Automated Systems & Technologies
25-26 May 2015 • St. Petersburg, Russia

WATERMARKING FRAMEWORK FOR IMAGE PROTECTION. A CASE STUDY.

Marius Rogobete[1,*], Ciprian Răcuciu[2]

[1]Alstom GRID, Software Development, Bucharest, Romania
[2]Titu Maiorescu University, Science and Information Technology, Bucharest, Romania
[*]marius.rogobete@gmx.de

Abstract

The copyright protection and general information of photos or video streaming are embedded and preserved using metadata, being essential to track and identify the digital images. For an efficient protection the ownership metadata should never be removed, but practically it can be relatively easy extracted.
This work proposes a watermarking method able to insert visual and hide information into image, instead metadata protection. The image creator or copyright holder is embedding visible and hide watermarks into image, using a specific framework.
A visible watermark inserted into the host image could be removable or permanently, based on the bijective or non-bijective embedding watermark function.
The permanently watermarking method is choosing when the image/video stream is distributed without control and avoids any attack.
The removable watermark allows the receiver far end to eliminate the visible watermark, whether it uses a framework that uses the invers embedding function. In this way, only the controlled receivers could profit by the clean photos/video stream.
More of this, the hide watermark could embed typical information that identify the owner.
When the framework's receptor module tries to eliminate visible watermark, it checks, first of all, the owner info embedded with hide watermark. If the hide information integrity is damaged (the frame/photo was changed) then the removing process is not done.
This research contains the description of watermark framework, several examples of bijective/non-bijective embedded functions and practical results with image quality comparison.

INTRODUCTION

The images description and protection is based, actually, on metadata info. It is a data structure that provides info about the digital image such as author, time and date of creation, purpose of image, network location, used standards, image properties (size, colour depth, resolution.), etc.
The metadata utilisation was extended to the image stream / video stream. Its standard is developed and updated by International Press Telecommunication Council (IPTC) which is universal accepted by news agencies, photographers, photo agencies, libraries, museums, and other related industries [8]. The specific structure together with metadata properties permit user to add reliable and precise data about images as is detailed in metadata standard documentation (IPTC Standard Photo Metadata 2014) [8] [9].
The main disadvantage of the metadata is given by the relatively simple method of image extraction from the whole entity of meta-container. This behaviour is kept even if advanced standard frameworks are used, as is Windows Imaging Component, which supports reading/writing specific metadata from/into image file [10].
The meaning of this weakness is it can be easy developed software applications able to change or eliminate metadata information from the container, keeping only the image. This problem is reported and proved by several IPTC studies: the main social media networks remove the metadata structure information from photos, keeping just the image [11].
The IPTC's tests resume that [12]:

- Facebook: metadata not shown anymore, all embedded metadata stripped-off from image files
- Google: primarily Exif metadata shown, all embedded fields are preserved
- Instagram: image taken by a smartphone, metadata edited with an app, then posted at Instagram - No metadata are shown, all metadata stripped-off from Save As files.
- Twiter: no metadata shown, all embedded metadata stripped-off from image files.

The IPTC Metadata Conferences didn't discuss the embedding method of the metadata [13], because the actually codecs concept defines only one technical way to add together the specific image and the metadata info: a concatenation method based on specific header [9]. For example the content diagram of a JPEG file that includes embedded metadata blocks (of type XMP, Extended) and metadata item, where the metadata is attached to a single frame as JPEG format does not support multiple image frames [14] (figure 1).
As any copyright protection is eliminated whether metadata is eliminated, the only way to keep almost permanently the copyright information is to embed it into image.
The watermarking technique allows overlying a translucent image over the host picture. It conveys ownership information directly on the media and it can deter attempts of copyright violations.

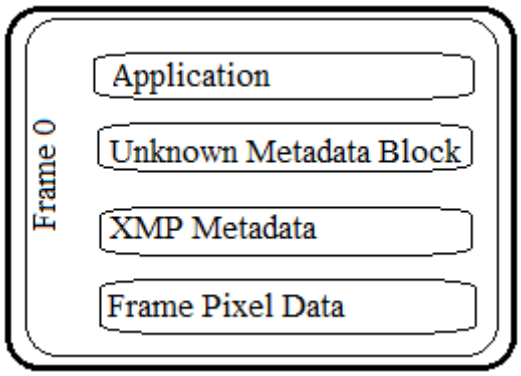


Figure 1. JPEG Image with Metadata [14]

