REVERSIBLE WATERMARK USING DIFFERENCE EXPANSION OF TRIPLETS

Adnan M. Alattar

Digimarc Corporation
19801 SW 72nd Ave., Suite 250
Tualatin, OR 97062
aalattar@digimarc.com

ABSTRACT

A new reversible watermarking algorithm based on the difference expansion of colored images has been developed. Since the watermark is completely reversible, the original image can be recovered exactly. The algorithm uses spatial and spectral triplets of pixels to hide pairs of bits, which allows the algorithm to hide a large amount of data. A spatial triplet is any three pixel values selected from the same spectral component, while a spectral triplet is any three pixel values selected from different spectral components. The algorithm is recursively applied to the rows and columns of the spectral components of the image and across all spectral components to maximize the hiding capacity. Simulation results show that the hiding capacity of the algorithm is very high and the resulting distortion is low.

1. INTRODUCTION

In digital watermarking or steganography, a hardly noticeable noise-like signal is usually embedded into a digital medium, such as an image, audio, or video data to protect it from illicit use and alteration, to authenticate its content and origin, or to enhance its value and enrich its information content [1]. Unlike metadata, which is often appended to the digital file, a watermark is bound with the fabric of the media and cannot be removed or easily destroyed. The watermarking process usually introduces irreversible degradation of the original medium. Although this degradation is slight, it may not be acceptable to some applications, such as military and medical uses, and, hence, there is a need for a reversible watermark.

If the embedding algorithm and all embedding parameters are available to the reader, it might be possible to calculate the watermark and subtract it from the marked medium in order to recover the original medium, once the watermark is detected and the payload is read. Unfortunately, these requirements are often not available, and, furthermore, most watermarking algorithms often employ some kind of non-linearity to optimize their performance. Therefore, a reversible watermark must be designed such that it can be removed to restore the original medium without any reference to information beyond what is available in the watermarked medium itself.

Several researchers have developed reversible watermarks [2-9]. In Jun Tian's work [9], Tian used a difference expansion transform of a pair of pixels to embed a large amount of data into gray scale images. His algorithm allows one bit to be embedded in every pair of pixels. In this paper, we extend Tian's algorithm using difference expansion of spatial and cross-spectral triplets, instead of pairs, to increase the hiding ability of the algorithm. This allows the algorithm to embed two bits in every triplet. In the next section, we define spatial and cross-spectral triplets and their difference expansion transforms. In Section (3), we describe the proposed embedding and recovery algorithms for a reversible watermark based on the difference expansion of triplets. In Section (4), we present simulation results of the proposed algorithm. And finally, in the last section, we summarize our conclusions.

2. DIFFERENCE EXPANSION OF TRIPLETS

Triplets: A triplet is a *1x3* vector whose entities are chosen from the pixel values in the colored image. Two kinds of triplets are considered in this paper. The first kind is purely spatial, while the other is cross-spectral. The spatial triplet is formed from three pixel values chosen from three different locations within the same color component according to a predetermined order. This order may serve as a first security key. The simplest way to form spatial triplets is to consider every three adjacent (column or row-wise) pixel values as a spatial triplet.

The cross-spectral triplet is also formed from three pixel values chosen from different color components according to a predetermined spatial order. This order may serve as a second security key. The order of the color components within the triplet is predetermined and may serve as yet a third security key. The cross-spectral triplets can be formed, simply, by assembling the RGB values of each pixel into a triplet.

For simplicity, we require that each color component is treated independently, and, hence, it has its own set of spatial triplets. Also, we require that spatial triplets do not overlap each other; i.e., each pixel exists in only one spatial triplet. Similarly, we require that cross-spectral triplets do not overlap each other. These requirements may be removed at the expense of complicating the algorithm due to the extra caution needed in deciding the processing order of the overlapped triplets.

