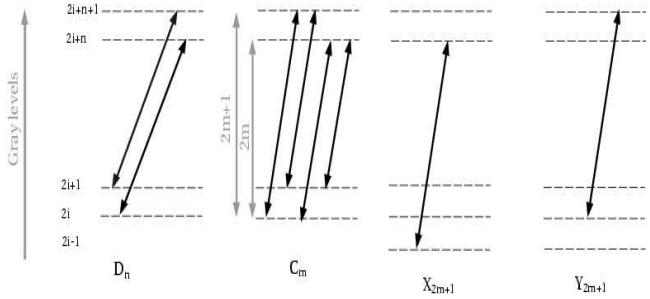


Figure 3: *Definitions of the several sets.*

It is obvious that, in a natural image, there is no reason for a sample pair to have its largest value to be odd rather than even. This leads us to the main assumption of the method on which is based the steganalysis:

$$E\{|Y_{2m+1}|\} = E\{|X_{2m+1}|\}.^2 \quad (1)$$

To summarize, C_m, which is closed under LSB embedding, is divided into four submultiset, which are:
X_{2m-1}, X_{2m}, Y_{2m} and Y_{2m+1}.
The variation of the number of elements of those multisets considering the message insertion will lead us to estimate, as we will see in the next section, the length p of the embedded message thanks to equality (1).

3.2 Detection of LSB Steganography

As presented by Dumitrescu *et al.* in [7] the effects of the LSB embedding on the multisets X_{2m-1}, X_{2m}, Y_{2m} and Y_{2m+1} can be be totally described with a modification pattern $\pi \in \{00, 01, 11, 10\}$. Those modifications are listed in the table 1. Let $\rho(\pi, P)$ be the probability that the sample pair of P is modified with π and p be the length of the embedded message. Then, if the message is randomly inserted in the image, this leads to the following probabilities:

- $\rho(00, P) = (1 - \frac{p}{2})^2$,
- $\rho(01, P) = \rho(10, P) = \frac{p}{2}(1 - \frac{p}{2})$,
- $\rho(11, P) = (\frac{p}{2})^2$.

	π			
	00	01	11	10
X_{2m-1}	X_{2m-1}	Y_{2m}	Y_{2m+1}	X_{2m}
X_{2m}	X_{2m}	Y_{2m+1}	Y_{2m}	X_{2m-1}
Y_{2m}	Y_{2m}	X_{2m-1}	X_{2m}	Y_{2m+1}
Y_{2m+1}	Y_{2m+1}	X_{2m}	X_{2m-1}	Y_{2m}

Table 1: Modifications of the several multisets under LSB embedding.

[2] $|.|$ denotes the number of elements of a set.

Finally, thanks to equality (1) the authors estimate the message's length with the following quadratic equation ((2) and (3)):

$$\frac{(|C_m|-|C_{m+1}|)p^2}{4} - \frac{(|D_{2m}|-|D_{2m+2}|+2|Y_{2m+1}|-2|X_{2m+1}|)p}{2} + |Y_{2m+1}| - |X_{2m+1}| = 0, \quad m \geq 1 \tag{2}$$

and

$$\frac{(2|C_0|-|C_1|)p^2}{4} - \frac{(2|D_0|-|D_2|+2|Y_1|-2|X_1|)p}{2} + |Y_1| - |X_1| = 0, \quad m = 0. \tag{3}$$

Where p denotes the proportion of modified pixels under LSB embedding.
In order to improve the accuracy of the estimated message lenght Dumitrescu *et al.* propose to extend equations (2) and (3) with $0 \leq m \leq 30$.
So instead of equalities (2) and (3), we will estimate p thanks to equalities (3) and (4):

$$\frac{(\sum_{m=1}^{30}|C_m|-\sum_{m=1}^{30}|C_{m+1}|)p^2}{4} - \frac{(\sum_{m=1}^{30}|D_{2m}|-\sum_{m=1}^{30}|D_{2m+2}|+2\sum_{m=1}^{30}|Y_{2m+1}|-2\sum_{m=1}^{30}|X_{2m+1}|)p}{2} + \sum_{m=1}^{30}|Y_{2m+1}| - \sum_{m=1}^{30}|X_{2m+1}| = 0. \tag{4}$$

In the following sections, we wrongly represent the sum of all the cardinals, $0 \leq m \leq 30$, $\sum_{m=0}^{30}|X_{2m+1}|$ by $|X_{2m+1}|$ and $\sum_{m=0}^{30}|Y_{2m+1}|$ by $|Y_{2m+1}|$.

