Color Image Processing Using Reduced Biquaternions with Application to Face Recognition in a PCA Framework

Moumen T. El-Melegy
Computer Vision Lab
Assiut University, Egypt
moumen@aun.edu.eg

Aliaa T. Kamal
Math Department
Assiut University, Egypt
aliaa_t@aun.edu.eg

Abstract

In this paper, we present the theory of reduced biquaternion algebra to represent color images and to develop efficient vector processing methods. We apply this theory to the field of face recognition in a principal component analysis (PCA) framework. We develop a novel PCA method based on reduced biquaternion to make full use of the face color cues. Moreover, we derive new mathematical results on the computation of the eigenvalues/eigenvectors of the data scatter matrix. We also extend this method to two-dimensional color PCA to combine the face spatial and color information. Experiments on several public-domain color face benchmark datasets demonstrate the higher performance of the proposed methods compared to regular PCA and like methods.

1 Introduction

The perception of color plays a paramount role to humans since they routinely use color to sense the environment, recognize objects, and convey information. However, early image processing research was devoted to processing binary then grayscale images. It was many years later when researchers have started to direct their attention to color images, motivated by the emergence of affordable color image acquisition hardware and the advent of affordable computation power to process these images. However the idea to think of a color image as a set of parallel, independent grayscale images dominated the first color processing techniques. Unfortunately, such an idea ignores the correlation between color components and the distribution of energy among them. A more recent category of color image processing methods processes all components of a color image simultaneously where each pixel can be treated as a vector. Such color vector processing methods are capable of implicitly accounting for the correlation that exists between the color components.

In this paper we are interested in *color vector processing* methods and their application to face recognition using principal component analysis (PCA)-like methods. Face recognition is a research domain that has traditionally received significant attention and still been a hot research topic with several important applications. One of the classical and successful approaches to face recognition is PCA. PCA is a well-established linear dimension-reduction technique. It finds the directions along which the original data are projected into a lower dimensional space with minimal reconstruction error. Based on PCA theory, Turk and Pentlant [1] presented in their 1991 seminal paper the eigenface method for face recognition. To apply this method, a color image should be transformed first into a grayscale image. As such, the color information is totally lost, the drawback on which some later research [21] stressed by proving that color cues indeed contribute to face recognition, especially when shape cues of the images are progressively degraded.

Years later, Yang et al [2] proposed the so-called two-dimensional principal component analysis (2DPCA), which enhances the original PCA method and does not transform the face image matrix into a vector before processing. This method showed better recognition accuracy due to the utilization of the natural spatial information of the face images. However the method was applicable only to the grayscale version of the color image thus still missing the color information.

In order to make use of the color information, some works have been recently developed. In 2015 Xiang et al [3] introduced two PCA-based methods to improve the accuracy on dealing with color images. The first method—called color PCA (CPCA)—represents a face image as a 2D matrix where each row represents a color channel and has as many columns as the number of image pixels. Despite its ability to deal with color information, it destroys the spatial information when the image pixels are flattened as a row. The second method—denoted here by C2DPCA—represents the three components of a pixel as a basic unit called color value. This method also defines a set of basic operations that can be applied on these color values in order to find the data scatter matrix and its eigenvectors. However some of these operations have no meaningful or well-justified definitions.

Adopting another strategy, several studies [4-6] have demonstrated that the concept of quaternions from the domain of hypercomplex numbers is well adapted to color images by encoding the pixel's color components into the

three imaginary parts of a quaternion (Q) number. As such, the color face image can be represented as a 2D quaternion matrix, on which a set of well-defined algebraic operations can be applied in the typical PCA framework. Although quaternions find popular uses in computer graphics and computer vision, in particular for calculations involving 3D rotations, they have a key drawback; their multiplication is noncommutative [13]. This leads to the existence of left and right eigenvalues (and eigenvectors), which generally have different values. Moreover, the eigenvalues of a quaternion matrix are infinite [13]. These facts complicate the process of finding the optimal projection directions in the PCA framework.

Inspired by the theory of hypercomplex algebras, in this paper we propose to use the concept and theory of reduced biquaternions, as another branch of hypercomplex algebras, to represent color images and develop efficient vector processing methods based on them.

A reduced biquaternion (B) shares with a quaternion the property of having one real part and three imaginary parts. Both of them have unique but different definitions of norm and conjugation and have individual representation forms. More importantly, B numbers have some key advantages over Qs. For one, their multiplication is commutative [13]. That means the right and left eigenvalues (eigenvectors) of a B matrix are equal and finite. In addition, the complexity of computing several operations on B numbers is lower than on Q numbers, e.g., multiplication and eigenvalues computation. This leads to faster algorithms compared to those for Qs. It is important to note here that Bs have the problem of not forming a division system [14]. However this disadvantage has almost no influence on many practical applications of Bs including the one we show in this paper.

Another key contribution of this paper is the derivation of several mathematical results on the computation of the eigenvalues/eigenvectors of a B matrix and their norm. Based on these derivations, we devise a lower-complexity (and thus faster) algorithm to find the best eigenvectors (projection directions) of the data scatter matrix.

The concept and theory of B algebra presented here can be applied for color image processing in a holistic manner, thus accounting for the correlation that exists between the color components of a pixel. While the developed theory and methods are applicable to several practical problems and applications, another distinctive feature of this paper is that we apply it here *for the first time* to the problem of color face recognition in a PCA framework. We develop first a 1D PCA method, called BPCA, which vectorizes a color matrix into a B vector before feature extraction and classification. Furthermore, we develop a 2D version, called 2DBPCA, to exploit the natural spatial information of the face image and needs no vectorization. Both developed methods are evaluated on three public-domain color face datasets and compared to several existing techniques in the literature in terms of accuracy and speed. The experimental results demonstrate the outperformance of the developed methods. It is important to stress here that while we are aware of the successful application of a B representation of color images for template matching and edge detection [15], it is to the best of our knowledge that the theory developed here has not been reported before. Additionally, we are not aware of any B-based PCA approach to color face recognition.

