Assiut University Journal of Multidisciplinary Scientific Research (AUNJMSR)
Faculty of Science, Assiut University, Assiut, Egypt.
Printed ISSN: 2812-5029
Online ISSN: 2812-5037
Vol. 51(3): 358 – 373 (2022)
https://aunj.journals.ekb.eg/

Face Recognition by Principal Component Regression using Hypercomplex Numbers

Moumen T. El-Melegy [1], Aliaa T. Kamal [2*], Khaled F. Hussain[2], H. M. El-Hawary[3]

Affiliation 1; Electrical Engineering Department, Faculty of Engineering, Assiut University, Assiut, Egypt
Affiliation 2; Computer Science Department, Faculty of Computers and Information, Assiut University, Assiut, Egypt
Affiliation 3; Mathematics Department, Faculty of Science, Assiut University, Assiut, Egypt
[*]Corresponding Author: aliaa_t@aun.edu.eg.

ARTICLE INFO
Article History:
Received: 2022-04-05
Accepted: 2022-06-16
Online: 2022-08-30

Keywords:
Face recognition, Linear regression classification, Principal component analysis, Hypercomplex numbers.

ABSTRACT

In this paper, we propose a classification by principal component regression (CbPCR) strategy, which depends on performing regression of each data class in terms of its principal components. This CbPCR formulation leads to a novel formulation of the Linear Regression Classification (LRC) problem that keeps the key information of the data classes while providing more compact closed-form solutions. We also extend this strategy to the 4D hypercomplex domains to take into account the color information of the image. Our experiments on two color face recognition benchmark databases prove the efficacy of the proposed strategy.

INTRODUCTION

Linear regression is one of the simplest and widely used machine learning algorithms that has received a lot of attention in many fields, such as pattern recognition, facial recognition, pose estimation, information security, and image/video processing. Linear regression is a mathematical test for evaluating and modeling the relationships between dependent and independent variables.

In 2010, Naseem et al [1] proposed a Linear Regression Classification (LRC) algorithm that represents each class's training images independently in a linear regression relationship. The algorithm depends on applying the least squares method to find the regression coefficient then decides the class label with the smallest reconstruction error. To enhance its performance, Huang and Yang [2] and Yani et al. [3] proposed to apply principal component analysis (PCA) [4] first to extract the vital information from images and reduce the feature vector dimensions. Then, the original data are transformed into a low-dimensional subspace. Finally, LRC is performed on the projected data.

The previous methods [1]–[3]works in principle on grayscale, single-channel images. They can work on color images after converting them to grayscale images, thus giving away to the important color information. Some methods (e.g., [5]) apply LRC to every color channel separately, then select the class having the smallest total prediction error over all color channels. Unlike all those methods, several

studies [6]–[12], have suggested the use of 4D hypercomplex numbers to represent color images. This depends on treating the color components of each image pixel as one entity thus considering the correlation between color components. Out of those studies, two have addressed the LRC problem. Zou et al. [9] proposes a quaternion LRC (QLRC) method that extends the classical LRC algorithm to quaternion space. QLRC relies on converting the quaternion quantities to real ones to circumvent using quaternion derivatives. In our recent paper [12], we develop closed-form solutions of QLRC from the principles of quaternion calculus. In addition, we propose new solutions based on reduced biquaternions (RBs), which represent another hypercomplex space with 3 imaginary components and one real. In addition to having commutative algebra—in contrast to quaternions—RBs may be represented using the so-called e_1-e_2 form [13] that can lead to more efficient and faster computation.

In this paper, we extend LRC by proposing a novel strategy for classification by performing regression of data in terms of its principal components. This leads to a novel formulation of the LRC problem, for which we derive a new closed-form solution. This Classification by Principal Component Regression (CbPCR) formulation keeps the key information of the data classes and removes superfluous and correlated details while providing more compact solution. We show in real experiments on public face recognition benchmark databases that our strategy outperforms the original LRC method [1] and its more recent variants [2, 3]. Another contribution of this paper is that we extend our proposed CbPCR formulation to both the quaternion and RB domains to process color images. To that end, we exploit an efficient algorithm we derive in [8] for computing the principal components (eigenvectors) of an RB matrix by casting it into an **x** + **y** selection problem [14, 15]. In our experimental results on public benchmark databases for *color* face recognition, we demonstrate the better performances of the new quaternion and RB-based CbPCR algorithms over existing and competing ones [5, 9, 12].

1. HYPERCOMPLEX DOMAINS

We first review here the basic fundamentals of the 4D hypercomplex domains of quaternions and RBs. We denote scalars and vectors using italic and bold lowercase letters, respectively, while matrices are indicated by bold uppercase letters. The number of dots on top of a symbol indicates its intended domain: real ($\mathbb{R}$), complex ($\mathbb{C}$), quaternion ($\mathbb{Q}$), or reduced biquaternion ($\mathbb{B}$). Symbols without any dots on top indicate real or complex quantities, where the intended domain is disambiguated by examining the context. A quaternion is represented by a symbol with one dot above, while an RB quantity has two dots on top.

1.1 Quaternions

A quaternion number consists of one real and three imaginary parts: $\dot{q} = q_r + q_i i + q_j j + q_k k$ (a quaternion number with no real part is called a pure quaternion). The three imaginary parts satisfy:

$$\begin{aligned} i^2 = j^2 = k^2 = -1, \qquad ij = -ji = k, \\ jk = -kj = i, \qquad ki = -ik = j. \end{aligned} \tag{1}$$

The quaternion conjugate is $\bar{\dot{q}} = q_r - q_i i - q_j j - q_k k$, and the quaternion norm is $|\dot{q}| = \sqrt{\dot{q}\bar{\dot{q}}}$. The Hermitian (conjugate transpose) satisfies $\left(\dot{\mathbf{P}}\dot{\mathbf{Q}}\right)^H = \dot{\mathbf{Q}}^H\dot{\mathbf{P}}^H$, see [16] for more details.