4. EVALUATION AND LIMITS OF THE ALGORITHM

We have tested this method on a free database of 108 images from the kodak image database ([1]).

4.1 Evaluation setup

The goal of this section is to describe how is implemented the algorithm in order to improve its performances.
The several steps of the used method are the following:

- The relative difference between X_{2m+1} and Y_{2m+1}, in order to verify the assumption (1), is also computed with:
$$\epsilon = \frac{2 * ||X_{2m+1}| - |Y_{2m+1}||}{|X_{2m+1}| + |Y_{2m+1}|}$$
- The message is randomly generated with k keys. In the practical experiments we generally use $k = 10$.
- This message is scattered in eleven ratios of the cover image, in such a way that $0\%, 10\%, 20\%, \ldots, 100\%$ of the low significant bitplane of the image is perturbed.
- Then the embedded message's length is estimated via the equations (2) and (3) on each ratio, m varies from 0 to 30.
- Finally, the MAE (Mean Absolute Error) is computed for each ratio of each key, by computing the difference between the real length and the estimated lenght of the message.

This method is used for each image from the database, as a final result we have got the mean of the MAEs, which is used to describe the general performance of the algorithm.

4.2 Results

From the tests we have noticed that the accuracy of the presented scheme is very good for almost 70% of the database (EAM$\leq$ 3%). Indeed the results obtain on the whole set of images (Table 2) show that the algorithm have very high estimation errors (EAM above 10%) for only 2% of the images, and have very low estimation errors (EAM under 1%) for 13% of the images.
In the last case, images correspond to homogeneous images like the image "jellies" presented in the figure 4.
In the whole set of images the estimation errors are relatively low, indeed the EAM is of $2,7\%$ and the equality 1 is respected, on the set we have $\epsilon = 0,023$.

MAE	number of images (%)
$\leq 1\%$	13
$\leq 2\%$	42
$\leq 3\%$	69
$\leq 4\%$	85
$\geq 6\%$	4
$\geq 10\%$	2

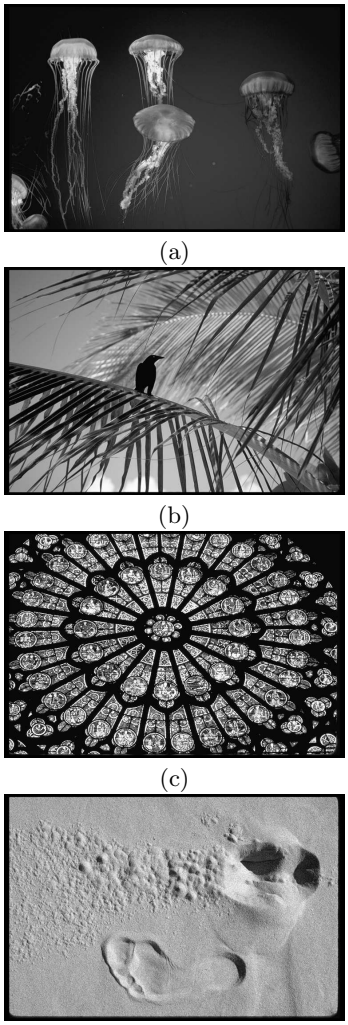
Table 2: Results on the whole set of images.

Figure 4 represents selected images that provide both low and high estimation errors.

Even if the scheme yields to very accurate estimations of the message's length, it is interesting to point out the image's features that lead to important estimation errors. These characteristics have to be identified in order to:
- detect images that are likely to counter steganalysis schemes, *e.g.* images that may contain a high ratio of undetectable hidden information;
- improve the original scheme by reducing the outlined drawbacks.

