A Digital Watermark Algorithm for QR Code

1Li Li, 2Rui-Ling Wang, 3Chin-Chen Chang
1College of Computer Science, Hangzhou Dianzi University, Hangzhou Zhejiang 310018, China, Lili2008@hdu.edu.cn
2College of Computer Science, Hangzhou Dianzi University, Hangzhou Zhejiang 310018, China, wrlingsx@126.com
3Department of Information Engineering and Computer Science, Feng Chia University, Taichung 40724, Taiwan, ccc@cs.ccu.edu.tw
doi: 10.4156/ijiip.vol2.issue2.4

Abstract

Technology that combines a 2D BarCode with a digital watermark is a topic of great interest in current research related to the security field. This paper presents a new digital watermark method for the QR Code (Quick Response Code). An invisible watermark is embedded in the QR Code image using watermark technology, and while this is being done, the DCT intermediate frequency coefficients are compared. To prevent the overflow of the QR Code in the DCT domain of the image, QR image need fuzzy processing and be added noise to. In order to resist image distortion after print and scan operations, the watermark is embedded repeatedly. The watermark is extracted by using the two maximum membership degree of fuzzy pattern recognition, without the original image. Experimental results have demonstrated the feasibility of the algorithm.

Key words: 2D BarCode, Digital Watermark, Fuzzy, Add Noise

1. Introduction

In recent years, information exchange and the transmission of digital images have become more and more convenient, with the rapid development of the Internet and digital storage technology. However, these technologies also make it possible for unscrupulous people to duplicate and distribute unauthorized images with low cost, especially images that have high commercial value, such as satellite image of the ground. To relieve tension and pressure from illegal trade in all industries, some experts have directed their attention to illegal trade, digital piracy, and illegal online trading.

The QR Code is a Matrix 2D BarCode that was developed by the Denso Company in Japan in September 1994. In addition to the advantages of 1D BarCode, it has specialties of rapid and wide identification, expresses Chinese and Japanese characters effectively, which other 2D BarCodes have been unable to do. QR Code has 40 versions, four error correcting levels. The single QR Code symbol can accommodate 1817 characters at most, and in character correction error of the data, the highest level can be roughly 30% of code words[10].

The digital watermark is a new kind of information security and protection technology. The basic idea is to embed the watermark signal that contains the electronic signature, date, trademark and access into the images, video, audio, text, and digital works, and to detect the watermark signal by a certain technology, to judge the validity of the copyright and to track and prosecute those who infringe on the digital works[4]. Of course, this hidden information can also be extracted from printed or scanned images. According to 2D BarCode anti-counterfeiting technology, combining the digital watermark with the 2D BarCode greatly improves the depth and width of the application of the 2D BarCode in the anti-counterfeiting field.

To resist print and scan attacks and to achieve the purpose of anti-counterfeiting, watermarking algorithms have been proposed in the literature [1], these algorithm are based on barcode rules, with watermark information embedded in the appropriate edge of the PDF417 code, but the user must consider the neighborhood of each pixel to identify the invisible watermark. Insufficiently, this method is suitable for the type of stacked barcode, and having not done print and scan attacks experiment. The feasibility of the method has not been verified. A method based on 2D BarCode encoding rules that uses error accumulation has been proposed by Niu Xiamu et al. [2], but it only works for detecting
information when a small amount of information exists and a high accuracy scanner is used. Based on the watermark technology of the DCT intermediate frequency, the photos are encoded in 2D BarCode[3]. The performance of the method was robust for scrambling the watermark signal, but the extracted watermark signal was of lesser quality after the print-scanning attacks, and the watermark image was fuzzy. A high accuracy printer is needed.

This paper presents an innovative watermark method. It is very important to discuss the selection of the noise. Based on the visual masking property of the human eyes, the watermark signal is embedded repeatedly, adding different noises to the QR Code gray image and using the comparison of the DCT intermediate frequency coefficients. This approach improved the invisible and robust to resist geometric distortion and the distortion of pixels after print and scanning. Experimental results show that the proposed method is feasible and effective in resisting print and scanning attacks, and the applicable scope is wide.

2. QR Code image preprocessing

QR Code symbols are an array composed of square modules, including a coding region and a function region that are composed of the viewfinder, separator, orientation, and graphics correction. There is a blank region around the symbols. Version 7 of the QR Code symbols is presented in Figure 1.