GENERAL PRESENTATION

Having the copyright information as a watermark image, it is embedded clearly, perceptibly into the primary image, using different kind of embedding functions.
These functions define the visual watermark properties as are transparency, colour, 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 any image quality, if is applied its inverse function [6].
The framework was designed to embed a visual watermark into any host image. The bijective function is applied whether the sender decides to remove the visual watermark from image on the receiver side, or non-bijective one if no receiver needs clean image. The visual watermark is an image (like an identification logo) that overlays the host image. Afterwards, the output watermarked image is distributed over media, using communication devices/channels, usually without any receiving control.
The receiver can display the image or even video stream sequence as it is, with the image watermark embedded (and the framework is not used on the receiver side).
But some receivers, who have specific plugins installed in the browser, will try to extract the watermark, in order to recover the original image [2]. When the embedding watermark process is reversible (the embedding function is bijective), a complete extraction is possible using specific inverse embedding function. On the user far end, the framework supplies the inverse watermark embedding function, in order to recover the original image.
If the receiver is not recognized (the license is missing) or the image is modified (the hide watermark is not identified) then the visual watermark is not extracted at all or it is incomplete eliminated [3] [4]. In this way the image integrity is checked (the signature from the hidden watermark), and the visual copyright protection is removed if the original image was not tampered.

EMBEDDING BIJECTIVE FUNCTION

The embedding watermark function is a linear, bijective function (1). The I_0 is the host image, W is the watermark image (of the same NxM size), d is an arbitrary constant value and results I_W, the image with watermark embedded in the embedding function (1) [6]:

$$\forall i_{0_{n,m}},,\left| i_{0_{n,m}} \in \{I_0 \cap W\} \Rightarrow i_{w_{n,m}} = \begin{cases} i_{0_{n,m}} + d, pentru \quad i_{W_{n,m}} \in (0,255) \\ (i_{0_{n,m}} + d) \bmod 256, pentru \quad i_{W_{n,m}} \notin (0,255)] \end{cases} \right. \tag{1}$$

where $i_{w_{n,m}} \in I_W, \quad n = 0,...,N-1, \quad m = 0,...,M-1, \quad d \in Z.$

The general algorithm with the simplest function $f(i_{w_{n,m}}) = i_{0_{n,m}} + d$ is:

$$i_{w_{n,m}} = \begin{cases} f(i_{0_{n,m}}) \quad for \quad w_{n,m} = 0 \\ i_{0_{n,m}} \quad for \quad w_{n,m} = 1 \end{cases} \tag{2}$$

The invers function, f^1 allows completly compensate the embedded watermark and to recover the original host image without lossing quality. Having the recovered image Q, the mathematical form is [6]:

$$q_{n,m} = \begin{cases} f^{-1}(i_{w_{n,m}}) \quad pentru \quad w_{n,m} = 1 \\ i_{w_{n,m}} \quad pentru \quad w_{n,m} = 0 \end{cases} \tag{3}$$

FRAMEWORK DESCRIPTION

The framework's block diagram (figure 2 and 3) has two main modules, the sender and the receiver. The owner distributes the picture/frame over internet and more receivers could access the images.