Due to the noncommutativity of quaternion multiplication, a quaternion matrix has left and right eigenvalues that may be different [13, 17]. Any quaternion matrix $\dot{\mathbf{Q}} \in \mathbb{Q}^{n\times n}$ can be defined as: $\dot{\mathbf{Q}} = \mathbf{Q}_1 + \mathbf{Q}_2 j$, where $\mathbf{Q}_1 = \mathbf{Q}_r + i\mathbf{Q}_i$ and $\mathbf{Q}_2 = \mathbf{Q}_j + i\mathbf{Q}_k$, with $\mathbf{Q}_r, \mathbf{Q}_i, \mathbf{Q}_j$, and $\mathbf{Q}_k$ being the real and the three imaginary parts of the quaternion matrix $\dot{\mathbf{Q}}$. The eigenvalues (and eigenvectors) can be calculated from the equivalent complex matrix [18]:

$$\mathcal{H}\left(\dot{\mathbf{Q}}\right) := \begin{bmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \\ -\overline{\mathbf{Q}_2} & \overline{\mathbf{Q}_1} \end{bmatrix} \in \mathbb{C}^{2n\times 2n}, \tag{2}$$

where the mapping $\mathcal{H}(.)$ transforms a quaternion matrix into an equivalent complex matrix. Thus, there are $2n$ eigenvalues (eigenvectors) for any $n \times n$ quaternion matrix.

1.2 Reduced Biquaternions

A reduced biquaternion number also has one real and three imaginary parts: $\ddot{p} = p_r + p_i i + p_j j + p_k k$, were

$$j^2 = 1, i^2 = k^2 = -1, kj = jk = i, \\ ji = ij = k, ki = ik = -j. \tag{3}$$

In contrast to quaternions, multiplication on the RB domain is commutative. There are two special numbers e_1 and e_2 [13] such that any RB number can be represented as: $\ddot{p} = p_1 e_1 + p_2 e_2$, where $e_1 = (1 + j)/2$, $e_2 = (1 - j)/2$ and $p_1 = (p_r + p_j) + i(p_i + p_k)$, $p_2 = (p_r - p_j) + i(p_i - p_k)$.

Expressing many operations in terms of the e_1-e_2 forms reduces their complexity. For example, direct RB multiplication requires 16 real multiplications while applying the e_1-e_2 form requires only 8. The RB norm and Hermitian can be defined in a similar way as quaternion numbers. The RB conjugate [19] is defined as:

$$\bar{\ddot{p}} = p_r - p_i i + p_j j - p_k k. \tag{4}$$

There are other definitions of conjugate [19]–[22] while (4) is the only definition satisfying $\bar{\ddot{p}} = \bar{p_1} e_1 + \bar{p_2} e_2$.

For any $n \times n$ RB matrix there are n^2 eigenvalues and eigenvectors (see [8, 13] for proof). The computation of n^2 eigenvalues and their corresponding eigenvectors would increase the computational cost required to find these eigenvalues. The time complexity of finding the t largest eigenvalues will be $O(n^2 t)$. We derive a more efficient algorithm for this purpose in [8] based on the well-known computer science problem **x** + **y** selection [14, 15] with time complexity of $O(n\, t + n \log n)$.

For an $m \times n$ RB matrix $\ddot{\mathbf{P}}$, the Frobenius norm $|\ddot{\mathbf{P}}| = \frac{1}{\sqrt{2}} |\mathcal{M}(\ddot{\mathbf{P}})|$ [12] where $\mathcal{M}(.)$ maps any RB matrix to its complex equivalent matrix:

$$\mathcal{M}(\ddot{\mathbf{P}}) := \begin{bmatrix} \mathbf{P}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_2 \end{bmatrix} \in \mathbb{C}^{2m \times 2n}, \tag{5}$$

where $\mathbf{P}_1$ and $\mathbf{P}_2 \in \mathbb{C}^{m \times n}$ and are defined as $\mathbf{P}_1 = (\mathbf{P}_r + \mathbf{P}_j) + i(\mathbf{P}_i + \mathbf{P}_k)$, $\mathbf{P}_2 = (\mathbf{P}_r - \mathbf{P}_j) + i(\mathbf{P}_i - \mathbf{P}_k)$.

2. PROPOSED METHODS

In this section, we review LRC [1] and derive the CbPCR for grayscale images and then for color images using the theory of quaternions and RBs.

2.1 Linear Regression Classification (LRC)

Suppose there are L classes in the training set where the *l-th* class consists of n_l samples. Each $m \times n$ grayscale image **X** is represented as a 1D vector $\mathbf{x} \in \mathbb{R}^{mn}$. $\mathbf{A}_l = [\mathbf{x}_1^l, \mathbf{x}_2^l, \dots, \mathbf{x}_{n_l}^l] \in \mathbb{R}^{mn \times n_l}$ is a matrix that represents samples from the l-th class. The goal is to infer the correct label of any new image $\mathbf{y} \in \mathbb{R}^{mn}$. LRC [1] seeks to represent **y** as a linear combination of the training images of each class by setting up the following real-valued linear regression problem:

$$\min_{\mathbf{c}_l \in \mathbb{R}^{n_l}} |\mathbf{y} - \mathbf{A}_l \mathbf{c}_l|^2, \quad l = 1, 2, \dots, L, \tag{6}$$

for which a closed-form solution is found via

$$\hat{\mathbf{c}}_l = (\mathbf{A}_l^T \mathbf{A}_l)^{-1} \mathbf{A}_l^T \mathbf{y}. \tag{7}$$

Then, the predicted vector $\hat{\mathbf{y}}_l$ is given by:

$$\hat{\mathbf{y}}_l = \mathbf{A}_l \hat{\mathbf{c}}_l, \quad l = 1, 2, \dots, L. \tag{8}$$

The test sample **y** is eventually assigned to the class with the minimal distance

$$\min_l |\mathbf{y} - \hat{\mathbf{y}}_l|, \quad l = 1, 2, \dots, L. \tag{9}$$

2.2 Classification by Principal Component Regression (CbPCR)

The idea in our proposed methods stems from that an image (column) in $\mathbf{A}_l$ can be represented as [4]

$$\tilde{\mathbf{x}}_g^l = \boldsymbol{\mu}_l + \sum_{b=1}^{t} w_g^b \mathbf{U}_l^b, \tag{10}$$

where $\boldsymbol{\mu}_l$ is the mean vector of the *l*-th class, and $\mathbf{U}_l \in \mathbb{R}^{mn \times t}$ represents the t largest principal components of the class scatter matrix $\bar{\mathbf{A}}_l \bar{\mathbf{A}}_l^T$ with $\bar{\mathbf{A}}_l = [(\mathbf{x}_1^l - \boldsymbol{\mu}_l), (\mathbf{x}_2^l - \boldsymbol{\mu}_l), \dots, (\mathbf{x}_{n_l}^l - \boldsymbol{\mu}_l)] \in \mathbb{R}^{mn \times n_l}$. That is, an image can be represented as a linear combination of the Eigen-components of the class-specific scatter matrix. The real weights $\{w_g^b\}_{b=1}^t$ represent the projections of the *g*-th image along these components. Huang and Yang [2] and Yani et al. [3] apply the standard LRC on these weights. Nevertheless, we setup our CbPCR model as

$$\min_{\mathbf{c}_l \in \mathbb{R}^t} |\mathbf{y} - \boldsymbol{\mu}_l - \mathbf{U}_l \mathbf{c}_l|^2, \quad l = 1, 2, \dots, L. \tag{11}$$

