Color Image Processing by Using Binary Quaternion-Moment-Preserving Thresholding Technique

Soo-Chang Pei, *Senior Member, IEEE,* and Ching-Min Cheng

***Abstract*— This paper presents a new moment-preserving thresholding technique, called the *binary quaternion-moment-preserving* (BQMP) thresholding, for color image data. Based on representing color data by the quaternions, the statistical parameters of color data can be expressed through the definition of quaternion moments. Analytical formulas of the BQMP thresholding can thus be determined by using the algebra of the quaternions. The computation time for the BQMP thresholding is of order of data size. By using the BQMP thresholding, quaternion-moment-based operators are designed for the applications of color image processing, such as color image compression, multiclass clustering of color data, and subpixel color edge detection. The experimental results show that the proposed operator for color image compression can have output picture quality acceptable to human eyes. In addition, the proposed edge operator can detect color edge in subpixel level. Therefore, the proposed BQMP thresholding can be used as a tool for color image processing.**

***Index Terms*— Binary thresholding, color image processing, quaternion-moment preserving.**

I. Introduction

GRAY-LEVEL moment-based operators have been successfully developed for image processing [1]–[3]. Delp and Mitchell [1] first proposed block truncation coding (BTC), which used a two-level moment-preserving quantizer to compress monochrome images. Unlike other image compression methods such as transform coding and vector quantization [4], BTC requires less computation. Tabatabai and Mitchell [2] also proposed an operator, which can compute edge location by fitting first three gray-level moments to the input data. Compared with traditional edge operators like Robert and Sobel [4], the precision to subpixel accuracy has been reported. In application to image segmentation, Tsai [3] has used the moment-preserving principle to select thresholds of input gray-level image. The threshold values are determined in such a way that gray-level moments of an input image are preserved in the output image. Good performance of Tsai's method has been declared by [5] as compared with other thresholding techniques. Generally speaking, these gray-level moment-based operators cluster the input one-dimensional (1-D) data points into different classes according to the moment-preserving principle.

When considering multidimensional color image data, the usage of gray-level moment-based operators to separately process individual dimension of data may not get desired results. For instance, in coding trichromatic images, three chromatic channels can be quantized with three separate gray-level BTC coders as suggested by [6]. Since there are three output bitmaps, the output compression ratio is unacceptable. To obtain single output bitmap, Wu and Coll [7] have considered quantizing only the intensity component of a color image. However, for the situation when there is large color discontinuity among pixels, there might be just a slight difference in intensity values of pixels. In this case, the quantization results of [7] may not get good performance. The same circumstances may happen when a gray-level moment-based operator [2] is applied to color edge detection. Consequently, it is desirable to design a color moment-based operator, which can directly process multidimensional color space of input data.

For two-dimensional (2-D) color space, Pei and Cheng [8] have introduced a complex-moment-preserving operator to threshold input data. This operator designates input data as complex numbers and extends moment-preserving principle of gray-level to that of complex number. In [8], performance comparison with other thresholding techniques shows that this approach is feasible. Recently, Yang *et al.* [9] presented a moment-based operator for color image sharpening. Although this operator can exercise on 1-D, 2-D, and three-dimensional (3-D) color data, it requires a different algorithm for each dimensional case. There are no analytical formulas for 3-D data and some sign decision operations are necessary in order to judge the correctness of the solution. Thus, [9] is not a unified approach for multidimensional color data.

In contrast to [9], we present in this paper new moment-based operators, which are based on a thresholding technique, called *binary quaternion-moment-preserving* (BQMP) thresholding. The BQMP thresholding generalizes conventional gray-level moment-based operators [1]–[3] to be multidimensional by expressing the input color space as a quaternion-valued space. Through the definition of quaternion moments of input color data, we extend the moment-preserving principle from 1-D gray-level data to 3-D color data. An analytic solution for the BQMP thresholding is also obtained by the use of quaternion arithmetic. With this approach, the operator of [8] is only a special case of the BQMP thresholding. The computation time for the BQMP

Manuscript received February 12, 1996; revised July 23, 1998. This work was supported by National Science Council, Taiwan, R.O.C., under Contract NSC 86-2221-E-002-018. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. John Goutsias.

S.-C. Pei is with the Electrical Engineering Department, National Taiwan University, Taipei, Taiwan, R.O.C. (e-mail: pei@cc.ee.ntu.edu.tw).

C.-M. Cheng is with the Telecommunication Laboratory, Chunghwa Telecommunication Co., Chung-Li, Taiwan 320, R.O.C.

Publisher Item Identifier S 1057-7149(99)03433-8.

thresholding is of order N, the data size. Moreover, the proposed quaternion-moment-based operators can be applied to the problems of color image processing, such as image compression, multiclass clustering and subpixel color edge detection.

In what follows, Section II first describes the algebraic operations of the quaternions and defines quaternion moments. Section III then presents the algorithm of the BQMP thresholding. The analysis of this algorithm is also conducted. In Section IV, we use the BQMP thresholding to design quaternion-moment-based operators for some applications of color image processing. The experimental results for these applications are presented for illustration. Finally, some conclusions are made in Section V.

II. Representation of Color Image Data

Human perception of gray level images is quite satisfactory, but color images seem to be perceptually richer. Color data representations could be 2-D or 3-D, contrary to the gray level data representations which are 1-D. For instance, the color image data from a frame grabber is usually in the RGB (red, green, blue) primary color space. Other representations of the color data can be generated from RGB spaces to facilitate the processing of color images. One example is the color data from the (r, g) space, $r = (R/(R+G+B))$ and $g = (G/(R+G+B))$. This space has been supported by some previous works to do color image segmentation [10]. To effectively compute the multidimensional color data, an efficient expression of the color data is necessary. Machuca and Phillips [11] have proposed vector fields as the theoretical model of color data. Through differential geometry and vector analysis, they obtained the algorithm for color edge detection. Different from [11], the expression selected in this paper is the quaternions discovered by W. R. Hamilton [12]. Employing this expression, an algebraic approach, which generalizes 1-D moment-preserving principle to be multidimensional, can be realized. In this section, we will introduce some preliminaries of quaternion arithmetic and the definition of quaternion moments used by the following sections.