The rest of this paper is structured as follows. Section 2 briefly reviews quaternion and reduced biquaternion. In Section 3, the proposed color face recognition methods will be presented. The experimental results of the color face recognition methods will be reported in Section 4. Finally, conclusions are drawn in Section 5.

2 Hypercomplex Algebras

A complex number is a number that consists of a real part and an imaginary part and expressed as:

$$f = l + mi, \tag{1}$$

where l and $m \in \mathbb{R}$ and i satisfy $i^2 = -1$. Let $\mathbb{C}$ denotes the set of complex numbers. Complex numbers found a connection between the pure algebra and the geometry of $\mathbb{R}^2$. This connection inspires to construct the *2n* dimensional hypercomplex numbers (u_{2n}) from two *n*–dimensional hypercomplex numbers ($u_n,\ v_n$) using:

$$u_{2n} = u_n + v_n e, \tag{2}$$

where $e^2 = 1\ or\ -1$. For example, the 4D hypercomplex numbers (h) can be constructed from complex numbers:

$$h = z_1 + z_2 j = a + bi + cj + dk,$$

where z_1 and $z_2 \epsilon\ \mathbb{C},\ a, b, c,$ and $d\ \epsilon\ \mathbb{R},\ i, j,$ and $k = \pm 1$. The first step towards this direction was made by William Hamilton in 1843 [7] discovering the four-dimensional (4D) quaternions (Q) after having tried for years with the case *n = 3*. A year later Hamilton proposed quaternions with complex coefficients called biquaternions that are not commutative in multiplication and don't form a division algebra. In 1990, the reduced biquaternions (B), were first introduced by Schütte and Wenzel [8]. They have four elements as conventional quaternions, and the multiplication rule is commutative. In 1992, Ell [9] defined the double-complex algebra with commutative multiplication. In 1996, Davenport [10] proposed the four dimensional commutative hypercomplex algebras (HCA_4). In 1998 and 1999, Sommer et al. extend the HCA to any 2^n dimensions [11].

The major difference between B, double-complex algebra, and HCA_4 is the choice of square root of 1, where $j^2 = 1$ in B, $i^2 = 1$ in double-complex algebra, while $k^2 = 1$ in HCA_4.

In our work we focus on quaternion and reduced biquaternion numbers. In next subsections we describe them in more details.

2.1 Quaternion numbers (Q)

A quaternion number $q \in \mathbb{Q}$, where $\mathbb{Q}$ is the set of all quaternions, consists of four components: one real part and three imaginary parts. It is represented as:

$$q = q_r + q_i i + q_j j + q_k k, \tag{3}$$

where q_r, q_i, q_j and $q_k \in \mathbb{R}$ and *i, j,* and *k* are satisfying:

$$i^2 = j^2 = k^2 = -1, \quad ij = -ji = k, \quad jk = -kj = i, \; ki = -ik = j. \tag{4}$$

Unlike real or complex numbers, multiplication of quaternion is not commutative as predicted from (4). The multiplication of p and q defined as:

$$\begin{aligned} pq = {} & (p_r q_r - p_i q_i - p_j q_j - p_k q_k) \\ & + (p_r q_i + p_i q_r + p_j q_k - p_k q_j) i \\ & + (p_r q_j - p_i q_k + p_j q_r + p_k q_i) j \\ & + (p_r q_k + p_i q_j - p_j q_i + p_k q_r) k. \end{aligned} \tag{5}$$

Quaternion is a division system, where the division of p and q exists and $p/q \in \mathbb{Q}$ given that $q_r^2 + q_i^2 + q_j^2 + q_k^2 \neq 0$.The norm of a quaternion is given by:

$$|q| = \sqrt{q_r^2 + q_i^2 + q_j^2 + q_k^2}, \tag{6}$$

which satisfies $|pq| = |p||q|$. The conjugate of a quaternion is denoted as $\bar{q}$ and is defined as (similar to complex numbers)

$$\bar{q} = q_r - q_i i - q_j j - q_k k \tag{7}$$

A quaternion matrix is a matrix whose all elements are quaternion numbers. Several properties can be defined for quaternion matrices. For example, the Hermitian transpose of quaternion matrix $\mathbf{P}$ is $\mathbf{P}^H = (\bar{\mathbf{P}})^T$. One such interesting property is that for an $n \times n$ quaternion matrix P there are left and right eigenvalues [12], since multiplication is not commutative. That is, we have

$$\begin{aligned} \mathbf{P}\mathbf{x}_r &= \mathbf{x}_r \lambda_r \\ \mathbf{P}\mathbf{x}_l &= \lambda_l \mathbf{x}_l, \end{aligned} \tag{8}$$

where λ_r, λ_l are right and left eigenvalues, $\mathbf{x}_r$ and $\mathbf{x}_l$ are right and left eigenvectors. $\lambda_l$and $\lambda_r \in \mathbb{Q}$ and they may not be equal. Moreover, the eigenvalues of a quaternion matrix are infinite. If λ is an eigenvalue of $\mathbf{P}$, then every element of the set $\tau = \{p\lambda p^{-1}$: p is any unit quaternion with $|p| = 1\}$ is also an eigenvalue of $\mathbf{P}$ [13]. The right eigenvalue can easily computed for any $n \times n$ quaternion matrix, while some authors [12, 14] showed that finding the left eigenvalues for matrices with $n \geq 4$ is rather difficult.