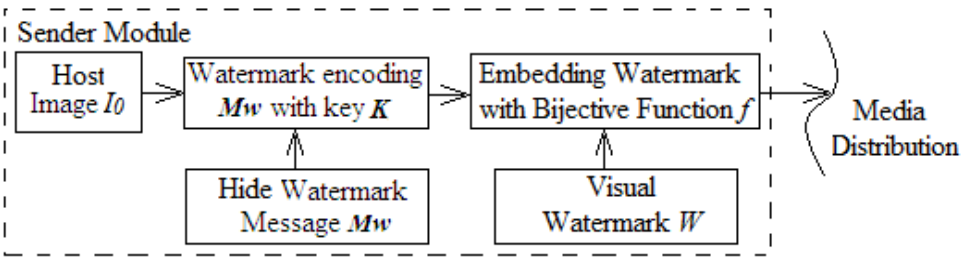


Figure 2. Sender module of the framework block diagram

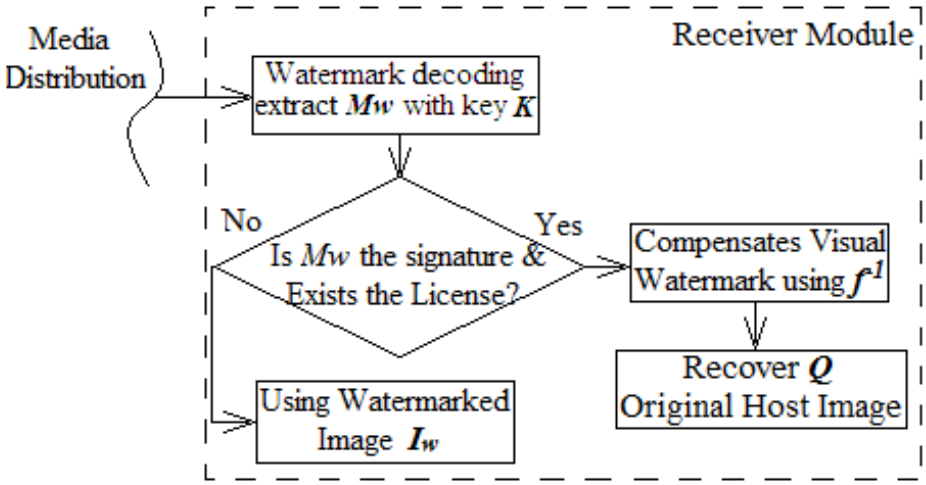


Figure 3. Receiver module of the framework block diagram

Embedding non-bijective function

On the sender side, the watermark ***W*** (figure 5) is embedded into the host image I_0 (figure 4) using bijective function (1). If the owner prefers to insert a permanent watermark, the embedding function should be a non-bijective one, e.g. mathematical form (4) [6]:

$$\forall i_{0_{n,m}} \Big| i_{0_{n,m}} \in \{I_0 \cap W\} \Rightarrow i_{w_{n,m}} \Rightarrow \begin{cases} R = cR + \dfrac{iR_{0_{n,m}} * (vR_{\max} - vR_{\min})}{255} \\ G = cG + \dfrac{iG_{0_{n,m}} * (vG_{\max} - vG_{\min})}{255} \\ B = cB + \dfrac{iB_{0_{n,m}} * (vB_{\max} - vB_{\min})}{255} \end{cases} \quad (4)$$

, where vR_{min}, vG_{min} and vB_{min} are minimum values on the channel band and vR_{max}, vG_{max} şi vB_{max} are maximum values, choose by the user. The low limits of the colors of inserted image are *cR, cG* and *cB*.

Sender module

For the bijective function, a semi-robust watermark message ***Mw*** is hidden into the host image using LSB method (but could be used also a cryptographic algorithm [5] [7]). ***Mw*** is practically the constant parameter, *d* in equation (1), of 8 bytes size, used in the inverse function. It could be different from frame to frame in case of images stream.

Figure 4. Host image I_0

Figure 5. The watermark image W used for embedding

When the sender distributes the image with watermark inside (***Iw***, figure 6), on the receiver side the image is analysed.

Receiver module

First of all is checked the hide watermark signature and if the license exists. Then the inverse embedding function is applied, in order to compensate the visual watermark and to recover the original host image ***Q*** (figure 7). If the signature or license is missing, then the inverse function is not applied and the watermarked image is used as it is.