Difference Expansion Transform: The forward difference expansion transform, $f(\cdot)$, for the triplet $t = (u_0, u_1, u_2)$ is defined as:

$$\begin{aligned} v_0 &= \left\lfloor \frac{u_0 + w u_1 + u_2}{N} \right\rfloor \\ v_1 &= u_2 - u_1 \\ v_2 &= u_0 - u_1 \end{aligned} \qquad (1)$$

where N and w are constants and $\lfloor \cdot \rfloor$ is the least nearest integer. For spatial-triplets we set $N=3$ and $w=1$. This makes the transform of a spatial-triplet equivalent to the average of the triplet and the differences between the center entity and its other two entities. Also, for cross-spectral triplets we set $N=4$ and $w=2$. This makes the transform of a cross-spectral triplet equivalent to converting the triplet from RGB to YUV.

The inverse difference expansion transform, $f^{-1}(\cdot)$, for the transformed triplet $t' = (v_0, v_1, v_2)$ is defined as:

$$
\begin{aligned}
u_1 &= v_0 - \left\lfloor \frac{v_1 + v_2}{N} \right\rfloor \\
u_0 &= v_2 + u_1 \\
u_2 &= v_1 + u_1
\end{aligned} \tag{2}
$$

Definition 1: The triplet $t = (u_0, u_1, u_2)$ is said to be expandable if for all values of $b_1 \in \{0,1\}$ and $b_2 \in \{0,1\}$

$$
\begin{aligned}
N(v_0 - 255) &\le (\tilde{v}_1 + \tilde{v}_2) \le (Nv_0 + N - 1) \\
N(v_0 - 255) + N\tilde{v}_2 &\le (\tilde{v}_1 + \tilde{v}_2) \le (Nv_0 + N - 1) + N\tilde{v}_2 \\
N(v_0 - 255) + N\tilde{v}_1 &\le (\tilde{v}_1 + \tilde{v}_2) \le (Nv_0 + N - 1) + N\tilde{v}_1
\end{aligned} \tag{3}
$$

Where:

$$
\begin{aligned}
\tilde{v}_1 &= 2 \times (u_2 - u_1) + b_1 \\
\tilde{v}_2 &= 2 \times (u_0 - u_1) + b_2
\end{aligned} \tag{4}
$$

Notice that each of $\tilde{v}_1$ and $\tilde{v}_2$ is one-bit left shifted version of the original value v_1 and v_2, respectively, but potentially with a different LSB (least significant bit). The conditions of equation (3), above, ensures that changing the LSBs of v_1 and v_2 according to equation (4) does not introduce an overflow or underflow in the values of u_0, u_1, and u_2 when the inverse transform is computed.

Definition 2: The triplet $t = (u_0, u_1, u_2)$ is said to be changeable if for all values of $b_1 \in \{0,1\}$ and $b_2 \in \{0,1\}$, $\tilde{v}_1$ and $\tilde{v}_2$ given by equation (5), below, satisfy equation (3).

$$
\begin{aligned}
\tilde{v}_1 &= 2 \times \left\lfloor \frac{u_2 - u_1}{2} \right\rfloor + b_1 \\
\tilde{v}_2 &= 2 \times \left\lfloor \frac{u_0 - u_1}{2} \right\rfloor + b_2
\end{aligned} \tag{5}
$$

Notice that $\tilde{v}_1$ and $\tilde{v}_2$ in the above equation are the same as the original v_1 and v_2, but with different LSBs. Also, notice that a changeable triplet remains changeable even after changing the LSBs of its v_1 and v_2. Also, from definitions 1 and 2, it can be observed that an expandable triplet is also changeable.

3. ALGORITHM FOR REVERSIBLE WATERMARK

Let $I(i, j, k)$ be an RGB image, and assume that:

1. the pixel values in the red component, $I(i, j, 0)$, are arranged into the set of spatial triplets $T_R = \{t_l^R, l = 1 \cdots L\}$ using the security key K_R
2. the pixel values in the green component, $I(i, j, 1)$, are arranged into the set of spatial triplets $T_G = \{t_n^G, n = 1 \cdots N\}$ using the security key K_G
3. the pixel values in the blue component, $I(i, j, 2)$, are arranged into the set of spatial triplets $T_B = \{t_p^B, p = 1 \cdots P\}$ using the security key K_B.
4. the pixel values of the red, green, and blue components of the image $I(i, j, k)$ are arranged into the cross-spectral triplets $T_S = \{t_r^S, r = 1 \cdots R\}$ using the security key K_S.