A. Quaternion Expression

The algebra of the quaternions is the generalization of complex numbers [12]. Considering a four-dimensional (4-D) real-valued data set $Q = \{(q_0(n), q_1(n), q_2(n), q_3(n))\}_{n=1}^{N}$, a quadruple data point $(q_0(n), q_1(n), q_2(n), q_3(n))$ can be expressed as a quaternion number $\hat{q}(n)$

$$\hat{q}(n) = q_0(n) + q_1(n) \cdot i + q_2(n) \cdot j + q_3(n) \cdot k \qquad (1)$$

where i, j, and k denote the operation units of quaternion number with the rules of operation as follows:

$$\begin{aligned} i^2 = j^2 = k^2 &= -1, \\ i \cdot j = -1 \cdot j \cdot i &= k, \\ j \cdot k = -1 \cdot k \cdot j &= i, \\ k \cdot i = -1 \cdot i \cdot k &= j. \end{aligned} \qquad (2)$$

Any vector $\boldsymbol{v} \in R^3$, can be expressed as a quaternion with q_0 set to be zero. For example, a color value (R, G, B) can be shown as a quaternion with $q_1 = R$, $q_2 = G$, $q_3 = B$, $q_0 = 0$. And any vector $\boldsymbol{v} \in R^2$ can be expressed like a complex number. A quaternion can also be denoted as $\hat{q}(n) = \langle \boldsymbol{a}, b \rangle$ where $\boldsymbol{a} = (q_1(n), q_2(n), q_3(n))$ and $b = q_0(n)$. The operation of quaternion number has the following properties.

1) The unit quaternion is defined as $\langle \mathbf{0}, 1 \rangle$, where $\mathbf{0} = (0, 0, 0)$.
2) The addition and subtraction rules of the quaternions are the same as for complex numbers.
3) Using the cross product ($\times$) of vector space, one can define multiplication of two quaternions, $\hat{q}$ and $\hat{q}'$, as

$$\begin{aligned} \hat{q} \cdot \hat{q}' &= \langle \boldsymbol{a}, b \rangle \cdot \langle \boldsymbol{a}', b' \rangle \\ &= \langle \boldsymbol{a} \times \boldsymbol{a}' + b \cdot \boldsymbol{a}' + b' \cdot \boldsymbol{a}, b \cdot b' - \boldsymbol{a} \cdot \boldsymbol{a}' \rangle. \end{aligned} \qquad (3)$$

4) The conjugate $\hat{q}^*$ of $\hat{q}$ is defined as

$$\hat{q}^* = -\langle \boldsymbol{a}, b \rangle = q_0 - (q_1 \cdot i + q_2 \cdot j + q_3 \cdot k) \qquad (4)$$

and the norm of the quaternion is denoted as $\|\hat{q}\|^2 = \hat{q} \cdot \hat{q}^*$.
5) The reciprocal of $\hat{q}$ is

$$(\hat{q})^{-1} = \frac{\hat{q}^*}{\|\hat{q}\|^2}. \qquad (5)$$

With the help of the $(\hat{q})^{-1}$, the division of the quaternions is denoted as

$$\frac{\hat{q}'}{\hat{q}} = \hat{q}' \cdot (\hat{q})^{-1}. \qquad (6)$$

B. Quaternion Moments

Based on the above definition of the quaternion, we will designate the quaternion moments as follows in order to explicitly express the statistical parameters of color data:

$$\begin{aligned} \hat{m}_1 &= \boldsymbol{E}[\hat{q}] \\ \hat{m}_2 &= \boldsymbol{E}[\hat{q} \cdot \hat{q}^*] \\ \hat{m}_3 &= \boldsymbol{E}[\hat{q} \cdot \hat{q}^* \cdot \hat{q}] \end{aligned} \qquad (7)$$

where $\boldsymbol{E}[\cdot]$ represents the expectation. The definitions of $\hat{m}_1$ and $\hat{m}_2$ are the extension of complex moments. And the definition of third order quaternion moment $\hat{m}_3$ is adopted from the high order statistics [13]. Equation (7) can be further expressed as

$$\begin{aligned} \hat{m}_1 &= \boldsymbol{E}[q_0] + \boldsymbol{E}[q_1] \cdot i + \boldsymbol{E}[q_2] \cdot j + \boldsymbol{E}[q_3] \cdot k \\ \hat{m}_2 &= \boldsymbol{E}[q_0^2 + q_1^2 + q_2^2 + q_3^2] \\ \hat{m}_3 &= \boldsymbol{E}[q_0^3 + q_1^2 q_0 + q_2^2 q_0 + q_3^2 q_0] \\ &\quad + \boldsymbol{E}[q_1^3 + q_0^2 q_1 + q_2^2 q_1 + q_3^2 q_1] \cdot i \\ &\quad + \boldsymbol{E}[q_2^3 + q_0^2 q_2 + q_1^2 q_2 + q_3^2 q_2] \cdot j \\ &\quad + \boldsymbol{E}[q_3^3 + q_0^2 q_3 + q_1^2 q_3 + q_1^2 q_3] \cdot k. \end{aligned} \qquad (8)$$