2.2 Reduced Biquaternion numbers (B)

A reduced biquaternion number $b \in \mathbb{B}$ (where $\mathbb{B}$ is the set of reduced biquaternions) is expressed in the form:

$$b = b_r + b_i i + b_j j + b_k k \tag{9}$$

where $b_r, b_i, b_j,$ and $b_k \in \mathbb{R}$ and *i, j,* and *k* conform to the rules:

$$i^2 = k^2 = -1, j^2 = 1, ij = ji = k, \quad jk = kj = i, ik = ki = -j \tag{10}$$

It is obvious from the previous equation that the multiplication rule of B is commutative, which is the major difference between Bs and Qs.

B is not a complete division system [8]. For example, for two special B numbers $e_1 = \frac{1+j}{2}, e_2 = \frac{1-j}{2}$ the form $c_1 e_1$or $c_2 e_2$ (where c_1 and c_2 are any complex numbers) is a divisor of zero and doesn't have a multiplication inverse. Thus there is no solution for the variable z in:

$$uz = 1, \text{ if } u = c_1 e_1 \text{or } c_2 e_2, \tag{11}$$

while z has infinite solutions in:

$$uz = 0, \text{ if } u = c_1 e_1 \text{or } c_2 e_2. \tag{12}$$

If the norm of B was defined similar to those of complex and quaternion numbers, it would be

$$|b|_\circ = \sqrt{b_r^2 + b_i^2 + b_j^2 + b_k^2}, \tag{13}$$

which unfortunately leads to $|bc|_\circ \neq |b|_\circ |c|_\circ$. Accordingly, the B norm is defined as [15]:

$$|b| = \left(\left(b_r^2 + b_i^2 + b_j^2 + b_k^2\right)^2 - 4\left(b_r b_j + b_i b_k\right)^2 \right)^{1/4} \geq 0, \tag{14}$$

which indeed satisfies $|bc| = |b||c|$ (see [15] for proof). However the norm as defined in (13) is sometimes used by authors (e.g., [13, 15]) due to its lower computation cost and similarity to the norm of complex numbers and quaternions. To avoid this confusion, we call it the weak norm and denote it by $|.|_\circ$, while the term 'norm' always refers to the definition in (14). The norm has similar but not identical properties to its counterpart in complex numbers, see [15, 16] for more details. In the same way, if the conjugate of $b \in \mathbb{B}$ was defined in the same way of the conjugate of quaternions, it would be $\bar{b} = b_r - b_i i - b_j j - b_k k$, which does not satisfy $b\bar{b} \in \mathbb{R}$. Therefore the conjugate of b is taken as [15]

$$\bar{b} = \frac{|b|^2}{b}, \tag{15}$$

where $b^{-1}$is the reciprocal (multiplicative inverse) of b and can be computed as shown below if $|b| \neq 0$. The conjugation of B is a nonlinear operation, so that $\overline{(b + c)} \neq \bar{b} + \bar{c}$.

The Hermitian transpose of a B matrix $\mathbf{P}$ (a matrix whose all elements are B numbers) is computed in the same way as quaternion, where $\mathbf{P}^H = (\bar{\mathbf{P}})^T$.

The B numbers can be represented in three forms [13, 15]: e_1-e_2 forms, matrix representations, and polar forms. While the polar forms are often used to describe the geometric meaning of B numbers, the former two can easily explain many concepts of B, such as addition,

multiplication, inverse and norm. In this work, we heavily employ the e_1-e_2 forms, which are irreducible representation for Bs.

Davenport [10] defined two nonzero numbers that are idempotent elements and defined as:

$$e_1 = (1+j)/2,\ e_2 = (1-j)/2, \text{ such that}$$
$$e_1 e_2 = 0, e_1^n = e_1, \text{ and } e_2^n = e_2. \quad (16)$$

A B number can be represented in the form

$$b = (b_r + ib_i) + j(b_j + ib_k) = c_a + jc_b \quad (17)$$
$$= c_1 e_1 + c_2 e_2,$$

where $c_a = b_r + ib_i,\ c_b = b_j + ib_k,\ c_1 = c_a + jc_b$ and $c_2 = c_a - jc_b$ are all complex numbers. The two elements e_1 and e_2 can be geometrically explained as the null cone of Minkowsky space [17]. This form can reduce the complexity of some operations. For example, we need only 8 real multiplications (compare to 16 real multiplications in case of Q multiplication) to calculate the multiplication of two Bs. Moreover, the reciprocal of b can be computed using e_1–e_2 forms as:

$$b^{-1} = c_1^{-1} e_1 + c_2^{-1} e_2, \quad (18)$$

where the reciprocal of b exists if and only if the inverse of c_1 and c_2 exists [8].

A reduced biquaternion matrix can be decomposed into four components and it is often represented using $e_1 - e_2$ forms. For example, a B $n \times n$ matrix $\mathbf{Y}$ can be represented in the form

$$\mathbf{Y} = (\mathbf{Y}_r + i\mathbf{Y}_i) + j(\mathbf{Y}_j + i\mathbf{Y}_k) = \mathbf{C}_a + j\mathbf{C}_b$$
$$= \mathbf{C}_1 e_1 + \mathbf{C}_2 e_2, \quad (19)$$

where $\mathbf{C}_a = \mathbf{Y}_r + i\mathbf{Y}_i, \mathbf{C}_b = \mathbf{Y}_j + i\mathbf{Y}_k, \mathbf{C}_1 = \mathbf{C}_a + j\mathbf{C}_b$, and $\mathbf{C}_2 = \mathbf{C}_a - j\mathbf{C}_b$ are all complex matrices. The addition and multiplication of two matrices are calculated by two additions and two multiplications of two matrices.

