Journal of *Imaging*

MDPI

Article

Image Fragile Watermarking through Quaternion Linear Transform in Secret Space

Marco Botta [1,*], Davide Cavagnino [1] and Victor Pomponiu [2]

[1] Dipartimento di Informatica, Università degli Studi di Torino, 10149 Turin, Italy; davide@di.unito.it
[2] Agency for Science, Technology and Research, 1 Fusionopolis Way, 487372 Singapore, Singapore; victor.pomponiu@gmail.com
* Correspondence: marco.botta@unito.it; Tel.: +39-011-670-6789

Received: 14 June 2017; Accepted: 5 August 2017; Published: 11 August 2017

Abstract: In this paper, we apply the quaternion framework for color images to a fragile watermarking algorithm with the objective of multimedia integrity protection (Quaternion Karhunen-Loève Transform Fragile Watermarking (QKLT-FW)). The use of quaternions to represent pixels allows to consider the color information in a holistic and integrated fashion. We stress out that, by taking advantage of the host image quaternion representation, we extract complex features that are able to improve the embedding and verification of fragile watermarks. The algorithm, based on the Quaternion Karhunen-Loève Transform (QKLT), embeds a binary watermark into some QKLT coefficients representing a host image in a secret frequency space: the QKLT basis images are computed from a secret color image used as a symmetric key. A computational intelligence technique (i.e., genetic algorithm) is employed to modify the host image pixels in such a way that the watermark is contained in the protected image. The sensitivity to image modifications is then tested, showing very good performance.

Keywords: data hiding; fragile watermarking; image authentication; color image processing; quaternions; genetic algorithm (GA); Karhunen-Loève Transform (KLT)

1. Introduction

The protection of digital media is one fundamental topic in the present age, in which practically every kind of content is represented in digital form. Without an integrity guard system, the transmission via open and unsecured networks of digital assets could not be verified. Researchers have developed and are still devising various techniques to solve the problem. For example, digital signature is a method to ensure authenticity and proof of origin for a digital media; Message Authentication Codes are another method to authenticate the integrity of a digital media for a restricted set of entities. Both methods require appending a certain amount of information to the protected digital object.

Another effective solution to defend digital objects from various attacks is digital watermarking [1]. Watermarking techniques insert a signal in the digital object itself with various purposes: content authentication, content integrity, copyright protection, traitor tracing, etc.

Depending on the application requirements, various watermarking methods (not necessarily excluding one other) have been devised, every one having specific properties. We briefly recall these characteristics in the following.

A watermarking algorithm may be *robust* or *fragile*: the first kind is intended to survive (intentional) modifications of the digital object aimed at its removal (while maintaining an acceptable quality of the resulting object); fragile watermarks have the opposite purpose: being destroyed at the minimal modification of the digital object, and possibly localizing the modified area. Therefore, robust watermarks are useful for copyright protection and track of origin, whilst fragile ones may be used for authentication and integrity check purposes. Some fragile watermarks have been devised

to accept minimal modifications (like high quality JPEG compression of an image), and are thus called semi-fragile; others have the ability, after detecting an altered area, to (partially) restore the original object.

Another property of watermarking methods is the capacity to recover the original (host) object from the watermarked one by authorized entities: an algorithm that possesses this ability is called *reversible*, and *non-reversible* (or *lossy*) otherwise.

The present work is devoted to fragile watermarking of color images, thus in the following we will restrict the subject to this type of data only.

The watermark signal may be inserted mainly into two different domains, namely the *spatial* and the *frequency domains*. The spatial domain refers to the pixels of the image, so the watermark signal is embedded by modifying the pixel values. On the contrary, the frequency domain generally refers to a transformed space, like the Fourier (DFT), Discrete Cosine (DCT), Karhunen-Loève (KLT) or Singular Value Decomposition (SVD), into which the pixels are projected producing a set of features (typically coefficients) used for watermark embedding. Embedding into frequency domains requires the inverse step to obtain pixels again: we note that some works ignore the possibility that the inverse operation returns floating point pixel values and the necessary rounding to integer values may, in general, remove part or all of the watermark. We already developed [2] and optimized a methodology based on Genetic Algorithms (GAs) to overcome this problem, using a modification of the approach suggested in [3]. The use of the KLT is peculiar with regard to other linear transforms like DFT and DCT because the kernel used by the KLT is not fixed but is instead computed from a set of samples (in our case, a set of subimages obtained from a secret key image): this allows the creation of a secret space of features that provide the necessary security for hiding a watermark signal.

In this paper, we consider bitmap images coded in RGB format having a bit-depth of 8 bits per channel. The objective is to adapt the methodology of a previously developed *fragile watermarking algorithm* for gray scale images to color images considering the holistic representation of the color information: we believe that the quaternion framework is a powerful representation, which could offer the optimal interplay between the robustness and imperceptibility components of the watermarking scheme and also improves on known ones [4]. In particular, [5] defines the computation of eigenvectors and eigenvalues for a set of color pixels, and proposes the Quaternion Karhunen-Loève Transform (QKLT). We think that the QKLT is a valuable tool that can be seemly incorporated into a fragile watermarking framework since the color information is processed as a whole and not as three independent channels, easing the protection of the integrity of an image even using a reduced number of watermark bits with regard to other approaches. The QKLT gives a formalization for the KLT of a color image combining the RGB values of a pixel. Moreover, the use of the QKLT defines a single approach in computing the secret space where the watermark is inserted, *differently* from the methods used in [4] which consider the three color channels as separate entities allowing many different KLT basis computation approaches that, even if correct and with very good performance, lack a motivation for the choice of one with regard to the other and lead to empirical combinations of the vectors of the three channels.

Thus, the main contribution of this work is the development of a fragile watermarking algorithm for color images which employs the channel color information as integrated producing very high quality images with a high level of security in detecting unauthorized modifications.

The quaternion framework we propose has been integrated into the modular watermarking architecture [4] by developing a new module that computes the suitable features for the watermarking process and brings improvements on various directions, as it will be detailed in the experimental results section. In particular, the use of quaternions allows to keep the transform space dimension limited to n^2 instead of $3n^2$ (where n^2 is the number of pixels per subimage) because quaternions incorporate the whole color information instead of keeping it separate and requiring to consider vectors three times bigger to take into account the correlation among color channels. The resulting algorithm

has smaller running times and a higher sensitivity in detecting tampering, while maintaining the same high quality of the watermarked images as the other approaches presented in [4].

