

REVERSIBLE WATERMARK USING DIFFERENCE EXPANSION OF QUADS

Adnan M. Alattar

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

ABSTRACT

A reversible watermarking algorithm with very high data hiding capacity has been developed for colored images. The algorithm allows the watermarking process to be reversed to restore the original image exactly. The algorithm hides triplets of bits in the difference expansion of quads of adjacent pixels. The necessary difference expansion transform and its inverse is derived for quads. Also, the necessary conditions to avoid under- and overflow are derived. The algorithm can also be applied recursively, to maximize the amount of data that can be hidden into an image. Simulation results show that the algorithm can hide a bit-rate as high as 3.3 bits/colored pixel while maintaining an image quality level of 33.5 dB.

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 destroyed easily. 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 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]. Tian [8] used a difference expansion transform of a pair of pixels to embed a large amount of data into grayscale images. His algorithm allows one bit to be embedded in every pair of pixels. Alattar [7] derived a difference expansion transform for triplets and extended Tian's algorithm to embed two bits in every triplet of pixels. In this paper, we derive a difference expansion transform for quads and extend Tian's algorithm further to embed two bits in every

quad. In the next section, we define quads 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 quads. In Section 4, we present simulation results of the proposed algorithm. In Section 5 we present some extensions to the basic algorithm. And finally, in the last section, we summarize our conclusions.

2. DIFFERENCE EXPANSION OF QUADS

Quads: A quad is a 1×4 vector formed from four pixel values chosen from four 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 quads is to consider every 2×2 adjacent pixel values as a quad. For simplicity, we require that each color component is treated independently, and, hence, it has its own set of quads. Also, we require that quads do not overlap each other; i.e., each pixel exists in only one quad. 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 quads. Figure 1 shows this configuration.

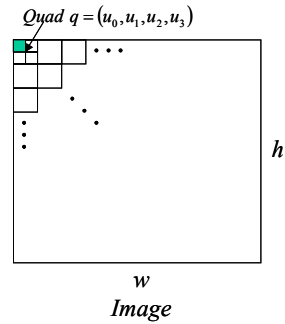


Figure 1. Quads Configuration in an Image

Difference Expansion Transform: The forward difference expansion transform, $f(\cdot)$, for the quad $q = (u_0, u_1, u_2, u_3)$ is defined as:

$$\begin{aligned} v_0 &= \left\lfloor \frac{a_0 u_0 + a_1 u_1 + a_2 u_2 + a_3 u_3}{a_0 + a_1 + a_2 + a_3} \right\rfloor \\ v_1 &= u_1 - u_0 \\ v_2 &= u_2 - u_1 \\ v_3 &= u_3 - u_2 \end{aligned} \quad (1)$$

where $\lfloor \cdot \rfloor$ is the least nearest integer.

The inverse difference expansion transform, $f^{-1}(\cdot)$, for the transformed quad $q' = (v_0, v_1, v_2, v_3)$ is defined as:

$$\begin{aligned}
u_0 &= v_0 - \\
&= \left\lfloor \frac{(a_1 + a_2 + a_3)v_1 + (a_2 + a_3)v_2 + a_3v_3}{a_0 + a_1 + a_2 + a_3} \right\rfloor \\
u_1 &= v_1 + u_0 \\
u_2 &= v_2 + u_1 \\
u_3 &= v_3 + u_2
\end{aligned} \tag{2}$$

Proof: To prove that equation (2) is the inverse of equation (1), one can substitute v_0, v_1, v_2 and v_3 from equation (1) into u_0 of equation (2). This gives