Forcing the gradient of the objective function (11) with respect to $\mathbf{c}_l$ to vanish, the closed form is found as:

$$\hat{\mathbf{c}}_l = (\mathbf{U}_l^T \mathbf{U}_l)^{-1} \mathbf{U}_l^T (\boldsymbol{y} - \boldsymbol{\mu}_l). \tag{12}$$

Since $\mathbf{U}_l$ is orthonormal,

$$\hat{\mathbf{c}}_l = \mathbf{U}_l^T (\mathbf{y} - \boldsymbol{\mu}_l). \tag{13}$$

The response vector $\hat{\mathbf{y}}_l$ is predicted as:

$$\hat{\mathbf{y}}_l = \mathbf{U}_l \hat{\mathbf{c}}_l + \boldsymbol{\mu}_l, \quad l = 1, 2, \dots, \mathrm{L}. \tag{14}$$

The distance between **y** and the predicted response vector $\hat{\mathbf{y}}_l$ is computed as:

$$d_l(\boldsymbol{y}) = |\mathbf{y} - \hat{\mathbf{y}}_l|, \quad l = 1, 2, \dots, \mathrm{L}. \tag{15}$$

The test image **y** is decided to belong to the class minimizing (15).

In order to reduce the computation burden of finding the largest principal components of the class scatter matrix, we follow the common practice [4] of finding first the eigenvectors $\mathbf{V}_l \in \mathbb{R}^{n_l \times t}$ of the matrix $\bar{\mathbf{A}}_l^T \bar{\mathbf{A}}_l$, and then the target eigenvectors are computed as:

$$\mathbf{U}_l = \bar{\mathbf{A}}_l \mathbf{V}_l. \; l = 1, 2, \dots, L. \tag{16}$$

The complete CbPCR procedure is presented in Algorithm 1**:**.

Algorithm 1:

Input: dataset consists of L classes, each containing n_l samples.

Output: The label of a new image $\mathbf{y} \in \mathbb{R}^{mn}$.

1. Represent each image in the training set as a 1D vector $\mathbf{x} \in \mathbb{R}^{mn}$.
2. For each class *l* do
    a. Represent samples from the *l*-th class as a matrix $\mathbf{A}_l \in \mathbb{R}^{mn \times n_l}$.
    b. Calculate the class scatter matrix $\bar{\mathbf{A}}_l \bar{\mathbf{A}}_l^T$, with $\bar{\mathbf{A}}_l$ represents the difference between the mean vector $\boldsymbol{\mu}_l$ of the *l-th* class and column of $\mathbf{A}_l$.
    c. Calculate eigenvectors from the class scatter matrix.
    d. Calculate the closed form solution (13).
    e. Calculate the reconstructed query color image $\hat{\mathbf{y}}_l$ from (14).
    f. Compute the distance $d_l(\boldsymbol{y})$ between **y** and the predicted response vector $\hat{\mathbf{y}}_l$ from (15).
3. The test image **y** is decided to belong to the class with the minimum distance *d*.

2.3 Quaternion-based CbPCR

Q-CbPCR is based on the algebra and calculus of quaternion matrices [23] to identify the class to which a new color image belongs. Each $m \times n$ training color image is portrayed as a 1D pure quaternion vector $\dot{\mathbf{x}} \in \mathbb{Q}^{mn}$. The columns of matrix $\dot{\mathbf{A}}_l \in \mathbb{Q}^{mn \times n_l}$ represent samples from the *l*-th class where each column represents the difference between the training image and the class mean $\dot{\boldsymbol{\mu}}_l$. Our goal is to infer the correct label of a query color image $\dot{\mathbf{y}} \in \mathbb{Q}^{mn}$ from the training data matrices and their labels. The test image $\dot{\mathbf{y}}$ is represented by setting up the following quaternion regression problem:

$$\min_{\dot{\mathbf{c}}_l \in \mathbb{Q}^t} \left| \dot{\boldsymbol{y}} - \dot{\boldsymbol{\mu}}_l - \dot{\mathbf{U}}_l \dot{\mathbf{c}}_l \right|^2, \quad l = 1, 2, \dots, L, \tag{17}$$

where $\dot{\mathbf{U}}_l$ encompasses the key eigenvectors of the quaternion scatter matrix $\dot{\mathbf{A}}_l \dot{\mathbf{A}}_l^H$ corresponding to the t largest eigenvalues in terms of the *quaternion* norm as computed by a QPCA technique [6].

Proposition 1: The closed-form solution of (17) is $\hat{\dot{\boldsymbol{c}}}_l = \dot{\boldsymbol{U}}_l^H (\dot{\boldsymbol{y}}_l - \dot{\boldsymbol{\mu}}_l)$.
Proof: See Appendix A.

The reconstructed query color image $\dot{\mathbf{y}}_l$ is computed by:

$$\hat{\dot{\mathbf{y}}}_l = \dot{\mathbf{U}}_l \hat{\dot{\mathbf{c}}}_l + \dot{\boldsymbol{\mu}}_l. \tag{18}$$

Eventually, $\dot{\mathbf{y}}$ is given the label of the class with the minimal *quaternion* norm

$$\min_l \left| \dot{\boldsymbol{y}} - \hat{\dot{\mathbf{y}}}_l \right|, \quad l = 1, 2, \dots, L. \tag{19}$$

The proposed quaternion-based representation is depicted in Figure 1, where a color face image is represented as a linear combination of the mean image and the best t eigenvectors of its class.

