Reversible Watermarking by Difference Expansion

Jun Tian
Digimarc Corporation
19801 SW 72nd Avenue
Tualatin, OR 97062, USA
1-503-495-4691

juntian@ieee.org

ABSTRACT

Reversible watermark has drawn lots of interest recently. Different from other types of digital watermarks, a reversible watermark has a special feature that the original digital content can be completely restored. In this paper we describe a high capacity and high quality reversible watermarking method based on difference expansion. A noticeable difference between our method and others is that we do not need to compress original values of the embedding area. We explore the redundancy in the digital content to achieve reversibility.

Categories and Subject Descriptors

I.4.9 [**Image Processing and Computer Vision**]: Applications.

General Terms

Algorithms, Security, Theory.

Keywords

Reversible watermarking, authentication, digital watermarking, difference expansion, Digimarc.

1. INTRODUCTION

In digital watermarking, an invisible (in some cases, visible) watermark is embedded into a digital content for the purpose of copyright communication and protection, content authentication, counterfeit deterrence, forensic tracking, connected content, or broadcast monitoring, etc. Reversible watermarking [1, 2, 3, 4, 6, 7, 8], is a special digital watermark with an intriguing feature that when watermarked content has been authenticated, one can remove the watermark to retrieve the original, unwatermarked content. Such reversibility to get back original content is highly desirable in sensitive imagery, such as military data and medical data.

In this paper, we describe a high capacity, high quality, reversible watermarking method for digital images. Our method can be applied to digital audio and video as well. It is a simplified version of our early work reported in [7]. We calculate the differences of neighboring pixel values, and select some

difference numbers for difference expansion. The original values of difference numbers, the location of expanded difference numbers, and a payload will all be embedded into the difference numbers, where the extra storage space is obtained by difference expansion.

In this paper we will consider grayscale images only. For color images, there are several options. One can decorrelate the dependence among different color components, and then reversibly watermark the decorrelated components. Or one can reversibly watermark each color component individually.

2. REVERSIBLE WATERMARKING

In reversible watermarking, we embed a watermark in a digital image I, and obtain the watermarked image I'. Before sending it to the content authenticator, the image I' might or might not have been tampered by some intentional or unintentional attack. If the authenticator finds that no tampering happened in I', i.e., I' is authentic, then the authenticator can remove the watermark from I' to restore the original image, which results in a new image I''. By definition of reversible watermark, the restored image I'' will be **exactly** the same as the original image I, pixel by pixel, bit by bit.

A basic approach of reversible watermarking is to select an embedding area in an image, and embed both the payload and the original values in this area (needed for exact recovery of the original image) into such area. As the amount of information needed to be embedded (payload and original values in the embedding area) is larger than that of the embedding area, most reversible watermarking techniques [1, 2, 3, 6] rely on lossless data compression on the original values in the embedding area, and the space saved from compression will be used for embedding the payload. In [7] we presented a different technique, called difference expansion, which removes the need of lossless compression on original values in the embedding area. The difference expansion technique discovers extra storage space by exploring the high redundancy in the image content.

In our method, we will embed the payload in the difference of pixel values. For a pair of pixel values (x, y) in a grayscale image, $x, y \in \mathbb{Z}, 0 \le x, y \le 255$, define their (integer) average l and difference h as

$$l = \left\lfloor \frac{x + y}{2} \right\rfloor, h = x - y \qquad (1)$$

where the symbol $\lfloor \cdot \rfloor$ is the floor function meaning "the greatest integer less than or equal to". The inverse transform of (1) is

$$x = l + \left\lfloor \frac{h+1}{2} \right\rfloor, \ y = l - \left\lfloor \frac{h}{2} \right\rfloor \qquad (2)$$

As grayscale values are bounded in [0,255], we have

$$0 \le l + \left\lfloor \frac{h+1}{2} \right\rfloor \le 255, 0 \le l - \left\lfloor \frac{h}{2} \right\rfloor \le 255$$

which is equivalent to

$$|h| \le \min(2(255-l), 2l+1) \qquad (3)$$

Thus to prevent overflow and underflow problems, the difference number h (after embedding) must satisfy Condition (3).

