Digital Image Watermark

with Quaternion Fourier Transform

Dogaru Mihail-Danut

Facultatea de Matematică și Informatică
Universitatea din București

January 30, 2025

Overview

Introduction

As multimedia content has increased over the last few decades, the digital security of data and intellectual property has become a serious issue. Digital watermarking is a technique used for securely embedding information into digital media content such as: image, video, audio.

Usecases:

- healthcare
- digital assets
- document authentification

Our Objectives

- Embed a binary image into an RGB image
- The host image should look the same to the naked eye
- Make the embedding pattern difficult to detect by third-parties
- Minimize the error between the initial watermark and the extracted watermark

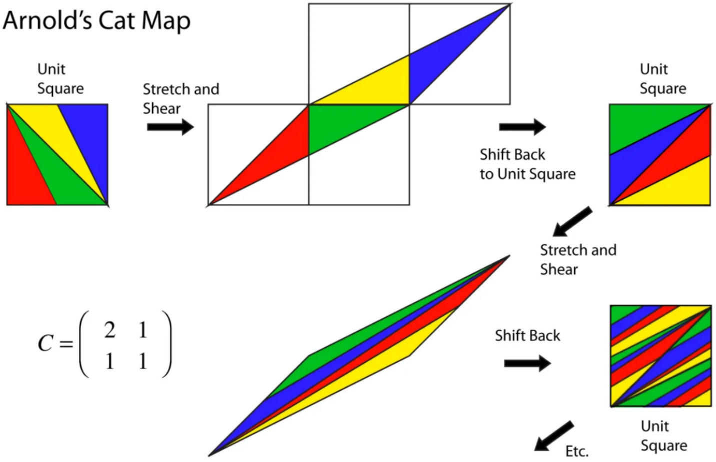
Preliminary - Arnold's Cat Map

To enhance security, we'll start by displacing the pixels of the watermark in a periodic pattern given by the transformation: $\Gamma : \mathbb{N}^2 \times \mathbb{N} \to \mathbb{N}^2$

$$\Gamma(\begin{bmatrix} x \\ y \end{bmatrix}, k) = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}^k \begin{bmatrix} x \\ y \end{bmatrix} (mod\ N)$$

where k is the security key for the watermark and N is the of the image. To decode the watermark, reapply the transform $P(N) - k$ times, where $P(N)$ is the period of an image of $N \times N$ pixels.

Preliminary - Arnold's Cat Map Visualisation



https://galileo-unbound.blog/2019/06/16/vladimir-arnolds-cat-map

Preliminary - Quaternions

Quaternions are four-dimensional hyper-complex numbers defined in the space $\mathbb{H}$, applied to mechanics in three-dimensional space defined as

$$a + bi + cj + dk$$

where $a, b, c, d \in \mathbb{R}$ and $1, i, j, k$ are vector basis on $\mathbb{H}$ with the following multiplicative laws:

$$ij = -ji = k, \;\; jk = -kj = i, \;\; ki = -ik = j, \;\; i^2 = j^2 = k^2 = -1$$

Preliminary - Quaternions for Images

Quaternions can be used for representing color images for multiple color spaces such as *RGB, YCbCr, CIE 1931* and so forth. For example an RBG pixel will be described as follows:

$$Image(x, y) = R(x, y)i + G(x, y)j + B(x, y)k$$

where *Image(x,y)* represents the pixel at row x and column y and *R/G/B(x,y)* is the R, G or B value of pixel (x, y).

Note that the real part is zero.

Preliminary - Fast Quaternion Fourier Transform

As shown in the work of S. Jiang et al. [Jiang, 2008] there is a fast algorithm for computing the 2D Quaternion Fourier Transform using the regular 2D *Fast Fourier Transform(FFT)*

$$\begin{aligned} F(u,v) = {} & i\big(\mathfrak{Re}(R_{RFT}) + \mu\mathfrak{Im}(R_{RFT})\big) \\ & + j\big(\mathfrak{Re}(G_{RFT}) + \mu\mathfrak{Im}(G_{RFT})\big) \\ & + k\big(\mathfrak{Re}(B_{RFT}) + \mu\mathfrak{Im}(B_{RFT})\big) \end{aligned}$$

where $\mathfrak{Re}$ denotes the real part, $\mathfrak{Im}$ the imaginary part and $R/G/B_{RFT}$ the regular Fourier Transform for each color channel.

Preliminary - Fast Quaternion Fourier Transform

Let $A(u, v)$ be the real part of the Quaternion Fourier Transform and $B(u, v)$, $C(u, v)$, $D(u, v)$ the three imaginary parts. We can rewrite the formula as such:

$$F(u, v) = A(u, v) + iB(u, v) + jC(u, v) + kD(u, v)$$

with the inverse:

$$\begin{aligned} f(m, n) = &\big(\mathrm{Real}(A_{IRFT}) + \mu \cdot \mathrm{Imag}(A_{IRFT})\big) \\ &+ i\big(\mathrm{Real}(B_{IRFT}) + \mu \cdot \mathrm{Imag}(B_{IRFT})\big) \\ &+ j\big(\mathrm{Real}(C_{IRFT}) + \mu \cdot \mathrm{Imag}(C_{IRFT})\big) \\ &+ k\big(\mathrm{Real}(D_{IRFT}) + \mu \cdot \mathrm{Imag}(D_{IRFT})\big) \end{aligned}$$

where $A/B/C/D_{IRFT}$ is the regular Inverse Fourier Transform for each component.