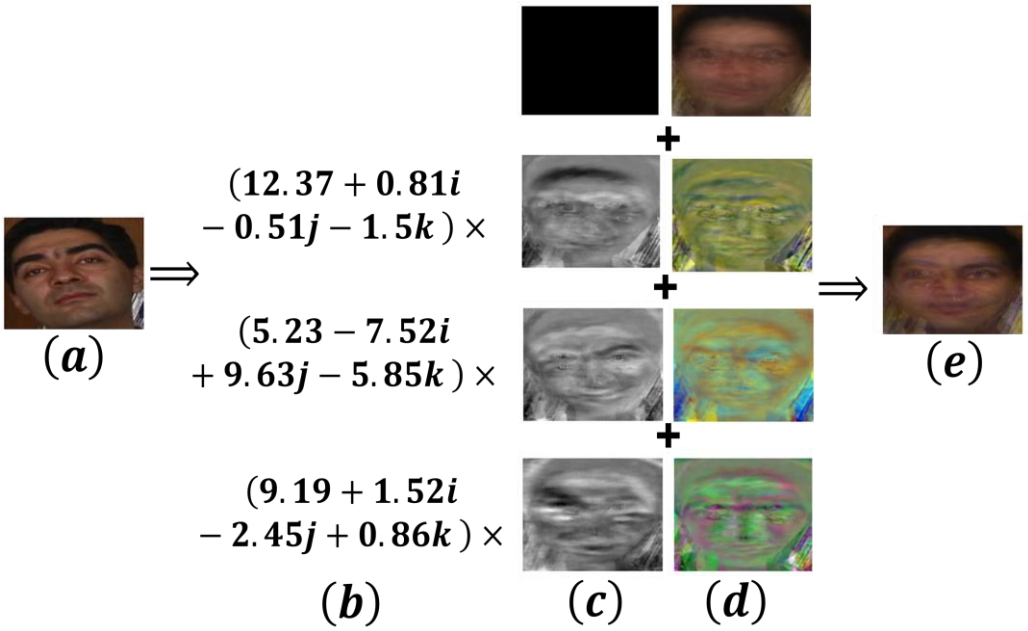


Figure 1: Quaternion-based color representation: Column (a) shows original color image. Column (b) gives the closed-form solution of (17) with t =3. Column (c) depicts the real parts of the principal components in $\dot{\mathbf{U}}_l$ as grayscale images. Column (d) depicts the imaginary parts of the principal components as color images. First row depicts the real (zero) and imaginary parts of the mean of the training images. Column (e) represents the reconstructed image from (18).

2.4 Reduced Biquaternion-based CbPCR

Analogously, RB-CbPCR relies on color image representation using RBs in place of quaternions. Our goal here is to find the correct label of a query color image $\ddot{\mathbf{y}} \in \mathbb{B}^{mn}$ from the given training data matrices $\ddot{\mathbf{A}}_l \in \mathbb{B}^{mn \times n_l}$ and their true labels. The proposed RB-CbPCR model is set up as:

$$\min_{\ddot{\mathbf{c}}_l \in \mathbb{B}^t} \left| \ddot{\boldsymbol{y}} - \ddot{\boldsymbol{\mu}}_l - \ddot{\mathbf{U}}_l \ddot{\mathbf{c}}_l \right|^2, \quad l = 1, 2, \dots, L. \tag{20}$$

where $\ddot{\boldsymbol{\mu}}_l$ is the mean of the l-th class, and $\ddot{\mathbf{U}}_l$ represents the eigenvectors of the RB scatter matrix $\ddot{\mathbf{A}}_l \ddot{\mathbf{A}}_l^H$ corresponding to the t largest eigenvalues in terms of the RB norm as obtained via our efficient RB-based PCA algorithm [8]. By Lemma 2 in [12], it can be proved that (20) is equivalent to:

$$\min \left| \mathcal{M}(\ddot{\mathbf{y}}) - \mathcal{M}(\ddot{\boldsymbol{\mu}}_l) - \mathcal{M}\left(\ddot{\mathbf{U}}_l\right) \mathcal{M}(\ddot{\boldsymbol{c}}_l) \right|^2, \tag{21}$$

where a closed-form solution is derived in Proposition 2.

Proposition 2: The closed-form solution of (20) is $\hat{\ddot{\mathbf{c}}}_l = \ddot{\mathbf{U}}_l^H (\ddot{\mathbf{y}} - \ddot{\boldsymbol{\mu}}_l)$.
Proof: See Appendix B.

The class-specific reconstructed test image is

$$\hat{\ddot{\mathbf{y}}}_l = \ddot{\mathbf{U}}_l \hat{\ddot{\mathbf{c}}}_l + \ddot{\boldsymbol{\mu}}_l. \quad (22)$$

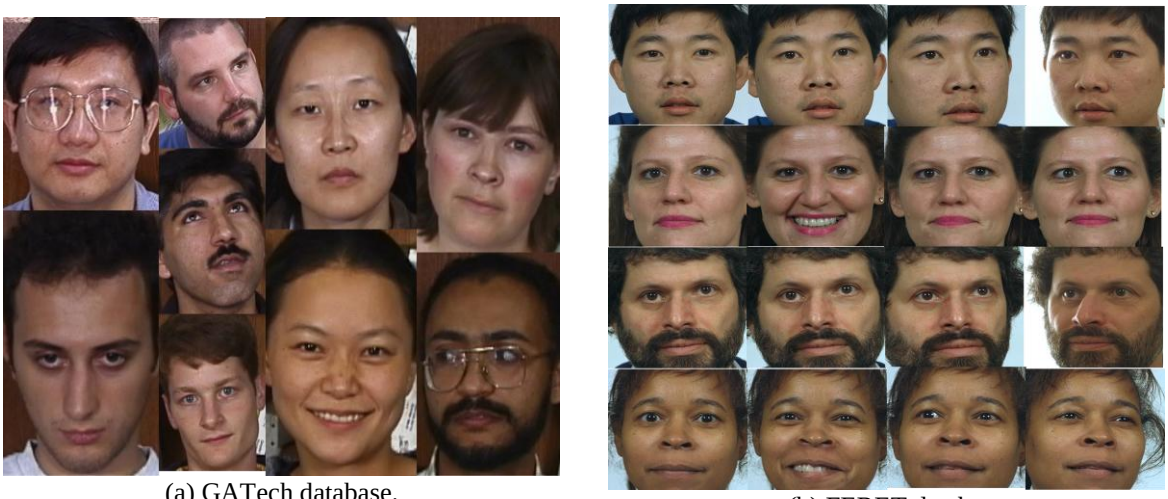
The test image $\ddot{\mathbf{y}}$ is finally labeled to the class with the minimal RB norm

$$\min_l \left| \ddot{\mathbf{y}} - \hat{\ddot{\mathbf{y}}}_l \right|, \quad l = 1, 2, \ldots, L. \quad (23)$$

3. EXPERIMENTS

In this section, the proposed methods are evaluated on two color face recognition benchmark databases: the GATech database [24] and the FERET database [25]. We compare the proposed CbPCR, Q-CbPCR, and RB-CbPCR methods with LRC [1], the quaternion-based QLRC method [9, 12], the RB-based RBLRC method [12], IPCRC [2, 3], and the CLRC method [5]. Note that LRC, IPCRC and CbPCR work on grayscale images, while the other methods operate on color images.