The least significant bit (LSB) of the difference number h will be the selected embedding area. As

$$h = \left\lfloor \frac{h}{2} \right\rfloor \cdot 2 + LSB(h)$$

with $LSB(h) = 0$ or 1, to prevent any overflow and underflow problems, we embed only in *changeable* difference numbers.

Definition *For a grayscale-valued pair* (x, y), *we say its difference number h is* changeable *if*

$$\left| \left\lfloor \frac{h}{2} \right\rfloor \cdot 2 + b \right| \le \min(2(255-l), 2l+1)$$

for both b=0 and 1.

Modifying changeable h (without compression) does not provide additional storage space. We gain extra storage space from *expandable* difference numbers.

Definition *For a grayscale-valued pair* (x, y), *we say its difference number h is* expandable *if*

$$|2 \cdot h + b| \le \min(2(255-l), 2l+1)$$

for both b=0 and 1.

In the binary representation of integers, an expandable h could add one extra bit b after its LSB, with $b=0$ or 1. More precisely, h could be replaced by a new difference number $h'=2h+b$, without causing an overflow or underflow. Thus for each expandable difference number, one could gain one extra bit. The reversible operation from h to h' is called *difference expansion*. An expandable h is also changeable. After difference expansion, the expanded h' is still changeable.

Note that the conditions on changeable and expandable difference number are both weaker than those in [7]. As a result, in this paper, more difference numbers will be changeable and/or expandable. Also note that if $h=0$ or -1, the conditions on changeable and expandable are exactly the same.

For a digital image, it is partitioned into pairs of pixel values. A pair consists of two neighboring pixel values or two with a small difference number. The pairing could be done horizontally, vertically, or by a key-based specific pattern. The pairing could be

through all pixels of the image or just a portion of it. (In practice, we can embed a payload with one pairing, then on the embedded image, we embed another payload with another pairing, and so on.) We apply the integer transform (1) to each pair.

Then we create five disjoint sets of difference numbers, EZ, NZ, EN, CNE, and NC:

1. EZ: expandable zeros. For all expandable $h \in \{0, -1\}$

2. NZ: not expandable zeros. For all non-expandable $h \in \{0, -1\}$

3. EN: expandable nonzeros. For all expandable $h \notin \{0, -1\}$

4. CNE: changeable, but not expandable. For all changeable, but non-expandable $h \notin \{0, -1\}$

5. NC: not changeable. For all non-changeable $h \notin \{0, -1\}$

Each difference number will fall into one and only one set.

The next step is to create a location map of all expanded (after embedding) difference numbers. We partition the set EN into two disjoint subset EN1 and EN2. For every h in EN1, it will be expanded; for every h in EN2, it will not (though it is expandable). A discussion on how to select expandable $h \notin \{0, -1\}$ for difference expansion will be presented later. We create a one-bit bitmap as the location map, with its size equal to the numbers of pairs of pixel values. For the difference number in either EZ or EN1, we assign a value "1" in the location map; for the difference number in either NZ, EN2, CNE, or NC, we assign a value "0". Thus a value "1" will indicate an expanded difference number. The location map (a one-bit bitmap) will be lossless compressed by a JBIG2 compression [5] or run-length coding. The compressed bit stream will be denoted as **L**. An end of message symbol is appended at the end of **L**.

We collect original LSB values of difference numbers in EN2 and CNE. For each h in EN2 or CNE, $LSB(h)$ will be collected into a bit stream **C**. An exception is when $h=1$ or -2, nothing will be collected.

With the location map **L**, the original LSB values **C**, and a payload **P** (which includes an authentication hash, for example, an SHA-256 hash), we combine them together into one binary bit stream **B**

$$B = L \cup C \cup P$$

Assuming b is the next bit in **B**, depending on which set h belongs to, the embedding (by replacement) will be

- EZ or EN1: $h = 2 \cdot h + b$

- EN2 or CNE: $h = \left\lfloor \frac{h}{2} \right\rfloor \cdot 2 + b$

- NZ or NC: no change on the value of h, b is passed to the next h

Table 1. Payload Size and PSNR of Reversibly Watermarked "Lena".