As we see, the quaternion-valued $\hat{m}_1$ represents the centroid of the input data, (q_0, q_1, q_2, q_3). The real-valued $\hat{m}_2$ expresses expected value of the vector length of (q_0, q_1, q_2, q_3). And the quaternion-valued $\hat{m}_3$ consists of the sum of joint-third-order

moments among q_0, q_1, q_2, and q_3, which reflects the measure of skewness of the input data. To calculate the quaternion moments of the data set Q, the expectation $\boldsymbol{E}[\cdot]$ of (8) is replaced by the sample mean value. For example,

$$\boldsymbol{E}[q_0] = \frac{1}{N}\sum_{n=1}^{N} q_0(n). \tag{9}$$

Although statistical distribution of a color image can be described by knowing each order of moment, it is known that the higher the order of moment is, the less important this moment is. Thus, preserving moments up to third order would not severely lose statistical distribution of the image. To derive analytical formulas for the BQMP thresholding in the next section, we will use moment-preserving principle on $\hat{m}_1, \hat{m}_2$, and $\hat{m}_3$ in order to maintain statistical characteristics of input image. Besides, without the loss of generality, we assume that the centroid of the input data is the origin, that is $\hat{m}_1 = \langle \mathbf{0}, 0 \rangle$. If this is not the case, we would first translate every data point inside Q, $\{(q_0(n), q_1(n), q_2(n), q_3(n))\}_{n=1}^{N}$, by the amount of $\Delta q_0 = \boldsymbol{E}[q_0]$, $\Delta q_1 = \boldsymbol{E}[q_1]$, $\Delta q_2 = \boldsymbol{E}[q_2]$, and $\Delta q_3 = \boldsymbol{E}[q_3]$. After the analytic solution of the BQMP thresholding are obtained, we then transform it back to the old coordinates by the inverse translation with $\Delta q_0 = -\boldsymbol{E}[q_0]$, $\Delta q_1 = -\boldsymbol{E}[q_1]$, $\Delta q_2 = -\boldsymbol{E}[q_2]$, and $\Delta q_3 = -\boldsymbol{E}[q_3]$. Therefore, the quaternion moments mentioned afterwards will mean central quaternion moments.

III. Binary Quaternion-Moment-Preserving Thresholding

The problem of the BQMP thresholding in a quaternion-valued data set Q is to select a hyperplane as a threshold, such that if all below-threshold data points and those above-threshold data points in Q are replaced by the representative values, $\hat{z}_0$ and $\hat{z}_1$, respectively, then the first three quaternion moments of Q are preserved in the resultant two-level data set G. If two thresholded classes of Q is designated as Q_0 and Q_1, G can be expressed as

$$G = \{\hat{z} | \hat{z} = \hat{z}_0, \text{ if } \hat{z} \in Q_0; \hat{z} = \hat{z}_1, \text{ if } \hat{z} \in Q_1\}$$

where the representatives of the thresholded classes by the BQMP thresholding, $\hat{z}_l$ for $l = 0, 1$, can be defined as

$$\hat{z}_l = z_{l0} + z_{l1} \cdot i + z_{l2} \cdot j + z_{l3} \cdot k, \quad \text{for } l = 0, 1.$$

We then represent the first three quaternion moments of the two-level data set G as

$$\begin{aligned} \hat{m}'_1 &= \sum_{l=0}^{1} p_l \cdot \hat{z}_l \\ \hat{m}'_2 &= \sum_{l=0}^{1} p_l \cdot \hat{z}_l \cdot \hat{z}_l^* \\ \hat{m}'_3 &= \sum_{l=0}^{1} p_l \cdot \hat{z}_l \cdot \hat{z}_l^* \cdot \hat{z}_l \end{aligned} \tag{10}$$

where p_0 and p_1 denote the fractions of the numbers of the below-threshold and the above-threshold data points in Q, respectively. And

$$p_0 + p_1 = 1. \tag{11}$$

If we let the first three quaternion moments of the two-level data set G equal to those of Q, $\hat{m}_l = \hat{m}'_l$, $l = 1, 2, 3$, we obtain the following moment-preserving equations:

$$p_0 \cdot \hat{z}_0 + p_1 \cdot \hat{z}_1 = \hat{m}_1 \tag{12a}$$

$$p_0 \cdot \hat{z}_0 \cdot \hat{z}_0^* + p_1 \cdot \hat{z}_1 \cdot \hat{z}_1^* = \hat{m}_2 \tag{12b}$$

$$p_0 \cdot \hat{z}_0 \cdot \hat{z}_0^* \cdot \hat{z}_0 + p_1 \cdot \hat{z}_1 \cdot \hat{z}_1^* \cdot \hat{z}_1 = \hat{m}_3. \tag{12c}$$

Thus, using the quaternion moments, the moment-preserving principle for gray level moment-based operators [1]–[3] can still be maintained for color image data. And the solutions of [1]–[3] can be generalized by the proposed BQMP thresholding. To efficiently calculate $\hat{z}_l$ for $l = 0, 1$, we adopt the concept of Prony method [14] to alleviate the computation complexity of (12). This concept embeds the nonlinear aspects of the exponential model into a polynomial factoring, which is defined by the following polynomial of $C(\hat{z})$.

$$C(\hat{z}) = \hat{z} \cdot \hat{z}^* + \hat{z}^* \cdot \hat{c}_1 + \hat{c}_0 \tag{13a}$$

$$= (\hat{z} - \hat{z}_0)(\hat{z} - \hat{z}_1) \tag{13b}$$

where c_1 and c_0 are two quaternion constants to be determined. In (13), we also suppose that $\hat{z}_l$ for $l = 0, 1$ are the roots of $C(\hat{z})$. If we substitute in the trivial expression

$$p_0 \cdot C(\hat{z}_0) + p_1 \cdot C(\hat{z}_1) = 0$$

the expression for $C(\hat{z})$ from (13a) and make use of (11), (12a), and (12b), we obtain:

$$\hat{m}_2 + \hat{c}_0 + \hat{m}_1^* \cdot \hat{c}_1 = 0 \tag{14}$$

Similarly, if we substitute in the trivial expression

$$p_0 \cdot \hat{z}_0 \cdot C(\hat{z}_0) + p_1 \cdot \hat{z}_1 \cdot C(\hat{z}_1) = 0$$

the expression for $C(\hat{z})$ from (13a) and make use of (12), we obtain

$$\hat{m}_3 + \hat{m}_1 \cdot \hat{c}_0 + \hat{m}_2 \cdot \hat{c}_1 = 0. \tag{15}$$