The main characteristics, novelties and improvements of the presented algorithm with regard to previous fragile watermarking algorithms are:

(a) representation of color pixels using quaternions, which translates in integrating the color information employing a linear transform that considers this *holistic* interpretation, and embedding with a *smaller impact* on the host image;
(b) *improved* running times for watermarking embedding with regard to a previous algorithm developed in the same framework;
(c) *increased* sensitivity to attacks;
(d) *very high quality* of the watermarked images, along with a good and *flexible localization potential* (satisfying almost all application contexts);
(e) *flexibility* in payload embedding for a chosen localization capability;
(f) application of the QKLT to the fragile watermarking domain.

The paper is organized as follows: firstly, we survey some works on the representation of color images in the quaternion framework and on image watermarking. Then, with the aim of making the paper as self-contained as possible, we devolve a section to briefly recall some basic concepts on quaternions, and afterwards we present the QKLT. The section on the watermarking algorithm QKLT-FW (Fragile Watermarking) presents the steps for embedding and extracting the watermark with the aim of detecting and localizing alterations to the image. Then, experimental results are shown along with a comparison with existing algorithms. In the final section we draw some conclusions on the developed algorithm and discuss the obtained performance.

Regarding notation, in this paper, we represent matrices, vectors and scalars by capital letters, bold lowercase letters and plain lowercase letters, respectively. All vectors are column-wise by default.

2. Related Works

2.1. Quaternion Signal Processing

In the last decade, quaternion signal processing (QSP) has started to be widely employed and several common signal processing transforms have been extended to the quaternion domain. For instance, the quaternion Fourier transform (QFT) was firstly introduced by Sangwine [6] and later extended with new results [7].

Other works proposed descriptors for the quaternion Singular Value Decomposition (QSVD), the quaternion Eigenvalue Decomposition (QEVD) and the QKLT.

The calculation and properties of the SVD for quaternion matrices (generated by vector-array signals) were extensively studied in [8].

According to the previous works, the main purpose of using the quaternion counterpart of these tools is that it can naturally account for the correlation among color channels, providing a holistic representation [9] of color images. Thus, the quaternion theory treats a color image as a vector field and processes it directly, without losing color information.

The paper [5] is the basis on which our work is founded. Le Bihan and Sangwine present the SVD, the Eigenvalue Decomposition and the KLT for quaternion matrices applied to color images: they call them QSVD, QEVD and QKLT respectively. Their work also refers to many previous papers on the topic of quaternion matrices.

An application where quaternion representation is finding an active field of research is digital watermarking. The next subsection of this paper will review the most representative and recent works devoted to quaternion-based watermarking.

2.2. Quaternion-Based Watermarking

The first application of quaternions for robust digital watermarking was within the Fourier framework. In particular, [10] applies the QFT defined in [11] to perform robust image watermarking; given that the QFT depends on a parameter μ (which is a pure quaternion satisfying $\mu^2 = -1$), a study aimed at finding the best μ value to achieve both invisibility and robustness to attacks is described in that paper. In [12], the watermarking algorithm inserts a robust watermark into the scalar part of some selected QFT coefficients, and the detection stage deals with attacks through a least squares support vector machine applied to pseudo-Zernike moments of the scalar QFT matrix of the possibly attacked image.

An et al. [13] also used the QFT for devising a robust watermarking scheme. To be able to hide large payloads, i.e., the number of features allocated for watermark embedding, an improved quantization index (QIM) algorithm was proposed to compress the watermark bits. Due to the use of QIM, the scheme is able to extract the watermark without the use of the host image, which greatly increases its applicability. The simulations carried out prove that the watermarking algorithm based on QFT and the improved QIM with distortion compensation attains a good tradeoff between invisibility and robustness.

Later, Tsui et al. [14] developed a pair of watermarking algorithms working in the La*b* space. The first one applies the Spatio-Chromatic Discrete Fourier transform (SCDFT) to the a* and b* pixel values, then a binary watermark is inserted into the yellow-blue component, maximizing the watermark intensity and keeping the distortion below a Just Noticeable Difference level. The second one embeds a quaternion watermark into the QFT coefficients of the image, taking into account the Human Visual System (HVS) characteristics.

To overcome the limitation of the previous works which spread the watermark information over a limited number of RGB color channels, Chen et al. [15] employ a full 4D discrete QFT (QDFT) of the host color image. In their complete framework, which introduces three schemes, they provide the symmetry preconditions of the unit quaternions necessary for the QDFT in order to achieve the correct watermark extraction in the case of no-attacks. The experimental results show that the proposed framework offers a good performance in terms of capacity and robustness against attacks. However, the imposed preconditions for the unit pure quaternion affects the payload of the watermark, i.e., the number of features allocated for watermark embedding. Furthermore, the study lacks a theoretical analysis of the probability of false detection and a thorough comparison with the existing works.

In [16], Shao et al. have explored a joint robust watermarking and encryption technology based on the quaternion gyrator transform (QGT) [17] and double random phase encoding (DRPE). The main idea is to encrypt the watermark via DRPE and then to insert the encrypted bits into the mid-frequency coefficients of the QGT of the host image. It is important to note that the scheme requires some side information, related to the host image, in order to recover the watermark.

The Quaternion Polar Harmonic Transform (QPHT) has been used in [18] to devise a robust watermarking scheme in order to increase the security of the watermark information. The transform is a parameterized version of the linear canonical transform with the parameters belonging to the real special linear group defined as the set of 2×2 real matrices having the determinant equal to 1. Due to the large space where the correct parameters for the forward and backward transform are lying, the proposed scheme has a high level of security. Moreover, the scheme shows satisfactory performance in terms of robustness, capacity and imperceptibility of the watermark.