Preliminary - QFT Over an Image

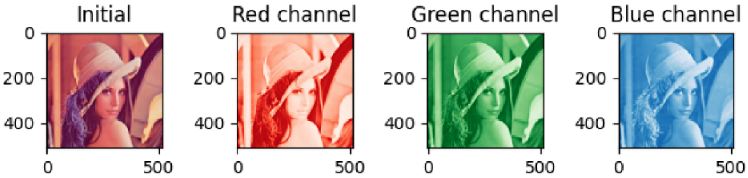


Figure: RGB Color channels separation for Lena.

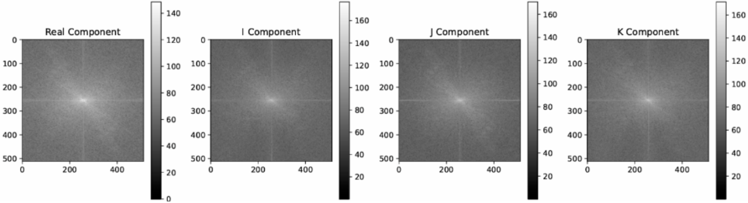


Figure: QFT spectrum components of Lena

Methodology - Constraints

- The watermarked binary image W must have a shape $N \times N$, with $N = 2k$, where $k \in \mathbb{N}$.
- The host RGB image I must have a shape $M \times M$, with $M = 8k$, where $k \in \mathbb{N}$.
- $M \geq 4N$.

Methodology - Watermark

Firstly, the pixels of W will be displaced by applying Arnold's Cat Map with an arbitrary $k \in \mathbb{N}$ that will be considered the *security key* of W, giving the new binary image W'.

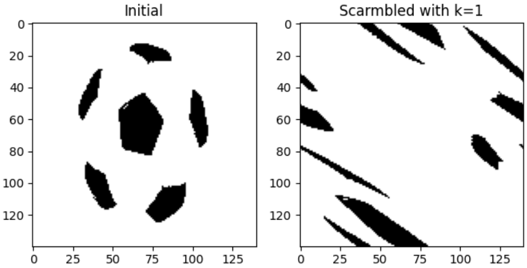


Figure: One scramble of an image.

Methodology - Watermark

Secondly, W' will be broken down into a block of size 2×2 resulting in the set of blocks $W_k = \{w_k(i,j) \mid i,j \in \{0,1\},\ k \in \{1,2\ldots,\frac{N^2}{4}\}\}$

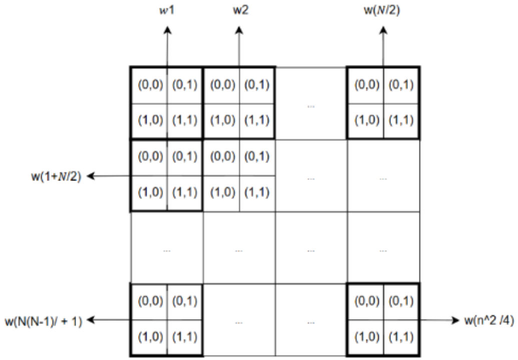


Figure: Scrambled watermark image blocks structure

Methodology - Host

Similarly, the host image H will be broken down into blocks of size 8×8 in the same way as the watermark resulting in a set of blocks
$H_k = \{h_k(i,j)|\ i,j \in \{0,1,\ldots 8\},\ k\ \in \{1,2\ldots,\frac{M^2}{64}\}\}.$

The Fast Quaternion Fourier Transform will be applied on each block of H_k, each resulting in one real part A_k and three imaginary parts B_k, C_k, D_k.

Methodology - Embedding Scheme

Each block of W_k corresponds to at least one block from H_k. The block W_k will be embedded in A_k as follows:

$$\begin{cases} a'_k(1,1) = sgn(w_k(0,0)) * \alpha \\ a'_k(1,2) = sgn(w_k(0,1)) * \alpha \\ a'_k(2,1) = sgn(w_k(1,0)) * \alpha \\ a'_k(2,2) = sgn(w_k(1,1)) * \alpha \end{cases}$$

where $|\alpha| > 10$. After reapplying qft, the a'_k values will contain noise from conversions and 10 is large enough in order to preserve it's sign that will later be use for extracting the watermark.