The eigenvalues and eigenvectors of any B matrix are finite since the multiplication is commutative. There are n^2 eigenvalues and eigenvectors of any B matrix that can be computed using $e_1 - e_2$ as [13]

$$\mathbf{Y}(\mathbf{x}_1 e_1 + \mathbf{x}_2 e_2) = (\lambda_1 e_1 + \lambda_2 e_2)(\mathbf{x}_1 e_1 + \mathbf{x}_2 e_2), \quad (20)$$

where λ_1 and λ_2 ($\mathbf{x}_1$ and $\mathbf{x}_2$) are the eigenvalues (eigenvectors) of $\mathbf{C}_1$ and $\mathbf{C}_2$, respectively.

Note that the eigenvalues of an $n \times n$ quaternion matrix requires the computation of the eigenvalues of an equivalent $2n \times 2n$ complex matrix [12], while the eigenvalues of an $n \times n$ B matrix is computed using two $n \times n$ complex matrices. Thus, the complexity of computing the eigenvalues of B matrix is lower than that of quaternion matrix.

3 Reduced Biquaternion Approach to Color Face Recognition

Each pixel in a color image consists of three channels (red, green, and blue). These channels can be represented as the three imaginary parts of a B number. Thus, any pixel of a color image can be represented as:

$$q = 0 + ri + gj + bk, \quad (21)$$

where $r, g,$ and $b \in \mathbb{R}$ and represent red, green, and blue (often after being normalized through division by 255) components of a pixel, respectively.

In this section, we use the B representation of color face image to propose color face recognition algorithms based on 1D PCA and 2D PCA.

3.1 1D Reduced Biquaternion Principal Component Analysis (BPCA)

The first approach we propose is based on the well-known PCA algorithm [1]. Let's consider a 2D $m \times n$color face image $\mathbf{I}$ represented as a 1D B vector $\mathbf{\Gamma} \in \mathbb{B}^{mn}$ obtained by stacking the columns of $\mathbf{I}$ on top of one another. Let the training set consists of M images each of size $m \times n$. For each image Γ_s in the training set, the B representation is:

$$\Gamma_s = 0 + \mathbf{r}_s i + \mathbf{g}_s j + \mathbf{b}_s k, \quad (22)$$
$$s = 1, 2, 3, \dots, M,$$

where **r, g,** and **b**are vectors that represent red, green, and blue values of image pixels. The training set can be represented as an $mn \times M$ B matrix $\mathbf{A}$. The main idea of PCA is to find a vector that best explains the distribution of face image within the whole image space. PCA depends on finding the covariance (*scatter*) matrix $\mathbf{G}_{1D} \in \mathbb{B}^{mn \times mn}$:

$$\mathbf{G}_{1D} = \frac{1}{M} \sum_{s=1}^{M} (\Gamma_s - \tilde{\Gamma})(\Gamma_s - \tilde{\Gamma})^H, \quad (23)$$

where $\tilde{\Gamma}$ is the average face of the training set and $(.)^H$ operator denotes the Hermitian transpose. The size $(mn \times mn)$ of $\mathbf{G}_{1D}$ will make finding its eigenvalues and eigenvectors computationally intractable for typical image sizes. Since the number of images in the training set is often much less than the dimension of the space $(M < mn)$, then a common practice [1] is recommended to reduce the computation cost. First, the eigenvalues and eigenvectors$\mathbf{v}_s \in \mathbb{B}^M$ of the $M \times M$ matrix $\mathbf{A}^T\mathbf{A}$ are computed. Then the covariance matrix $\mathbf{G}_{1D}$ will have the same eigenvalues, and its eigenvectors $\mathbf{u}_s \in \mathbb{B}^{mn}$ (also known as the eigenfaces) are computed by pre-multiplying the vectors $\mathbf{v}$ by $\mathbf{A}$:

$$\mathbf{u}_s = \mathbf{A}\mathbf{v}_s, s = 1, 2, \dots, t, \quad (24)$$

where $t \leq M$ is the number of selected eigenvectors corresponding to the largest eigenvalues in terms of the norm as defined in (14), see the implementation note below.

A new face image $\mathbf{\Gamma}$ projected into the face space by

$$\omega_s = \mathbf{u}_s^T(\Gamma - \tilde{\Gamma}), \quad s = 1, 2, \dots, t. \quad (25)$$

These weights form a feature vector $\mathbf{\Omega} = [\omega_1, \omega_2, \dots, \omega_t]^T \in \mathbb{B}^t$ that describes the contribution of eigenfaces in representing $\mathbf{\Gamma}$. Suppose that $\mathbf{\Omega}^j$ denotes the

feature vector representing the *j*-th training face image, then the class of the *j*-th training image that minimizes

$$d(\Gamma, \Gamma_j) = \left(\sum_{s=1}^{t} |\omega_s - \omega_s^j|^2 \right)^{1/2}, j = 1, \dots, M, \quad (26)$$

is the class that $\mathbf{\Gamma}$ belongs to (nearest neighbor classification), where the norm is defined as given in (14).

This proposed B-based representation of face images is illustrated in Figure 1, where each image can be represented as a linear combination of the best *t* eigenvectors in addition to the mean image.

Implementation note: As discussed above, finding the eigenvectors of $\mathbf{G}_{1D}$ would involve computing the eigenvectors of the $M \times M$ matrix $\mathbf{A}^T\mathbf{A}$. Being a B matrix, this latter matrix, as mentioned in the previous section, will have M^2(not just M) B eigenvalues and eigenvectors. This would increase the computational cost of finding them and identifying their best subset, especially if the training set is large in size. Since we are actually interested in finding only the *t* largest eigenvalues along with their corresponding eigenvectors, all M^2 eigenvalues are computed then a quick selection algorithm [22] can be employed to return those *t* largest in terms of the norm. The time complexity of this will be $O(M^2t)$.We propose to use a more efficient, faster algorithm for the same sake. To outline this algorithm, we need to prove first a key result.

