

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSI*net*

Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Invertible Transform-Based Reversible Text Watermarking

¹Zhijie Liu, ¹Xingming Sun, ¹Yuling Liu, ²Lincong Yang, ¹Zhangjie Fu, ¹Zhihua Xia and ¹Wei Liang
¹School of Computer and Communication, Hunan University, Changsha, 410082, China
²School of Journalism and Communication, Hunan University, Changsha, 410082, China

Abstract: In some critical application areas, it is not allowed to modify the contents of the text, like military, legal and literature fields. Therefore, restoring the original contents of the text becomes a practical and important issue for text watermarking. This study aims to deal with this problem. We firstly present the concept of reversible text watermarking and then bring forth an effective scheme to achieve reversible text watermarking. Based on the synonym substitution method, the proposed algorithm applies an invertible transform to embed watermark, extract watermark and recover the original contents of the text. By using the reversible watermarking scheme, one can not only protect the interests of the author of the text, but also get the original contents of the text, if the receiver has the wish and right to revert the original contents of the text. Moreover, the scheme improves the payload capacity via using high embedding level, compressing the watermark or repeating the algorithm more than one time.

Key words: Reversible text watermarking, invertible transform, natural language watermarking, synonym substitution

INTRODUCTION

With the rapid development of network, there are many multi-media contents in the network. However, text is still the main medium for transferring and distributing information of the internet (Topkara *et al.*, 2005). Thus, how to prevent the text information from pirating is a key task, in terms of which, the technique of text watermarking is very useful and it has been studied a lot in recent years (Qadir and Ahmad, 2006).

Currently, there are mainly three kinds of text watermarking schemes: schemes taking texts as binary images (Ge *et al.*, 2002), schemes using characteristics of the text (Brassil *et al.*, 1994) and schemes using natural language processing (Topkara *et al.*, 2006). The former two do not change the contents of the text, but they can easily be attacked by retyping; the third one called natural language watermarking has a great advantage over the former two with reference to robustness. So, natural language watermarking is the most prominent scheme for text watermarking. However, every coin has two sides; natural language watermarking has the defect that it may twist the meaning of the words, sentences or even the whole text (we name it as meaning distortion). One typical method of natural language watermarking is the synonym substitution by which the original words are replaced by their synonyms, which often causes the change of semantic meaning.

The meaning distortion generated by natural language watermarking is often quite small and imperceptible, but in military, legal and literature fields, even a very slight change of the contents of the text is undesirable. For example, if a legal paper uses natural language watermarking to embed the author's information, it may create a meaning distortion and mislead a judge to make a false verdict. The practical demands expedite the research on a watermarking scheme which can revert the contents of the text as well as embed watermark and extract watermark. There exists such a scheme for image, called reversible watermarking (Feng *et al.*, 2006). Reversible watermarking is a novel watermarking technique which enables the encoder to embed some secret information into the image and the decoder to extract the embedded information exactly and restore the total original image. Reversible watermarking has been researched for several years for image and many successful results have been obtained. Up to now, there are three categories of reversible watermarking algorithms available: the lossless compression approaches (Celik *et al.*, 2005), the histogram approaches (Yang *et al.*, 2004) and the expansion approaches (Tian, 2002; Thodi *et al.*, 2007; Coltuc and Chassery, 2006; Coltuc, 2007; Coltuc and Chassery, 2007; Chrysochos *et al.*, 2009). Yet, to the best of our knowledge, no research was done on reversible text watermarking. In this study we firstly put forward the

concept of reversible text watermarking and then we offer an effective method to study the reversible text watermarking, because the algorithms for image are not suitable for text out of the differences between text and image. In our reversible watermarking scheme, we use an invertible transform which applies the feature of the floor function to embed watermark, extract watermark and revert the contents of the text. The experimental result shows that we can improve the embedding capacity by using high embedding level or compressing the watermark or repeating the scheme more than one time.

REVERSIBLE TEXT WATERMARKING AND THE INVERTIBLE TRANSFORM

Here, we present three matters: the definition of reversible text watermarking, the three main differences between image and text and the details of the invertible transform used in our proposed scheme.