![Figure 1. QR Code symbols structure diagram](image)

Figure 2(a) depicts a 384×384 QR Code encoded with “CARD:N:VENU;COR:HANGZHOU DIANZI UNIVERSITY”. Binary images, which are black-and-white images whose each pixel only says “0” or “1”. When embedding watermark information into the binary image, unlike the gray image and the color image that have only isolated pixels, neighborhood pixels are needed to avoid vision abnormalities. Compared with the gray image and the color image, the binary image is short of visual redundancy. But in practical applications, QR Code with some noise will be identified correctly, and in paper printing and use, QR Code always introduces some noise. This noise does not affect the QR Code identification, so QR Code binary image is pretreated in order to embed the watermark information in the transform domain. Due to the absence of visual redundancy in the QR Code, first fuzzy then add noise to QR Code image to reduce the correlation between the image pixels in QR Code image, reduce visual distortion of the QR Code image with watermark and prevent DCT domain overflow. Because the different types of noise affect the original image differently, they must be tested individually. In the experiment, version 3 decoding software with a correction level of 7% was used.

The progress is as follows:
1. The QR Code gray image will be obtained by a Gaussian blur preprocessing, and the result is presented in Figure 2-(b).
2. The QR Code gray image after Gaussian blur preprocessing is added the Gaussian noise of 0 mean, and different variance $\sigma^2$, then , it was judged whether the code can be identified correctly. QR Code with Gaussian noise of 0 means and a variance of 0.001 is presented in Figure 2-(c).
3. The QR Code gray image after Gaussian blur preprocessing has random noise of 0 mean, and different variance $\sigma^2$ added, and then it was judged whether the code can be identified correctly. QR Code with random noise of 0 means and a variance of 0.001 is presented in Figure 2-(d).
4. The QR Code gray image after Gaussian blur preprocessing has salt and pepper noise added, and then it is judged whether the code can be identified correctly. QR Code with salt and pepper noise of a density of 0.04 is presented in Figure 2-(e), but it cannot be identified correctly. Salt and pepper noise is seriously harmful to the quality of the QR Code image, and it is a less useful process than the expected grayscale, so it is only presented as an analogy.

5. According to the first four steps, a QR Code grayscale image, which can be identified correctly can be obtained.

![QR Code image preprocessing](image)

Figure 2. QR Code image preprocessing

3. Embedding and extracting algorithm

The digital watermarking scheme structure diagram is presented in Figure 3. This paper consists of three measures to resist the distortion from print and scan, i.e. digital watermark structure, watermark embedding algorithm, and watermark extracting algorithm. But some attention must be paid to the readability of the QR Code image itself in order to protect the quality of the images in the watermark embedding strength and the scheme design of extraction.

![Digital watermarking structure diagram](image)

Figure 3. Digital watermarking structure diagram

3.1. Embedding watermark

Due to the fact that different printers and scanners affect the DCT quantitative coefficients of the digital image differently, so it is difficult to find the quantitative relationship between DCT coefficients. But print and scan is less useful to the order relationship between them, so the method of comparing DCT coefficients is used to embed the watermark.

To resist the geometrical attack from the print and scan, the watermark signal should be expanded periodically. This increases the redundancy of the watermark signal and makes the signal cyclical[5]. It can extract enough information to prove the existence of watermarking after geometric attacks.

The embedding progress is as follows:
1. Initially, the original image is divided into non-overlapping 8×8 blocks, and the technique
evaluates the DCT of each block. The array of DCT coefficients is presented in Table 1, and the grid with letters represent the position where the watermark is embedded.

### Table 1. Array of 8×8 DCT coefficients

<table>
<thead>
<tr>
<th>DCT coefficients</th>
<th>a(1,1)</th>
<th>a(1,2)</th>
<th>a(1,3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>a(2,1)</td>
<td>a(2,2)</td>
<td>a(2,3)</td>
<td></td>
</tr>
<tr>
<td>a(3,1)</td>
<td>a(3,2)</td>
<td>a(3,3)</td>
<td></td>
</tr>
<tr>
<td>a(4,1)</td>
<td>a(4,2)</td>
<td>a(4,3)</td>
<td></td>
</tr>
<tr>
<td>a(5,1)</td>
<td>a(5,2)</td>
<td>a(5,3)</td>
<td></td>
</tr>
<tr>
<td>a(6,1)</td>
<td>a(6,2)</td>
<td>a(6,3)</td>
<td></td>
</tr>
<tr>
<td>a(7,1)</td>
<td>a(7,2)</td>
<td>a(7,3)</td>
<td></td>
</tr>
</tbody>
</table>