Yang et al. [19] also apply the quaternion algebra and Polar Harmonic Transform (PHT) to introduce an invariant color image watermarking scheme. The selection of PHT was motivated by its appealing properties compared to others moment-based transforms, e.g., (pseudo) Zernike moments: noise robustness, low computational complexity, better reconstruction accuracy and numerical stable solutions. An in-depth analysis of invariance properties (rotation, scaling and translation) of the QPHT moments is given in the paper along with the criterion used for the selection of the watermarking

coefficients. More precisely, only the set of coefficients that offer the highest reconstruction accuracy are chosen by using the following relation

$$S = \left\{ M^R_{n,l} \text{ and } M^L_{n,l},\ l = 4m, m \in \mathbb{Z} \right\}, \tag{1}$$

where $M^R_{n,l}$ and $M^L_{n,l}$ denote the QPHT right and left coefficients respectively, n is the order and l is the repetition parameter. The encrypted watermark bits are inserted by adaptively modulating (via quantization) the selected coefficients. For instance, for the right coefficients the embedding rule is

$$\left| M'^R_{n,l} \right| = \begin{cases} 2\Delta \cdot \text{round}\left(\frac{|M^R_{n,l}|}{2\Delta} \right) + \frac{\Delta}{2},\ w_k = 1 \\ 2\Delta \cdot \text{round}\left(\frac{|M^R_{n,l}|}{2\Delta} \right) - \frac{\Delta}{2},\ w_k = 0 \end{cases} \quad (0 \leq k < P \times Q)\ , \tag{2}$$

where $|\cdot|$ denotes the modulus operator, round$(\cdot)$ represents the round operator, Δ is the watermark strength factor, w_k is the bit of the watermark of size $P \times Q$. One of the drawbacks of the scheme is the inability to fully extract the watermark from the watermarked coefficient $\left| M'^R_{n,l} \right|$ since the QPHT coefficients can be obtained approximately for a digital image.

2.3. Image Authentication

The problem of image authentication has been also addressed by Al-Otum [20]. The paper proposes a semi-fragile watermarking scheme based on the DWT. The watermark, i.e., the authentication information, is implanted into the DWT coefficients of all image blocks of a color channel, randomly chosen. In order to better capture the characteristics of the image, a modified DWT (a reminiscent of the wavelet packet decomposition [21]) is used to compute the approximation of the horizontal, vertical and diagonal components. The method is semi-blind since it requires some auxiliary information (i.e., quantization thresholds are computed and passed to the detector) in order to extract the watermark, which limits its applicability. A security issue with this scheme is that for each block the authentication value, i.e., weighted mean values of the difference-color features, is computed only from the approximation horizontal and vertical components, ignoring the diagonal coefficients.

Besides detecting whether it has been tampered by common signal processing operations, the Lin et al. [22] scheme adds the recovery functionality of the affected image regions. To achieve these goals, they make use of several tools chosen to meet the requirements of watermarking schemes: lattice-based embedding into the DCT coefficients to lower the impact of the hidden data and a secret sharing approach to reconstruct the watermark with recovery capability.

In [23], a watermarking method is introduced which authenticates the compressed indexed representation of a color image. The authentication watermark is embedded into the LBSs of indexes of the compressed color images. To overcome the issue that arises when the modified index LSB coincides with the watermark bit, the scheme adopts an improved tamper verification procedure which consists of introducing interdependency relationships among pixels in each row or column.

In [24], the authors exploit the standard deviation information to devise an authentication method. Two sorts of information are embedded into the image: an authentication watermark and some image information that enables the recovery of tampered blocks. The two watermarks follow different insertion procedures. The authentication bits are just inserted into the LSBs of the image. To insert the recovery data, the scheme proceeds by firstly using the standard deviation to classify the image blocks into three classes. Afterwards, each block is prepared for embedding by mapping it to the DCT domain followed by quantization. Interestingly, the amount of information to be embedded in each block is adaptively modified and is determined by its class.

In the following section, we briefly introduce two algorithms for the authentication of images, which will be used to perform a comparison with our method (these algorithms do not make use of quaternion representations, but have very high Peak Signal-to-Noise Ratio (PSNR) and sensitivity).

A secure method for fragile image watermarking was introduced in [25]. The algorithm contrasts Vector Quantization attacks, may detect tampering and authenticates an image. The image is subdivided into a hierarchy of blocks. The LSBs of a block contain authentication information (a Message Authentication Code, MAC, or a signature) of the block itself and part of the MAC (or signature) of the upper layers in the hierarchy obtained by merging the block with other blocks (like in a quadtree decomposition). Thus, a block not in the lowest level of the hierarchy is authenticated by bits contained in the LSBs of the blocks at the lowest level composing it. This algorithm has a 100% tamper detection capability, but suffers a low PSNR due to the large quantity of data necessary to store a MAC or a signature.

MIMIC 9 is a modular framework developed for the fragile watermarking of color images. It embeds a watermark in a transformed secret space (defined by the Karhunen-Loève transform), using several different embedding techniques, such as LSB embedding and syndrome coding.

3. Quaternions

Quaternions are a representation of numbers in a hypercomplex space. A quaternion q is defined in a four-dimensional *vector space* $\mathbb{H}$ as $q = s\vec{\mathbf{1}} + v_i\vec{\boldsymbol{i}} + v_j\vec{\boldsymbol{j}} + v_k\vec{\boldsymbol{k}}$, where the basis of the vector space is composed by the vectors $\vec{\mathbf{1}}$, $\vec{\boldsymbol{i}}$, $\vec{\boldsymbol{j}}$, $\vec{\boldsymbol{k}}$. The number s is called the scalar part and $\vec{\boldsymbol{v}} = v_i\vec{\boldsymbol{i}} + v_j\vec{\boldsymbol{j}} + v_k\vec{\boldsymbol{k}}$ constitutes the vector part, so a quaternion is also written as $q = (s,\ \vec{\boldsymbol{v}})$. The basis vectors have the property that

$$\vec{\boldsymbol{i}}^{2} = \vec{\boldsymbol{j}}^{2} = \vec{\boldsymbol{k}}^{2} = -\vec{\mathbf{1}}, \tag{3}$$

which has several implications, like the non-commutativity of multiplication. The four operations are performed in the usual way of vector spaces, taking into account the previous property. When $\vec{\boldsymbol{v}} = \vec{\mathbf{0}} = (0,\ 0,\ 0)$ the quaternion is said *scalar*, whilst when $s = 0$ it is called *pure*.

The L_2 norm (or simply the norm) of a quaternion $q = (s,\ \vec{\boldsymbol{v}})$ is $\| q \| = \sqrt{s^2 + v_i^2 + v_j^2 + v_k^2}$ and the L_1 norm is $\| q \|_1 = |s| + |v_i| + |v_j| + |v_k|$.

The multiplicative inverse of a quaternion q is computed as $q^{-1} = \bar{q}/\| q \|^2$ where $\bar{q} = (s,\ -\vec{\boldsymbol{v}})$ is the complex conjugate of q. Due to the non-commutative property of multiplication, it is possible to divide a quaternion a by q in two possible ways, namely aq^{-1} and $q^{-1}a$.

A very good introduction to quaternions is [26]. In the following, we will use matrices having quaternion elements. An overview on this topic can be found in [27].