Payload Size (bits)	39566	63676	84066	101089	120619	141493	175984	222042	260018
Bit Rate (bpp)	0.1509	0.2429	0.3207	0.3856	0.4601	0.5398	0.6713	0.8470	0.9919
PSNR (dB)	44.20	42.86	41.55	40.06	37.66	36.15	34.80	32.54	29.43

After all bits in $\textbf{\textit{B}}$ are embedded, we apply the inverse integer transform (2) to obtain the embedded image.

The bit stream $\textbf{\textit{B}}$ has a bit length of ($|\textbf{\textit{L}}|+|\textbf{\textit{C}}|+|\textbf{\textit{P}}|$). Assume the total number of 1 and –2 in EN2 and CNE is N, as each expanded pair will give one extra bit, the total hiding capacity will be ($|\textbf{\textit{C}}|+N+|EZ|+|EN1|$). Accordingly, to have $\textbf{\textit{B}}$ successfully embedded, we must have

$$|L|+|C|+|P| \leq |C|+N+|EZ|+|EN1| \quad (4)$$

i.e.,

$$|L|+|P| \leq N+|EZ|+|EN1| \quad (5)$$

Note that if the bit stream $\textbf{\textit{C}}$ is losslessly compressed before embedding, then Condition (4) becomes

$$|L|+\alpha|C|+|P| \leq |C|+N+|EZ|+|EN1|$$

where α is the achieved compression rate, $0 < \alpha \leq 1$.

The partition of expandable $h \notin \{0,-1\}$ into EN1 and EN2 will be subject to Condition (5). We present two partition methods here, one for mean square error (MSE) consideration, and the other for visual quality consideration.

Assume after difference expansion, an expanded pair (x, y) becomes (x', y'), with the average number unchanged, then

$$(x-x')^2+(y-y')^2 \approx 2(y-y')^2 =$$
$$2\left(\left\lfloor\frac{h}{2}\right\rfloor-\left\lfloor\frac{h'}{2}\right\rfloor\right)^2 = 2\left(\left\lfloor\frac{h}{2}\right\rfloor-\left\lfloor\frac{2\cdot h+b}{2}\right\rfloor\right)^2 \approx \frac{h^2}{2}$$

Thus to minimize the MSE, we should select h with small magnitudes for difference expansion. For example, one can pick a threshold T, and partition EN into EN1 and EN2 by checking whether the magnitude of h is less than or greater than T.

For the visual quality consideration, we can define a hiding ability of an expandable difference number, as follows.

Definition *For an expandable difference number h, if k is the largest integer such that*

$$|k \cdot h+b| \leq \min(2(255-l),2l+1)$$

for all $0 \leq b \leq k-1$*, then we say the* hiding ability *of h is* $\log_2 k$.

The hiding ability tells us how many bits could be embedded into the difference number h, without causing overflow or underflow. For an expandable h, it will be at least $\log_2 2 = 1$, as $k \geq 2$.

Although in this paper we are not going to embed more than one

bit into the difference number, the hiding ability could be used as a guide on selecting expandable difference numbers for difference expansion. In general, selecting an expandable difference number with large hiding ability will degrade less on the visual quality than an expandable difference number with small hiding ability. A large hiding ability implies that the average of two pixel values is close to mid tone, while their difference is close to zero.

For decoding, we do the pairing using the same pattern as in the embedding, and apply the integer transform (1) to each pair.