2. Embed the watermark using the method of comparing DCT intermediate frequency coefficients. The seven groups of coefficients in each 8×8 block represent a 1-bit watermark. $B(j)$ represents a no. j-bit watermark, $a(i,1)$, $a(i,2)$, $a(i,3)$, where(i=1,2,...,7), that represents adjacent coefficients in the DCT intermediate frequency coefficients. The algorithm is as follows:

1) When $B(j) = 1$, the absolute maximum coefficient is placed in $a(i,2)$, the absolute larger coefficient is placed in $a(i,1)$, and the absolute minimum coefficient is placed in $a(i,3)$.

2) When $B(j) = 0$, the absolute minimum coefficient is placed in $a(i,2)$, the absolute maximum coefficient is placed in $a(i,1)$, and the absolute larger coefficient is placed in $a(i,3)$.

Due to the selection of the DCT intermediate frequency coefficients, they are relatively close. We should choose proper parameters $d = f(k)$ to control the embedding strength, the terms $d_1 = k$, $d_2 = k/2$, respectively, represents the embedding strength. Because the relationship order between the coefficients may have changed after print and scan attacks, we need introduce a threshold $k$ to control the watermark strength, to reduce the False Accept Rate, and improve robustness. The algorithm is as follows:

$$\begin{align*}
\text{If } & \quad \max(a(1,1), a(2,2), a(3,3)) + \min(a(1,1), a(2,2), a(3,3)) > k \\
\text{if } & \quad \min < k/2 \\
& \quad \text{max} = \text{max} + d_1; \\
& \quad \text{else} \quad \text{max} = \text{max} + d_2; \\
& \quad \text{min} = \text{min} - d_2; \\
\text{else } & \quad \text{embed watermark}.
\end{align*}$$

3. To resist the geometric attack from the print and scan attacks, use the watermark periodically to constitute a new watermark, and embed the original watermark 16 times.

4. Each block was processed with IDCT to obtain the watermarked image.

### 3.2. Extracting watermark

To print and scan the watermarked image, and correct the image, the extracting process is as follows:

1. The watermarked image is divided into non overlapping 8×8 blocks, and each block was processed with DCT.

2. The algorithm is as follows.

   If $B(i,j) = 1$;
   else $B(i,j) = 0$;

   $k_1$ for adjustment factor, $0 < k_1 < 1$.

3. The 7-bit watermark was extracted, as follows: $B(1,j)$, $B(2,j)$,...,$B(7,j)$, and judge whether $B(j)$ would be 0 or 1, according to the maximum membership degree.

4. Due to the requirement that the original watermark be embedded 16 times, again we should judge whether $B(j)$ would be 0 or 1 according to maximum membership degree.
4. Experimental results and analysis

In the experiment, the QR Code images (size of 384×384) with different types of noise would be the cover images in Figure 2, and the binary image was chosen for the watermark image, the size of which is 12×12 (Figure 4). Due to the requirement that the watermark be embedded 16 times in order to identify the QR Code correctly and to reduce the correlation between the image pixels to be good for detecting and extracting, we set the threshold for k = 100, watermark strength for d1 = 100, d2 = 50, and the adjustment factor for k1 = 0.48.

![Figure 4. Watermark image](image)

In the experiment, first, the gray image after fuzzy was chosen for the cover image. Then different types of noises were added by the same variance of 0.001 to the binary image after fuzzy, and the watermark was embedded in an image with Gaussian noise, an image with random noise, and an image with salt and pepper noise. Salt and pepper noise is seriously harmful to the quality of the QR Code image, and visual distortion was large. Therefore, we only deal with the first two types. Print and scan the watermarked image to obtain 10 samples, then, extract the watermark from each of the 10 samples. Correlation test data are shown in Table 2 and Figure 5. As shown in Table 3, in the cases in which the watermarked images with three types of noise were used without any treatment, the peak signal to noise ratio (PSNR) showed that salt and pepper noise causes the largest harm to the quality of the image, results in the least useful grayscale of the cover image, and prevents the identification of the QR Code. The watermarked image was printed by an HP Color Laserjet 4700 printer, scanned by an HP Scanjet 4890 scanner with 600 dpi to obtain the digital image, corrected using Adobe Photoshop software, and restored with a sampling resolution of 72 dpi and an image size of 384×384. In Table 3, the experimental results show that the extraction results from the watermarked images with noise would be much better than the results from the gray image after fuzzy, after print, and after scan. It was the best extraction result for the watermarked image with random noise after print and scan, and it had the biggest correlation value.