Moreover, let the set $T = \{t_q, q = 1 \cdots Q\}$ represent any of the above set of triplets T_R, T_G, T_B, and T_S, and K represent its associated security key. Also, let $T' = \{t'_q, q = 1 \cdots Q\}$ be the transformation of T under the difference expansion transform $f(\cdot)$ (i.e. $T' = f(T)$ and $T = f^{-1}(T')$). Also, let $t_q = (u_0, u_1, u_2)$ and its difference expansion transform be $t'_q = (v_0, v_1, v_2)$.

The triplets in T can now be classified into three groups according to the definitions given in Section (2), above. The first group, S_1, contains all expandable triplets whose $v_1 \le T_1$ and $v_2 \le T_2$, where T_1 and T_2 are predefined thresholds. The second group, S_2, contains all changeable triplets that are not in S_1. The third group, S_3, contains the rest of the triplets (not changeable). Also, let S_4 denote all changeable triplets (i.e., $S_4 = S_1 \cup S_2$).

Let's now identify the triplets of S_1 using a binary location map, M, whose entries are 1s and 0s, where the 1 symbol indicates the S_1 triplets and the 0 symbol indicates S_2 or S_3 triplets. Depending on how the triplets are formed, the location map can be 1- or 2-dimensional. For example, if spatial-triplets are formed from adjacent pixels, column wise, the location map forms a binary image that has the same number of rows and one-third the number of columns as the original image. However, if a random key is used to identify the locations of the entries of each triplet, then the location map is a binary stream of ones and zeros. The security key and an indexing table are needed in this case to map the zeros and ones in this stream to the actual locations in the image. Such a table must be predefined and assumed to be known to both the embedder and the reader.

3.1. Embedding of Reversible Watermark

The embedding algorithm can be summarized using the following steps:

1. For every $T \in \{T_R, T_G, T_B, T_S\}$, do the following:
   a. Form the set of triplets T from the image $I(i,j,k)$ using the security key K.
   b. Calculate T' using the difference expansion transform, $f(\cdot)$ (see equation (1)).
   c. Use T' and the conditions in equation (3) to divide T into the sets S_1, S_2, and S_3.
   d. Form the location map, M; then compress it using a lossless compression algorithm, such as JBIG or an arithmetic compression algorithm, to produce sub-bitstream B_1. Append a unique identifier, *EOS*, symbol to B_1 to identify its end.
   e. Extract the LSBs of v_1 and v_2 of each triplet in S_2. Concatenate these bits to form sub-bitstream B_2.
   f. Assuming the watermark to be embedded forms a sub-bitstream B_3, and concatenate sub-bitstreams B_1, B_2, and B_3 to form the bitstream B.
   g. Sequence through the member triplets of S_1 and S_2 as they occur in the image and through the bits of the bit-stream B in their natural order. For S_1, expand the triplets as described in equation (4). For S_2, expand the triplets as in equation (5). The values of b_1 and b_2 are taken sequentially from the bitstream.
   h. Calculate the inverse difference expansion transform of the resulting triplets using $f^{-1}(\cdot)$ (see equation (2)) to produce the watermarked S_1^w and S_2^w.
   i. Replace the pixel values in the image, $I(i,j,k)$, with the corresponding values from the watermarked triplets in S_1^w and S_2^w to produce the watermarked image $I^w(i,j,k)$.
   j. Set $I(i,j,k) = I^w(i,j,k)$.

It should be noted here that the size of bitstream B must be less than or equal to twice the size of the set S_4. To meet this condition, the values of the threshold T_1 and T_2 must be properly set. Strategies for setting these thresholds are detailed in [9].

