Lossless Authentication Watermarking Based on Adaptive Modular Arithmetic

Hengfu YANG\textsuperscript{1,2}, Xingming SUN\textsuperscript{1}, Guang SUN\textsuperscript{1,3}, Zuwei TIAN\textsuperscript{1,2}

\textsuperscript{1} School of Computer and Communication, Hunan University, No. 252, Lushan South Road, Changsha, 410082, China
\textsuperscript{2} Department of Information Science & Engineering, Hunan First Normal University, No. 1015, Fenglin 3rd Road, Changsha, 410205, China
\textsuperscript{3} Information Management Department, Hunan Financial and Economic College, No. 139, Fenglin 2nd Road, Changsha, 410205, China

sunnudt@163.com, hengfuyang@hotmail.com, hengfuyang@163.com

Abstract. Reversible watermarking schemes based on modulo-256 addition may cause annoying salt-and-pepper noise. To avoid the salt-and-pepper noise, a reversible watermarking scheme using human visual perception characteristics and adaptive modular arithmetic is proposed. First, a high-bit residual image is obtained by extracting the most significant bits (MSB) of the original image, and a new spatial visual perception model is built according to the high-bit residual image features. Second, the watermark strength and the adaptive divisor of modulo operation for each pixel are determined by the visual perception model. Finally, the watermark is embedded into different least significant bits (LSB) of original image with adaptive modulo addition. The original image can be losslessly recovered if the stego-image has not been altered. Extensive experiments show that the proposed algorithm eliminates the salt-and-pepper noise effectively, and the visual quality of the stego-image with the proposed algorithm has been dramatically improved over some existing reversible watermarking algorithms. Especially, the stego-image of this algorithm has 11.2942 dB higher PSNR value than that of modulo-256 addition based reversible watermarking scheme, on average.

Keywords

1. Introduction

Common irreversible watermarking techniques\textsuperscript{[1-4]} introduce some amount of distortion to the original image and the distortion is permanent and not reversible. There are, however, some applications (such as medical, astronomical, and military image systems) where any distortion introduced to the image is not acceptable\textsuperscript{[5]}. In these above fields, it is expected to reverse the marked media back to the original cover media after the hidden data are retrieved for some legal or other considerations\textsuperscript{[5-7]}. In order to avoid permanent distortion, a new branch of the watermarking technique, called reversible, distortion-free, or lossless watermarking, has been developed. In fact, Barton’s patent, filed in 1994, may be the earliest one\textsuperscript{[8]}. Recently, there are a number of lossless data hiding algorithms that have been reported in the literature, and attention on lossless data hiding scheme are increasing\textsuperscript{[8-22]}. According to embedding strategies, reversible data hiding can be classified into four types. The first type of algorithm losslessly compressed selected features from the original image to obtain enough space, which were then replaced with the watermark payload\textsuperscript{[9-11]}. But the capacity depended on the adopted image compress algorithm. The second type of reversible scheme used difference expansion to embed information\textsuperscript{[12-14]}. The main advantage was high embedding capacity, but this type of algorithm was lack of capacity control due to embedding of a location map which contained the location information of all selected expandable difference values. The third types of algorithms are lossless data hiding techniques based on histogram modification\textsuperscript{[15-17]}. They used image histogram to hide message bits and achieve reversibility. This type of algorithm had low computational cost because that there were no need any transform for data embedding, all processing is performed in spatial domain in most histogram-based methods, but the embedding capacity of this type is low. The fourth type of algorithm is the method based on modular arithmetic\textsuperscript{[18-22]}. This type of reversible watermarking algorithms used modulo-256 addition to embed the watermark into the original watermark. The type of algorithm had low computational consumption, but it cased annoying salt-and-pepper noise in the stego-image.