From (14) and (15), $\hat{c}_0$ and $\hat{c}_1$ can be expressed as follows:

$$\hat{c}_1 = \frac{(\hat{m}_3 - \hat{m}_1 \cdot \hat{m}_2)}{(\hat{m}_1 \cdot \hat{m}_1^* - \hat{m}_2)} \tag{16}$$

$$\hat{c}_0 = -(\hat{m}_1^* \cdot \hat{c}_1 + \hat{m}_2). \tag{17}$$

Since $\hat{m}_1 = 0$ in the proposed BQMP thresholding, $\hat{c}_0$ and $\hat{c}_1$ will be reduced to $-\hat{m}_2$ and $-(\hat{m}_3/\hat{m}_2)$, respectively. And $\hat{c}_1$ and $\hat{c}_0$ can be represented as quaternion numbers

$$\begin{aligned} \hat{c}_1 &= c_{10} + c_{11} \cdot i + c_{12} \cdot j + c_{13} \cdot k \\ \hat{c}_0 &= c_{00}. \end{aligned} \tag{18}$$

After $\hat{c}_0$ and $\hat{c}_1$ are obtained, roots of $C(\hat{z}_l) = 0$ can thus be solved. by substituting $\hat{z}_l$ and (18) into (13) and letting (13) equal to zero.

$$\begin{aligned} a_1 \cdot z_{l3}^2 + a_2 \cdot z_{l3} + c_{00} &= 0 \\ z_{l0} &= (w \cdot u + v) \cdot z_{l3} \\ z_{l1} &= (w \cdot v - u) \cdot z_{l3} \\ z_{l2} &= w \cdot z_{l3} \end{aligned} \tag{19}$$

where

$$\begin{aligned} u &= \frac{c_{10}\cdot c_{12} - c_{11}\cdot c_{13}}{c_{13}^2 + c_{12}^2} \\ v &= \frac{c_{11}\cdot c_{12} + c_{10}\cdot c_{13}}{c_{13}^2 + c_{12}^2} \\ w &= \frac{c_{11}\cdot v + c_{10}\cdot u + c_{12}}{c_{10}\cdot v - c_{11}\cdot u + c_{13}} \\ a_1 &= (1+w^2)\cdot(1+u^2+v^2) \\ a_2 &= w\cdot(u\cdot c_{10} + v\cdot c_{11} + c_{12}) \\ &\quad - u\cdot c_{11} + v\cdot c_{10} + c_{13}. \end{aligned} \tag{20}$$

From (19), the two solutions, z_{03} and z_{13}, can be obtained directly by using quadratic formula

$$\begin{aligned} z_{03} &= \frac{-1}{2\cdot a_1}\cdot(a_2 - \sqrt{a_2^2 - 4\cdot a_1\cdot c_{00}}) \\ z_{13} &= \frac{-1}{2\cdot a_1}\cdot(a_2 + \sqrt{a_2^2 - 4\cdot a_1\cdot c_{00}}). \end{aligned} \tag{21}$$

Since $c_{00} = -m_2$, it can be readily shown that the values of z_{03} and z_{13} are real. After z_{l3} is obtained, the corresponding z_{l0}, z_{l1}, and z_{l2} can be acquired from (19). Besides, p_0 and p_1 can be obtained by using (12).

In implementing the BQMP thresholding, we choose the hyperplane l, which is perpendicular to the line segment $\overline{\hat{z}_0\hat{z}_1}$ such that l split the quaternion space into two halves and the half space containing $\hat{z}_0$ has p_0 portion of pixels, as the decision boundary to separate the input data set into two classes. However, if we want to implement the BQMP thresholding in a much quicker way, a hyperplane l' can be chosen alternatively to be perpendicular to and bisecting the line segment $\overline{\hat{z}_0\hat{z}_1}$. This approach has also been adopted by Lin and Tsai [15] in their two-class clustering method. When this computation-saving decision boundary l' is used, we understand that the fractions of the numbers of data points in two separated classes do not necessarily match with the values of p_0 and p_1 computed by moment-preserving equations (12). Nevertheless, the trade-off between reducing the computation complexity and sticking to the estimated p_0 and p_1 is worthy, especially when a fast thresholding method is desired. As experimental results shown in this and next section, hyperplane l' would not produce significant degradation of performance. The linear equation, which formulates the decision boundary l', is thus described as

$$q_0 = s - t_1\cdot q_1 - t_2\cdot q_2 - t_3\cdot q_3 \tag{22}$$

with

$$\begin{aligned} s &= \frac{(z_{00}^2 + z_{01}^2 + z_{02}^2 + z_{03}^2 - z_{10}^2 - z_{11}^2 - z_{12}^2 - z_{13}^2)}{2\cdot(z_{00} - z_{10})} \\ t_1 &= \frac{(z_{01} - z_{11})}{(z_{00} - z_{10})} \\ t_2 &= \frac{(z_{02} - z_{12})}{(z_{00} - z_{10})} \\ t_3 &= \frac{(z_{03} - z_{13})}{(z_{00} - z_{10})}. \end{aligned}$$

This segmentation of two classes is equivalent to the nearest-neighbor clustering of two centers, $\hat{z}_0$ and $\hat{z}_1$. As we notice, the computation complexity of the BQMP thresholding is mainly dominated by the calculation of quaternion moments $\hat{m}_l$, for $l = 1, 2, 3$, whose computation time is of order N, the data size.

A. The Algorithm

In summary, the procedures of the proposed BQMP thresholding can be described as follows.

1) Express the value of each data point by a quaternion number.
2) Obtain two representatives, $\hat{z}_0$ and $\hat{z}_1$, of input data by solving the moment-preserving equations (12).
3) Choose the line l' perpendicular to and bisecting the line segment $\overline{\hat{z}_0\hat{z}_1}$ as the decision boundary.
4) Construct a class-indicating bit-map such that each data point is coded as a "one" or a "zero" depending on whether or not that the pixel resides in the half containing $\hat{z}_0$.

