CHAPTER III
PROBLEM ANALYSIS

III.1 Arnold Transform

Expression of Arnold Transform can be shown as the following equation:

$x' = a_1x + a_2y + mod L$

$y' = a_3x + a_4y + mod L$

The parameters may fulfill, a1 * a4 – a2 * a3 = ± 1 and the parameter L x L is the size of the logo image. Scrambled image can be get from an adaptive logo-scrambling performs with iterative procedure. However, Arnold Transform only valid for square images. The Arnold Transformation is used to scramble the digital images and has many applications, especially in digital watermarking.

This technique can rotate logo for angles 90, 180, 270. The ability to scramble and descramble may increase the security in watermarking.

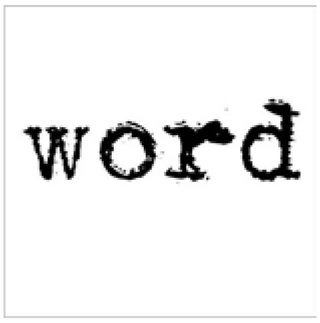

Figure 3.1 Logo Image of Size 64x64

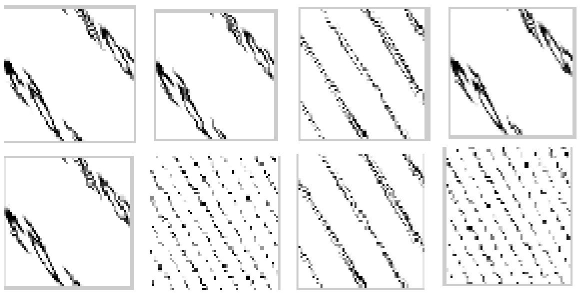

Figure 3.2 Results of Arnold Transform applied on logo image size of 64x64

III.2 Morphological Haar Wavelet Transform

In mathematics, expression of Haar transform matrix shown as the following:

$T = HFH^T$

F = N x N image matrix

H = N x N transformation matrix

T = the result of Haar Transform

H contains the Haar basis function, its h_k (z), it is defined in a continuous closed interval of z ∑ [0,1].

k = 0,1,2,….N – 1.

$N = 2^n$.

To generate H matrix, define an integer variable *k*, as k = 2p + q – 1 (0 ≤ p ≤ n – 1 when p = 0, q = 0 or 1).

Define of Haar basis function:

$$h_0(z) = h_{00}(z) = \frac{1}{\sqrt{N}}, z \in [0,1]$$

And

$$h_k(z) = h_{pq}(z) = \frac{1}{\sqrt{N}} \begin{cases} 2^{p/2} & (q-1)/2^p \le z < (q-0.5)/2^p \\ -2^{p/2} & (q-0.5)/2^p \le z < q/2^p \\ 0 & \text{others}, z \in [0,1] \end{cases}$$

To embed a pictures we refer to two section of synthesis operators, a family of $\omega^s{}_j$ of signal synthesis operator and a family of $\omega^s{}_j$ which detail synthesis operators mapping $V_j + 1$ back into V_j. The equation define as:

$$\psi_j^a(x)(m,n) = \min(x(2m,2n), x(2m+1,2n), x(2m+1,2n), x(2m+1,2n+1))$$

$$\omega_j^a(x)(m,n) = (\omega_{j,v}^a(x)(m,n), \omega_{j,h}^a(x)(m,n), \omega_{j,d}^a(x)(m,n))$$

$\psi_j^a, \omega_{j,v}^a, \omega_{j,h}^a, \omega_{j,d}^a$ represent the scaled signal and the vertical, horizontal, and diagonal detail signals, which are given by below equation:

$$\omega_{j,v}^{a}(x)(m,n)=1/2((x(2m,2n)-x(2m,2n+1)+x(2m+1,2n)-x(2m+1,2n+1))$$

$$\omega_{j,h}^{a}(x)(m,n)=1/2((x(2m,2n)-x(2m+1,2n)+x(2m,2n+1)-x(2m+1,2n+1))$$

$$\omega_{j,d}^{a}(x)\,(m,n)=1/2((x(2m,2n)-x(2m+1,2n)-x(2m,2n+1)+x(2m+1,2n+1))$$

The synthesis operators are now given by below equation:

$$\psi_{j}^{s}(x)(2m,2n)=\psi_{j}^{s}(x)(2m,2n+1)=\psi_{j}^{s}(x)\,(2m+1,2n)=\psi_{j}^{s}(x)\,(2m+1,2n+1)=x(m,n)$$