Quaternions and Color Images

Following the use of quaternions to represent and process color image pixels introduced in [28], a color pixel expressed in the RGB space by the triple (r, g, b) is represented by the pure quaternion $(0,\ r\vec{\boldsymbol{i}} + g\vec{\boldsymbol{j}} + b\vec{\boldsymbol{k}})$, so every pixel is considered as an element of $\mathbb{H}$.

4. The Quaternion Karhunen-Loève Transform

The general definition of a linear transform is a mapping between different bases of vector spaces. Given a d-dimensional column vector $\boldsymbol{x}$, a linear transform is defined by a $d \times d$ orthonormal matrix Φ (called kernel, whose rows form a vector basis) that maps $\boldsymbol{x}$ to $\boldsymbol{y}$, i.e., the same vector expressed in a different basis, by means of the relation (forward transform) $\boldsymbol{y} = \Phi\boldsymbol{x}$. The vector $\boldsymbol{x}$ may be again obtained from $\boldsymbol{y}$ by means of the inverse transform $\boldsymbol{x} = \Phi^{-1}\boldsymbol{y}$, that given the orthonormality of Φ may be also written as $\boldsymbol{x} = \Phi'\boldsymbol{y}$ (where Φ' is the conjugate transpose of Φ). Differently from some common transforms which have fixed kernels for a fixed d, like Discrete Cosine, Fourier, Hadamard, and Haar transforms, the discrete Karhunen-Loève Transform, KLT, (also known as Hotelling Transform, Principal Component Analysis, or Eigenvector Transform) computes a kernel from a set of vectors.

The ability of the KLT is to orient the basis Φ along the d directions of maximum data expansion of the vectors used to compute the kernel.

The QKLT is based on the Quaternion Eigenvalue Decomposition (QED) [5].

An eigenvector associated to a matrix $\boldsymbol{A}$ is defined as a vector $\boldsymbol{x}$ which multiplied by the matrix results in a multiple of the vector itself, i.e., in general $\boldsymbol{Ax} = \lambda\boldsymbol{x} = \boldsymbol{x}\lambda$, where λ is the associated eigenvalue. But in $\mathbb{H}$ the product is not commutative, so two possible kinds of eigenvectors can be defined, namely left eigenvectors/eigenvalues for which $\boldsymbol{Ax}_L = \lambda_L\boldsymbol{x}_L$ and right eigenvectors/eigenvalues satisfying $\boldsymbol{Ax}_R = \boldsymbol{x}_R\lambda_R$. As stated in [5] and papers cited therein, left eigenvalues pose many theoretical problems, so in our work we will use right eigenvectors/eigenvalues only.

It is possible to compute the right eigenvalues and their associated eigenvectors of a quaternion matrix $\boldsymbol{A}$ with the decomposition $\boldsymbol{A} = \boldsymbol{VEV}'$ as stated in [5], where the columns of $\boldsymbol{V}$ contain the eigenvectors and $\boldsymbol{E}$ is a diagonal matrix containing the eigenvalues. If $\boldsymbol{A}$ is hermitian (i.e., $\forall a_{ij} \in \boldsymbol{A}$ at row i and column j, $a_{ij} = \overline{a_{ji}}$), as is the case with the covariance matrix we will compute on a color image, then the eigenvalues are real, i.e., $\lambda_R \in \mathbb{R}$ (in general, instead, $\lambda_R \in \mathbb{C}$).

Consider a set of column vectors $\boldsymbol{U} = \{\boldsymbol{x}_i\}$, compute the average vector, $\boldsymbol{m_U} = E[\boldsymbol{x}_i]$ where $E[\cdot]$ is the expected value operator, and successively the Hermitian covariance matrix $\boldsymbol{C_U} = E\left[(\boldsymbol{x_i} - \boldsymbol{m_U})(\boldsymbol{x_i} - \boldsymbol{m_U})'\right]$. Decomposing $\boldsymbol{C_U}$ as $\boldsymbol{C_U} = B\ \Gamma\ B'$ returns, as previously stated, the eigenvalues in the diagonal of Γ and the associated eigenvectors forming an *orthonormal basis* as columns of B. It is useful to give an ordering to the eigenvectors: sorting in non-increasing order of the eigenvalues' norm results in moving in the first positions the eigenvectors having, on average, more importance in the reconstruction of the vectors in $\boldsymbol{U}$. The KLT (or QKLT, if the vectors' components are quaternions) of a vector $\boldsymbol{z}$ (of size d) may then be written as $\boldsymbol{y} = B^T(\boldsymbol{z} - \boldsymbol{m_U})$, where the transpose operator ^{T} moves the eigenvectors in rows: the d components of $\boldsymbol{y}$ are called coefficients of the transform and the position in $\boldsymbol{y}$ is called *order* of the coefficient. Obviously, the inverse QKLT is computed as $\boldsymbol{z} = \left(B^T\right)^{-1}\boldsymbol{y} + \boldsymbol{m_U}$. A more extensive discussion of the QKLT is presented in [5] and references cited therein.

5. Genetic Algorithms

A GA is a method of computation inspired by the evolution of living beings.

When the parameters of a problem may be encoded in a data structure (called individual) and a function exists to evaluate the quality of an individual in representing a solution to the problem, then a GA may be employed. The GA explores the space of the possible outcomes evolving a population of individuals (initially randomly created, as in real evolution) evaluating them through a *fitness function*, and reproducing the best ones to converge to an individual that represents a (local) optimal solution.

The GA executes a cycle for a maximum number of times (each iteration is called *epoch*) or until a viable solution is found, according to the following high level Algorithm 1:

Algorithm 1: GENETIC EVOLUTION

```
INITIALIZE POPULATION
solution = ϕ
e = 0
best fitness = THRESHOLD
while best fitness ≥ THRESHOLD and e < MAX_EPOCHS
        REPRODUCE POPULATION
        EVALUATE FITNESS OF EVERY INDIVIDUAL
        IF best fitness < THRESHOLD THEN
          solution = individual having best fitness
        ELSE e = e + 1
        UPDATE POPULATION
RETURN solution
```

The population is evolved by first reproducing the individuals and then picking the ones that will be part of the population for the next epoch.

Reproduction operates by selecting pairs of individuals (we made this with tournament selection, in which two pairs are chosen, and the individual with the best fitness in each pair is selected) and applying with probability p_c a crossover operator, which exchanges random subsets of genes in corresponding positions. The resulting individuals have a probability p_m to have a mutation of one of its genes (this aims to create potentially better individuals avoiding to fall into local optima).