4.3 Limits and analysis

As we said in the past section, the studied scheme is really efficient on most of images, but in several cases the Mean Absolute Error becomes significant (refer to table 3). In this case, our analysis leads to the fact that important estimation errors are due to two different drawbacks:
- the non-equality of the cardinals,
- a non-adapted distribution of the joint statistics of the image.



(a) (b) (c) (d)

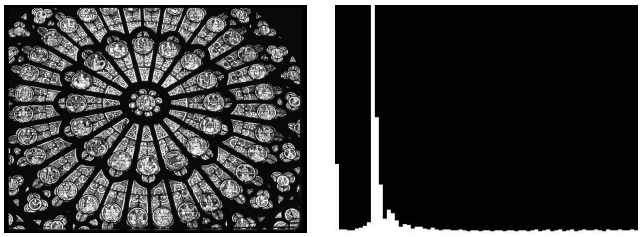
Figure 4: ***Several tested images: jellies (a), crow (b), notredamewindow (c) and sandprints (d).***

Image	MAE (%)	ϵ
jellies	0.54	0.017
crow	0.87	0.004
notredamewindow	12.01	0.22
sandprints	15.18	0.046

Table 3: Examples of errors.

4.3.1 Non-equality of the cardinals

Because the presented scheme use an *a priori* hypothesis that $E\{|Y_{2m+1}|\} = E\{|X_{2m+1}\}$, this hypothesis has to be checked. However there exists natural images that do not respect this hypothesis: for example image having large uniform areas such as shadows or sky may contain isolated peaks in their histogram.
As we can see in the table 3, for the image "notredamewindow" the relative difference between X_{2m+1} and Y_{2m+1} is not insignificant.
Indeed, there is a factor 10 between this value and mean of the relative differences (Table 3). The histogram of this image is depicted on Figure 5. On this histogram it is possible to notice the presence of a singular peak defined by an isolated double peak for pixel values equal to 9 and 10. Thus, such a configuration leads to an important increase of the value of $|Y_1|$ and consequently will break assumption (1). Our analysis has outlined the fact that this drawback is the explanation if only about 10% of a wrong estimation of the message length, nevertheless a solution is proposed in section 5.1.



(a) (b)

Figure 5: ***Image notrewindow (a) and its histogram's region of interest (b).***

4.3.2 Joint statistics of the image

For several images the studied scheme is not accurate even if assumption (1) is verified, in this case the problem must come from another reason, and more particularly, it comes from the empirical choice of the values of m.
Indeed, let take the image "sandprints", in this image we can notice that $X_{2m+1} \simeq Y_{2m+1}$, and nevertheless the Mean Absolute Error is far from being insignificant (table 3).
In section 3.1 we have seen that Dumitrescu's scheme is based on joint statistics (statistics on adjacent sample pairs), so it is interesting to analyse the statistics of images that lead to high estimation errors in order to compare with those providing reliable estimations.
Such an analysis is possible by using the *coocurrence matrix* which represents the joint statistics of the image. The coocurence matrix takes into account the probability for a sample to have a grey level, knowing its adjacent sample value, and is computed as following:

$$MC_t(a,b) = Card\{(s, s+t) \in I^2 \setminus V[s] = a, V[s+t] = b\} \quad (5)$$

With $MC_t(a,b)$ the element of the coocurrence matrix in (a,b), s a sample of the image, $V[s]$ the value of the sample, I the image and t the distance between the two pixels.
In order to be coherent with the studied algorithm the pixels analysed are adjacent, in that case t is equal to (1,0).
In order to observe the influence of the statistics of the images on results, we have computed the coocurrence matrices of the whole database. In the Figure 6 we give coocurrence matrices of the images presented in Figure 4.