And

$$\omega_{j}^{s}(y)(2m,2n)=\max(y_{v}(m,n)+y_{h}(m,n),y_{v}(m,n)+y_{d}(m,n),y_{h}(m,n)+y_{d}(m,n),0)$$

$$\omega_{j}^{s}(y)(2m+1,2n)=\max(y_{v}(m,n)-y_{h}(m,n),\,y_{v}(m,n)-y_{d}(m,n),-y_{h}(m,n)-y_{d}(m,n),0)$$

$$\omega_{j}^{s}(y)(2m,2n+1)=\max(y_{h}(m,n)-y_{v}(m,n),-y_{v}(m,n)-y_{d}(m,n),\,y_{h}(m,n)-y_{d}(m,n),0)$$

$$\omega_{j}^{s}(y)(2m+1,2n+1)=\max(-y_{v}(m,n)-y_{h}(m,n),y_{d}(m,n)-y_{v}(m,\,n),\,y_{d}(m,n)-y_{h}(m,n),0)$$

III.3 Watermark Embedding

First, using Arnold Scrambling Algorithm to improve the security of watermark. As example, picture with 64x64 pixel grayscale Baboon image as the watermark is used.

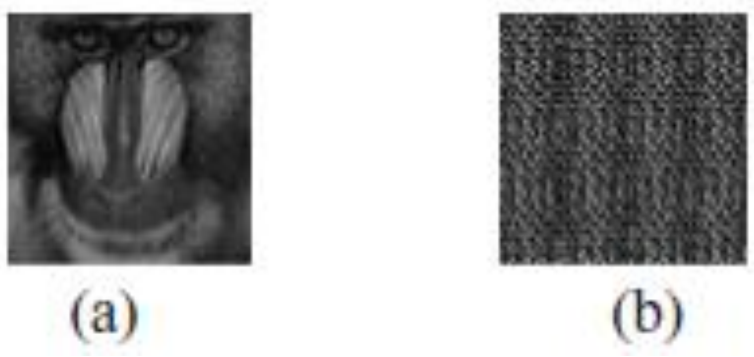

Figure 3.3 Watermark embedding using Arnold Scrambling Algorithm

Then, composed the original image by L-level MHWT. After that the result can get the approximate subgraph D*l* obtained by the following formula:

$$p(i,j)=(1-NVF(i,j))\times S_1+NVF(i,j)\times S_2$$

NVF is the Noise Visibility Function, S1 and S2 are the maximum allowable distortion values of expected texture region and flat region.

The result of picture is, the approximate multi-scale images retain the basic texture features of the original image by MHWT, so this region can be embedded with watermark. For the concrete steps, it can be following:

1. Composed the original image into L-level wavelet into L-level wavelet to obtain the image Dl with the size of M*l* x N*l*, based on the 2-D MHWT.
2. Do the first step again to obtain the relevant level scale image *Wi*, with the size of *Mi x Ni.*
3. Get the permissible distortion value matrix P from the original image using NVF.
4. Select the maximum $\beta \times M\,i \times N\,i$ values of $p(i, j)$ from D*l*, where β is in the range of $[1,2, \ldots, \frac{Ml \; x \; Nl}{Mi \; x \; Ni}]$
5. Select the number of $M\ i\ x\ N\ i$ pixel points from mentioned values above as embedding watermark position, use the generated random sequence under the key K.
6. Corresponding formula, the decomposed watermarks are embedded in the corresponding resolution of the decomposed original image to which the watermark can adapt itself.

   $$D'_l(i, j) = D_l(i, j) + \alpha(i, j) \cdot W_i(i, j)$$

   Where α is the embedding strength of watermark, W*i* (*i*, *j*) is the relevant information of watermark.
7. Apply Morphological Haar Synthesis Transform on the D*'l* image, include the detail signal images, to produce the watermarked host image.

   The watermark embedding carried out in the above steps is shown in image below.

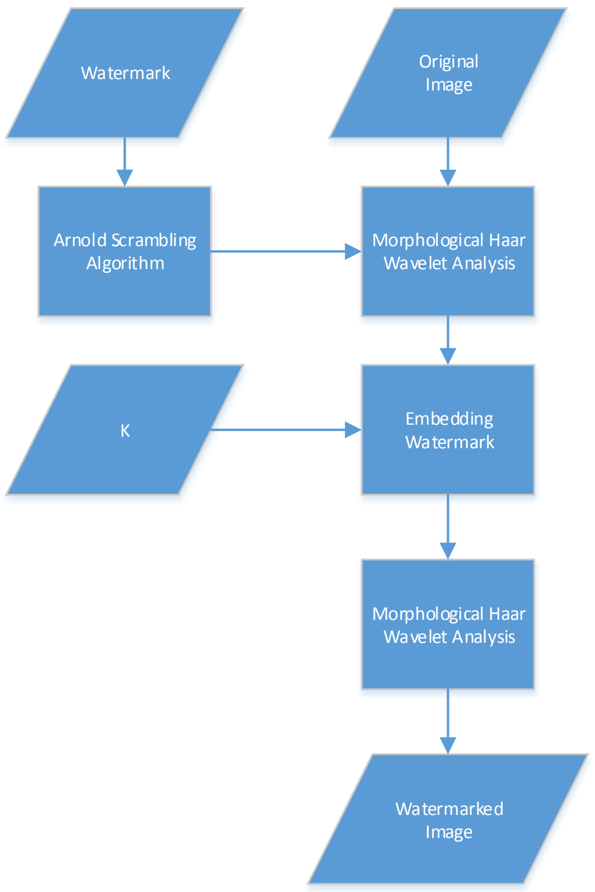
Watermark
Original Image
Arnold Scrambling Algorithm
Morphological Haar Wavelet Analysis
K
Embedding Watermark
Morphological Haar Wavelet Analysis
Watermarked Image