Result 1: If $\lambda = \lambda_1 \mathrm{e}_1 + \lambda_2 \mathrm{e}_2$, where $\lambda_1$and λ_2 are complex numbers, then $|\lambda|^2 = |\lambda_1||\lambda_2|$, where $|.|$denotes the norm defined in (14).

Proof: Let $\lambda_1 = a + bi,$ and $\lambda_2 = c + di$. Then,

$$\lambda = \frac{1}{2}[(a + c) + (b + d)i + (a - c)j + (b - d)k].$$

Using the norm defined in (14),

$$|\lambda| = \left[\frac{1}{4}(a^2 + b^2 + c^2 + d^2)^2 - \frac{1}{4}\left(a^2 + b^2 - (c^2 + d^2)\right)^2\right]^{1/4}$$

$$= \left[\frac{1}{4}(|\lambda_1|^2 + |\lambda_2|^2)^2 - \frac{1}{4}(|\lambda_1|^2 - |\lambda_2|^2)^2\right]^{1/4}$$

$$\text{Thus, } |\lambda|^2 = |\lambda_1||\lambda_2| \quad \blacksquare \quad (27)$$

The algorithm we propose to use is based on the idea that taking the logarithm of the norm of the eigenvalues of the B matrix, from (27), will produce the sum of the logarithms of the norms of the eigenvalues of two complex matrices. This will transform the problem here into a classical, challenging problem in computer science called X+Y sorting [22, 23], which is often formally stated as follows: Given two finite sets X and Y, the problem is to order all pairs (x, y) in the Cartesian product X × Y by the key x + y.

We first outline the proposed algorithm, and then derive its performance complexity.

Algorithm 1: To find the *t* largest eigenvalues of an$M \times M$ B matrix $\mathbf{Y}$ and their corresponding eigenvectors.

a) Represent the matrix $\mathbf{Y}$ using the e_1-e_2 form as explained in(19): $\mathbf{Y} = \mathbf{C}_1 e_1 + \mathbf{C}_2 e_2$, where $\mathbf{C}_1$ and $\mathbf{C}_2$ are two complex matrices.
b) Compute the M complex eigenvaluesof $\mathbf{C}_1$ in the set $\mathcal{S}_\alpha$and their corresponding eigenvectors $\{\mathbf{u}_s\}_{s=1}^M$, and the M complex eigenvalues $\mathcal{S}_\beta$of $\mathbf{C}_2$ and their corresponding eigenvectors $\{\mathbf{v}_s\}_{s=1}^M$.
c) Order the two sets $\mathcal{S}_\alpha$ and $\mathcal{S}_\beta$ independently in decreasing order in terms of the usual norm of complex numbers using a quicksort algorithm [22].
d) Compute the set $\mathcal{S}_\alpha^o = \{\log(|\alpha| + \epsilon) \,|\alpha \in \mathcal{S}_\alpha, \epsilon > 0\}$ and $\mathcal{S}_\beta^o = \{\log(|\beta| + \epsilon) \,|\beta \in \mathcal{S}_\beta, \epsilon > 0\}$. ϵ is a small number to avoid the logarithm of 0.
e) Run the X+Y *k*th selection algorithm [24] on the two sets $\mathcal{S}_\alpha^o$ and $\mathcal{S}_\beta^o$ to obtain the *t* largest pairwise sums of all pairs in the Cartesian product of the two sets $\mathcal{S}_\alpha^o$and $\mathcal{S}_\beta^o$. Let the obtained pairs form the set $\mathcal{S}_\lambda$.
f) Compute the set of eigenvalues $\{\lambda_z\}$, of $\mathbf{Y}$ from all $\lambda_z = \alpha e_1 + \beta e_2, \forall(\alpha, \beta) \in \mathcal{S}_\lambda, z = 1, \dots, t$.
g) Compute the eigenvectors $\{\mathbf{x}_z\}$ of $\mathbf{Y}$ from $\mathbf{x}_z = \mathbf{u}e_1 + \mathbf{v}e_2$, for every $\mathbf{u}$ and $\mathbf{v}$ corresponding to $\forall(\alpha, \beta) \in \mathcal{S}_\lambda$, where $z = 1, \dots, t$.

Result 2: Algorithm 1 has a time complexity of $O(Mt + M\log M)$.

Proof: The algorithm first sorts the two *M*-sized sets of the eigenvalues of each matrix $\mathbf{C}_1$ and $\mathbf{C}_2$ using a quicksort algorithm [22]. This requires $O(M\log M)$ complexity. The X+Y selection algorithm [24] works on the two sorted sets to select the *k*th $(1 \le k \le t)$ largest element of $\mathcal{S}_\alpha^o + \mathcal{S}_\beta^o$ over their Cartesian product in time $O(M)$ [24]. For obtaining all t elements, the complexity is $O(Mt)$. In total, the overall time complexity of the algorithm is $O(Mt + M\log M)$. This is considerably less than the original $O(M^2t)$ complexity. ■

3.2 2D Reduced Biquaternion Principal Component Analysis (2DBPCA)