We can notice two important points:

- For images presenting a low estimation error, the distribution of the maxima of joint probabilities are well located along the diagonal, it means that the way of

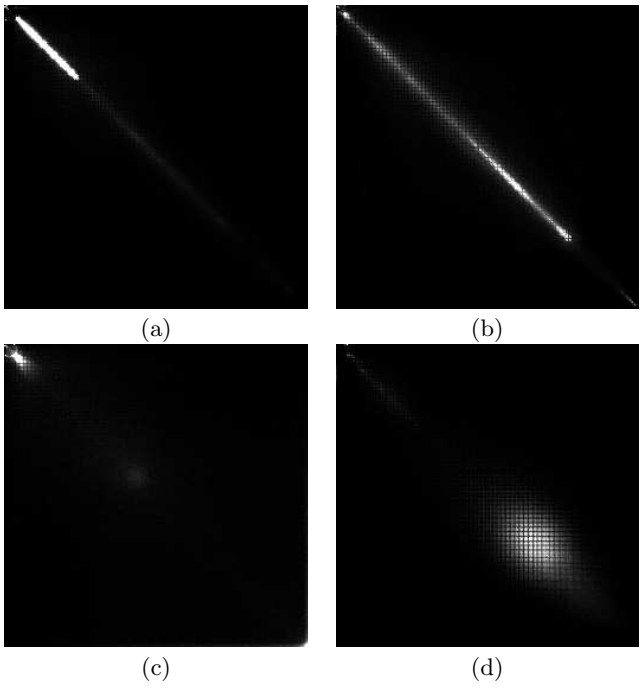


Figure 6: ***Coocurrence matrices of Tested Images: jellies (a), crow (b), notredamewindow (c) and sandprints (d).***

summation on m is efficient. Indeed the samples take into account will be enough representative to have a good estimation of the message's length.

- For images that do not work the distribution of the maxima of joint probabilities are not located at all, therefore the summation is done on pixel pair which have not a difference of value enough represented. This leads us to take care of the distribution of the coocurrence matrix. Indeed it seems to be optimum to only take into account sample pairs that are largely represented in the coocurrence matrix because such pairs represent pixels which belong to textured areas of the image.

5. IMPROVEMENTS

The goal of this section is to proposed some improvements to the studied scheme, in order to compensate the related drawbacks.

5.1 Segmentation of the marginal probability distribution

As we have seen the difference of cardinals which break the assumption (1) leads to huge estimation errors.
If the cause of the non equality of the cardinals is that there is a double peak for pixel values equal to x and y in the histogram, we propose to not take care, during the steganalysis, of sample pair that have values (x, y). This will restore the equality between cardinals, and lead to a better estimation of the message's length.

5.2 Segmentation of the joint probability distribution

As we have seen in section 3.2 the values of m are empirically chosen from 0 to 30, but for some images it seems to not be adequate, so the goal of this section is to introduce an algorithm that take the statistics of the analysed image into account. The implementation of our algorithm is the following (also see Figure 7):

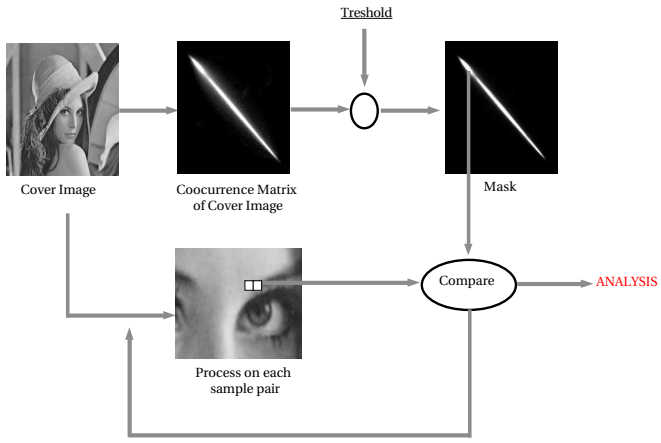


Figure 7: ***Algorithm used.***

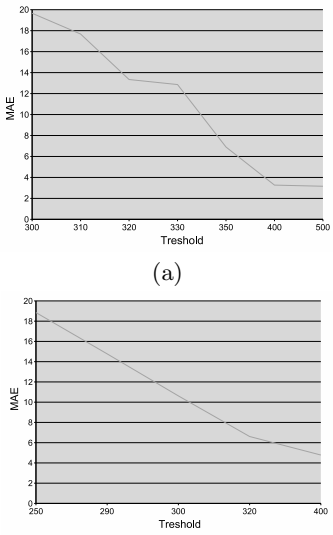
- The way of embedding the message is the same that in section 4.1 (*i.e.* 11 ratios of the image are used for 10 different keys).
- For each embedded message we compute the coocurrence matrix of the cover image.
- Then this coocurrence matrix is threshold in order to keep only the largest values (we take 10 values of threshold).
- We only estimate the message's length on sample pairs that have their difference of values contain in the segmented matrix.