This paper focused on lossless data hiding method based on modular arithmetic. Honsinger et al.\textsuperscript{[18]} presented a lossless fragile watermarking technique for image authentication by applying modulo-256 addition. But this method may introduce visual artifacts, similar to salt-and-pepper noise, into the watermark image when pixels with grayscales close to zero are flipped to values close to 255 and vice versa. Macq\textsuperscript{[19]} described a modification to the
patchwork algorithm [23] to achieve lossless watermark embedding which uses addition modulo-256. However, this method also suffers from annoying salt-and-pepper noise. De Vleeschouwer et al. [20] proposed a lossless data embedding algorithm based on patchwork theory [23], which eliminated salt-and-pepper noise by adopting circle interpretation of bijective transformations, but the embedding capacity of the scheme is low. Similarly, S. Weng et al. [19] utilized symmetric modulo operation to avoid salt-and-pepper artifacts, but this method has embedded capacity because the cover image may have some useless blocks which cannot be used to embed the watermark. In a word, the watermarking scheme with modulo operations has two main defects: perceptible artifacts in the form of salt-and-pepper noise are easily introduced into the watermarked image and the degradation of watermark detection especially on images whose histograms are relatively spread from the values from 0 to 255 [22].

In order to avoid salt-and-pepper visual artifacts caused by modulo-256 addition based lossless watermarking algorithms and to provide good perceptual transparency, we exploit visual perception characteristics and adaptive modular arithmetic to achieve a reversible watermarking scheme (called the AMA method hereafter) in this paper.

2. Visual perception model and the adaptive divisor

It is known that the robustness and the imperceptibility of the watermark are contradictory to each other. The best method to achieve the tradeoff between the aforementioned two requirements is to take the human visual perception into account while embedding the watermark [23,27]. To eliminate the salt-and-pepper noise caused by the modular arithmetic-based reversible watermarking, in our scheme we only use adaptive least significant bits for each pixel to apply modular arithmetic (called adaptive modular arithmetic). We make use of the image features, including the luminance masking, texture masking and edge masking, to create a spatial visual perception model, and then use the model to deduce an adaptive divisor for reversible watermarking based on adaptive modular arithmetic.

As to luminance masking, human eyes usually have different sensitivity to different luminance, specifically, more sensitivity to noise in the areas with middle level luminance, and less sensitivity to noise in those areas with high and low brightness. As a result, the calculation of luminance masking can be formulated as

\[
\alpha(i,j) = \left\lfloor \frac{x(i,j) - 255}{2} \right\rfloor + \left\lfloor x(i,j) - 127.5 \right\rfloor,
\]

where \(x(i,j)\) is the pixel value at the spatial position \((i,j)\) in the host image \(I\).

On the one hand, human eyes are more sensitive to the noise in smooth image areas, but less sensitive to that in texture image areas. On the other hand, the smooth image regions have small entropy value, while the texture image areas have big entropy value. Hence, the texture masking we proposed here can use the entropy value of the sliding window as the texture masking (All the calculations are based on the image block of size \((2l+1)\times(2l+1)\), therefore in the implementation, we will use a sliding window of size \((2l+1)\times(2l+1)\), where \(1\leq l\leq 4\)). Larger entropy value corresponds to the texture or edge image region, while smaller entropy value corresponds to the smooth image region. Let \(H(i,j)\) be the entropy of subblock centered by the pixel \(I(i,j)\). The entropy \(H(i,j)\) can therefore be used to depict the texture characteristics of the pixel \(I(i,j)\). This maximum entropy is achievable when all of the gray-levels have the same probability. In other words, an image block receives its maximum entropy when it contains the same number of all of the gray values in that block. For a 256-level image block with size of \((2l+1)\times(2l+1)\), the maximum entropy \(H_{\text{max}}\) is evaluated by

\[
H_{\text{max}} = -\sum_{i=1}^{2l} \sum_{j=1}^{2l} \frac{1}{(2l+1)^2} \log_2 \left( \frac{1}{(2l+1)^2} \right) = 2\log_2 (2l+1).
\]

So, the normalized entropy \(\beta(i,j)\) can be obtained by the following formula

\[
\beta(i,j) = \frac{H(i,j)}{H_{\text{max}}}.
\]