Next we create two disjoint sets of difference numbers, C, and NC:

1. C: changeable. For all changeable h

2. NC: not changeable. For all non-changeable h

Then we collect all LSBs of difference numbers in C and form a binary bit stream $\textbf{\textit{B}}$. From $\textbf{\textit{B}}$, we first decode the location map. With the location map, we restore the original values of difference numbers as follows (assuming b is the next bit in $\textbf{\textit{B}}$):

- if $h \in C$, the location map value is 1, then $h = \left\lfloor\frac{h}{2}\right\rfloor$, b is passed to the next h

- if $h \in C$, the location map value is 0, and $0 \leq h \leq 1$, then h=1, b is passed to the next h

- if $h \in C$, the location map value is 0, and $-2 \leq h \leq -1$, then h=-2, b is passed to the next h

- if $h \in C$, the location map value is 0, and $h \geq 2$ or $h \leq -3$, then $h = \left\lfloor\frac{h}{2}\right\rfloor \cdot 2 + b$

- if $h \in NC$, the location map value should be 0 (otherwise a decoding error on a tampered image), no change on h, b is passed to the next h

After all difference numbers have been restored, we apply the inverse integer transform (2) to reconstruct a restored image. If the embedded image has not been tampered, then the restored image will be identical to the original image. To authenticate the content of the embedded image, we extract the embedded payload $\textbf{\textit{P}}$ from $\textbf{\textit{B}}$ (which will be the remaining after restoring difference numbers). Compare the authentication hash in $\textbf{\textit{P}}$ with the hash of the restored image. If they match exactly, then the image content is authentic, and the restored image will be exactly the same as the original image. (Most likely a tampered image will not go through to this step because some decoding error could happen when restoring difference numbers.)

Figure 1. Original "Lena" Image

Figure 2. Reversibly Watermarked, with a 101089 Bits Payload, PSNR = 40.06 dB

3. EXPERIMENTAL RESULTS

The original, unwatermarked "Lena" image, which is 512 by 512, 8 bits per pixel (bpp), grayscale, is shown in Figure 1. We embed one payload with a vertical pairing; then on the embedded image, we embed another payload with a horizontal pairing. The decoding will also be two parts, first decode the horizontally embedded payload, then decode the vertically embedded payload. The final restored image will be identical to the original, unwatermarked image, pixel by pixel, bit by bit.

The embedded payload size (sum of two payloads' sizes), its corresponding bit rate, and PSNR of the watermarked image are listed in Table 1. We include the watermarked image with a payload of 101089 bits in Figure 2. As difference expansion increases the magnitudes of difference numbers, the watermark has the effect similar to mild sharpening in the mid tone regions.

4. CONCLUSIONS

In this paper we describe a high capacity, high quality, reversible watermarking method. We partition an image into pairs of pixel values, select expandable difference numbers for difference expansion and embed a payload which includes an authentication hash. By exploring the redundancy in the image, reversibility is achieved. As difference expansion brings extra storage space, compression is not necessary for our method. Of course employing compression can either increase the hiding capacity or reduce the visual quality degradation of watermarked images.

5. ACKNOWLEDGMENTS

6. REFERENCES

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

[2] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber. Reversible data hiding. In *Proc. of International Conference on Image Processing*, volume II, pages 157-160, Sept. 2002.

[3] J. Fridrich, M. Goljan, and R. Du. Lossless data embedding - new paradigm in digital watermarking. *EURASIP Journal on Applied Signal Processing*, 2002(2):185-196, Feb. 2002.

[4] 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.

[5] P. G. Howard, F. Kossentini, B. Martins, S. Forchhammer, and W. J. Rucklidge. The emerging JBIG2 standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(7):838-848, 1998.

[6] T. Kalker and F. M. J. Willems. Capacity bounds and constructions for reversible data hiding. In *Proc. of the 14th International Conference on Digital Signal Processing*, volume 1, pages 71-76, July 2002.

[7] J. Tian. Wavelet-based reversible watermarking for authentication. In E. J. Delp III and P. W. Wong, editors, *Security and Watermarking of Multimedia Contents IV*, volume 4675 of Proc. of SPIE, pages 679-690, Jan. 2002.

[8] C. D. Vleeschouwer, J. F. Delaigle, and B. Marq. Circular interpretation of histogram for reversible watermarking. In *Proc. of IEEE 4th Workshop on Multimedia Signal Processing*, Oct. 2001.