All experiments are carried out on a pc with an Intel i7 CPU at 2.5GHz with 8GB RAM using MATLAB 2015. Quaternion computations are done using the quaternion MATLAB toolbox [26], while RB computations are carried out using our own MATLAB toolbox.



(a) GATech database. (b) FERET database.

Figure 2: Sample images from GATech and FERET databases.

3.1 Experiments on GATech database

The GATech database [24] consists of 50 (but only 38 are available to us) subjects with 15 images per subject taken in two or three sessions. It experiences several variations in facial expression, pose, illumination, and scale; see Figure 2(a). Following [27], we use 10 images from each subject for training and the remaining 5 for testing.

We study two factors on the recognition performance: the image size and the number of principal components t. Figure 3 graphs the rank-1 recognition accuracy by all methods versus the image size ranging from 5% to 100% (size 54×39 pixels) in steps of 5%. We fix t=7 per class in our proposed methods and use 266 principal components from the whole training data in the IPCRC method. Expectedly, the performances of the grayscale-based methods are generally worse than those of color-based ones. CbPCR achieves 3% and 4.7% improvement over LRC and IPCRC, respectively. IPCRC is the worst among the three and achieves a peak accuracy of 84.2% compared with respective peak accuracies of 88.9% and 85.8% by CbPCR and LRC.

The color methods offer better performances than the grayscale-based methods except for CLRC which has worse accuracy than CbPCR and a close-performance to LRC with a peak accuracy of 85.8%. The new grayscale-based CbPCR method has a better performance than QLRC and RBLRC till image size 50% then has almost the same performance afterwards. The new Q-CbPCR and RB-CbPCR have a close-performance that is better than all other methods. Both have a peak accuracy of 88.95% yielding about 2.28% improvement over QLRC and RBLRC.

Figure 4 shows the rank-1 recognition accuracy of all methods against $t = 1$ to 9 fixing the image size at 54×39 pixels. We take the number of principal components in IPCRC as the number of classes

(38) times t. The accuracies of CLRC, LRC, QLRC, and RBLRC are not dependent on t. IPCRC is the worst overall. As more principal components are used, the proposed CbPCR, Q-CbPCR, and RB-CbPCR offer higher accuracies. For $t \leq 4$, CbPCR and RB-CbPCR have similar performance, while Q-CbPCR tops both methods. Afterwards, the performances of Q-CbPCR and RB-CbPCR are better than that of CbPCR. Q-CbPCR tends to offer a slightly better accuracy than RB-CbPCR, where the former has a peak accuracy of 88.95% versus 87.37% for the latter. CbPCR has 86.84% peak accuracy.

Finally, we study the recognition time (in seconds) by taking the average of running each algorithm 10 times on image size 54×39. We use t =7 in our proposed methods and 266 principal components in IPCRC. As shown in Table 1, CLRC is the slowest while the grayscale-based LRC and IPCRC are the fastest due to their simpler computation. RB-CbPCR is around 1.8x faster than Q-CbPCR. This is due to the faster computations of RB operations taking advantage of the e_1-e_2 form and to the faster computation of the RB principal components by our efficient RB-based PCA algorithm [8].

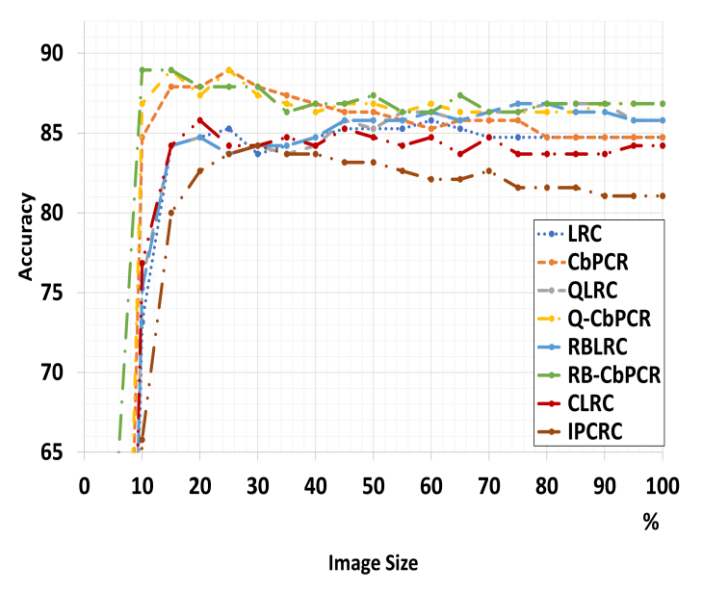


Figure 3: Rank-1 recognition accuracy on the GATech database by all methods for various image sizes. (Better viewed in color)

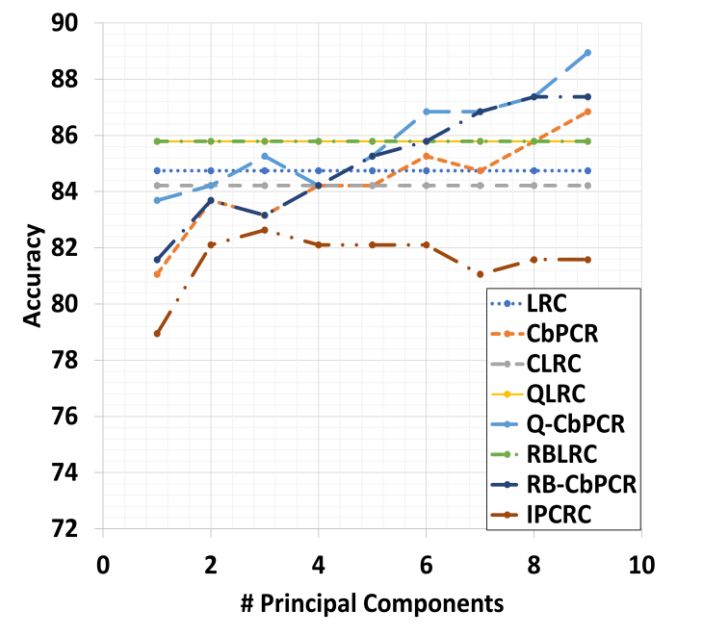


Figure 4: Rank-1 recognition accuracy on the GATech database by all methods against number of principal components. (Better viewed in color).