After reproduction has taken place, all the individuals are evaluated according to the fitness function to create the new population for the next epoch: strategies like tournament selection, total or partial replacement may be applied in this phase. Also, if an individual has a fitness below a pre-defined THRESHOLD then the cycle is terminated and the individual given as output (we assume that the smaller the fitness the better the individual).

6. The QKLT-FW Watermarking Algorithm

This section describes the various components of the watermarking algorithm. We call this algorithm Quaternion Karhunen-Loève Transform Fragile Watermarking (QKLT-FW).

The input to the algorithm are the host color image I_h to be watermarked, and a secret key, in the form of a color image I_k. The images, both in bitmap format, are divided into contiguous non-overlapping sub-images (or blocks) of size $n \times n$: if the dimensions of I_k are not a multiple of n then the remaining part ($<n$) is not considered, whilst for simplicity we assume that for I_h of size $N \times M$ both N and M are multiples of n leading to $N_B = N \times M/n^2$ blocks. The output of the algorithm is a fragile watermarked image I_f.

The idea of the whole method is to hide a key-and-host-image dependent binary watermark into a set of secret features (QKLT coefficients) defined by the key image, considering the *color pixels as pure quaternions*. The host image is divided into sub-images (of size $n \times n$) and a portion of the watermark is embedded in each of them (see Figure 1). The alteration of a sub-image of I_f will, with high probability (we will discuss this point in a following section), modify the part of the watermark embedded in it, allowing the detection and localization of the attack by simple comparison of the expected and extracted watermarks.

In the following, we describe the steps to be performed to watermark an image with QKLT-FW.

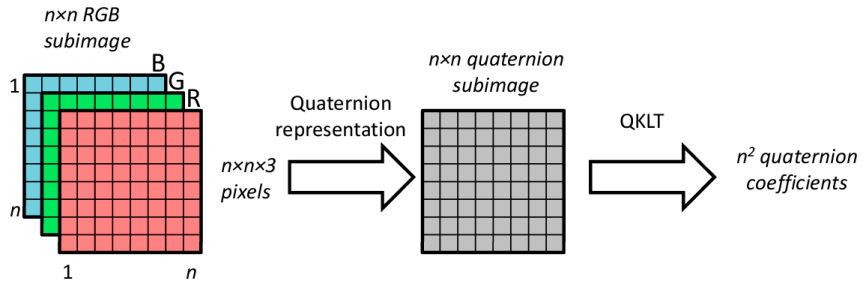

Figure 1. Quaternion representation of a sub-image and Quaternion Karhunen-Loève Transform (QKLT) coefficients derivation.

6.1. QKLT-FW Basis Generation

This step is performed by a unit that, receiving as input a key image I_k, divides it into non-overlapping blocks of size $n \times n$ and returns a QKLT orthonormal basis and an average sub-image as previously described: obviously the basis is composed of n^2 basis vectors each composed of n^2 quaternions, and the average sub-image is a vector of n^2 quaternions.

This computation may be performed only once for a set of images to be watermarked using the same key. Due to the dependence of the watermark on the host image, in principle a large amount of images may be watermarked with the same key without any possible leak on the computed secret basis.

6.2. QKLT-FW Integrity Data Generation

The secret bit string W used as watermark is made dependent on the host image to avoid transplantation and cut-and-paste attacks. It is obvious that this step must be executed for every host image to be watermarked with a particular key. Also, W should depend on the key (image), otherwise the bit string will not be secret (this is not strictly necessary, given the use of a secret space of embedding, but improves the whole security avoiding a possible search of embedded known bit strings); thus $W = S(I_h,\ I_k)$.

We implemented S (that may, and must, be public) as a procedure that:

- selects a set of pixels K_p in I_k in pre-defined positions;
- uses the values of these pixels to address a set of pixels H_p in I_h;
- applies the values of the pixels in H_p to address a set of pixels in I_k which in turn are used as seed of a cryptographic hash function (c.h.f.) like SHA-3;
- generates W by iteratively applying the c.h.f. until the required length is reached.

The pixels in the set H_p are not modified by the embedding algorithm because if only one of them is altered, a different watermark W_A will be computed during verification, leading to a completely altered image. Indeed, this is the result obtained from an attack that modifies a pixel in H_p: the localization property is lost but not the alteration detection.

If cropping is considered a possible attack (e.g., in case the protected images may be of any size), it is necessary to make the watermark dependent also on the image size (i.e., concatenating the height and the width of the image to the seed of the c.h.f.). A cropped (or enlarged) image will produce a different watermark during verification, thus many blocks will be flagged as forged: the localization will be lost but the attack will still be detected (effectively, cropping changes the relative position of a block with regard to borders).

6.3. QKLT-FW Embedding

To watermark a host image I_h composed of N_B sub-images, the algorithm inserts s watermark bits in every block of $n \times n$ color pixels (we call this payload s bits-per-block, or s bpb): thus, the previous step *Integrity data generation* will build a string W of size $s \cdot N_B$ bits.

To perform the insertion, different methods may be used. In the MIMIC framework [4], various embedding techniques were presented, but we briefly discuss only the two that are used here in conjunction with the QKLT:

- Bit Collect Module (BCM): s QKLT coefficients are selected to store each one a bit of the s watermark bits;
- Syndrome Coding Module (SCM): the s watermark bits per block are recorded as the syndrome of a word of r bits; these r bits are stored in r QKLT coefficients. Many possible codes may be used to perform syndrome coding, e.g., Hamming, Hadamard, Golay, BCH. See [4] for a deeper discussion on this topic.

In the present implementation, we chose to store one bit in one QKLT coefficient, having chosen a-priori the orders of these coefficients. From previous studies, we found that the order does not particularly influence the performance of the whole algorithm, so we presently use contiguous coefficients starting from the third (a key dependent choice is also an option, but we feel this a little improvement in security against an increased implementation complexity). A (quaternion) coefficient c is considered to carry the bit value b in position p (fixed a-priori) computed as

$$b = odd\big(int\big(2^{-p} \parallel c \parallel\big)\big), \tag{4}$$

where $\parallel c \parallel$ is the norm of the coefficient, *int* is a function that truncates a real number to its integer part, and *odd* is a function that returns 1 if its argument is odd and 0 otherwise.