As an example, suppose a 4-D data set Q is given and arranged as the following manner:

$$\begin{aligned} Q = \{&(8,12,19,9),(7,13,14,13),(11,14,19,10),\\ &(13,10,16,9),(80,117,53,98),(88,118,63,103),\\ &(105,106,88,89),(109,117,71,106),\\ &(90,112,49,92),(99,113,34,84),\\ &(10,17,17,18),(113,115,77,117),\\ &(100,117,34,89),(102,115,43,93),\\ &(12,13,12,17),(110,112,70,122)\} \end{aligned}$$

so

$$\begin{aligned} \hat{z}_0 &= (12.050176, 10.863585, 21.460031, 15.350859) \\ \hat{z}_1 &= (101.780\,290, 119.593\,189, 56.309\,682, 100.843\,550) \end{aligned}$$

and the two-class bitmap is

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix}.$$

B. Statistical Analysis of Performance

In this subsection, we analyze the performance of the proposed BQMP thresholding on a two-cluster 3-D data set. Since it is commonly assumed that a region of high local concentration of patterns, called *core*, is associated with each cluster, we use the normal distribution to approximate the probability distribution of each cluster. In view of the central-limit theorem [16], we understand that this approximation has a good degree of reliability. Thus, we investigate the BQMP thresholding for the case that input data set is composed of two equiprobable 3-D normal distributions. For each distribution, we assume that three components have equal variances. The covariance matrix, $\boldsymbol{C}_l$, of each distribution is given by

$$\boldsymbol{C}_l = \begin{pmatrix} \sigma_l^2 & 0 & 0 \\ 0 & \sigma_l^2 & 0 \\ 0 & 0 & \sigma_l^2 \end{pmatrix} \quad \text{for } l = 0, 1$$

where σ_l represents variance. The probability density function of these normal distributions are then denoted as

$$f_l(\boldsymbol{q}) = (2\pi^{3/2}|\boldsymbol{C}_l|^{1/2})^{-1}\exp(-\tfrac{1}{2}(\boldsymbol{q}-\boldsymbol{m}_l)^t\boldsymbol{C}_l^{-1}(\boldsymbol{q}-\boldsymbol{m}_l)) \quad \text{for } l=0,1.$$

where $\boldsymbol{q}=[q_1,q_2,q_3]$ is the vector representing the data point. And $\boldsymbol{m}_l=[\mu_l,\mu_l,\mu_l]$, for $l=0,1$, are the mean vectors for each distribution. The value of $|\boldsymbol{C}_l|$ is the determinant of covariance matrix $\boldsymbol{C}_l$. According to the moment theorem for normal distribution [19], quaternion moments $\hat{m}_2$ and $\hat{m}_3$ can be calculated as

$$\begin{aligned}\hat{m}_2 &= \tfrac{3}{2}\cdot\sigma_0^2+\sigma_1^2+\tfrac{3}{4}\cdot(\mu_0-\mu_1)^2\\ \hat{m}_3 &= \tfrac{5}{4}\cdot(\sigma_0^2-\sigma_1^2)\cdot(\mu_0-\mu_1)\cdot i\\ &\quad+\tfrac{5}{4}\cdot(\sigma_0^2-\sigma_1^2)\cdot(\mu_0-\mu_1)\cdot j\\ &\quad+\tfrac{5}{4}\cdot(\sigma_0^2-\sigma_1^2)\cdot(\mu_0-\mu_1)\cdot k.\end{aligned} \tag{23}$$

Then, $\hat{z}_0$ and $\hat{z}_1$ can be obtained by using (19) and (21).

$$\begin{aligned}z_{0v} &= \frac{-1}{6}\cdot(3\cdot c_{13}-\sqrt{9\cdot c_{13}^2-12\cdot c_{00}}\\ z_{1v} &= \frac{-1}{6}\cdot(3\cdot c_{13}+\sqrt{9\cdot c_{13}^2-12\cdot c_{00}}\\ &\quad \text{for } v=1,2,3.\end{aligned} \tag{24a}$$

where

$$\begin{aligned}c_{00} &= -\left(\frac{3}{2}\cdot\sigma_0^2+\sigma_1^2+\frac{3}{4}\cdot(\mu_0-\mu_1)^2\right)\\ c_{13} &= \frac{-5\cdot(\sigma_0^2-\sigma_1^2)\cdot(\mu_0-\mu_1)}{(6\cdot\sigma_0^2+4\cdot\sigma_1^2+3\cdot(\mu_0-\mu_1)^2)}.\end{aligned} \tag{24b}$$

To determine the total probability of classification error, we have to calculate P_{e0}, the probability of classifying an input data point as class one when it is not, and P_{e1}, the probability of classifying an input data point as class zero when it is not. Using the decision boundary equation (22), these error probabilities, P_{e0} and P_{e1}, can be expressed as [18]:

$$\begin{aligned}P_{e0} &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{s-t_1\cdot q_1-t_2\cdot q_2}^{+\infty} \cdot f_0(q_0,q_1,q_2)\,dq_0\,dq_1\,dq_2\\ &= \frac{1}{(4\pi\sigma_0^2)}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \cdot\operatorname{erfc}\left(\frac{s-t_1\cdot q_1-t_2\cdot q_2-\mu_0}{\sigma_0\sqrt{2}}\right)\\ &\quad\cdot\exp\left(\frac{(q_1-\mu_0)^2+(q_2-\mu_0)^2}{-2\sigma_0^2}\right)dq_1\,dq_2\end{aligned} \tag{25}$$

$$\begin{aligned}P_{e1} &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{s-t_1\cdot q_1-t_2\cdot q_2} \cdot f_1(q_0,q_1,dq_2)\,dq_0\,dq_1\,dq_2\\ &= \frac{1}{(4\pi\sigma_1^2)}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \cdot\left(1+\operatorname{erf}\left(\frac{s-t_1\cdot q_1-t_2\cdot q_2-\mu_1}{\sigma_1\sqrt{2}}\right)\right)\\ &\quad\cdot\exp\left(\frac{(q_1-\mu_1)^2+(q_2-\mu_1)^2}{-2\sigma_1^2}\right)dq_1\,dq_2\end{aligned} \tag{26}$$