Table 1: Average recognition CPU time (in seconds) over 10 runs of different methods on the GATech database

LRC	CLRC	CbPCR	QLRC	Q-CbPCR	RBLRC	RB-CbPCR	IPCRC
0.018	2.860	0.105	1.631	2.011	1.182	1.121	0.083

3.2 Experiments on FERET database

The FERET database [25] contains more than 14,000 face images having pose and light variations. In this study, we consider a subset consisting of 115 subjects with 4 images from each subject captured in 3 poses, see Figure 2(b). Two faces are captured at 0° while the other two faces are captured at 15° and −15°. We choose one frontal image and one with the head rotated 15° for training, while testing is performed on the other 2 images.

Figure 5 shows the rank-1 recognition accuracy for various image sizes varying from 5% to 100% (48×32 pixels) in steps of 5%. In our proposed methods we use $t = 1$, while we select 115 principal components in IPCRC. Expectedly, the grayscale-based methods have lower accuracy. IPCRC and LRC are close to each other with respective peak accuracies of 79.6% and 80.9%. CbPCR achieves 1.79% and 1.18% improvements over LRC and IPCRC, respectively. QLRC and RBLRC exhibit nearly the same performance with a top accuracy of 81.74%. CLRC performs better than QLRC and RBLRC with a peak of 82.1%. Both Q-CbPCR and RB-CbPCR offer the best *overall* accuracy of 82.61% at 20% image size. Q-CbPCR shows a slightly better performance than RB-CbPCR for larger image sizes.

Moreover, we assess the average recognition time (in seconds) by running each algorithm 10 times on image size 48 × 32 using t =1 in our proposed methods and 115 principal components in IPCRC. As shown in Table 2, LRC and IPCRC are the fastest while CLRC is the slowest. Q-CbPCR and RB-CbPCR take less time than QLRC and RBLRC. Moreover, RB-CbPCR is about 1.5x faster than Q-CbPCR.

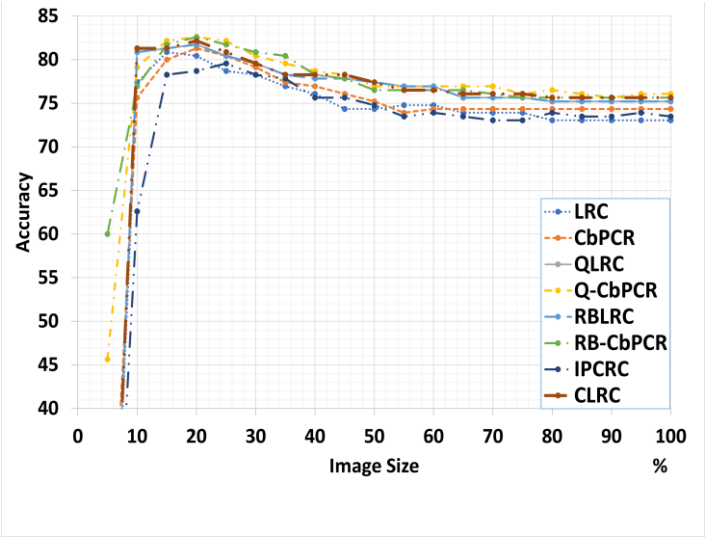


Figure 5: Rank-1 recognition accuracy on the FERET database by all methods for various image sizes. (Better viewed in color)

Table 2: Average recognition CPU time (in seconds) over 10 runs of different methods on the FERET database

LRC	CLRC	CbPCR	QLRC	Q-CbPCR	RBLRC	RB-CbPCR	IPCRC
0.038	2.642	0.145	1.910	1.782	1.494	1.224	0.06

4. Conclusion

We have proposed a novel formulation of LRC based on principal component regression. This

formulation keeps the key information of the data classes while providing more compact closed-form solutions. We have also extended it to the quaternion and RB domains to take into account the color information. The specific contributions of this paper are:

- We have proposed a CbPCR strategy that depends on performing regression of each data class in terms of its principal components.
- We have extended this CbPCR strategy to the hypercomplex domains of quaternions and RBs to consider color images.
- We have developed CbPCR closed-form solutions from the principles of real, quaternion and RB domains.
- Experiments on two color face recognition benchmark databases have showed that the proposed Q-CbPCR and RB-CbPCR have the highest overall accuracy among eight different methods including very recent ones [5, 9, 12]. Moreover, RB-CbPCR is about 1.8x faster than Q-CbPCR. The grayscale-based CbPCR algorithm has even outperformed some color-based algorithms in the literature in addition to the original grayscale LRC method [1] and its more recent variants [2, 3].

The main limitation of the proposed methodology is its reliance on the assumption of linear relationship between each class's training images and an unknown, test image if it belongs to the same class. This assumption may not hold in all practical cases. Getting around this problem is the focus of our future research.

APPENDIX A: PROOF OF PROPOSITION 1

The gradient of (17) with respect to $\dot{\mathbf{c}}_l$ is:

$$\frac{\partial}{\partial \dot{\mathbf{c}}_l}\left|\dot{\boldsymbol{y}}-\dot{\boldsymbol{\mu}}_l-\dot{\mathbf{U}}_l\dot{\mathbf{c}}_l\right|^2=\frac{\partial}{\partial \dot{\mathbf{c}}_l}\mathrm{tr}\left(\left(\dot{\boldsymbol{y}}-\dot{\boldsymbol{\mu}}_l-\dot{\mathbf{U}}_l\dot{\mathbf{c}}_l\right)^H\left(\dot{\boldsymbol{y}}-\dot{\boldsymbol{\mu}}_l-\dot{\mathbf{U}}_l\dot{\mathbf{c}}_l\right)\right)$$

$$=\frac{\partial}{\partial \dot{\mathbf{c}}_l}\mathrm{tr}\left((\dot{\boldsymbol{y}}-\dot{\boldsymbol{\mu}}_l)^H(\dot{\boldsymbol{y}}-\dot{\boldsymbol{\mu}}_l)-\dot{\mathbf{c}}_l^H\dot{\mathbf{U}}_l^H(\dot{\boldsymbol{y}}-\dot{\boldsymbol{\mu}}_l)-(\dot{\boldsymbol{y}}-\dot{\boldsymbol{\mu}}_l)^H\dot{\mathbf{U}}_l\dot{\mathbf{c}}_l+\dot{\mathbf{c}}_l^H\dot{\mathbf{U}}_l^H\dot{\mathbf{U}}_l\dot{\mathbf{c}}_l\right)$$