Reversible text watermarking: In accordance with the concept of the reversible watermarking for image, we define reversible text watermarking as follows: reversible text watermarking is the technology for embedding covert information into the text and restoring the original contents of the text as well as extracting the embedded information from the text. And the reversible text watermarking is based on natural language watermarking which alters the contents of the text. Here, restoring the original contents of the text refers to the recovery of the original words and sentences in the text having been changed through the watermarking process.

The technique of reversible watermarking meets the requirements of robustness, imperceptibility and readily embedding and extracting. Besides, it owns the following features differing from the traditional nonreversible watermarking:

- Blind embedding and extracting, which means the original contents is reverted while extracting the watermark
- High embedding capacity, which indicates adequate space in the text to embed covert information

The procedures of embedding and extracting are almost the same as traditional nonreversible watermarking schemes, except that the original contents can be reverted while extracting the watermark.

Differences between text and image: Image is a matrix made up of pixels which are organized in a predefined order, while text refers to a list of words and punctuations. And there exist some differences between them which can be used for reversible text watermarking:

- The range for pixel values is a constant value for a certain gray level, e.g., for gray level 8, the range of pixel values is (0, 255). However, the range of synonym indexes (we name the index of a word in its synonym set as synonym index) does not follow such a law, e.g., for a word having 4 synonyms, the range of the synonym indexes is (0, 3), while for a word having 8 synonyms, the range of the synonym indexes is (0, 7)
- Vast changes to pixel values affect the imperceptibility of image, but the alterations of the synonym indexes do not change the feelings of human beings
- The threshold of pixel values is much bigger than that of synonym indexes

The method which is demonstrated earlier is based on the above three differences.

Invertible transform: Inspired by Coltuc and Chassery (2006, 2007), we choose to use the following invertible transform for the proposed algorithm.

Let (x_1, x_2) be a pair of integer values whose domain is:

$$D = (0, 2^L - 1) \times (0, 2^L - 1)$$

where, L is the embedding level which controls the length of an embedding unit and n is a positive integer constant. Then, our transform is defined as:

$$\begin{cases} x_1' = (n+1)x_1 - nx_2 \\ x_2' = -nx_1 + (n+1)x_2 \end{cases} \quad (1)$$

Here, we should make sure (x_1, x_2) is in the specified domain D . Therefore, we have the following equations as our constraints:

$$\begin{cases} 0 \leq x_1' \leq 2^L - 1 \\ 0 \leq x_2' \leq 2^L - 1 \end{cases} \quad (2)$$

The inverse transform of Eq. 1 is defined as follows:

$$\begin{cases} x_1 = \left\lfloor \frac{n+1}{2n+1} x_1' + \frac{n}{2n+1} x_2' \right\rfloor \\ x_2 = \left\lfloor \frac{n}{2n+1} x_1' + \frac{n+1}{2n+1} x_2' \right\rfloor \end{cases} \quad (3)$$

where, $\lfloor x \rfloor$ is the floor function which returns the greatest integer less than or equal to x .

The proof that Eq. 3 is the inverse transform of Eq. 1 can be performed via getting off the floor function, which is greater than $x-1$ and less than or equal to x .

In the following contexts, we set n to 1 to illustrate the proposed algorithm.

Let's define:

$$(x_1', x_2') = \left(2 \times \left\lfloor \frac{x_1'}{2} \right\rfloor, 2 \times \left\lfloor \frac{x_2'}{2} \right\rfloor \right)$$

to replace (x_1', x_2') in Eq. 3 and assume $(x_1', x_2') \in D$: if $\text{LSB}(x_1') + \text{LSB}(x_2') = 0$, then, (x_1, x_2) is exactly calculated from Eq. 3; if $\text{LSB}(x_1') + \text{LSB}(x_2') \neq 0$, then, (x_1, x_2) is decreased by (1,1) from Eq. 3.

Thereby, we can use the LSBs of (x_1, x_2) to embed information. Firstly, we transform (x_1, x_2) into (x_1', x_2') by Eq. 1. If $(x_1', x_2') \in D$ and $\text{LSB}(x_1') + \text{LSB}(x_2') = 0$ then:

$$2 \times \left\lfloor \frac{x_1'}{2} \right\rfloor + 1 \rightarrow x_1$$

($y \rightarrow x$ means to assign y to x) and let $\text{LSB}(x_2')$ be available for coding one bit information; if $(x_1', x_2') \in D$ and $\text{LSB}(x_1') + \text{LSB}(x_2') \neq 0$, then:

$$2 \times \left\lfloor \frac{x_1'}{2} \right\rfloor \rightarrow x_1$$

and let $\text{LSB}(x_2')$ be available for coding one bit information; if $(x_1', x_2') \notin D$, then record $\text{LSB}(x_2')$ for recovering the original x_1 and

$$2 \times \left\lfloor \frac{x_1'}{2} \right\rfloor \rightarrow x_1$$

There may be some successive un-embeddable pairs which do not satisfy (Eq. 2). In order to recover the original pairs, we need to embed the LSBs of the first elements in all successive un-embeddable pairs as well. In order to improve the embedding capacity, we can just embed the LSBs of the first elements of the successive un-embeddable pairs at odd positions and let the other pairs at even positions be unchanged. By doing so, we can save $c/2$ bits available for embedding covert information, where, c is the number of the successive un-embeddable pairs.

PROPOSED METHOD

As stated earlier, reversible text watermarking refers to natural language watermarking. And the most popular natural language watermarking scheme is synonym substitution. For a word w having synonyms and watermark bits b in binary representation, the synonym substitution method just replaces w with its synonym

whose synonym index is b . Assuming the synonym number for word w , which is the number of words in the synonym set of w , is n , the synonym substitution method can embed $\lfloor \log_2 n \rfloor$ bits information into the position of w .

Our proposed method is based on the synonym substitution scheme.

Regardless of the three differences between text and image, we still treat the synonym indexes as the pixel values through a simulation process described in the following subsection. After that, we can embed watermark, extract watermark and recover the original contents of the text.

The general procedure of our scheme consists of the following 3 steps:

- Convert the synonym indexes of words in the text into pixel values
- Use the above mentioned invertible transform and the synonym substitution scheme to embed watermark, extract watermark and restore the original synonym indexes of words
- Get the pixel values back into synonym indexes and replace synonym indexes with words

In the following subsections, we expand the 3 steps in details.

Preparation work: As required by the general process of our scheme, we should perform the following pixel value simulation procedure before embedding watermark or extracting watermark: Concatenate all synonym indexes of words in the text into a single string and divide the concatenated string into groups of embedding level L before performing step 2. The algorithm can be performed as algorithm 1 describes.

Algorithm 1: Pixel value simulation

Input: Text T , synonym database DB , embedding level L

Output: Synonym words list $synonym$, simulated pixel values list P whose gray level is L

Body:

1. Split T into words by adding the words whose synonym number is greater than 1 into synonyms;
 2. For each word w in synonyms:
 - 2.1. Find the synonym index of w in DB as i ;
 - 2.2. Get the synonym number of w as n ;
 - 2.3. Calculate the binary representation of i and save it as $binInd$ of which adds '0' in the front if the bits count is not $\lfloor \log_2 n \rfloor$;
 - 2.4. Append the $binInd$ into the $binList$;
 3. For each element in $binList$:
 - 3.1. Concatenate all of them into one string str ;
 4. Split str into pieces of L bits, get the decimal values and put them into P .
-

After embedding the watermark into the simulated pixel values or recovering the original simulated pixel

values, we should turn the pixel values into the synonym indexes and get the text by using the corresponding words for the synonym indexes (algorithm 2). The algorithm is described below.

Algorithm 2: Synonym indexes recovering

Input: Synonym words list synonyms, simulated pixel values list P, synonym database DB, embedding level L

Output: watermarked text T_w or the original text T

Body:

1. **For** each element p in P:
 - 1.1. Get the binary representation of p as binStr of which adds 0 in the front if the length of binStr is not L;
 - 1.2. Concatenate binStr into one string str;
2. **For** each word w in synonyms:
 - 2.1. Get the synonym number of w as n;
 - 2.2. Get the first $\lfloor \log_2 n \rfloor$ from str as synonym index i;
 - 2.3. Get the synonym for w whose synonym index is i and replace w in the text;
3. **Return** T_w or T;

Algorithm 1 and 2 correspond to the first and last steps of the aforementioned general procedure of our scheme.

Watermarking embedding: As described in the invertible transform subsection, for a pair (x_1, x_2) , we use LSB of the first element to indicate whether one bit information is embedded into LSB of the second element (algorithm 3). The watermark embedding algorithm is shown as the following one.