3.2. Reading Watermark and Restoring Original Image

To read the watermark and restore the original image, the following steps must be followed:

1. For every $T \in \{T_S, T_R, T_G, T_B\}$, do the following:
   a. Form the set of triplets T from the image $I^w(i,j,k)$ using the security key K.
   b. Calculate T' using the difference expansion transform, $f(\cdot)$ (see equation (1)).
   c. Use T' and the conditions in equation (3) to divide the triplets in T into the two sets $\hat{S}_4$ and S_3. $\hat{S}_4$ has the same triplets as S_4, which was constructed during embedding, but the values of the entities in each triplet may be different. Similarly, S_3 is the same set constructed during embedding, since it contains non-changeable triplets.
   d. Extract the LSBs of $\tilde{v}_1$ and $\tilde{v}_2$ of each triplet in $\hat{S}_4$, and concatenate them to form the bitstream B, which is identical to that formed during embedding.
   e. Identify the *EOS* symbol and extract sub-bitstream B_1. Then, decompress B_1 to restore the location map M, and, hence, identify the member triplets of the set S_1 (expandable triplets). Collect these triplets into set $\hat{S}_1$.
   f. Identify the member triplets of S_2. They are the members of $\hat{S}_4$ who are not members of $\hat{S}_1$. Form the set $\hat{S}_2 = \hat{S}_4 - \hat{S}_1$.
   g. Sequence through the member triplets of $\hat{S}_1$ and $\hat{S}_2$ as they occur in the image and through the bits of the bit-stream B in their natural order after discarding the bits of B_1. For $\hat{S}_1$, restore the original values of v_1 and v_2 as follows:

$$v_1 = \left\lfloor \frac{\tilde{v}_1}{2} \right\rfloor, \quad v_2 = \left\lfloor \frac{\tilde{v}_2}{2} \right\rfloor \tag{6}$$

   For $\hat{S}_2$, restore the original values of v_1 and v_2 according to equation (5). The values of b_1 and b_2 are taken sequentially from the bitstream.
   h. Calculate the inverse difference expansion transform of the resulting triplets using $f^{-1}(\cdot)$ (see equation (2)) to restore the original S_1 and S_2.
   i. Replace the pixel values in the image $I^w(i,j,k)$ with the corresponding values from the restored triplets in S_1 and S_2 to restore the original image $I(i,j,k)$.

j. Discard all the bits in the bit-stream B, which were used to restore the original image. Form the sub-bitstream B_3 from the remaining bits. Read the payload and authenticate the image using the watermark contained in B_3.

k. Set $I^w(i,j,k) = I(i,j,k)$.

5. EXPERIMENTAL RESULTS

In summary, we implemented and tested the algorithm detailed in Section (3). We applied the algorithm recursively to the columns and the rows of each color component. We assembled the spatial triplets from adjacent pixels and assembled the spectral-triplets in RGB order. We used a random binary sequence derived from a uniform noise as a watermark signal. We tested the algorithm with three 512x512 RGB images. These images are Lena, Baboon, and Fruits. We set $T_1 = T_2$ in all experiments. The Peak Signal to Noise Ratios (PSNR) of the watermarked images are listed in Table (1), along with the payload sizes we were able to embed into these images. The table indicates that the achievable embedding capacity depends on the nature of the image itself. Some images can bear more bits with lower distortion in the sense of PSNR than others. Images with a lot of low frequency contents like Lena and Fruits produce more expandable triplets with lower distortion than high frequency images such as Baboon, and, hence, can carry more watermark data at higher PSNR. However, Baboon can visually obscure the introduced artifacts much better. In general, a bit-rate as high as 3.5 bits/colored pixel with a quality level of 27 dB is achievable.