According to quaternion derivatives [23]:

$$\frac{\partial}{\partial \dot{\mathbf{c}}_l}\left|\dot{\boldsymbol{y}}-\dot{\boldsymbol{\mu}}_l-\dot{\mathbf{U}}_l\dot{\mathbf{c}}_l\right|^2=-\frac{1}{2}\left(\overline{\dot{\mathbf{U}}_l^H(\dot{\boldsymbol{y}}-\dot{\boldsymbol{\mu}}_l)}\right)+\frac{1}{2}\left(\overline{\dot{\mathbf{U}}_l^H\dot{\mathbf{U}}_l\dot{\mathbf{c}}_l}\right)$$

Nulling the gradient with respect to $\dot{\mathbf{c}}_l$ at the target solution $\hat{\dot{\mathbf{c}}}_l$ leads to:

$$\dot{\mathbf{U}}_l^H(\dot{\boldsymbol{y}}-\dot{\boldsymbol{\mu}}_l)=\dot{\mathbf{U}}_l^H\dot{\mathbf{U}}_l\hat{\dot{\mathbf{c}}}_l$$

$$\hat{\dot{\mathbf{c}}}_l=\left(\dot{\mathbf{U}}_l^H\dot{\mathbf{U}}_l\right)^{-1}\dot{\mathbf{U}}_l^H(\dot{\boldsymbol{y}}-\dot{\boldsymbol{\mu}}_l)$$

Since $\dot{\mathbf{U}}_l$ is orthonormal

$$\Rightarrow \hat{\dot{\mathbf{c}}}_l=\dot{\mathbf{U}}_l^H(\dot{\boldsymbol{y}}-\dot{\boldsymbol{\mu}}_l) \qquad \blacksquare$$

APPENDIX B: PROOF OF PROPOSITION 2

For notation brevity, let's first drop the class specific index *l*. Assume $\ddot{\boldsymbol{y}}=\mathbf{y}_1e_1+\mathbf{y}_2e_2$, $\ddot{\boldsymbol{\mu}}=\boldsymbol{\mu}_1e_1+\boldsymbol{\mu}_2e_2$, $\ddot{\mathbf{U}}=\mathbf{U}_1e_1+\mathbf{U}_2e_2$, and $\ddot{\mathbf{c}}=\mathbf{c}_1e_1+\mathbf{c}_2e_2$. By Lemma 1 and Lemma 2 in [12],

$$\left|(\ddot{\boldsymbol{y}}-\ddot{\boldsymbol{\mu}})-\ddot{\mathbf{U}}\ddot{\mathbf{c}}\right|^2=\frac{1}{2}\left|\mathcal{M}(\ddot{\boldsymbol{y}}-\ddot{\boldsymbol{\mu}})-\mathcal{M}\left(\ddot{\mathbf{U}}\right)\mathcal{M}(\ddot{\mathbf{c}})\right|^2.$$

According to the properties of calculus on complex domain,

$$\frac{\partial}{\partial \mathbf{c}_1}\left|\mathcal{M}(\ddot{\boldsymbol{y}}-\ddot{\boldsymbol{\mu}})-\mathcal{M}\left(\ddot{\mathbf{U}}\right)\mathcal{M}(\ddot{\mathbf{c}})\right|^2$$

$$=\frac{\partial}{\partial \mathbf{c}_1}\mathrm{tr}\left(\left(\mathcal{M}(\ddot{\boldsymbol{y}}-\ddot{\boldsymbol{\mu}})-\mathcal{M}\left(\ddot{\mathbf{U}}\right)\mathcal{M}(\ddot{\mathbf{c}})\right)^H\left(\mathcal{M}(\ddot{\boldsymbol{y}}-\ddot{\boldsymbol{\mu}})-\mathcal{M}\left(\ddot{\mathbf{U}}\right)\mathcal{M}(\ddot{\mathbf{c}})\right)\right)$$

$$=2\left(\mathbf{U}_1^H\mathbf{U}_1\mathbf{c}_1-\mathbf{U}_1^H(\mathbf{y}_1-\boldsymbol{\mu}_1)\right)$$

Similarly,

$$\frac{\partial}{\partial \mathbf{c}_2}\left|\mathcal{M}(\ddot{\boldsymbol{y}}-\ddot{\boldsymbol{\mu}})-\mathcal{M}\left(\ddot{\mathbf{U}}\right)\mathcal{M}(\ddot{\mathbf{c}})\right|^2=2\left(\mathbf{U}_2^H\mathbf{U}_2\mathbf{c}_2-\mathbf{U}_2^H(\mathbf{y}_2-\boldsymbol{\mu}_2)\right)$$

The solution $\hat{\mathbf{c}}_1$ is found by setting the gradient of the objective function with respect to $\mathbf{c}_1$to zero

$$\hat{\mathbf{c}}_1 = (\mathbf{U}_1^H\mathbf{U}_1)^{-1}\mathbf{U}_1^H(\mathbf{y}_1 - \boldsymbol{\mu}_1)$$

Similarly, the solution $\hat{\mathbf{c}}_2$is:

$$\hat{\mathbf{c}}_2 = (\mathbf{U}_2^H\mathbf{U}_2)^{-1}\mathbf{U}_2^H(\mathbf{y}_2 - \boldsymbol{\mu}_2).$$