Algorithm 3: Watermark embedding

Input: Simulated pixel values list P, synonym database DB, embedding level L, watermark W

Output: Updated simulated pixel values list P

Body:

1. Partition P into pairs with 2 elements: The i-th pair is represented as $p_i = (x_i, x_{i+1})$ where, N is the number of elements in P;
2. Set $p_i'' = (x_i'', x_{i+1}'')$ as the binary representation pairs after embedding information and initialize (" ", " ") $\rightarrow p_i''$;
3. Set the count of continuous un-embeddable pairs as c and initialize $0 \rightarrow c$;
4. **For** each pair p_i :
 - 4.1. Transform p_i into pair $p_i = (x_i, x_{i+1})$ through (1);
 - 4.2. **If** $p_i'' \in D$:
 - 4.2.1. **If** $c \% 2 = 0$, put $LSB(x_i)$ into the watermark, set:

$$2 \times \left\lfloor \frac{x_i}{2} \right\rfloor \rightarrow x_i \text{ and } c + 1 \rightarrow c$$

4.2.2. **If** $c \% 2 = 1$, set $c + 1 \rightarrow c$;

4.3. **If** $p_i'' \in D$:

4.3.1. **If** $LSB(x_i') + LSB(x_{i+1}') \in D$, use:

$$\left(2 \times \left\lfloor \frac{x_i'}{2} \right\rfloor + 1, 2 \times \left\lfloor \frac{x_{i+1}'}{2} \right\rfloor + b \right)$$

to replace p_i where, b is the current embedding bit and $0 \rightarrow c$;

4.3.2. **If** $LSB(x_i') + LSB(x_{i+1}') \notin D$, use:

Algorithm 3: Continued

$$\left(2 \times \left\lfloor \frac{x_i'}{2} \right\rfloor, 2 \times \left\lfloor \frac{x_{i+1}'}{2} \right\rfloor + b \right)$$

to replace p_i where, b is the current embedding bit and $0 \rightarrow c$;

- 4.4. Insert $LSB(x_i)$, $LSB(x_{i+1})$, to the start of x_i'', x_{i+1}'' and set $(x_i'' >> 1, x_{i+1}'' >> 1) \rightarrow (x_i, x_{i+1})$;
- 4.5. Set $L - 1 \rightarrow L$;
- 4.6. **If** $L > 1$, **goto** 4.1;
5. Remove all the elements in the list P;
6. Get the decimal values of p_i'' and add it to the result list P;

Watermark extracting and the original contents of the text recovering: In the phase of extracting watermark and recovering the original contents of the text, we perform the reverse operations used in the watermark embedding phase (algorithm 4). The algorithm can be executed as below.

Algorithm 4: Watermark extracting and original contents recovering

Input: Simulated pixel values list P, synonym database DB, embedding level L

Output: Extracted watermark W, updated simulated pixel values list P

Body:

1. Partition P into pairs with 2 elements: The i-th pair is represented as $p_i'' = (x_i'', x_{i+1}'')$:

$$i = 1, \dots, \left\lfloor \frac{N}{2} \right\rfloor$$

where, N is the number of elements in P;

2. Set $p_i = (x_i, x_{i+1})$ as the original pairs and initialize $(MSB(x_i''), (MSB(x_{i+1}''))) \rightarrow$, where, $MSB(x)$ means to get the most significant bit of x;
3. Set the count of continuous un-embeddable pairs as c and initialize $0 \rightarrow c$;
4. Set the current embedding level as l and initialize $0 \rightarrow c$;
5. **For** each pair p_i'' :
 - 5.1. **If** $LSB(x_i) \neq 0$:
 - 5.1.1. **If** $c = 0$, collect $LSB(x_{i+1})$ as one bit of W and $0 \rightarrow 1$;
 - 5.1.2. **If** $c \neq 0$, set $c + 1 \rightarrow c$;
 - 5.2. **If** $LSB(x_i) = 0$, set:

$$\left(2 \times \left\lfloor \frac{x_i}{2} \right\rfloor, 2 \times \left\lfloor \frac{x_{i+1}}{2} \right\rfloor \right) \rightarrow (x_i', x_{i+1}')$$