In general, a sub-image does not already contain the required watermark string of s bits: the duty of the GA is to compute a modification Ξ of the sub-image such that it stores the correct bit string. The modification Ξ is, in the present case, a vector of n^2 components specifying the alterations to be applied to the RGB values of the n^2 pixels: we found that for an 8 bit-per-channel image, it is sufficient to have the set of possible alterations as small as $\{-3, -2, -1, 0, 1, 2, 3\}$, leading, as we will show, to a very good PSNR. Thus, the GA evolves a population of individuals each composed by n^2 genes: the absolute value of the alteration indexes the channel (i.e., 1 is red, 2 is green and 3 is blue) whilst the sign specifies if the pixel must be incremented or decremented by 1 (0 meaning no modification to the pixel). The fitness function of the GA takes into account two main properties of the resulting sub-image: the presence of the watermark and the PSNR with regard to the original. The GA runs for a maximum amount of epochs or until a viable solution is found.

When all the blocks have been processed by the GA, then the watermark has been embedded into I_h producing I_f.

We may summarize a high level behavior of the GA as:

- $I_f = I_h$
- For every sub-image Σ of I_f
    a. have a population of individuals representing modifications Ξ_i
    b. apply, in turn, every modification Ξ_i to Σ obtaining Σ_i
    c. compute the QKLT of every Σ_i and extract the bits according to (4)
    d. if, for some Σ_i, the watermark is stored and the PSNR is high (i.e., above a threshold) then save Σ_i in place of Σ and proceed to the next sub-image
    e. otherwise evolve the population Ξ_i and go to step b.
- Output I_f

6.4. QKLT-FW Extraction

This step uses the key image I_k and the watermarked (and possibly altered) image I_f. From I_k, the QKLT basis and the average sub-image are derived. Then, the QKLT is performed on the N_B blocks

of I_f, selecting the chosen coefficients and extracting s bits from every block (using BCM or SCM and Equation (4)). The extracted watermark W_E is the concatenation of the $s \cdot N_B$ bits.

6.5. QKLT-FW Verification

From both I_k and I_f the watermark bit string, W is computed as shown in the *QKLT-FW Integrity data generation* step: this string is compared with the one extracted in the *QKLT-FW Extraction* phase W_E. Differing bits in homologous positions mean an alteration in the corresponding sub-image, signaling that an attack to the integrity of I_f has been performed.

6.6. Public and Secret

As the specific GA algorithm used and its parameters are instrumental for embedding only, they are not required by the verifier and are not part of the secret embedded, thus knowledge of the GA configuration would not compromise the security of the method and so its parameters may be left public.

The use of QKLT, the block size, the order and number of coefficients used and the bit embedding position may also be left public, but they give a hint in brute force attacks: anyway the space of all possible basis images is so large, for not naive (i.e., small) n, that the attack is unfeasible. The indexes used to address the pixels in I_k to create K_p may be public, and the suggestion is to keep the size of the set H_p small to reduce the probability that an attack alters the pixels in this set, leading to a loss of localization.

Finally, the key image *must* be kept secret: a compromised key image invalidates all the images authenticated with it.

7. Experimental Methodology

The parameters we used to evaluate the performance of the algorithms are the PSNR, the Structural Similarity index (SSIM) and the sensitivity.

For a d bit-per-pixel channel image, the PSNR of a watermarked image I_f with regard to the host image I_h is defined in the quaternion framework, considering the quaternion $m = (0,\ (2^d-1)\vec{i} + (2^d-1)\vec{j} + (2^d-1)\vec{k})$ that represents the peak pixel value, as:

$$\mathrm{PSNR} = 10\log_{10}\frac{\| m \|^2}{\text{m.s.e.}} = 10\log_{10}\frac{\| m \|^2}{\frac{1}{NM}\sum_{z=1}^{NM} \| I_h{}^{(z)} - I_f{}^{(z)} \|^2}, \tag{5}$$

where m.s.e. is the mean squared error between the host and the watermarked images, and $I_h{}^{(z)}$ and $I_f{}^{(z)}$ are their z-th pixel quaternion representation.

The SSIM defined in [29] measures the difference between two images taking into account the characteristics of the Human Visual System. Its values range in the interval $[-1,\ 1]$, with a value of 1 expressing that two images are identical. As it resulted to be greater than 0.998 in all the experiments, we do not report the SSIM value explicitly for each setting in the result tables.

Sensitivity of level D refers to the percentage of detected altered image blocks when a single pixel of that block is modified by $+D$ or $-D$ intensity levels in a single channel. To compute the sensitivity of level D we initialize to 0 two counters TOTBLOCKS and DETECTED, and considering all the watermarked images of our experiments (as we will see, 500 images) for every image the respective watermark is generated and the following nested cycles are performed:

FOR EVERY BLOCK IN THE IMAGE
  FOR EVERY PIXEL IN THE BLOCK
    TOTBLOCKS = TOTBLOCKS + 1;
    ALTER THE PIXEL ADDING D AND, IF THE MODIFICATION IS POSSIBLE (That is, no overflow takes place), CHECK THE BLOCK USING THE VERIFICATION PROCEDURE TO TEST IF THE ATTACK IS DETECTED, IN WHICH CASE DO DETECTED = DETECTED + 1;
    ALTER THE PIXEL ADDING $-D$ AND, IF THE MODIFICATION IS POSSIBLE (That is, no overflow takes place), CHECK THE BLOCK USING THE VERIFICATION PROCEDURE TO TEST IF THE ATTACK IS DETECTED,IN WHICH CASE DO DETECTED = DETECTED + 1;

When all the images have been examined, the ratio DETECTED/TOTBLOCKS represents the sensitivity of level D.

In this paper, we report the results for sensitivity levels 1 and 2, in order to show that QKLT-FW is very sensible to the smallest possible pixel alterations.

It should be pointed out that this kind of test is deeper that any image processing or compression algorithm that may be used to attack an image because any such algorithm, if it performs an alteration to a pixel, will be at least 1 intensity level: therefore, it is obvious that the detection of image processing or compression attacks can only lead to better performance than the worst cases examined by us.

The GA parameters were set as the following: a population size of 100 individuals, a crossover probability of 0.9 and a mutation probability of 0.04; the termination condition was 2000 generations total or the best solution did not change for 10 generations.

Firstly, we show the performance of the proposed algorithm in terms of PSNR, embedding time (on a Linux workstation equipped with 4GB RAM and an Intel(R) Xeon(R) E5410 2.33GHz processor) and sensitivity, in order to support our claims (b), (c) and (d) stated in the introduction. The set of images was composed of 500 images selected from the databases [30,31]; the images were cropped to 256×256 pixels to cut the computation times.

Table 1 reports the averages of PSNR, execution times and sensitivity for some system parameter settings (payload, embedding method, syndrome coding) of QKLT-FW. Moreover, the insertion position p (see Equation (4)) was fixed to 0.