As for edge masking, human eyes are very sensitive to the information distortion in the edge image area. Hence the watermark embedding must not lead to significant distortion in that image area. Image area with prominent edges has larger variance value, while smooth image area has smaller variance value. We can therefore use the variance of the image blocks to indicate the edge feature. Using the monotonic logarithm function for range compression we achieve the following expression for the edge masking.

\[
V'(i,j) = \log_{10} (V(i,j)),
\]

where \(V(i,j)\) is the variance (mean square error) of the image block centered by the pixel \(I(i,j)\). The maximum variance is in the block with a checkerboard pattern with the adjacent pixels having the maximum and
minimum permissible gray value. The maximum variance \( V_{\text{max}} \) is defined as
\[
V_{\text{max}} = \log_{10} \left( \frac{2^{2^2 + 2^2} - 2^{2^2 + 2^2 + 1}}{(2^2 + 1)^2} G^2 \right)
\]

where \( G \) is the maximum permissible gray value. Now, the normalized variance \( \gamma(i,j) \) as follows
\[
\gamma(i,j) = \frac{V'(i,j)}{V_{\text{max}}} = \frac{V'(i,j)}{\log_{10}(G/2)^2}.
\]

A large entropy value corresponds to the texture or edge image area, and the texture masking created by the entropy value also includes the image edge parts. In order to prevent the watermark embedding from corrupting the edge easily and causing severe distortion to the host image, we must ensure low watermark strength to be embedded in the image edge areas. Based on all the above considerations, the effect of visual perception characteristics is estimated by the formula
\[
\tau(i,j) = \alpha(i,j) \times \left( \beta(i,j) - \gamma(i,j) \right).
\]

At last, in order to improve the watermark invisibility and to enhance the controllability of the strength of the embedded watermark, we obtained the final visual perception factor (used as watermark strength) as follows by normalizing the visual perception effect \( \tau(i,j) \).
\[
\omega(i,j) = \text{round} \left( \left( 2^{(n-r)} - 1 \right) \times \tau(i,j) - \min(\tau) \right) / \max(\tau) + 1 \right). \tag{8}
\]

where \( 1 \leq r \leq 7 \) and \( r \) represents the MSBs number of each pixel used to calculate the visual perception masking. Function \( \text{round}(x) \) returns the nearest integer to the argument \( x \), and \( \max(x) \) and \( \min(x) \) returns the maximum and minimum of the array \( x \) respectively. The factor \( \omega(i,j) \) is the final visual perception factor of the pixel \( I(i,j) \), and \( 1 \leq \omega(i,j) \leq 2^{(n-r)} - 1 \).

After final perception factor has been obtained, the adaptive LSBs number \( \lambda(i,j) \) in pixel \( I(i,j) \) for modular addition operation can be derived as
\[
\lambda(i,j) = \left\lfloor \log_2^{\omega(i,j)} \right\rfloor + 1. \tag{9}
\]

where \( \lfloor x \rfloor \) denotes the floor function meaning “the greatest integer less than or equal to the real number \( x \)”.

The adaptive LSBs number \( \lambda(i,j) \) satisfies \( 1 \leq \lambda(i,j) \leq 8 - r \) because of Eq. (8).

At last, the adaptive divisor \( \phi(i,j) \) of pixel \( I(i,j) \) for modular arithmetic during data embedding can be computed by
\[
\phi(i,j) = 2^{\lambda(i,j)} \tag{10}
\]

Taking Peppers, Lena, Bone, and Baboon images for example, Fig.1 illustrates the magnified visual perception factor. The darker region is considered visually less sensitive to noise and has relatively lower masking values. Meanwhile, the white part in the masking image is more sensitive to distortion and has higher masking values. From Fig.1, we see that the highly-textured image regions or regions with higher and lower brightness have bigger masking values, while the smooth image areas and the prominent edge areas have lower values, indicating the proposed visual perception model can well depict the visual perception characteristics of the host images.
3. Lossless watermarking scheme based on adaptive modular arithmetic

3.1 Lossless data embedding

To eliminate the salt-and-pepper noise artifacts, we use adaptive LSBs of each pixel (not all the bits) for watermark embedding. Reversibility is achieved by adaptive modulo addition. The watermark embedding procedure is illustrated in Fig.2 and the detailed embedding algorithm is described as follows.

