USING HIDE WATERMARK IN VISUAL WATERMARK EXTRACTION. ADVANTAGES. ALGORITHM

Marius ROGOBETE1
Ciprian RÂCUCIU2
Marian-Dorin PIRLOGĂ3
Florin MEDELEANU4

1 Eng.Alstom GRID Romania, 040231, Bucharest, Romania, marius.rogebete@gmx.de
2 Prof. eng. PhD, ‘Titu Maiorescu’ University, Bucharest, Romania
3 Inf. Military Technical Academy, Bucharest, Romania
4 Eng. Military Technical Academy, Bucharest, Romania

Abstract: Any embedded watermark as a guest image into a host image should use an embedding function that offers specific characteristics to the visual watermark, on the side of the host image’s owner or sender, for security reason. On the other side, of the receiver, the host image is visual marked by the watermark object. The sender could offer to the receiver a software tool that purposes to eliminate the watermark in such a way that the output image to be clear, 100% as the original one. This process of visual watermark extraction is based on the inverse embedding function. This function could be different from stream to stream or even from image to image. The function identification could be done using the hide watermark information, embedded into the host image. The algorithm is presented together with the main advantages of the method.

INTRODUCTION
The watermark image that overloads the host image should be fixed theoretically without removing possibility, like the classical method of visual embedding watermark.

This scenario is extremely advantageous for the owners that purely protect their images over media, but it has some disadvantages when, on the image receiver side, the pictures/stream should be displayed without visual watermark embedded. For this last case the embedded watermark should be extract without any visible damage of the main image.

Into the host image a visible watermark is inserted using a specific embedding watermark function that operates directly on pixel value. When the function is bijective, the watermark could be removed from the embedded image without any damage of the original picture. But, using non-bijective functions, the watermark image is permanently embedded and the original image could not be perfectly retrieved.

EMBEDDING FUNCTION
A simple bijective (reversible) function form (1) is:

\[ f(i_{w,m}) = a \cdot p_{w,m} + b \] (1)

With

\[ i_{w,m} = \begin{cases} f(i_{o,m}) & \text{for } w_{o,m} = 1 \\ i_{o,m} & \text{for } w_{o,m} = 0 \end{cases} \] (2)

where

\[ i_{o,m} \in I_o, \quad n = 0,\ldots,N-1, \quad m = 0,\ldots,M-1, \quad m,n,a,b \in \mathbb{Z} \]

\[ w_{o,m} \in W \Rightarrow w_{o,m} = 1 \]

and \( I_o \) is the watermarked image; \( I' \) is the visual watermark.

The invers function, \( f^{-1} \) allows completely compensate the embedded watermark and to recover the original host image without losing quality. Having the recovered image \( K \), the mathematical form is [6]:

\[ k_{w,m} = \begin{cases} f^{-1}(i_{w,m}) & \text{for } w_{o,m} = 1 \\ i_{w,m} & \text{for } w_{o,m} = 0 \end{cases} \] (3)

where \( k_{w,m} \in K, \quad n = 0,\ldots,N-1, \quad m = 0,\ldots,M-1, \quad m,n \in \mathbb{Z} \).

METHOD PRESENTATION
The embedding functions define the visual watermark properties as are: transparency, color, position and, most important for this research work, the capabilities of the watermark to be complete extracted or, in other words, to restore the original content of the host image such the recovered image to be identical to the original image [1], pixel by pixel. When the embedding function is bijective it is possible to recover the original image without losing the image quality, if is applied its inverse function [6].

The visual watermark extracting process is necessary when the owner decide to deliver the image without embedded image/logo to specific users.

This process can be performed when just bijective function is applied to embed the visual watermark. But, before to extract the visual watermark, is necessary to decide if the user has rights to proceed with.

The hide watermark has like main goal to hide specific message \( M_w \) and a map of watermark position, \( G_w \), in cover data (the original image \( I \), figure 1), to obtain new data \( I' \), practically indistinguishable from \( I \).

Authentication and Copyright
The bijective function parameters are extracted from the hide watermark that contains the owner identification string \( S_i d \), and the

Figure 1. The host image \( I_o \).
constant parameters, \(a\) and \(b\) in equation (3), used by the inverse function.

On the receiver side is checked if the hide watermark signature (\(\text{Sid}\)) is the same with the license signature. If the user is licensed, is extracted from \(M_w\) the inverse function constant parameters \(a\) and \(b\). The inverse embedding function is applied, in order to compensate the visual watermark and to recover the original host image \(I\), which is identical with \(I\). When the signature or license is missing the inverse function is not applied and the output image is the watermarked image.

**Watermark’s Border**

In order to apply the inverse function, we need the geometrical locus of the watermark, \(G_w\). It defines a set of pixels that should be processed, applying the inverse embedding function. The watermark’s geometrical locus is defined by the watermarked area only (bordered in figure 2).

![Figure 2](image)

Figure 2. The watermarked image with the watermark’s border highlighted.

The geometrical locus of the watermark (figure 3) is embedded into watermarked image as supplementary info, using stenographic techniques. Therefore, the final watermarked image will contain:
- Visible watermark \(W\);
- Embedded string message \(M_w\) formatted from:
  - license key substring \(\text{Sid}\);
  - constant parameters substring \(a, b\);
- Map of watermark position \(G_w\) (as a black and white image).

Thus, after the visual image was added to the host image, the hidden watermark is embedded (4), where \(I_w\) is the image broadcasted over media and \(I_0\) is the original host image:

\[
I_M = I_0 \oplus W \oplus M_w \oplus G_w
\]

(4)

It is used the LSB matching techniques, a simple stenographic method because it doesn’t need any attack protection. Basically it is used a semi-robust techniques as the forgery detection is blind, any image modification will modify the hide watermarks (message and watermark map) that will produce an incorrect visual watermark elimination.

![Figure 3](image)

Figure 3. Geometrical locus of the watermark into the host image

**THE ALGORITHM FORM**

The general watermarking algorithm is presented in figure 4.

![Figure 4](image)

Figure 4. Watermarking algorithm

Finally the embedding algorithm has the follow logical structure (figure 5):

![Figure 5](image)

Figure 5. The block scheme for complete embedding algorithm

**Conclusions**

The presented method uses hide watermark to attach copyright information and constant parameters to the host image. Also, the geometrical locus of the watermark’s pixels is attached to the watermarked image. Using this info, the method is able to apply the inverse of the embedding watermark’s function and to extract the embedded visual watermark.

The copyright protection but also image forgery protection is ensured on the owner side, based on a complex algorithm that integrate the steganography techniques for messages and image together with the watermarking technique for visual embedding image. The protection is assured on the watermark extraction side, as the stego message and stego image are directly used for visual watermarking compensation. Thus, any image modification will produce wrong visual watermark compensation, resulting in a damaged image on the receiver side.

**Bibliography**