Table 1. Results for some settings of Quaternion Karhunen-Loève Transform Fragile Watermarking (QKLT-FW) (the best performances with regard to the parameter are highlighted in boldface).

Payload (bpb)	Insertion Module	PSNR (dB)	Time (s)	Sensitivity ±1 (%)	Sensitivity ±2 (%)
8	BCM	61.97 ± 0.1	22.65 ± 1.67	67.29 ± 8.22	88.61 ± 2.82
12	SCM (Golay {24,12,8})	62.23 ± 0.11	121.5 ± 8.77	90.53 ± 2.91	99.25 ± 0.23
12	BCM	62.25 ± 0.18	34.7 ± 2.4	93.07 ± 1.05	99.78 ± 0.06
16	BCM	59.44 ± 0.1	87.68 ± 18.06	82.75 ± 7.07	97.36 ± 1.11

We also performed a test using smaller blocks of size 6×6, embedding 6 bpb using BCM as Insertion module: the PSNR resulted in 62.3961 ± 0.23853 dB with a computation time of 45.8534 ± 19.3611 s. As it may be seen, the quality is very high with a slightly increased computation time due to the augmented number of blocks: this is because the overhead is due to the computation of the genetic algorithm and the reduced dimension of the block does not completely compensate for this. This is only an example of the flexibility of the proposed algorithm, as the size and shape of the blocks can be set according to the localization resolutions, payload and running times required by the application.

As a visual example of the results of the watermarking and verification processes (i.e., what the naive user perceives), the Appendix A reports the F16 watermarked image with blocks of size 8×8, an attack to it and how the algorithm detects the tampered region(s).

Then, we compared the performance with those resulting from running the algorithms [25], implemented for color images, and [4] on the same set of images (as MIMIC was already compared with others in [4] and resulted to be qualitatively better). Table 2 shows the comparison among these

algorithms. In the case of [25], the intrinsic properties of the algorithms forced the specific values of bpb and block size: on the contrary, both MIMIC and QKLT-FW revealed a better flexibility allowing more combinations of block sizes and payloads. Note that the sensitivity (of any level) of [25] is 100%, thanks to the use of MACs for the protection of the blocks. Due to the MAC size, [25] requires embedding a larger number of authentication bits reducing the PSNR.

Table 2. Comparison among different embedding algorithms.

Algorithm	Payload (bpb)	Block Size (pixel)	PSNR (dB)	Time (s)	Sensitivity ±2 (%)
[25]	213.125	256	56.71	~2	100
MIMIC SRM [4]	12	64	62.7	37.32 ± 1.96	93.43
MIMIC BCM192 [4]	12	64	58.47	72.97 ± 0.83	96.49
QKLT-FW	12	64	62.25	34.70 ± 2.4	99.79

As can be seen in Table 2, QKLT-FW exhibits an improved performance over the baseline systems considered. Indeed, with an *equivalent* watermarked image *quality*, the quaternion approach *improves* with regard to *running time* and *sensitivity*. It is worth pointing out that the quaternion framework produces an integrated information with regard to the color components (or multi-channel components), allowing the application of all the properties of quaternions for color image processing, in particular the QKLT.

8. Conclusions

In summary, we emphasize that with a simple data type modification from real value to quaternion, the proposed system shows a better performance for detecting and localizing the content alterations. As already noted in [4], [25] has the advantages of a sensitivity of 100% and of a predictable computing time, but its main drawbacks are a lower PSNR and the space required to store the authentication data, that implies the use of large blocks needed to save the authentication information, reducing the localization capability.

The proposed algorithm based on QKLT has thus the following properties and advantages:

- high PSNR and high SSIM, resulting in very high quality of the watermarked images, both objective and subjective (see Table 1);
- high sensitivity to modifications because even single pixel modifications of two intensity levels may be detected in more than 99.7% of the cases (see Table 2): this makes the probability that any real attack goes undetected close to zero in practice;
- flexible and good localization capability, as shown working on blocks of different sizes, namely 6×6 and 8×8 color pixels, and different payloads;
- easily integrated into the MIMIC framework as a new Watermark Distilling Unit, improving the running times of previously developed algorithms in the same framework (as can be seen in Table 2).

It should be noted that, in some cases, the GA may not converge to a solution due to the intrinsic stochastic approach that embeds the watermark in every image block; it is possible to cope with this problem by running the GA multiple times on the blocks reporting a convergence problem, also reducing the tightness of some constraints (e.g., on the possible modification the GA may perform on the pixel values).

As for MIMIC [4], the security of the method is based on the secrecy of the image from which the KLT basis is derived: from that image a secret embedding space is derived, so the transform coefficients cannot be determined, in particular their less significant bits. Moreover, the watermark string is dependent both on the key image, on the host image, and on the host image size: this prevents copy-and-paste attacks, transplantation attacks, VQ attacks, cropping and embedding attacks. Note

that in the MIMIC framework, a trivial substitution attack is always possible: changing a block with a random one will go undetected 1 every 2^s attempts, if s bpb are embedded. But, in an image with U blocks, the probability of not detecting any modification is $1/2^{Us}$: this is a very small number, even for an image with a small number of blocks. We stress the fact that the use of quaternions as color pixel representation opens up the possibility of applying the presented approach to color images with alpha channel (i.e., four-dimensional pixels), as an integrated approach. This is one of our future research directions.

Author Contributions: All authors equally contributed to the design and implementation of the described algorithms and experiments, as well as to write and proofread the paper.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

In this appendix we report, as example, the result of watermarking one of the image used in the experiments: Figure A1 shows the watermarked image (we do not report the original image because, given the very high PSNR no differences with Figure A1 may be appreciated by the human eye), whilst Figure A2 presents the watermarked image altered by a tamper on the right part of the sign (the number 1100 has been modified, by copying and pasting two areas, to 1010).

In Figure A3, the output of the verification procedure on the image in Figure A2 is presented: the forged area is correctly evidenced by marking the blocks which contain at least one modified pixel.

Figure A1. Watermarked color image, publicly available from the McGill Calibrated Color Image Database [30] (http://tabby.vision.mcgill.ca/html/welcome.html), PSNR = 67.02 dB (with zoom on detail).

Figure A2. Tampered image (with zoom on tampered detail).

Figure A3. Verified image, with (nineteen) tampered blocks evidenced as crossed areas.