**Input:** Original image $I$ of size $m \times n$, secret key $k$, and MSBs number $r$ for visual perception calculation.

**Output:** Watermarked image $I'$ of size $m \times n$.

**Step 1:** Extract $r$ (say 4) MSBs of the original image $I$ to obtain its corresponding high-bit residual image $I_r$.

**Step 2:** Compute the final visual perception factor $\omega(i,j)$ from the high-bit residual image $I_r$ using Eq. (8), and then use Eq. (9) to derive the adaptive LSBs number $\lambda(i,j)$ in pixel $I(i,j)$ for modular addition operation.

**Step 3:** Obtain the adaptive divisor $\phi(i,j)$ of pixel $I(i,j)$ for modular arithmetic by Eq. (10).
Step 4: Produce the LSB image $I_l$ and the MSB image $I_h$ of original image by extracting $\lambda(i,j)$ LSBs and $8-\lambda(i,j)$ MSBs of the pixel $I(i,j)$ respectively.

Step 5: Generate a pseudo random sequence $PN$ with number 1 and -1 by secret key $k$ as the watermark sequence $W$.

\[ W = \{ w(i,j) | w(i,j) \in \{-1, 1\}, 0 \leq i < m, 0 \leq j < n \}. \] (11)

Step 6: Embed the watermark into the LSB image $I_l'$ by adaptive modular addition operation to generate the watermarked LSB image $I_l''$,

\[ I_l''(i,j) = (I_l'(i,j) + \omega(i,j) \times w(i,j)) \bmod \phi(i,j). \] (12)

Step 7: Obtain the watermarked image $I'$ by adding the MSB image $I_h'$ to the LSB image $I_l''$.

\[ I' = I_h' + I_l''. \] (13)

3.2 Recovery of the original image and watermark detection

The procedure of recovering the original image is the inverse operation of the embedding process, and can be presented below.

Input: Watermarked image $I'$, and secret key $k$, and MSBs number $r$ for visual perception calculation.

Output: Original image $I$.

Step 1: Extract $r$ MSBs from the watermarked image $I'$ to obtain the high-bit residual image $I_r'$, note that $I_r'$ is the same as $I_l'$ because that the watermark is not embedded into the $r$ MSBs of each pixel.

Step 2: By using the similar method in the Sep 2 and Step 3 of the watermark insertion process, we can obtain the final visual perception factor $\omega(i,j)$, the adaptive LSBs number $\lambda(i,j)$, and the adaptive advisor $\phi(i,j)$ of pixel $I'(i,j)$ for modular arithmetic according to the high-bit residual image $I_r'$.

Step 3: Generate the LSB image $I_l'$ and the MSB image $I_h'$ of original image by extracting $\lambda(i,j)$ LSBs and $8-\lambda(i,j)$ MSBs of the pixel $I'(i,j)$ respectively.

Step 4: Use the shared secret key $k$ to generate a pseudo random sequence $PN$ with number 1 and -1 as the watermark signal $W$.

Step 5: Obtain the original LSB image $I_l$ by the following formula.

\[ I_l(i,j) = (I_l'(i,j) - \omega(i,j) \times w(i,j)) \bmod \phi(i,j). \] (14)

Step 6: Recover the original image $I$ by the original LSB image $I_l$ to the MSB image $I_h$. $I_h'$ is the same as $I_h$ because that the watermark is not embedded into the MSBs image $I_h$.

\[ I = I_h' + I_l = I_h + I_l. \] (15)

Because the watermark is embedded into the LSB image $I_l'$ instead of the MSB image $I_h'$ as long as the watermarked image is not illegally altered, the final visual perception factor and the adaptive advisor calculated in the image recovery phrase are the same as the ones obtained in the watermark embedding process. Therefore, the original image can be perfectly recovered without distortion.