Methodology - Embedding Scheme

To maintain symmetry and ensure the pure quaternion form is correct before applying the inverse, we must also adjust the coefficients symmetrically corresponding to those modified in the previous step.

$$\begin{cases} a'_k(6,6) = -a'_k(2,2) \\ a'_k(6,7) = -a'_k(2,1) \\ a'_k(7,6) = -a'_k(1,2) \\ a'_k(7,7) = -a'_k(1,1) \end{cases}$$

Methodology - Embedding Scheme

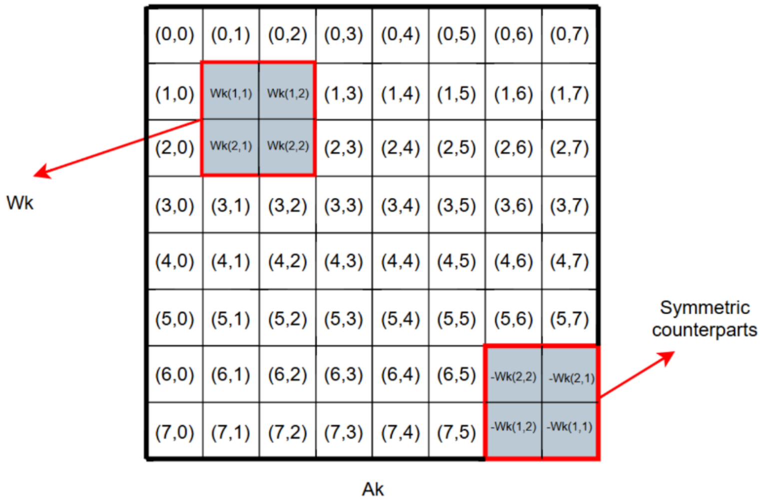


Figure: Embedding of Wk into Ak

Methodology - Reconstruction

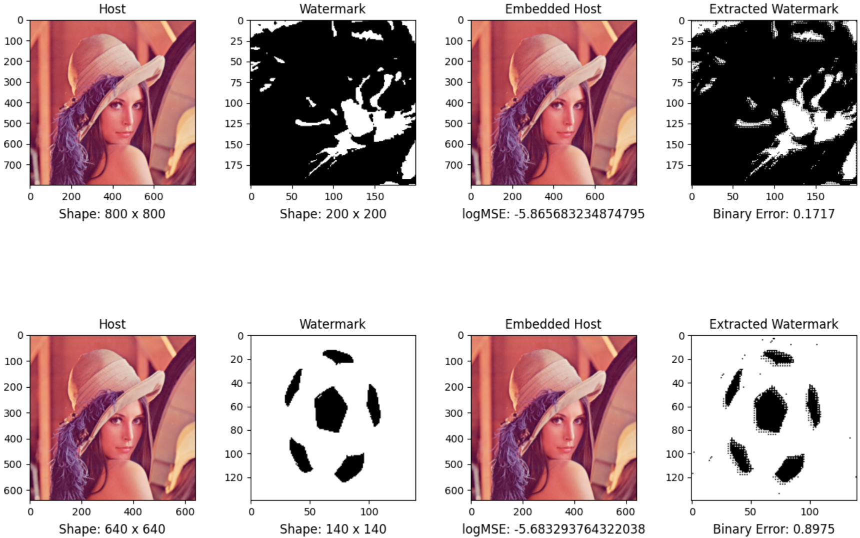
Take each real coefficient components A_k with its corresponding imaginary components B_k, C_k, D_k and perform the Fast Quaternion Inverse Transform.

Lastly, join back all the new blocks generated by the inverse transform, in the shape of the initial image W ($M \times M$) and convert the new quaternion values into RGB.

Methodology - Extraction

- Deconstruct the image into blocks again and apply QFT on each block
- From the real part, check the sign of the elements from the embedded positions $(- \rightarrow 0, + \rightarrow 1)$
- Construct a 2×2 matrix with the 0's and 1's extracted from each block.
- Join together all the new blocks into an image
- De-scramble the watermark according to the key

Experiments



Conclusions

In conclusion, digital watermarking has proven to be a safeguarding technique for multimedia data. With advancements in technology and evolving security challenges, digital watermarking will play an important role in protecting digital content.

The method presented above has proven it's efficiency in hiding and restoring binary images onto RGB color images if the shape requirements are being fulfilled.

Libraries

- NumPy - For mathematical collections and operations
- quaternions - A quaternion dtype over NumPy
- skimage - For metrics over images
- cv2 - For image basic processing
- matplot - For plottings

References

Hitzer, E. M. (2007).
Quaternion fourier transform on quaternion fields and generalizations.
Advances in Applied Clifford Algebras, 17:497–517.

Jiang, S. (2008).
Novel algorithms for fast hypercomplex fourier transform and hypercomplex correlation with applications.
Acta Electronica Sinica, 36(1):100.

Moxey, C., Sangwine, S., and Ell, T. (2002).
Color-grayscale image registration using hypercomplex phase correlation.
In *Proceedings. International Conference on Image Processing*, volume 2, pages II–Ii.

Wang, X.-y., Wang, C.-p., Yang, H.-y., and Niu, P.-p. (2013).
A robust blind color image watermarking in quaternion fourier transform domain.
Journal of Systems and Software, 86(2):255–277.