To summarize we only sum the sets' cardinals when m is enough represented in the coocurrence matrix.

5.3 Threshold of the matrix of coocurrence

The choice of the threshold of the coocurrence matrix is important in this method. Therefore we have computed the EAM for several thresholds, in order to understand the meaning of this value.
For several thresholds the estimation of the message's length is not efficient at all. For example if we take a very low value, it is the same as to use all the coefficients of the matrix of coocurrence.
On an other side if we take a too much big value, we will compute the message's length on too few sample pairs of pixels. We have to find a compromise for the choice of the threshold.

The results shown in the graphic 8 describe the variation of the the MAE according to the threshold of the coocurrence matrix of the two images "duneprints" (on the left) and "sandprints" (on the right).



(a)

(b)

Figure 8: ***MAE as a fonction of the treshold.***

We must notice that if we take a threshold too much big, the results are skewed because too few pairs of pixels are taken into account, therefore we have to be careful with choice of this threshold.

5.4 Experimental results

We have improved both algorithm from sections 5.1 and 5.2 on the images presenting a high MAE.

5.4.1 Segmentation of the marginal probability distribution

Here the image on which the scheme was tested is "notredamewindow". The results are really good, indeed the error estimation is divided by nearly 3 (Table 4).

Image	MAE (%)	
	before algo	after algo
notredamewindow	12.01	4.23

Table 4: Estimation errors before and after the segmentation of the histogram.

5.4.2 Segmentation of the joint probability distribution

We have tested this scheme on the three images of the kodak database that have the worst results. For each image we have chosen the best threshold (*i.e.* the one which provides best estimations in Section 5.3), and the estimation errors lie in the Table 5.

Image	MAE (%)	
	before algo	after algo
notredamewindow	12.01	4.23
sandprints	15.18	4.2
duneprints	10.14	3.2

Table 5: Estimation errors before and after the segmentation of the coocurrence matrix.

The estimations of the embedding message's length are by this way, hugely improved, for this kind of images.
It is clear that an automatic choice of the the threshold would be a best improvement, and at the moment we work on an adaptive contrast function (based on the contrast function computed thanks to the coocurrence matrix and presented by Haralick in [6]) that could determine what should be the optimum threshold for a given image.

6. CONCLUSIONS AND PERSPECTIVES

The dumitrescu's scheme, that is very efficient on most of images, does not take the image texture into account, and by this way, is not always reliable.
But thanks to texture image processing our perspectives are to compensate this drawback using the coocurence matrix to differentiate image which provide low or high MAE, and thanks to a contrast function decide which threshold is to take.
An other track to follow is to transform from high frequency to low frequency the seventh Highest Significant Bitplanes of images that do not have good estimations.

7. REFERENCES

[1] Kodak database. ftp://ftp.kodak.com/www/images/pcd/.

[2] D.Upham. Jpeg-jsteg, modification of the independent jpeg group's jpeg software for 1-bit steganography in jfif output files. ftp://ftp.funet.fi/pub/crypt/steganography/, 1992-1997.

[3] H. Farid. Detecting hidden messages using higher-order statistical models. In *International Conference on Image Processing*, NY, 2002.

[4] J.Fridrich, M.Goljan, and R.Du. Reliable detection of LSB Steganogeraphy in color and grayscale images. In *ACM Workshop on Multimedia and Security*, pages 27–30, 2001.

[5] J.Fridrich, M.Goljan, and R.Du. Detecting LSB Steganogeraphy in color and grayscale images. In *Magazine of IEEE Multimedia Special Issue on Security*, pages 22–28, October 2001.

[6] R.M.Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. In *IEEE Transactions on Systems, Man, and Cybernetics*, 1973.

[7] S.Dumitrescu, X.Wu, and Z.Wang. Detection of LSB steganography via sample pair analysis. In *IEEE transactions on Signal Processing*, pages 1995–2007, 2003.