This technique is based on the 2DPCA algorithm [2]. In this algorithm we present a 2D $m \times n$color face image $\mathbf{I}$ as a 2D B matrix $\mathbf{\Phi} \in \mathbb{B}^{m \times n}$, where each element is a B number that represents a pixel. As such each image in the training set can be represented as:

$$\mathbf{\Phi}_s = 0 + \mathbf{R}_s i + \mathbf{G}_s j + \mathbf{B}_s k, \quad s = 1, 2, \dots, M. \quad (28)$$

where $\mathbf{R}, \mathbf{G}$, and $\mathbf{B} \in \mathbb{R}^{m \times n}$, M is the number of images in training set. The idea is to project the image matrix $\mathbf{\Phi}$ onto an n-dimensional reduced biquaternion column vector $\mathbf{v}$ by the following transformation:

$$\mathbf{u} = \mathbf{\Phi}\mathbf{v} \quad (29)$$

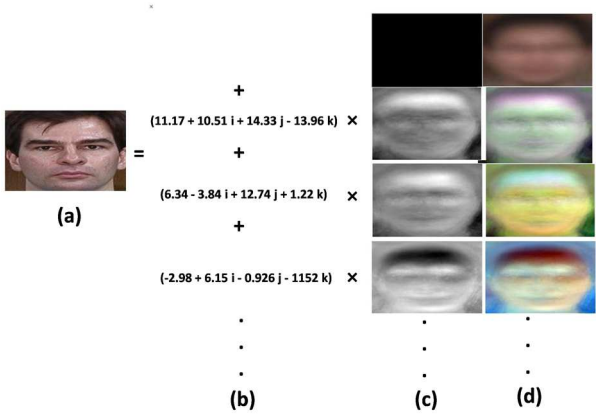

Figure 1: B-based face representation: Column (a) represents original color image. Column (b) shows the weights as computed from Eq.(25). Column (c) is the real components of the eigenfaces represented as grayscale images. Column (d) is the imaginary components of the eigenfaces represented as color images. First row shows the real and imaginary parts of the mean face image $\widetilde{\Gamma}$ of the training data.

to obtain an m-dimensional projected vector **u**. The main problem here is how to determine a good projection vector **v.** The 2DPCA algorithm proves that the best projection vector **v** is the eigenvector of the image covariance matrix $\mathbf{G}_{2D}$ corresponding to the largest eigenvalue. $\mathbf{G}_{2D}$ can be evaluated as:

$$\mathbf{G}_{2D} = \frac{1}{M}\sum_{s=1}^{M}\left(\mathbf{\Phi}_s - \widetilde{\mathbf{\Phi}}\right)^H\left(\mathbf{\Phi}_s - \widetilde{\mathbf{\Phi}}\right), \tag{30}$$

where $\mathbf{\Phi}_s$ is the s^{th} training image, $\widetilde{\mathbf{\Phi}}$ is the average image of all training samples, and $(.)^H$ operator denotes the Hermitian transpose. Selecting only one optimal projection vector is not enough, therefore a set of the optimal projection vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_t$ corresponding to the largest t eigenvalues are selected. Then the t principal components $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_t$ of the image $\mathbf{\Phi}$ are obtained from:

$$\mathbf{u}_s = \left(\mathbf{\Phi} - \widetilde{\mathbf{\Phi}}\right)\mathbf{v}_s, \quad s = 1, 2, \ldots, t. \tag{31}$$

It should be noted that the principal component $\mathbf{u}_s \in \mathbb{B}^m$ is a B vector, while the principal components of BPCA (from (25)) are scalar B numbers.

For classification, suppose we have an image $\mathbf{\Phi}$ with principal components $\mathbf{u_1}, \mathbf{u_2}, \ldots, \mathbf{u}_t$. Suppose that $\mathbf{u}_1^j, \mathbf{u}_2^j, \ldots, \mathbf{u}_t^j$ are the principal components of the j-th training image $\mathbf{\Phi}_j$. Then the class of the j-th training image that minimizes

$$d\left(\mathbf{\Phi}, \mathbf{\Phi}_j\right) = \left(\sum_{t=1}^{t}\sum_{l=1}^{m}\left|\mathbf{u}_{tl} - \mathbf{u}_{tl}^j\right|^2\right)^{1/2}, \tag{32}$$

where $|.|$ denotes the norm described in(14), is the class that $\mathbf{\Phi}$ belongs to (i.e., nearest neighbor classification).

4 Experimental Results

The performance of our color face recognition methods is evaluated using three standard, public color face databases. The GATech database [18] was used to examine the performance when the facial expression, lighting condition, and scale are varied. The FERET database [19] is used to evaluate the system performance where there is a variation in viewing direction. The FEI database [20] is employed to test the performance under conditions where the pose and expression are varied. Eight face recognition methods are compared in our experiments covering two groups of methods. The first group includes 1D PCA-based methods, such as the grayscale PCA[1], CPCA[3], the quaternion-based QPCA [4,6] (our implementation is based on the quaternion Matlab toolbox [25]), and the proposed reduced biquaternion-based method (BPCA). The second group consists of 2D PCA-based methods: grayscale 2DPCA [2], C2DPCA [3], and the proposed 2DBPCA. We added to this group our own development of a new version (called 2DQPCA) of the QPCA [4,6] algorithm based on the 2DPCA algorithm. We compare between all these methods in terms of recognition accuracy and run times. All methods are implemented in Matlab 2015 and run on an Intel core i7 CPU 2.5GHz, 8GB RAM pc. Due to space constraints, we report in this paper the results on the FERET and FEI databases.

4.1 Experiment on FEI Database

The FEI Database [20] contains 2800 images of 200 individuals, each with several head poses spanning about 180 degrees. Each image is 640×480 pixels. In this experiment we used 65 individuals (because the quaternion Matlab toolbox [25] in QPCA could not handle more subjects; other implemented methods did not have this constraint) with the head poses indicated in Figure 2, where figures c, e, h, k, and m are selected for training, and the remaining samples are selected for testing. All images are resized to 30×40 pixels.

The accuracy results of the 1D methods are graphed in Figure 3 when the number of eigenfaces is varied from 10 to 90 in steps of 20. This figure indicates that the performances of BPCA and CPCA are much better than QPCA and PCA. The performance of BPCA is slightly better than CPCA, where the average recognition accuracy of BPCA was 92.85% versus 91.73% for CPCA. Figure 4 shows the results of the 2D methods. 2DPCA still has the worst accuracy. C2DPCA and 2DQPCA have close performances achieving a peak accuracy of 95.85% at 3 eigenvalues. 2DBPCA has a solid performance peaking to an accuracy of 98% using 7 eigenvalues.