- 5.2.1. **If** $(x_i', x_{i+1}') \in D$, collect $LSB(x_{i+1})$ as one bit of W and update (x_i, x_{i+1}) from Eq. 3 using (x_i', x_{i+1}') as input and $0 \rightarrow 1$;
- 5.2.2. **If** $(x_i', x_{i+1}') \notin D$:
 - 5.2.2.1. **If** $c \% 2 = 0$, revert x_i from W and x_{i+1} as x_{i+1}' and $c + 1 \rightarrow c$;
 - 5.2.2.2. **If** $c \% 2 = 1$, set $c + 1 \rightarrow c$;
- 5.3. Set $l + 1 \rightarrow l$;
- 5.4. **If** $l \leq L$, $(x_i'' << 1, x_{i+1}'' << 1) \rightarrow (x_i'', x_{i+1}'')$, $(2 \times x_i + MSB(x_i''), (2 \times x_{i+1} + MSB(x_{i+1}''))) \rightarrow$ and **goto** 5.1;
6. Let the output be $P = \{p_i\}$.

RESULTS

Our scheme was examined in several tests. We chose six groups of texts whose number of words having

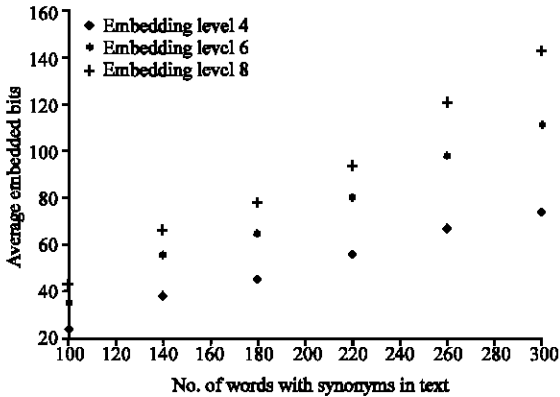


Fig. 1: Average embedded bits of the proposed scheme with embedding level 4, embedding level 6 and embedding level 8

synonyms were about 100, 140, 180, 220, 260 and 300, respectively. The synonyms used in our tests were extracted from the famous synonyms library WordNet.

Applying our proposed method with different embedding levels, results obtained from the test texts are shown in Fig. 1, The Fig. 1 demonstrates that the average embedded bits increase with the embedding level: The average embedded bits at the embedding level 6 is about 1.2 times than that at the embedding level 4; the average embedded bits at embedding level 8 is 1.7 times than that at the embedding level 4 and 1.2 times than that at the embedding level 6.

We take the embedding level 8 as an example to make a comparison about the average embedded bits obtained by performing the proposed algorithm once and twice and by adopting the Golomb Coding to compress the watermark. Besides, we also compare the results between that obtained through using the proposed algorithm and that obtained through the traditional synonym substitution method, all the test results are shown in Fig. 2.

The increases are obvious when we use compression. Through Golomb Coding, the average embedded bits at the embedding level 8 increase by 1.5 times comparing with that at the embedding level 8. Moreover, the embedding capacity can be improved by repeating our proposed scheme. At the embedding level 8, the average embedded bits obtained by performing the algorithm twice are about 1.7 times than that obtained by performing the algorithm once.

The essential feature of our proposed scheme is to recover the original contents of the text. To achieve this purpose, we need to embed extra information used to do the recovery. Hence, the embedding capacity is much lower than that of the traditional synonym substitution

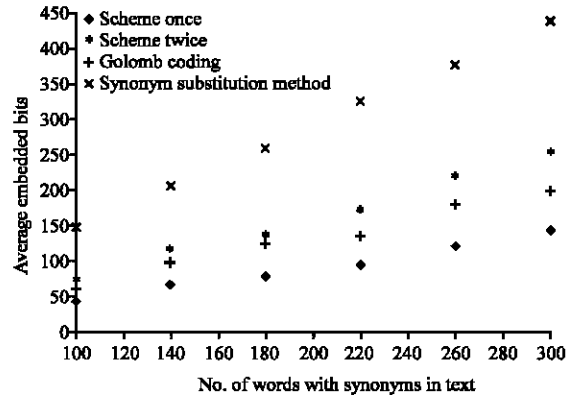


Fig. 2: Average embedded bits at the embedding level 8 by performing the proposed scheme once, performing the proposed scheme twice, adopting Golomb coding and the traditional synonym substitution method

method. However, it can still embed some information to identify the author of the text and we can improve the embedding capacity by using a lossless compression algorithm to compress the watermark or executing the algorithm more than one time.