To detect the presence of the watermark, we adopt the similar watermark detection method in ref. [22]. The detector can be written as

\[ \Lambda = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} (I_l'(i,j) - I_l(i,j)) \times w(i,j). \] (16)

where

\[ T = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} (I_l'(i,j)). \]

The reader is referred to ref. [22] for more details.

4. Experimental results and performance analysis

In this section, the proposed reversible watermarking algorithm has been applied to many different types of images with different characteristics. We choose some 512×512 8-bit gray-scale images shown in Fig.3 as the test sample. The watermark signal is produced by a pseudo random sequence initiated by the secret key $k$. In all experiments, we set the MSBs number for visual perception calculation $r=4$, and the slide window size in Eq. (2) $l=2$.

(a) Bone

(b) Chest
4.1 Transparency

During modulo-256 addition, a very bright pixel with a large gray value (close to 256) will possibly be changed to a very dark pixel with a small gray value (close to 0), and vice versa. As a result, the marked images generated by the modulo-256 addition based lossless watermarking algorithms usually suffer from salt-and-pepper noise. The salt-and-pepper noise becomes severe for those images that contain a number of dark and bright pixels (e.g., medical images as shown in Fig.4 e and g). Fig.4 shows the test results of lossless data hiding scheme based on modulo-256 addition (called the M2A method) [18] with watermark strength of 10 and embedded bit rate of 1 bpp (bit per pixel). From Fig.4, it is easy to notice that the annoying salt-and-pepper noise which heavily influences the visual quality of the marked images. Especially, the peak signal to noise ratios (PSNR) of the marked image shown in Fig.4a and Fig.4b are as low as 6.9803 dB and 18.9180 dB respectively.

Fig. 3. Some test image examples.

Fig. 4. Marked images using the M2A method.

Fig.5 shows the experimental results of adopting the AMA method for 1 bpp payload size. There are no visible perceptible artifacts in Fig.5, indicating that a significant performance improvement has been achieved as compared with ref. [18]. This is because our lossless watermarking approach uses adaptive modular arithmetic rather than modulo-256 addition.
Fig. 5. Marked images using the proposed method.

In particular, the effectiveness of the final visual perception factor can be illustrated by difference images between the original images and its corresponding watermarked ones (generated by the presented algorithm) with luminance enhancement by 15 times as shown in Fig. 6. It is clear that the watermark is mainly embedded into highly activated image regions, indicating that the watermark embedding is adaptive to the original image characteristics. This may also be attributed to the full use of the visual perception characteristics.

Fig. 6. Difference images with luminance enhancement by 15 times

4.2 Performance on capacity versus distortion

In our experiments, the peak signal noise ratio (PSNR) is used to evaluate the visual quality of the watermarked image, and the bit per pixel (bpp) is used to judge the
hiding capacity. The PSNR values of embedded images for different types of images with different payload size are shown in Fig. 7. We can easily see that the PSNR value can reach as high as at least 33 dB for 1bpp embedded rate. The proposed algorithm has the best performance for Lena image, which it is able to hide 26214 bits (0.1 bpp) with the watermarked image quality of 52.2426 dB. The results demonstrate that the proposed algorithm has high embedding capacity while maintaining good invisibility for different types of images.

![Image of Fig. 7](image.png)

**Fig. 7.** Hiding payload size versus distortion performance using the proposed algorithm for different types of images

To objectively judge the performance of the proposed algorithm, we can deduce the expectation of the PSNR value of stego-image generated by the proposed algorithm.