Colored Lena		**Baboon**		**Fruits**	
Payload (bits)	**PSNR (dB)**	**Payload (bits)**	**PSNR (dB)**	**Payload (bits)**	**PSNR (dB)**
305,194	35.80	115,050	30.14	299,302	35.36
420,956	34.28	187,248	28.54	497,034	33.00
516,364	33.12	256,334	27.20	582,758	32.45
660,618	31.44	320,070	26.10	737,066	31.14
755,096	30.28	408,840	24.73	824,760	30.06
837,768	29.10	505,150	23.34	853,846	29.49
941,420	27.01	656,456	21.20	888,850	28.52

Table 1. Embedded payload size vs. PSNR for colored images

	Gray-scale Lena		**Gray-scale Barbara**		
	Tian's Alg.	**Prop. Alg.**		**Tian's Alg.**	**Prop. Alg.**
PSNR (dB)	**Payload (bits)**	**Payload (bits)**	**PSNR (dB)**	**Payload (bits)**	**Payload (bits)**
29.4	260,018	**298,872**	23.6	247,629	**279,756**
32.5	222,042	**236,318**	31.2	159,000	**202,120**
34.8	175,984	**189,468**	32.8	138,621	**187,288**
36.2	**141,493**	131,588	34.1	120,997	**167,986**
37.7	**120,619**	107,416	37.4	81,219	**108,608**
40.1	**101,089**	49,588	40.2	**60,577**	45,500
41.6	**84,066**	19,108	42.8	**39,941**	19,384

Table 2. Comparison results between the proposed algorithm and Tian's using gray scale images.

We also compared the performance of the algorithm with that of Tian's described in [1] using gray scale images. The results are listed in Table (2), above, for Lena and Barbara images. The table indicates that, for the test images we used, our algorithm outperforms Tian's at lower PSNR, but Tian's algorithm outperforms ours at higher PSNR.

6. CONCLUSIONS

In this paper, a high capacity algorithm based on the difference expansion of triplets has been developed for embedding a reversible watermark with reasonable level of image distortion. Test results of the algorithm indicate that the amount of data one can embed into an image depends highly on the nature of the image. The algorithm has the potential of embedding a large amount of data at medium PSNR. Initial results indicate that the algorithm might be more suitable than Tian's algorithm for lower PSNR embedding. Further investigations are needed to fully determine the performance of the algorithm with various settings of the internal thresholds.

ACKNOWLEDGEMENTS

The author would like to thank Jun Tian, John Stach, Steve Decker, and Joel Meyer at Digimarc Corporation for their helpful discussion and feedback.

REFERENCES

[1] I. J. Cox, M. L. Miller, J. A. Bloom, *Digital Watermarking*, Morgan Kaufmann Publishers, San Francisco, CA, 2001.

[2] J. M. Barton, "Method and apparatus for embedding authentication information within digital data," *United States Patent*, 5,646,997, 1997.

[3] C. W. Honsinger, P. W. Jones, M. Rabbani, and J. C. Stoffel, "Lossless recovery of an original image containing embedded data," *United States Patent*, 6,278,791, 2001.

[4] B. Macq, "Lossless multiresolution transform for image authenticating watermark," *in Proceedings of EUSIPCO*, Tampere, Finland, Sept. 2000.

[5] M. Goljan, J. Fridrich, and R. Du, "Distortion-free data embedding for images," *in 4th Information Hiding Workshop*, Apr. 2001, pp. 27-41.

[6] J. Fridrich, M. Goljan, and R. Du, "Lossless data embedding – new paradigm in digital watermarking," *EURASIP Journal on Applied Signal Processing*, vol. 2002, no. 2, pp. 185-196, Feb 2002.

[7] C. De Vleeschouwer, J. F. Delaigle, and B. Macq, "Circular interpretation of histogram for reversible watermarking," *in Proceedings of IEEE 4th Workshop on Multimedia Signal Processing*, Oct. 2001.

[8] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Reversible data hiding," *in Proceedings of the IEEE International Conference on Image Processing*, Sept. 2002, vol. II, pp. 157-160.

[9] J. Tian, ``Reversible watermarking by difference expansion," in *Proceedings of Workshop on Multimedia and Security: Authentication, Secrecy, and Steganalysis*, J. Dittmann, J. Fridrich, and P. Wohlmacher, Eds., Dec. 2002, pp. 19-22.