CONCLUSION

This study described the concept and features of reversible text watermarking and proposed a practical scheme to achieve the reversible text watermarking. In this proposed scheme, we applied the synonym substitution scheme as the cornerstone and introduced an invertible transform to perform the embedding and extracting procedures.

Though the embedding capacity of our proposed scheme is lower than that of the synonym substitution scheme, our scheme as a reversible watermarking scheme can revert the original contents of the text. Hence, it is of great significance to be applied in such domains as legal, military and literature areas.

In the future, we will provide more flexible invertible transform to improve the payload capacity.

ACKNOWLEDGMENTS

The research project was partially or fully sponsored by National Basic Research Program 973 (Grant No. 2006CB303000), special for National Basic Research Program 973 (Grant No. 2009CB326202, 2010CB334706), National Natural Science Foundation of China (Grant No. 60736016, 60873198, 60973128, 60973113), and Science Program of Institutes of Higher Education of Hunan Province (Grant No. 09w023).

REFERENCES

- Brassil, J., S. Low, N. Maxemchuk and L. O'Garman, 1994. Electronic marking and identification techniques to discourage document copying. Proceedings of the IEEE INFOCOM Networking for Global Communications, Jun. 12-16, IEEE Press, Piscataway, New Jersey, pp: 1278-1287.
- Celik, M.U., G. Sharma, A.M. Tekalp and E. Saber, 2005. Lossless generalized-ISB data embedding. IEEE Trans. Image Process., 14: 253-266.
- Chrysochos, E., E.E. Varsaki, V. Fotopoulos and A.N. Skodras, 2009. High capacity reversible data hiding using overlapping difference expansion. Proceedings of the 10th Workshop on Image Analysis for Multimedia Interactive Services, May 6-8, IEEE, London, UK., pp: 121-124.
- Coltuc, D. and J.M. Chassery, 2006. High capacity reversible watermarking. Proceedings of the Image Processing 2006 IEEE International Conference, Oct. 8-11, IEEE, Atlanta, GA., USA., pp: 2565-2568.
- Coltuc, D., 2007. Improved capacity reversible watermarking. Proceedings of the Image Processing ICIP 2007, IEEE International Conferenc, Sept. 16-Oct. 19, IEEE, San Antonio, TX., USA., pp: III-249-III-252.
- Coltuc, D. and J.M. Chassery, 2007. Very fast watermarking by reversible contrast mapping. IEEE Signal Process. Lett., 14: 255-258.
- Feng, J.B., I.C. Lin, C.S. Tsai and Y.P. Chu, 2006. Reversible watermarking: current status and key issues. Int. J. Network Security, 2: 161-171.
- Ge, H., F. Su and Y. Zhu, 2002. Color image text watermarking using wavelet transform and error-correcting code. Proceedings of the Signal Processing, 2002 6th International Conference, Aug. 26-30, IEEE, Piscatway, USA., pp: 1584-1587.
- Qadir, M.A. and I. Ahmad, 2006. Digital text watermarking: secure content delivery and data hiding in digital documents. Aerospace Electronic Syst. Magazine, 21: 18-21.
- Thodi, D.M. and J.J. Rodriguez, 2007. Expansion embedding techniques for reversible watermarking. IEEE Trans. Image Process., 16: 721-730.
- Tian, J., 2002. Reversible watermarking by difference expansion. Proceedings of the ACM Multimedia 2002 Workshop W2-Workshop on Multimedia and Security: Authentication, Secrecy and Steganalysis, Dec. 1-6, Association for Computing Machinery, Juan-Les-Pins, France, pp: 157-160.
- Topkara, M., C.M. Taskiran and E.J. Delp, 2005. Natural language watermarking. Security, Steganography Watermarking Multimedia Contents, 5681: 441-452.
- Topkara, M., G. Riccardi, H. Dilek and M.J. Atallah, 2006. Natural language watermarking: challenges in building a practical system. Proc. SPIE, 6072: 106-117.
- Yang, B., S. Martin, X. Niu, B. Christoph and S. Sun, 2004. Reversible image watermarking by histogram modification for integer DCT coefficients. Proceedings of the IEEE 6th Workshop on Multimedia Signal Processing, Sept. 29- Oct. 1, IEEE, Siena, Italy, pp: 143-146.