Given the MSBs number \( r \) for visual perception calculation, the adaptive LSBs number \( \lambda(i,j) \) in pixel \( I(i,j) \) for modular addition operation is considered to obey uniform distribution over range \([1, 8-r] \) for an ideal natural image. So, the expectation of the adaptive LSBs number \( \lambda(i,j) \) can be calculated by the formula,

\[
E(\lambda(i,j)) = \frac{1 + 8 - r}{2} = \frac{9 - r}{2}.
\]

Let \( \delta(i,j) = x'(i,j) - x(i,j) \) be the hidden error between \( x'(i,j) \) and \( x(i,j) \). When \( E(\lambda(i,j)) \) LSBs in each pixel \( I(i,j) \) are used to data hiding, \( \delta(i,j) \) can be also considered to obey uniform distribution over range \([-2^{E(\lambda(i,j))} + 1, 2^{E(\lambda(i,j))} - 1] \) if the watermark signal obeys uniform distribution. So the expected mean square error (MSE) caused by the presented scheme can be computed as follows,

\[
E(MSE) = E(\delta^2(i,j))
= D(\delta(i,j)) - (E(\delta(i,j)))^2
= D(\delta(i,j))
= \left(\frac{2^{E(\lambda(i,j))} - 1 - \left(2^{E(\lambda(i,j))} + 1\right)}{12}\right)^2
= \left(\frac{2^{E(\lambda(i,j))} - 1}{3}\right)^2
\]

Finally, given the embedded rate \( \eta \) (bpp), the expectation of PSNR value obtained by the proposed scheme can be get by the following formula,

\[
PSNR = 10 \log_{10} \left(\frac{255^2}{\eta \times MSE}\right).
\]

where \( 0 < \eta \leq 1 \).

Tab. 2 list the expected PSNR value versus hiding capacity (embedded rate) using the proposed algorithm with different MSBs number \( r \) for visual perception calculation. It can be seen that the proposed algorithm can achieve satisfactory expected PSNR value for different hiding capacity with different MSBs number \( r \). For
example, for \( r =4 \), it can obtain the stego-image quality of 46.0001 dB with 26214 bits (0.1 bpp), and 39.5402 dB with 262144 bits (1 bpp).

<table>
<thead>
<tr>
<th>( r )</th>
<th>0.1 bpp</th>
<th>0.2 bpp</th>
<th>0.3 bpp</th>
<th>0.4 bpp</th>
<th>0.5 bpp</th>
<th>0.6 bpp</th>
<th>0.7 bpp</th>
<th>0.8 bpp</th>
<th>0.9 bpp</th>
<th>1 bpp</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>39.3802</td>
<td>36.3699</td>
<td>34.6000</td>
<td>33.3190</td>
<td>32.3905</td>
<td>31.5987</td>
<td>30.9292</td>
<td>30.3493</td>
<td>29.8378</td>
<td>29.3802</td>
</tr>
<tr>
<td>2</td>
<td>42.6337</td>
<td>39.6234</td>
<td>37.8625</td>
<td>36.6131</td>
<td>35.6440</td>
<td>34.8227</td>
<td>34.1827</td>
<td>33.6028</td>
<td>33.0913</td>
<td>32.6337</td>
</tr>
<tr>
<td>3</td>
<td>46.0001</td>
<td>42.9898</td>
<td>41.2288</td>
<td>39.9795</td>
<td>39.0104</td>
<td>38.2185</td>
<td>37.5491</td>
<td>36.9692</td>
<td>36.4576</td>
<td>36.0001</td>
</tr>
<tr>
<td>4</td>
<td>49.5402</td>
<td>46.5299</td>
<td>44.7690</td>
<td>43.5196</td>
<td>42.5505</td>
<td>41.7587</td>
<td>41.0892</td>
<td>40.5093</td>
<td>39.9977</td>
<td>39.5402</td>
</tr>
<tr>
<td>5</td>
<td>53.3596</td>
<td>50.3493</td>
<td>48.5884</td>
<td>47.3390</td>
<td>46.3699</td>
<td>45.5781</td>
<td>44.8086</td>
<td>44.1287</td>
<td>43.5178</td>
<td>43.0596</td>
</tr>
<tr>
<td>6</td>
<td>57.6605</td>
<td>54.6502</td>
<td>52.8893</td>
<td>51.6399</td>
<td>50.6708</td>
<td>49.8790</td>
<td>49.2095</td>
<td>48.6296</td>
<td>48.1180</td>
<td>47.6605</td>
</tr>
<tr>
<td>7</td>
<td>62.9020</td>
<td>59.8917</td>
<td>58.1308</td>
<td>56.8814</td>
<td>55.9123</td>
<td>55.1205</td>
<td>54.4510</td>
<td>53.7811</td>
<td>53.3596</td>
<td>52.9020</td>
</tr>
</tbody>
</table>