Figure 3.4 Flowchart watermark embedding

III.4 Watermark Detection

The inverse procedure of the watermark embedding is Watermark Extraction. Non-blind watermarking algorithm and the original host image is required to extract the watermark. The similarity between the original watermark and the extracted watermark using the correlation coefficient factor ρ given below in equation:

$$\rho(w,\hat{w}) = \frac{\sum_{i=1}^{N} w_i \hat{w}_i}{\sqrt{\sum_{i=1}^{N} w_i^2}\sqrt{\sum_{i=1}^{N} \hat{w}_i^2}}$$

w & ŵ = the original and extracted watermarks.

N = the number of pixels in watermark.

ρ = the number between [0-1].

The correlation coefficient factor value around 0.75 and above is considered acceptable. The watermark detection is conducted in the steps as shown in below picture.

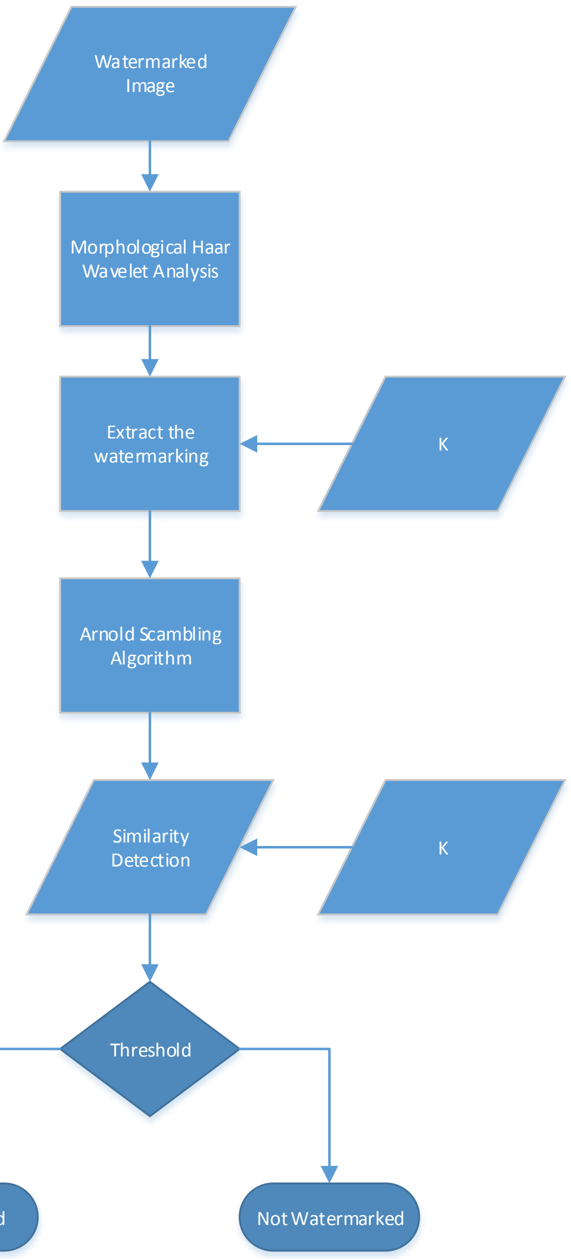

Figure 3.5 Flowchart watermark detection

III.5 Result of Embedding

Combination of Haar Wavelet Transform and Arnold Scrambling Algorithm won’t make watermark easy to detect. However, both of Algorithm are dependent each other. Without Haar Wavelet Transform, Arnold Scrambling Algorithm only can make the watermark reducted to detected. Without Arnold Scrambling Algorithm, Haar Wavelet Transform can’t make the secure of Watermarked Image.

The reason why Haar Wavelet Transform need Arnold Scrambling Algorithm because it can improve the security to make the watermark hard to detect when it embedded with original image. Arnold Scrambling Algorithm also used in decode the image which want to be decoded. It’s because Arnold Scrambling Algorithm will detect the similarity of the image which want to be decoded.