Fig. 6. Watermarked image I_w

Fig. 7. The recovered image Q after watermark extraction

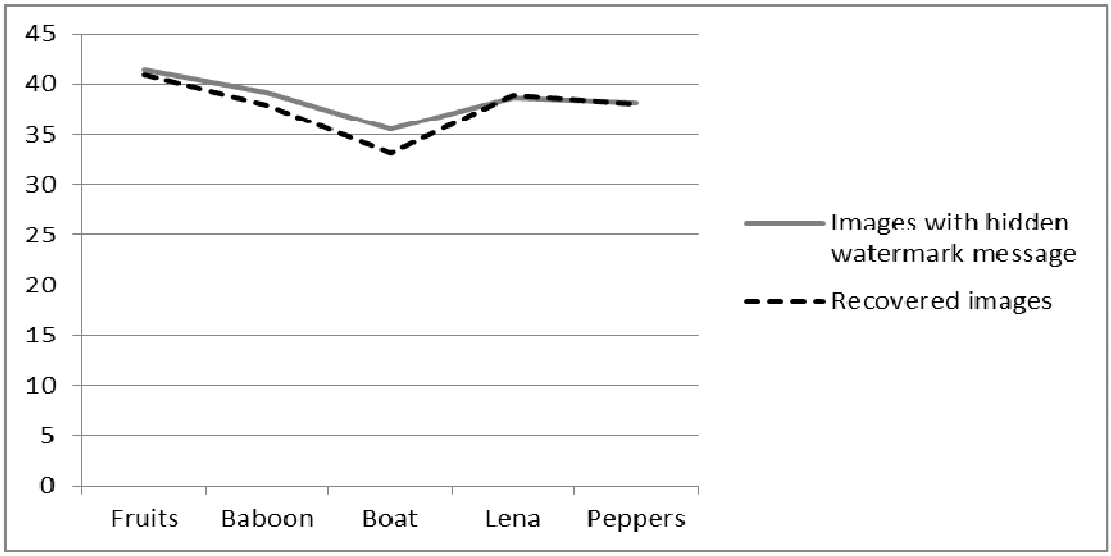
RESULTS ANALYSIS

The hide watermarking distortion transferred to the original image during embedding process results in PSNR (peak signal-to-noise ratio) loss, presented in table 1. There are five typical images compared; in the column one is the PSNR between original image and the image with hide watermark hided. In the second column the PSNR is computed between recovered image $\boldsymbol{Q}$ and original image $\boldsymbol{I_0}$.

Image	Image with hide watermark [dB]	Recovered image (Q) [dB]
Fruits	41.4	40.9
Baboon	39.2	37.9
Boat	35.5	33.2
Lena	38.7	38.9
Peppers	38.1	38.0

Table 1. PSNR values on images with hide message (column 1) and recovered image (column 2).

The analysis denotes a good quality image on output of recovering process, showing that the visual watermark is pretty complete compensated (graphic 1).
Anyway, more significant in any visual inspection is the quality in terms of HVS (human visual system) perception, which shows identical recovered image with the original one.



Graphic 1. The comparaison between PSNR values of hidden watermark image and the recovered image for five pictures.

CONCLUSIONS

The proposed framework is able to cover the metadata weakness in image forgery and copyright protection. Whereas the metadata could be easy detached by the image, the robustness of the presented method is given by the hide and visual watermarking techniques that are together applied to protect against tampering, to detect the certified user and, finally, to eliminate the visual watermarking.

But the framework can change the watermark embedding function parameter for every frame of a stream and to send it to the receiver into hide message, embedded into image, that is extracted on the receiver side and applied with the inverse function of the embedding visual watermark. If the parameter is not the same like on the sender side, the visual watermark will not be properly compensated.

Comparing the results in quality terms, the compensated image resulted after watermark extraction has very good quality, practically the same as the original image, in HVS perception.

REFERENCES

[1] M Rogobete, L Răcuciu, "First and second order image statistics in specific image artifact detection", International Conference on Innovative Technologies, IN-TECH 2012

[2] M Rogobete, C Răcuciu, E. Rădoi "Original Methodology and Algorithm able to Identify Visible Noisy in Image and Video Stream", International Conference for Education and Creativity, 7th Edition, Bucharest, 2013

[3] M Rogobete, C Răcuciu, "Using Potential Field Analysis into Image Artifact Detection Field", Indian Journal of Research, May, 2014

[4] W Jiao, Y Fang, G He, "An Integrated Feature Based Method For Sub-Pixel Image Matching", The International Archives of the Photogrammetry, 2008 – Citeseer.

[5] Marius Rogobete, Ciprian Răcuciu - "Cryptographic Extension Key for Watermark Encoding" " Titu Maiorescu – 04.11.2014, International Conference for Education and Creativity

[6] Marius Rogobete, Ciprian Răcuciu - "Visual Watermark Embedded Functions" Titu Maiorescu – 04.11.2014, International Conference for Education and Creativity

[7] Marius Rogobete, Ciprian Răcuciu - "An Improved Cryptographic Method in Watermark Encoding", Indian Journal of Research, Volume IV, Issue III, March 2015

[8] "IPTC Standard Photo Metadata 2014" , www.iptc.org

[9] "Dublin Core Metadata for Resource Discovery - RFC 2413", www.ietf.org

[10] Peter Krogh," The DAM Book: Digital Asset Management for Photographers", O'Reilly Media Inc., 2009

[11] "Social Media sites: photo metadata test results", www.iptc.org/about-iptc/media-releases

[12] "Social Media Test Results", www.embeddedmetadata.org/social-media-test-results.php

[13] http://www.phmdc.org/index2014post.php

[14] "WIC Metadata Overview", https://msdn.microsoft.com