Tab. 2. Expected PSNR value versus hiding capacity using the proposed algorithm with different MSBs number \( r \)

### 4.3 Capacity versus distortion performance comparisons

We also compared the proposed AMA method with ref. [9] (called GLSB), [13] (called EET), [17] (called SCSS), and [18] (called M2A) in terms of the payload capacity versus the stego-image quality. The comparison experimental results of embedding capacity versus invisibility among the proposed watermarking method and existing reversible marking methods on Lena image is shown in Fig. 8. By embedding the same size payload, the watermarked Lena image by our AMA method has better watermark transparency than the other algorithms [9, 13, 17 and 18] do. Especially, the PSNR value achieved by the presented algorithm is about 9.9864 to 14.1117 dB higher than that by the modulo-256 addition based method [18]. On average, it can achieve 11.2942 dB higher PSNR value than the modulo-256 addition based method.

![Fig. 8. Capacity versus distortion comparison among different methods](image)

### 5. Conclusion

By exploiting the luminance masking, texture masking and edge masking based on the image features, a novel spatial visual perception model is created, and then an adaptive divisor for modular arithmetic is deduced on the basis of the model. At last, a novel reversible watermarking scheme is presented by using visual perception model and adaptive modular arithmetic. The proposed algorithm is able to eliminate the salt-and-pepper visual artifacts in the watermarked images effectively, and achieves better transparency in contrast with some existing reversible watermarking algorithms. Especially, the proposed algorithm can achieve better stego-image visual quality than modulo-256 addition based reversible watermarking scheme. It can obtain 11.2942 dB higher...
PSNR value than modulo-256 addition based reversible watermarking schemes, on average.

Acknowledgements

This work was supported by National Natural Science Foundation of China (60736016, 60873198, 60973128, and 60973113), Scientific Research Fund of Hunan Provincial Education Department of China (08C018) and National Basic Research Program of China (2006CB303000, 2009CB326202).

References


About Authors ...

Hengfu YANG was born in Hunan, China, 1974. He received the M.S. degree in Computer Application from Guizhou University, China, in 2003. He is currently
pursuing a Ph.D. degree in Computer Application at School of Computer and Communication from Hunan University, China. His research interests include Multimedia, Image Processing, Digital Watermarking, and Information Hiding.

**Xingming SUN** was born in Hunan, China, 1963. He received the B.S. degree in Mathematics from Hunan Normal University, China, in 1984, the M.S. degree in Computing Science from Dalian University of Science and Technology, China, in 1988, and the Ph. D. degree in Computing Science from Fudan University, China, in 2001. He is currently a Professor in School of Computer and Communication, Hunan University, China. His research interests include Network & Information Security, Digital Watermarking, Wireless Sensor Network Security, and Natural Language Processing.

**Guang SUN** was born in Shandong, China, 1973. He received the M.S. degree in Software Engineering, from Hunan University, China, in 2005. He is currently a Ph.D. candidate in the School of Computer and Communication at Hunan University, China. His research interests include Network & Information Security, Software Watermarking, Software Birthmarking.

**Zuwei TIAN** was born in Hunan, China, 1973. He received the M.S. degree in Computer Software and Theory from National University of Defense Technology, China, in 2004. He is currently pursuing a Ph.D. degree in Computer Application at School of Computer and Communication from Hunan University, China. His research interests include compilers principles, Information Security, Software Watermarking.