$$\begin{aligned}
u_0 &= \left\lfloor \frac{a_0u_0 + a_1u_1 + a_2u_2 + a_3u_3}{a_0 + a_1 + a_2 + a_3} \right\rfloor - \\
&\quad \left\lfloor \frac{(a_1 + a_2 + a_3)(u_1 - u_0) + (a_2 + a_3)(u_2 - u_1) + a_3(u_3 - u_2)}{a_0 + a_1 + a_2 + a_3} \right\rfloor \\
&= \left\lfloor \frac{a_0u_0 + a_1u_1 + a_2u_2 + a_3u_3}{a_0 + a_1 + a_2 + a_3} \right\rfloor - \\
&\quad \left\lfloor \frac{a_0u_0 + a_1u_1 + a_2u_2 + a_3u_3 - u_0}{a_0 + a_1 + a_2 + a_3} \right\rfloor \\
&= \left\lfloor \frac{a_0u_0 + a_1u_1 + a_2u_2 + a_3u_3}{a_0 + a_1 + a_2 + a_3} \right\rfloor - \\
&\quad \left\lfloor \frac{a_0u_0 + a_1u_1 + a_2u_2 + a_3u_3}{a_0 + a_1 + a_2 + a_3} \right\rfloor + u_0 = u_0
\end{aligned} \tag{3}$$

Now, the reversibility concerning u_1, u_2 , and u_3 can be proven by simple mathematical manipulation of v_1, v_2 , and v_3 in equation (1).

Definition 1: The quad $q = (u_0, u_1, u_2, u_3)$ is said to be expandable if for all values of b_1, b_2 , and $b_3 \in \{0,1\}$

$$\begin{aligned}
0 &\leq v_0 - \left\lfloor \frac{\tilde{v}_1 + \tilde{v}_2 + \tilde{v}_3}{a_0 + a_1 + a_2 + a_3} \right\rfloor \leq 255 \\
0 &\leq \tilde{v}_1 + u_0 \leq 255 \\
0 &\leq \tilde{v}_2 + u_1 \leq 255 \\
0 &\leq \tilde{v}_3 + u_2 \leq 255
\end{aligned} \tag{4}$$

Where:

$$\begin{aligned}
\tilde{v}_1 &= 2 \times v_1 + b_1 \\
\tilde{v}_2 &= 2 \times v_2 + b_2 \\
\tilde{v}_3 &= 2 \times v_3 + b_3
\end{aligned} \tag{5}$$

Notice that each of \tilde{v}_1, \tilde{v}_2 , and \tilde{v}_3 is a one-bit left-shifted version of the original value v_1, v_2 , and v_3 , respectively, but potentially with a different LSB (least significant bit). The conditions of equation (4), above, ensures that changing the LSBs of v_1, v_2 , and v_3 according to equation (5) does not introduce an overflow or underflow in the values of $\tilde{u}_0, \tilde{u}_1, \tilde{u}_2$, and \tilde{u}_3 when the inverse transform is computed.

Definition 2: The quad $q = (u_0, u_1, u_2, u_3)$ is said to be changeable if for all values of b_1, b_2 , and $b_3 \in \{0,1\}$ \tilde{v}_1, \tilde{v}_2 , and \tilde{v}_3 given by equation (6), below, satisfy equation (4).

$$\begin{aligned}
\tilde{v}_1 &= 2 \times \left\lfloor \frac{v_1}{2} \right\rfloor + b_1 \\
\tilde{v}_2 &= 2 \times \left\lfloor \frac{v_2}{2} \right\rfloor + b_2 \\
\tilde{v}_3 &= 2 \times \left\lfloor \frac{v_3}{2} \right\rfloor + b_3
\end{aligned} \tag{6}$$

Notice that \tilde{v}_1, \tilde{v}_2 , and \tilde{v}_3 in the above equation are the same as the original v_1, v_2 , and v_3 , but with different LSBs. Also, notice that a changeable quad remains changeable even after changing the LSBs of its v_1, v_2 , and v_3 . Also, from definitions 1 and 2, it can be observed that an expandable quad is also changeable.