We also study the time (in seconds) taken for feature extraction and classification of test images by each method. Expectedly, the grayscale-based methods (PCA and 2DPCA) are the fastest due to the fewer and simpler computations involved. At 90 eigenfaces, PCA and CPCA need 17.09s and 70.20s, respectively, while BPCA and QPCA take 136.20s and 134.31s, respectively. It is

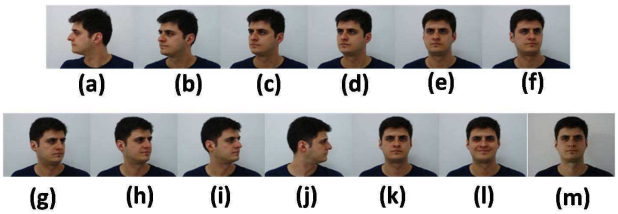

Figure 2: Face images taken from different viewing angles for one person in the FEI database.

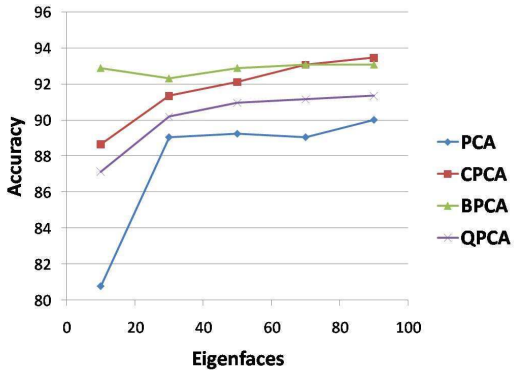

Figure 3: Accuracy of 1D methods on FEI database.

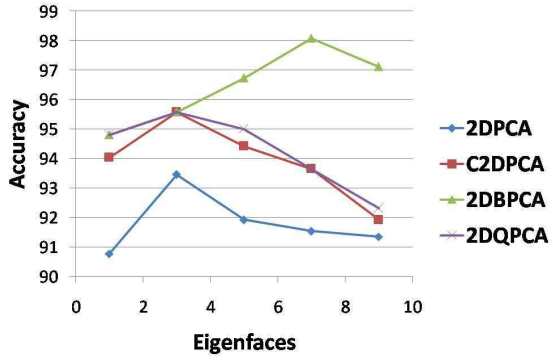

Figure 4: Accuracy of 2D methods on FEI database.

important to stress that without using Algorithm 1 (i.e., using plain selection on M^2 eigenvalues) BPCA would take 221.37s. 2D methods are generally slower than the 1D methods since the principal components of 1D methods are scalars while those of 2D methods are vectors. Using only 9 eigenfaces, 2DPCA needs 21.23s whereas C2DPCA, 2DBPCA, and 2DQPCA need 973.35s, 791.37s, and 948.86s, respectively.

4.2 Experiments on FERET Database

The FERET database [19] contains a large amount of face images. We construct a subset of the database consisting of 115 individual, each having four images. Figure 5shows some samples of this database. Two images (fa and rb) are selected for training while fb and rc are selected for testing. All images are resized to 48×32 pixels.Figure 6shows the recognition accuracies of the 1D methods when the number of eigenfaces is changed till 90 in steps of 20. This experiment shows that PCA still has the worst accuracy, while CPCA and QPCA have better and very close performances. BPCA starts off with a close performance to PCA, but improves rapidly after 30 eigenfaces and beyond. The accuracy of BPCA reaches 74.35% at 90 eigenfaces, while the respective accuracies of CPCA and QPCA are only 71.3% and 70.44% using the same number of eigenfaces. The 2D group performance is illustrated in Figure 7when the number of eigenfaces is changed from 1 to 9 in steps of 2. One can notice that 2DBPCA has the highest overall accuracy. The remaining methods have significantly lower performances that tend to decrease somewhat when adding more eigenfaces.

Similar to the previous experiment, PCA and CPCA are still the fastest where they take 5.52s and 27.58s using 90 eigenfaces while BPCA and QPCA need 56.27s and 41.81s, respectively. 2DBPCA is (about 1.25x) faster than 2DQPCA as 2DBPCA takes 254.49s using 9 eigenfaces while 2DQPCA take 317.26s.

5 Conclusions

In this paper, we have introduced the theory of reduced biquaternions to represent color images and to develop efficient vector processing methods. *The representation is quite new and expands the set of useful tools for color image processing*. Moreover, we have derived several new mathematical results on the computation of the eigenvalues/eigenvectors of a B matrix and their norm where we proved that the squared norm of eigenvalue of a B matrix equals product of norms of two complex numbers. Also we developed a lower-complexity, faster algorithm to find the best eigenvectors based on a novel transformation of the problem into X+Y sorting problem. We have applied the developed theory—*for the first time*— to the problem of color face recognition in a PCA framework. After developing a 1D PCA method, we have devised a 2D version to exploit the natural spatial information of the face images.

Our extensive experimental results on three public-domain benchmark face datasets show that the proposed method of 2DBPCA has achieved the best overall performance among all methods. The method exploits the natural spatial information of the face images as well as the color information. It also proved to be faster than the quaternion-based counterpart method [4, 6]. Our current research is directed to building upon the developed B theory to develop several novel color image processing solutions. One underway direction is to apply the B theory into a collaborative and sparse representation framework [5] for better color face recognition.

Acknowledgement. This research is supported by Information Technology Industry Development Agency of Egypt (project # ARP2016.R21.10).