According to (5),

$$\mathcal{M}(\hat{\ddot{\mathbf{c}}}) = \begin{bmatrix} (\mathbf{U}_1^{\mathrm{H}}\mathbf{U}_1)^{-1}\mathbf{U}_1^{\mathrm{H}}(\mathbf{y}_1 - \boldsymbol{\mu}_1) & 0 \\ 0 & (\mathbf{U}_2^H\mathbf{U}_2)^{-1}\mathbf{U}_2^H(\mathbf{y}_2 - \boldsymbol{\mu}_2) \end{bmatrix}.$$

which is simplified to

$$= \begin{bmatrix} \mathbf{U}_1^{\mathrm{H}}\mathbf{U}_1 & 0 \\ 0 & \mathbf{U}_2^H\mathbf{U}_2 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{U}_1^{\mathrm{H}} & 0 \\ 0 & \mathbf{U}_2^H \end{bmatrix} \begin{bmatrix} (\mathbf{y}_1 - \boldsymbol{\mu}_1) & 0 \\ 0 & (\mathbf{y}_2 - \boldsymbol{\mu}_2) \end{bmatrix},$$

$$\therefore \hat{\ddot{\mathbf{c}}} = \left(\ddot{\mathbf{U}}^H\ddot{\mathbf{U}}\right)^{-1}\ddot{\mathbf{U}}^H(\ddot{\mathbf{y}} - \ddot{\boldsymbol{\mu}})$$

Since $\ddot{\mathbf{U}}$ is orthonormal

$$\therefore \hat{\ddot{\mathbf{c}}} = \ddot{\mathbf{U}}^H(\ddot{\mathbf{y}} - \ddot{\boldsymbol{\mu}}) \qquad \blacksquare$$

REFERENCES

[1] I. Naseem, R. Togneri, and M. Bennamoun, "Linear regression for face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32 (2010), no. 11, pp. 2106–2112.

[2] S.-M. Huang and J.-F. Yang, "Improved principal component regression for face recognition under illumination variations," *IEEE Signal Process. Lett.*, vol. 19 (2012), no. 4, pp. 179–182.

[3] Y. Zhu, C. Zhu, and X. Li, "Improved principal component analysis and linear regression classification for face recognition," *Signal Processing*, vol. 145 (2018), pp. 175–182.

[4] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cogn. Neurosci.*, vol. 3 (1991), no. 1, pp. 71–86.

[5] W.-J. Yang, C.-Y. Lo, P.-C. Chung, and J. F. Yang, "Weighted Module Linear Regression Classifications for Partially-Occluded Face Recognition," *Digit. Image Process. Adv. Appl. IntechOpen*, 2021.

[6] N. Le Bihan and S. J. Sangwine, "Quaternion principal component analysis of color images," in *Proceedings 2003 International Conference on Image Processing (Cat. No. 03CH37429)*, 2003, vol. 1, pp. I–809.

[7] L. Shi, "Exploration in quaternion colour." Doctoral dissertation, School of Computing Science-Simon Fraser University, 2005.

[8] M. T. El-Melegy and A. T. Kamal, "Color image processing using reduced biquaternions with application to face recognition in a PCA framework," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 3039–3046.

[9] C. Zou, K. I. Kou, L. Dong, X. Zheng, and Y. Y. Tang, "From grayscale to color: Quaternion linear regression for color face recognition," *IEEE Access*, vol. 7 (2019), pp. 154131–154140.

[10] J. Miao and K. I. Kou, "Quaternion matrix regression for color face recognition," *arXiv Prepr. arXiv2001.10677*, 2020.

[11] S. Gai and X. Huang, "Reduced Biquaternion Convolutional Neural Network for Color Image Processing," *IEEE Trans. Circuits Syst. Video Technol.*, 2021.

[12] M. El-Melegy and A. Kamal, "Linear Regression Classification in the Quaternion and Reduced Biquaternion Domains," *IEEE Signal Process. Lett.*, p. 1, 2022.

[13] S. C. Pei, J. H. Chang, J. J. Ding, and M. Y. Chen, "Eigenvalues and singular value decompositions of reduced biquaternion matrices," *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 55 (2008), no. 9, pp. 2673–2685.

[14] L. H. Harper, T. H. Payne, J. E. Savage, and E. Straus, "Sorting x+ y," *Commun. ACM*, vol. 18 (1975), no. 6, pp. 347–349.

[15] J.-L. Lambert, "Sorting the sums (xi+ yj) in O (n2) comparisons," *Theor. Comput. Sci.*, vol. 103 (1992), no. 1, pp. 137–141.

[16] Y. Tian, "Matrix theory over the complex quaternion algebra," *arXiv Prepr. math/0004005*, 2000.

[17] L. Huang and W. So, "On left eigenvalues of a quaternionic matrix," *Linear Algebra Appl.*, vol. 323 (2001), no. 1–3, pp. 105–116.
[18] F. Zhang, "Quaternions and matrices of quaternions," *Linear Algebra Appl.*, vol. 251 (1997), pp. 21–57.
[19] H. H. Kösal, "Least-squares solutions of the reduced biquaternion matrix equation AX= B and their applications in colour image restoration," *J. Mod. Opt.*, vol. 66 (2019), no. 18, pp. 1802–1810.
[20] H.-. Schutte and J. Wenzel, "Hypercomplex numbers in digital signal processing," in *1990 IEEE International Symposium on Circuits and Systems (ISCAS)*, 1990, pp. 1557–1560 vol.2.
[21] K. Ueda and S. Takahashi, "Digital filters with hypercomplex coefficients," *Electron. Commun. Japan (Part III Fundam. Electron. Sci.*, vol. 76 (1993), no. 9, pp. 85–98.
[22] S. C. Pei, J. H. Chang, and J. J. Ding, "Commutative reduced biquaternions and their Fourier transform for signal and image processing applications," *IEEE Trans. Signal Process.*, vol. 52 (2004), no. 7, pp. 2012–2030.
[23] D. Xu and D. P. Mandic, "The theory of quaternion matrix derivatives," *IEEE Trans. Signal Process.*, vol. 63 (2015), no. 6, pp. 1543–1556.
[24] A. V Nefian, "Georgia Tech face database," *Georg. Inst. Technol.*, 1999.
[25] P. J. Phillips, H. Wechsler, J. Huang, and P. J. Rauss, "FERET database and evaluation procedure for face-recognition algorithms," *Image Vis. Comput.*, vol. 16 (1998), no. 5, pp. 295–306.
[26] S. J. Sangwine and N. Le Bihan, "Quaternion and octonion toolbox for Matlab." 2013.
[27] M. Zhao, Z. Jia, and D. Gong, "Improved two-dimensional quaternion principal component analysis," *IEEE Access*, vol. 7 (2019), pp. 79409–79417.