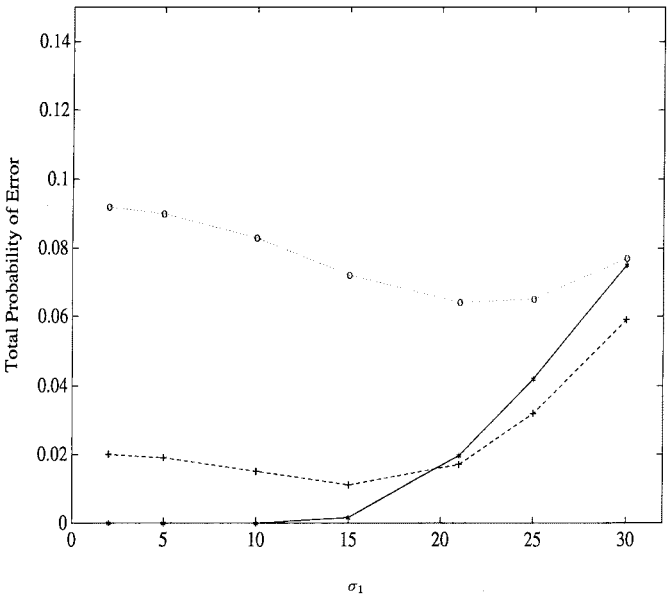


Fig. 1. Total error probability distributions of the BQMP thresholding versus the variances, σ_0 and σ_1, when μ_0, and μ_1 are fixed to be 100 and 150, respectively. The '$*$,' '$+$,' and '$\circ$' curves correspond to $\sigma_0=11$, $\sigma_0=20$, and $\sigma_0=31$, respectively.

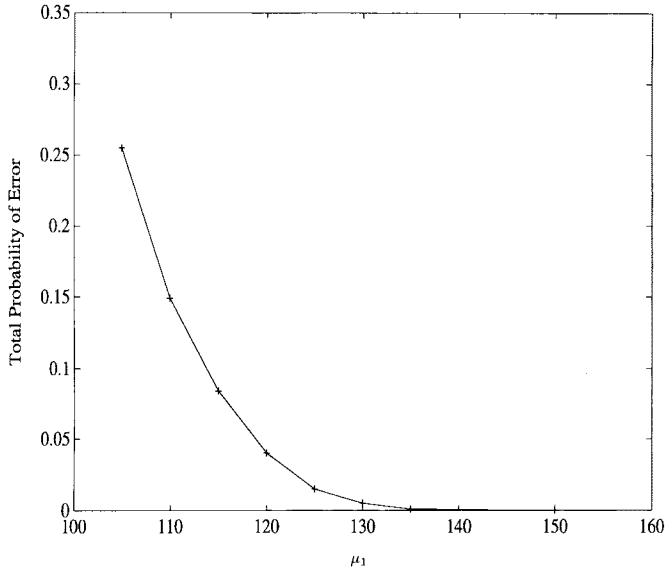
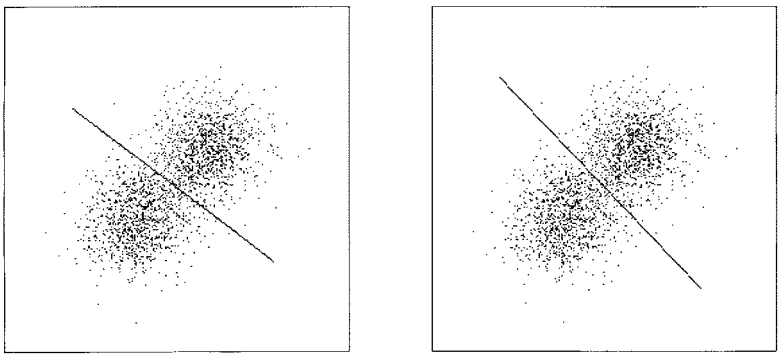


Fig. 2. Total error probability distributions of the BQMP thresholding versus the mean value, μ_0 and μ_1, when σ_0, and σ_1 are fixed to be 5 and 10, respectively. The '$+$' curve corresponds to $\mu_0=100$.

where $f_0(q_0,q_1,q_2)$ and $f_1(q_0,q_1,q_2)$ are the probability density functions for each class, respectively. And function erf($\cdot$) is the error function. Function erfc($\cdot$) is the error complementary function [20]. Once P_{e0} and P_{e1} are obtained, the total error probability can be denoted as

$$P_e=\tfrac{1}{2}(P_{e0}+P_{e1}). \tag{27}$$

The behaviors of the proposed BQMP thresholding is depicted in Figs. 1 and 2. In Fig. 1, μ_0, and μ_1 are set as 100 and 150, respectively, and the variances of two distributions are changed. We have observed that the minimum total error probability happened when the variance values of two distributions were near. In Fig. 2, σ_0, and σ_1 are fixed to 5 and



(a) (b)

Fig. 3. Thresholding results of the generated 2-D data specified in Table I. (a) Proposed BQMP thresholding operator. (b) Bayes classifier.