6 References

1. Turk, M. and A. Pentland, *Eigenfaces for recognition.* J. Cognitive Neuroscience, 1991. **3**(1): p. 71-86.

Figure 5: Sample images of FERET database. (fa) and (fb) represent frontal view, while (rb) and (rc) represent head turned about 15 degree in different directions.

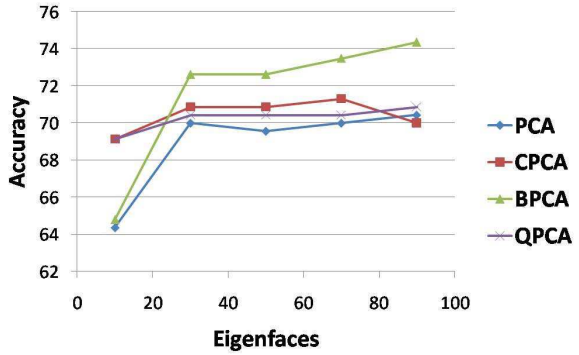

Figure 6: Accuracy of 1D methods on FERET database.

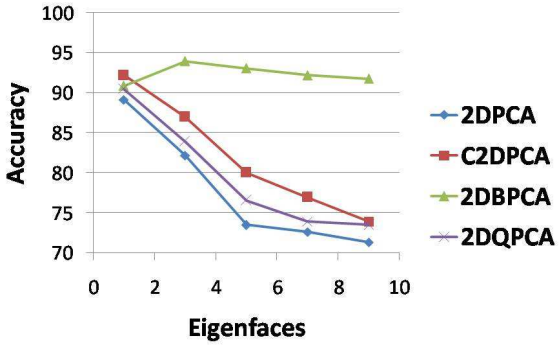

Figure 7: Accuracy of 2D methods on FERET database.

2. Jian, Y., et al., *Two-dimensional PCA: a new approach to appearance-based face representation and recognition.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2004. **26**(1): p. 131-137.
3. Xiang, Yang, and Q. Chen, *Color face recognition by PCA-like approach.* Neurocomputing, 2015. **152**: p. 231-235.
4. Bihan, N.L. and S.J. Sangwine. *Quaternion principal component analysis of color images*. in *Proceedings 2003 International Conference on Image Processing (Cat. No.03CH37429)*. 2003.
5. Zou, C., K.I. Kou, and Y. Wang, *Quaternion Collaborative and Sparse Representation With Application to Color Face Recognition.* IEEE Transactions on Image Processing, 2016. **25**(7): p. 3287-3302.
6. Shi, L., *Exploration in quaternion colour*. 2005, School of Computing Science-Simon Fraser University.
7. Hamilton, W.R. and W.E. Hamilton, *Elements of quaternions*. 1866, London,: Longmans, Green, & co. vi p., 1 l., lix, 762 p.
8. Schutte, H.D. and J. Wenzel. *Hypercomplex numbers in digital signal processing*. in *IEEE International Symposium on Circuits and Systems*. 1990.
9. Ell, T.A. *Quaternion-Fourier transforms for analysis of two-dimensional linear time-invariant partial differential systems*. in *Proceedings of 32nd IEEE Conference on Decision and Control*. 1993.
10. Davenport, C.M., *A commutative hypercomplex algebra with associated function theory*, in *Clifford algebras with numeric and symbolic computations*. 1996, Birkhauser Boston Inc. p. 213-227.
11. Bülow, T., *Hypercomplex spectral signal representations for the processing and analysis of images*. 1999, Christian-Albrechts-Universität: Kiel.
12. Huang, L. and W. So, *On left eigenvalues of a quaternionic matrix.* Linear Algebra and its Applications, 2001. **323**(1): p. 105-116.
13. Pei, S.C., et al., *Eigenvalues and Singular Value Decompositions of Reduced Biquaternion Matrices.* IEEE Transactions on Circuits and Systems I: Regular Papers, 2008. **55**(9): p. 2673-2685.
14. Wood, R.M.W., *Quaternionic Eigenvalues.* Bulletin of the London Mathematical Society, 1985. **17**(2): p. 137-138.
15. Soo-Chang, Ja-Han, and D. Jian-Jiun, *Commutative reduced biquaternions and their Fourier transform for signal and image processing applications.* IEEE Transactions on Signal Processing, 2004. **52**(7): p. 2012-2031.
16. Mitrinović, D.S., J.E. Pečarić, and A.M. Fink, *Classical and new inequalities in analysis*. Mathematics and its applications East European series. 1993, Dordrecht ; Boston: Kluwer Academic. xvii, 740 p.
17. Naber, G.L., *The geometry of Minkowski spacetime : an introduction to the mathematics of the special theory of relativity*. Second edition. ed. Applied mathematical sciences. 2012, New York: Springer. xvi, 324 pages.
18. *GATech database*. Available from: http://www.anefian.com/research/face_reco.htm.
19. Phillips, P.J., et al., *The FERET database and evaluation procedure for face-recognition algorithms.* Image and Vision Computing, 1998. **16**(5): p. 295-306.
20. *FEI face database*. Available from: http://fei.edu.br/~cet/facedatabase.html
21. A.Yip, P.Sinha, *Role of color in face recognition*, Journal of Vision, , Vol.2(7), November 2002.
22. Donald Knuth. *The Art of Computer Programming*, Vol. 3: Sorting and Searching, 3rd Edition. Addison-Wesley, 1997.
23. Jean-Luc Lambert, *Sorting the sums* $(x_i + y_j)$ *in* $O(n^2)$ *comparisons,* Theoretical Computer Science. Vol. 103 (1): 137–141, 1992.
24. A. Mirzaian, E. Arjomandi, Selection in X + Y and matrices with sorted rows and columns, Information Processing Letters, Vol. 20(1), Issue 1, Pages 13-17, 1985.
25. Quaternion and octonion toolbox for Matlab. Available from: http://qtfm.sourceforge.net/.