In view of the initial experimental results, the image was tested with random noise, and variances of 0.0002, 0.005, and 0.02. It was hoped that the extraction result would be better. The test data are shown in Table 4 and Figure 6. As shown in Table 5, the extraction result from the watermarked image with random noise with a variance of 0.005 was the best: the mean of the correlation data was 0.9221, the best extraction value was 0.9849, and the QR Code can be identified correctly. In Table 5, the experimental results indicate that, although the extraction result from the watermarked image with random noise with a variance of 0.02 was reasonably good, the QR Code cannot be identified correctly after print and scan, so it is difficult to use the QR Code in practice in this case.

In the experiment, we also embedded the watermark in the image with random noise 32 times, but the size of the cover image and the size of the watermark image were too small, so we gave up the result although the result was 100%. In practice, it is sufficient to embed watermark 16 times to extract the watermark. So when using different gray images, we can consider using different times to expand the watermark in order to get good extraction results. In the experiment, the scanner directly affected the extraction result directly.

| Table 2. Correlation test data of samples in different types of noise after print and scan |
|---|---|---|---|---|---|---|---|---|---|---|
| Noise type | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | 10  | Mean |
| Without noise | 0.0508 | 0.5620 | 0.6544 | 0.6686 | 0.5831 | 0.5901 | 0.6024 | 0.5426 | 0.5586 | 0.6022 | 0.5414 |
| Gaussian noise | 0.8414 | 0.9239 | 0.9114 | 0.9239 | 0.7929 | 0.9397 | 0.7929 | 0.7176 | 0.7504 | 0.8040 | 0.8398 |
| Random noise  | 0.8310 | 0.9325 | 0.8455 | 0.9267 | 0.8543 | 0.9462 | 0.8210 | 0.7943 | 0.8544 | 0.9081 | 0.8714 |
A Digital Watermark Algorithm for QR Code
Li Li, Rui-Ling Wang, Chin-Chen Chang
International Journal of Intelligent Information Processing Volume 2, Number 2, June 2011

Figure 5. Comparison diagram for test data using different noises

Table 3. Different types of noise by print and scan attacks

<table>
<thead>
<tr>
<th>Noise type</th>
<th>Add noise (variance)</th>
<th>Cover image with noise</th>
<th>Histogram</th>
<th>PSNR (dB)</th>
<th>Watermarked image by attacks</th>
<th>Optimal extraction</th>
<th>Mean of correlation</th>
<th>Decode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without noise</td>
<td>-</td>
<td></td>
<td></td>
<td>52.145 5</td>
<td></td>
<td></td>
<td>0.541 4</td>
<td>Yes</td>
</tr>
<tr>
<td>Gaussian noise</td>
<td>0.001</td>
<td></td>
<td></td>
<td>51.122 0</td>
<td></td>
<td></td>
<td>0.839 8</td>
<td>Yes</td>
</tr>
<tr>
<td>Random noise</td>
<td>0.001</td>
<td></td>
<td></td>
<td>51.693 8</td>
<td></td>
<td></td>
<td>0.871 4</td>
<td>Yes</td>
</tr>
<tr>
<td>Salt&amp;pepper noise</td>
<td>density: 0.04</td>
<td></td>
<td></td>
<td>36.677 9</td>
<td></td>
<td></td>
<td>0.867 9</td>
<td>No</td>
</tr>
</tbody>
</table>
Table 4. Correlation test data of sample of the watermarked image with random noise by print and scan attacks