TABLE I
THE STATISTICAL PARAMETERS OF THE TWO 2-D DISTRIBUTIONS FOR THE EXAMPLE OF FIG. 3 AND PERFORMANCE COMPARISON OF TWO DIFFERENT CLASSIFIERS

	Generated Data			Baysian Classifier			BQMP Thresholding		
	Mean Vector	Covariane Matrix		Mean Vector	Covariane Matrix		Mean Vector	Covariane Matrix	
Population 1 (1050)	150.23	386.4	51.4	149.99	369,8	30.3	149.96	370.3	21.2
	150.17	51.4	404.7	150.14	30.3	378.0	150.59	21.2	348.1
Population 2 (1050)	99.85	402.0	65.1	98.53	365.0	21.1	99.01	390.9	32.0
	99.55	65.1	417.9	98.07	21.1	366.9	98.02	32.0	372.9
Error Rate				0.049			0.052		
Confusion Matrices				1004	46		996	54	
				57	993		56	994	

10, respectively, and the mean vectors of two distributions are changed to indicate the closeness of two distributions. We have noticed that the more far away two distributions are, the smaller the total error probability is.

We next show the empirical results of applying the proposed BQMP thresholding to two two-cluster examples. For these examples, we have used a random number generator based on normal distributions [17] to create a 2-D or 3-D subset S_A, and used the same generator to create another 2-D or 3-D subset S_B. These two subsets are then merged together to form a data set Q. The first example shown in Fig. 3 contains two identical 2-D normal-distribution data subsets with parameters specified in Table I. The second example displayed in Fig. 4 has two identical 3-D normal-distribution data subsets with parameters specified in Table II. The number of data points in each subset of Figs. 3 and 4 is 1050 points. For these examples, we utilize 1) the proposed thresholding algorithm and 2) the optimum Bayes classifier [18] to carry out the task of classifying data set Q into two classes. The job of the Bayes classifier is to find an optimum decision boundary that minimizes the average risk or cost. However, to implement this parametric classifier, *a priori* probability distributions of classes have to be known beforehand. The estimation of class statistical parameters is not an easy task. Nonparametric clustering methods such as the proposed thresholding algorithm have thus been designed to overcome this difficulty. When these classifiers are applied to the above artificially generated data sets, the results are evaluated by means of the total error probability, or equivalently the classification error rate, which is estimated as the ratio of the number of misclassified data points to the total number of available data points. A confusion matrix is also a well-known indicator that can be used to evaluate clustering results. In Fig. 3, the respective computed decision boundaries are shown as the solid straight lines. To display the 3-D plot of Fig. 4, we have projected the classified results in the $q_0 - q_1$ plane and $q_0 - q_2$ plane. As we have observed, the classified results of Figs. 3 and 4 by the proposed algorithm are close to those of the optimum Bayes classifier. These clustering results are further enhanced by the error rate and confusion matrices in Tables I and II. These tables compare the statistical properties in each classified subset for the test examples, too.

IV. APPLICATIONS

In this section, we present some applications of the BQMP thresholding to do color image processing. The color components of input image are chosen from the RGB space. The color value of each pixel is denoted as $(I_1, I_2, I_3) = (red, green, blue)$, which is then expressed as a quaternion number. Through the usage of the BQMP thresholding,

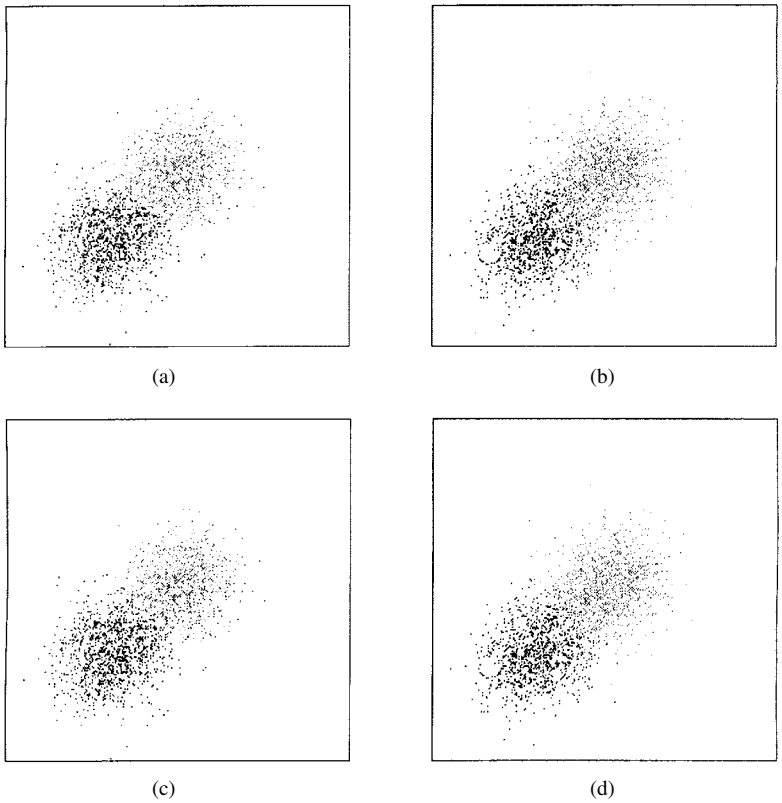


Fig. 4. Thresholding results of the generated 3-D data specified in Table II. (a) Proposed BQMP thresholding operator on $q_0 - q_1$ plane. (b) Proposed BQMP thresholding operator on $q_0 - q_2$ plane. (c) Bayes classifier on $q_0 - q_1$ plane. (d) Bayes classifier on $q_0 - q_2$ plane.