3. ALGORITHM FOR REVERSIBLE WATERMARK

3.1. Embedding of Reversible Watermark

The embedding algorithm can be summarized using the following steps:

1. For every color component k , do the following:
 - a. Form the set of quads Q from the image $I(i, j, k)$ using the security key K .
 - b. Calculate $Q' = f(Q)$ using the difference expansion transform, $f(\cdot)$ (see equation (1)).
 - c. Use Q' and the conditions in equation (4) to divide Q into four sets S_1, S_2, S_3 , and S_4 . The set S_1 contains all expandable quads whose $v_1 \leq T_1, v_2 \leq T_2$, and $v_3 \leq T_3$, where T_1, T_2 , and T_3 are

- predefined thresholds. The set S_2 contains all changeable quads that are not in S_1 . The set S_3 contains the rest of the quads (not changeable). The set S_4 contains all changeable quads (i.e., $S_4 = S_1 \cup S_2$).
- d. Form the location map, M to identify the locations of the quads in S_1 , S_2 , and S_3 . The 1 symbol in M indicates the locations of S_1 quads and the 0 symbol indicates the locations of S_2 or S_3 quads. Then, compress M 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 , v_2 , and v_3 of each quad 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 quads of S_1 and S_2 as they occur in the image and through the bits of the bitstream B in their natural order. For S_1 , expand the quads as described in equation (5). For S_2 , expand the quads as in equation (6). The values of b_1 , b_2 , and b_3 are taken sequentially from the bitstream.
 - h. Calculate the inverse difference expansion transform of the resulting quads 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 quads in S_1^w and S_2^w to produce the watermarked image $I^w(i, j, k)$.

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 , T_2 , and T_3 must be properly set. Strategies for setting these thresholds are detailed in [8].

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 color component k , do the following:
 - a. Form the set of quads Q from the image $I^w(i, j, k)$ using the security key K .

- b. Calculate Q' using the difference expansion transform, $f(\cdot)$ (see equation (1)).
- c. Use Q' and the conditions in equation (4) to divide the quads in Q' into the two sets \hat{S}_4 and S_3 . \hat{S}_4 has the same quads as S_4 , which was constructed during embedding, but the values of the entities in each quad may be different. Similarly, S_3 is the same set constructed during embedding, since it contains non-changeable quads.
- d. Extract the LSBs of \tilde{v}_1 , \tilde{v}_2 , and \tilde{v}_3 of each quad 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 quads of the set S_1 (expandable quads). Collect these quads into set \hat{S}_1 .
- f. Identify the member quads of S_2 . They are the members of \hat{S}_4 that are not members of \hat{S}_1 . Form the set $\hat{S}_2 = \hat{S}_4 - \hat{S}_1$.
- g. Sequence through the member quads of \hat{S}_1 and \hat{S}_2 as they occur in the image and through the bits of the bitstream B in their natural order after discarding the bits of B_1 . For \hat{S}_1 , restore the original values of v_1 , v_2 , and v_3 as follows:
$$v_1 = \left\lfloor \frac{\tilde{v}_1}{2} \right\rfloor, \quad v_2 = \left\lfloor \frac{\tilde{v}_2}{2} \right\rfloor, \quad v_3 = \left\lfloor \frac{\tilde{v}_3}{2} \right\rfloor \quad (7)$$
- h. Calculate the inverse difference expansion transform of the resulting quads 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 quads in S_1 and S_2 to restore the original image $I(i, j, k)$.
- j. Discard all the bits in the bitstream 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 .

4. EXPERIMENTAL RESULTS

We implemented and tested the algorithm detailed in Section 3. We applied the algorithm to each color component. We assembled the quads from 2x2 adjacent pixels as shown in Figure 1. We used a random binary sequence derived from a uniformly distributed noise as a watermark signal. We tested the algorithm with three 512x512 RGB images. These images are *Lena*, *Baboon*, and *Fruits*. We set $T_1 = T_2 = T_3$ in all experiments. The payload size embedded into each of the test images (all color components) is plotted against the peak signal-to-noise ratios (PSNR) of the resulting watermarked image in Figure 2. The plot 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 and high correlation, like *Lena* and *Fruits*, produce more expandable triplets with lower distortion (in the PSNR sense) than high frequency images, such as *Baboon*, and, hence, can carry more watermark data at higher PSNR. With *Fruits*, the algorithm is able to embed 867 kbits with image quality of 33.59 dB. It is also able to embed 321 kbits with high image quality of 43.58 dB. Nevertheless, with *Baboon* the algorithm is able to embed 802 kbits at 24.73 dB and 148 kbits at 36.6 dB.