References

1. Cox, I.J.; Miller, M.L.; Bloom, J.A.; Fridrich, J.; Kalker, T. *Digital Watermarking and Steganography*, 2nd ed.; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 2008.
2. Botta, M.; Cavagnino, D.; Pomponiu, V. Fragile Watermarking using Karhunen-Loève transform: The KLT-F approach. *Soft Comput.* **2015**, *19*, 1905–1919. [CrossRef]
3. Aslantas, V.; Ozer, S.; Ozturk, S. Improving the performance of DCT-based fragile watermarking using intelligent optimization algorithms. *Opt. Commun.* **2009**, *282*, 2806–2817. [CrossRef]
4. Botta, M.; Cavagnino, D.; Pomponiu, V. A Modular Framework for Color Image Watermarking. *Signal Process.* **2016**, *119*, 102–114. [CrossRef]
5. Le Bihan, N.; Sangwine, S.J. Quaternion principal component analysis of color images. In Proceedings of the 2003 International Conference on Image Processing, Barcelona, Spain, 14–17 September 2013; Volume 1, pp. 809–812. [CrossRef]
6. Sangwine, S.J. Fourier transforms of colour images using quaternion, or hypercomplex numbers. *Electron. Lett.* **1996**, *32*, 1979–1980. [CrossRef]
7. Ell, T.A.; Sangwine, S.J. Hypercomplex Fourier transforms of color images. *IEEE Trans. Image Process.* **2007**, *16*, 22–35. [CrossRef] [PubMed]
8. Le Bihan, N.; Mars, J. Singular value decomposition of quaternion matrices: A new tool for vector-sensor signal processing. *Signal Process.* **2004**, *84*, 1177–1199. [CrossRef]
9. Angulo, J. Geometric algebra colour image representations and derived total orderings for morphological operators—Part I: Colour quaternions. *J. Vis. Commun. Image Represent.* **2010**, *21*, 33–48. [CrossRef]
10. Bas, P.; Le Bihan, N.; Chassery, J.-M. Color image watermarking using quaternion Fourier transform. In Proceedings of the 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, Hong Kong, China, 6–10 April 2003; Volume 3, pp. 521–524.
11. Sangwine, S.J.; Ell, T.A. Hypercomplex Fourier transforms of color images. In Proceedings of the ICIP, Thessaloniki, Greece, 7–10 October 2001; Volume 1, pp. 137–140.
12. Wang, X.-Y.; Wang, C.-P.; Yang, H.-Y.; Niu, P.-P. A robust blind color image watermarking in quaternion Fourier transform domain. *J. Syst. Softw.* **2013**, *86*, 255–277. [CrossRef]
13. An, M.; Wang, W.; Zhao, Z. Digital watermarking algorithm research of color images based on quaternion Fourier transform. In Proceedings of the SPIE 8917: Multispectral Image Acquisition, Processing, and Analysis, Wuhan, China, 26 October 2013; pp. 1–7.
14. Tsui, T.K.; Zhang, X.-P.; Androutsos, D. Color Image Watermarking Using Multidimensional Fourier Transforms. *IEEE Trans. Inf. Forensics Secur.* **2008**, *3*, 16–28. [CrossRef]
15. Chen, B.; Coatrieux, G.; Chen, G.; Sun, X.; Coatrieux, J.L.; Shu, H. Full 4-D quaternion discrete Fourier transform based watermarking for color images. *Digit. Signal Process.* **2014**, *28*, 106–119. [CrossRef]
16. Shao, Z.; Duan, Y.; Coatrieux, G.; Wu, J.; Meng, J.; Shu, H. Combining double random phase encoding for color image watermarking in quaternion gyrator domain. *Opt. Commun.* **2015**, *343*, 56–65. [CrossRef]

17. Shao, Z.; Wu, J.; Coatrieux, J.L.; Coatrieux, G.; Shu, H. Quaternion gyrator transform and its application to color image encryption. In Proceedings of the ICIP, Melbourne, Australia, 15–18 September 2013; pp. 4579–4582.
18. Qi, M.; Li, B.-Z.; Sun, H. Image watermarking using polar harmonic transform with parameters in SL(2,R). *Signal Process. Image Commun.* **2015**, *31*, 161–173. [CrossRef]
19. Yang, H.-Y.; Wang, X.-Y.; Niu, P.-P.; Wang, A.-L. Robust Color Image Watermarking Using Geometric Invariant Quaternion Polar Harmonic Transform. *ACM Trans. Multimed. Comput. Commun. Appl.* **2015**, *11*, 40:1–40:26. [CrossRef]
20. Al-Otum, H. Color image authentication using a zone-corrected error-monitoring quantization-based watermarking technique. *Opt. Eng.* **2016**, *55*, 083103. [CrossRef]
21. Akansu, A.N.; Haddad, R.A. *Multiresolution Signal Decomposition: Transforms, Subbands, and Wavelets*; Academic Press: Cambridge, MA, USA, 1992.
22. Lin, S.; Lin, J. Authentication and recovery of an image by sharing and lattice-embedding. *J. Electron. Imaging* **2010**, *19*, 043008. [CrossRef]
23. Lo, C.; Hu, Y.; Chen, W.; Chang, I. Probability-based image authentication scheme for indexed color images. *J. Electron. Imaging* **2014**, *23*, 033003. [CrossRef]
24. Wang, X.; Zhang, D.; Guo, X. Authentication and recovery of images using standard deviation. *J. Electron. Imaging* **2013**, *22*, 033012. [CrossRef]
25. Celik, M.U.; Sharma, G.; Saber, E.; Tekalp, A.M. Hierarchical watermarking for secure image authentication with localization. *IEEE Trans. Image Process.* **2002**, *11*, 585–595. [CrossRef] [PubMed]
26. Vince, J. *Quaternions for Computer Graphics*; Springer: Berlin, Germany, 2011.
27. Zhang, F. Quaternions and matrices of quaternions. *Linear Algebra Its Appl.* **1997**, *251*, 21–57. [CrossRef]
28. Pei, S.-C.; Cheng, C.-M. A novel block truncation coding of color images by using quaternion-moment preserving principle. In Proceedings of the IEEE International Symposium on Circuits and Systems, Atlanta, GA, USA, 23–26 May 1996; Volume 2, pp. 684–687.
29. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [CrossRef]
30. Olmos, A.; Kingdom, F.A.A. A biologically inspired algorithm for the recovery of shading and reflectance images. *Perception* **2004**, *33*, 1463–1473. [CrossRef] [PubMed]
31. Fred Kingdom's Laboratory, McGill Vision Research. Available online: http://tabby.vision.mcgill.ca/ (accessed on 18 December 2014).