<table>
<thead>
<tr>
<th>Variance</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without noise</td>
<td>0.050 8</td>
<td>0.562 0</td>
<td>0.654 4</td>
<td>0.668 6</td>
<td>0.583 1</td>
<td>0.590 1</td>
<td>0.602 4</td>
<td>0.542 6</td>
<td>0.558 6</td>
<td>0.602 2</td>
<td>0.541 4</td>
</tr>
<tr>
<td>0.002</td>
<td>0.722 0</td>
<td>0.848 6</td>
<td>0.668 6</td>
<td>0.687 6</td>
<td>0.662 4</td>
<td>0.911 4</td>
<td>0.801 2</td>
<td>0.640 5</td>
<td>0.665 4</td>
<td>0.650 6</td>
<td>0.725 8</td>
</tr>
<tr>
<td>0.005</td>
<td>0.831 0</td>
<td>0.932 5</td>
<td>0.845 5</td>
<td>0.926 7</td>
<td>0.854 3</td>
<td>0.946 2</td>
<td>0.821 0</td>
<td>0.794 3</td>
<td>0.854 4</td>
<td>0.908 1</td>
<td>0.871 4</td>
</tr>
<tr>
<td>0.02</td>
<td>0.901 0</td>
<td>0.955 7</td>
<td>0.954 2</td>
<td>0.984 9</td>
<td>0.927 8</td>
<td>0.955 7</td>
<td>0.894 5</td>
<td>0.908 1</td>
<td>0.854 4</td>
<td>0.884 3</td>
<td>0.922 1</td>
</tr>
</tbody>
</table>

Table 5. Random noise by print and scan attacks

<table>
<thead>
<tr>
<th>Noise type (variance)</th>
<th>Add noise</th>
<th>The cover image with noise</th>
<th>Histogram</th>
<th>PSNR(dB)</th>
<th>Watermarked image by attacks</th>
<th>Optimal extraction</th>
<th>Mean of correlation</th>
<th>Decode</th>
</tr>
</thead>
<tbody>
<tr>
<td>without noise</td>
<td>-</td>
<td><img src="image" alt="QR code" /></td>
<td><img src="image" alt="Histogram" /></td>
<td>52.145 5</td>
<td><img src="image" alt="Watermarked image" /></td>
<td><img src="image" alt="Optimal extraction" /></td>
<td>0.541 4</td>
<td>Yes</td>
</tr>
<tr>
<td>Random noise 0.000 2</td>
<td></td>
<td><img src="image" alt="QR code" /></td>
<td><img src="image" alt="Histogram" /></td>
<td>52.902 6</td>
<td><img src="image" alt="Watermarked image" /></td>
<td><img src="image" alt="Optimal extraction" /></td>
<td>0.725 8</td>
<td>Yes</td>
</tr>
<tr>
<td>Random noise 0.001</td>
<td></td>
<td><img src="image" alt="QR code" /></td>
<td><img src="image" alt="Histogram" /></td>
<td>51.693 8</td>
<td><img src="image" alt="Watermarked image" /></td>
<td><img src="image" alt="Optimal extraction" /></td>
<td>0.871 4</td>
<td>Yes</td>
</tr>
<tr>
<td>Random noise 0.005</td>
<td></td>
<td><img src="image" alt="QR code" /></td>
<td><img src="image" alt="Histogram" /></td>
<td>49.348 3</td>
<td><img src="image" alt="Watermarked image" /></td>
<td><img src="image" alt="Optimal extraction" /></td>
<td>0.922 1</td>
<td>Yes</td>
</tr>
<tr>
<td>Random noise 0.02</td>
<td></td>
<td><img src="image" alt="QR code" /></td>
<td><img src="image" alt="Histogram" /></td>
<td>44.498 9</td>
<td><img src="image" alt="Watermarked image" /></td>
<td><img src="image" alt="Optimal extraction" /></td>
<td>0.914 8</td>
<td>No</td>
</tr>
</tbody>
</table>
5. Conclusions

This paper discusses the fact that different types of noise have different effects on the QR Code image, and on the method of comparison of DCT intermediate frequency coefficients to embed watermark. Technology that combines 2D BarCode with digital watermark is used in security field. The experimental results prove that the digital watermark method is feasible and effective. The preprocessing of the QR Code image is very important for experimental success. The cycle of the watermark information improved the performance of the anti-print and anti-scan attacks and enhanced the security and secrecy of the QR Code in practical applications. It is difficult to forge the 2D BarCode when digital watermarking technology is used with the 2D BarCode.

In conclusion, the technology combining the 2D BarCode with digital watermarking has developed very quickly, and it is anticipated that, with further research and the development of appropriate hardware, the combined technologies will have wide application.

6. Acknowledgement

This work is partially supported by the National Natural Science Foundation of China (No.60703087), the Project of Science and Technology Department of Zhejiang Province (No.2008C13080). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

7. References