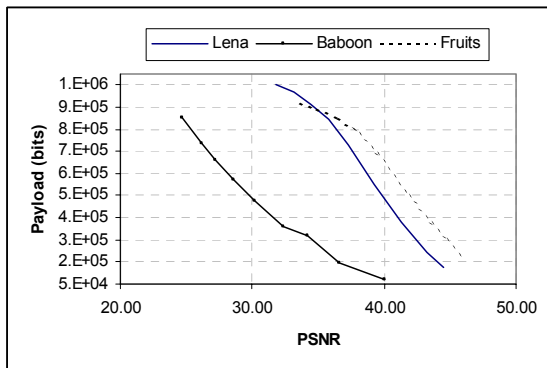


Figure 2. Embedded payload size vs. PSNR for colored images

We also compared the performance of the proposed algorithm with that of Tian's described in [8] using grayscale *Lena* and *Barbara* images. As expected, the results indicate that our quad-based algorithm outperforms Tian's at PSNR higher than 35dB, but Tian's algorithm marginally outperforms ours at lower PSNR.

We also compared the performance of the proposed quad-based algorithm with that of the spatial tripled-based algorithm described in [7]. The results reveal that the achievable payload size for the quad-based algorithm is about 300,000 bits higher than for the spatial triplets-based algorithm at the same PSNR, and the PSNR is about 5 dB higher for the quad-based algorithm than for the spatial triplet-based algorithm at the same payload size. Also, the quad-based algorithm has finer control over the payload size and the PSNR than the spatial triplet-based algorithm. For example, it was possible to produce images at PSNRs in the 38 dB to 46 dB range with quad-based algorithm, but not with spatial triplet-based algorithm. This result is because 2x2 spatial quads have higher

correlation than 1x3 spatial triplets and because the spatial quad-based algorithm uses a smaller location map.

We also compared our proposed algorithm with that of Celik [6] using grayscale *Lena* and *Barbara* images. The results indicate that our quad-based algorithm is superior to Celik's at almost all PSNRs.

5. CONCLUSIONS

In this paper, a high capacity algorithm based on the difference expansion of quads 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. The test results also indicate that the performance of the spatial quad-based algorithm is superior to that of the spatial triplet-based algorithm, Tian's algorithm, and Celik's algorithm. Further investigations are needed to fully determine the performance of the algorithm with various settings of the internal thresholds.

ACKNOWLEDGMENT

The author thanks John Stach and Kyle Smith at Digimarc Corporation for their assistance in this paper.

REFERENCES

- [1] I. J. Cox, M. L. Miller, J. A. Bloom, *Digital Watermarking*, Morgan Kaufmann Publishers, San Francisco, CA, 2001.
- [2] C. W. Honsinger, P. W. Jones, M. Rabbani, and J. C. Stoffel, "Lossless recovery of an original image containing embedded data," U.S. Patent 6,278,791, 2001.
- [3] B. Macq, "Lossless multiresolution transform for image authenticating watermark," in *Proc. of EUSIPCO*, Tampere, Finland, Sept. 2000.
- [4] C. De Vleeschouwer, J. F. Delaigle, and B. Macq, "Circular interpretation of histogram for reversible watermarking," in *Proc. of IEEE 4th Workshop on Multimedia Signal Processing*, Oct. 2001.
- [5] 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.
- [6] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Reversible data hiding," in *Proc. of the IEEE International Conference on Image Processing*, vol. II, Sept. 2002, pp. 157-160.
- [7] A. M. Alattar, "Reversible Watermark Using Difference Expansion of Triplets," *Proceedings of the 2003 IEEE International Conference on Image Processing, ICIP'2003*, Barcelona, Spain, September 14-17, 2003, pp. 501-504.
- [8] J. Tian, "Reversible watermarking by difference expansion," in *Proc. of Workshop on Multimedia and Security: Authentication, Secrecy, and Steganalysis*, J. Dittmann, J. Fridrich, and P. Wohlman, Eds., Dec. 2002, pp. 19-22.
- [9] T. Kalker and F.M. Willems, "Capacity bounds and code constructions for reversible data-hiding," in *Proc. of Electronic Imaging 2003, Security and Watermarking of Multimedia Contents V*, Santa Clara, California, Jan. 2003.