TABLE II
THE STATISTICAL PARAMETERS OF THE TWO 3-D DISTRIBUTIONS FOR THE EXAMPLE OF FIG. 4 AND PERFORMANCE COMPARISON OF TWO DIFFERENT CLASSIFIERS

	Generated Data		Baysian Classifier		BQMP Thresholding	
	Mean Vector	Covariane Matrix	Mean Vector	Covariane Matrix	Mean Vector	Covariane Matrix
Population 1 (1050)	129.65 130.91 129.89	404.8 58.2 73.8 58.2 380.8 70.5 73.8 70.5 410.5	129.39 130.67 129.57	384.8 39.8 58.9 39.8 365.3 54.6 58.9 54.6 398.1	129.26 131.12 127.64	389.8 29.7 110.2 29.7 339.6 92.8 110.2 92.8 502.8
Population 2 (1050)	78.79 80.43 80.45	405.3 63.3 84.4 63.3 422.1 68.2 84.4 68.2 394.0	77.78 79.41 79.51	384.0 42.8 65.1 42.8 397.9 49.2 65.1 49.2 375.3	77.81 77.86 81.37	387.0 31.1 115.1 31.1 371.6 84.7 115.1 84.7 452.5
Error Rate			0.030		0.050	
Confusion Matrices			1021 29 34 1016		1000 50 57 993	

quaternion-moment-based operators are generated for each presented application.

A. Color Image Compression

Based on the BQMP thresholding, a quaternion-moment-based operator can be designed to do color image compression by using the concept of BTC. Different from just thresholding intensity component of input color image as [7], the proposed operator can cluster the color points of a pixel block into two classes even though the intensity value fluctuation of the color points is small. In implementation, the input color image is first divided into several square pixel blocks, which is 4×4 in the following experiment. Then, for each pixel block, the BQMP thresholding is used to segment the pixel block into two pixel classes with a corresponding binary bitmap to indicate the membership of each pixel within the block.

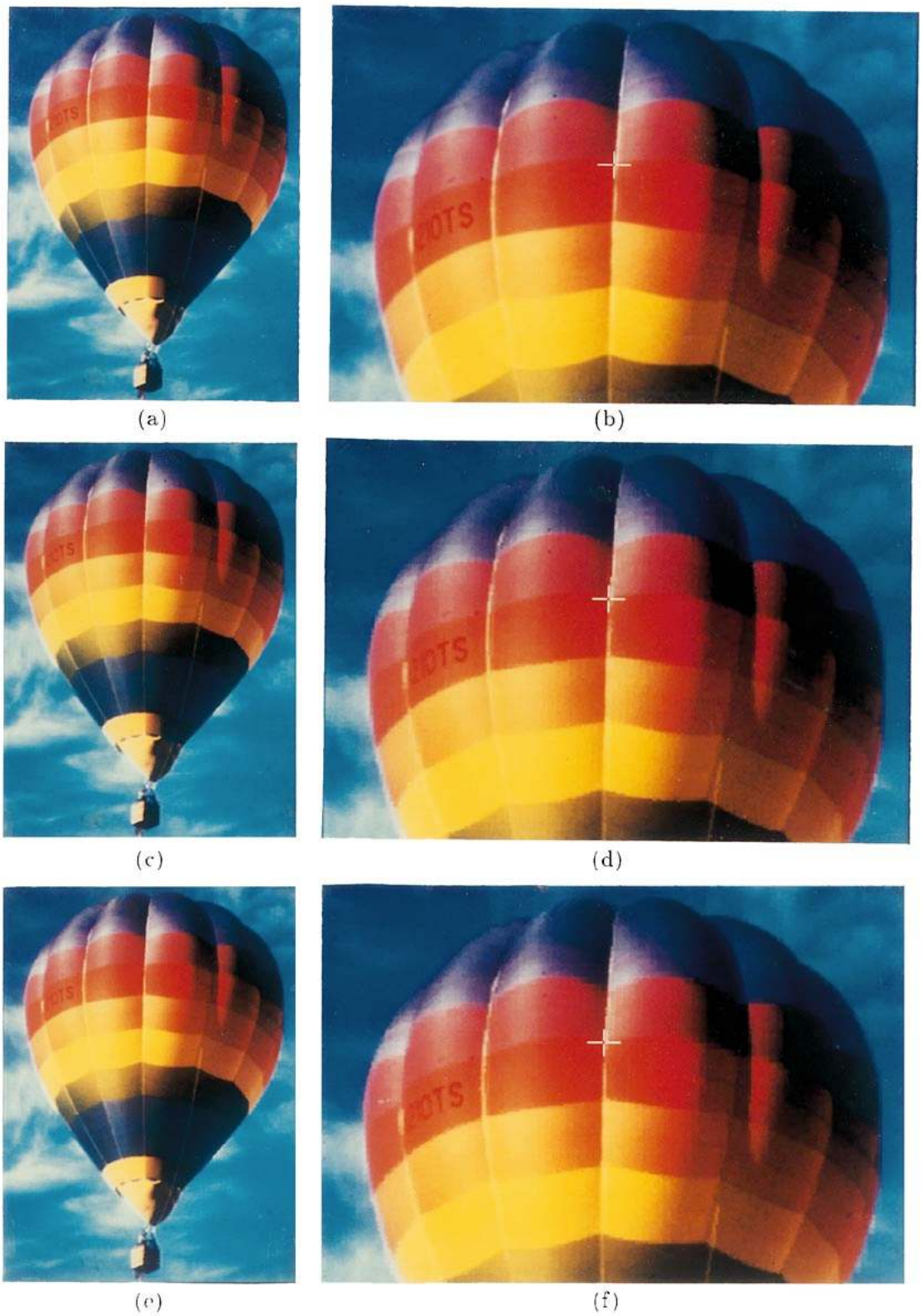


Fig. 5. Original and reconstructed balloons image. (a) Original image. (b) Expanded view of (a). (c) Reconstructed image by the quaternion moment-based BTC operator. (d) Expanded view of (c). (e) Reconstructed image by Wu and Coll's algorithm [7]. (f) Expanded view of (e).

The reconstructed colors of each segmented pixel class is chosen as the centroid of color points of each class. In decoding, the reconstructed image is built by using binary bitmap of each block and the reconstructed colors of each block.

To illustrate the performance of the proposed operator, we have tested a color image of balloons, which is the RGB image shown in Fig. 5(a) with 360 $\times$ 480 size coded at 8 b/pixel/component. Fig. 5(b) illustrates the expanded view of original image. For the purpose of comparison, we also used

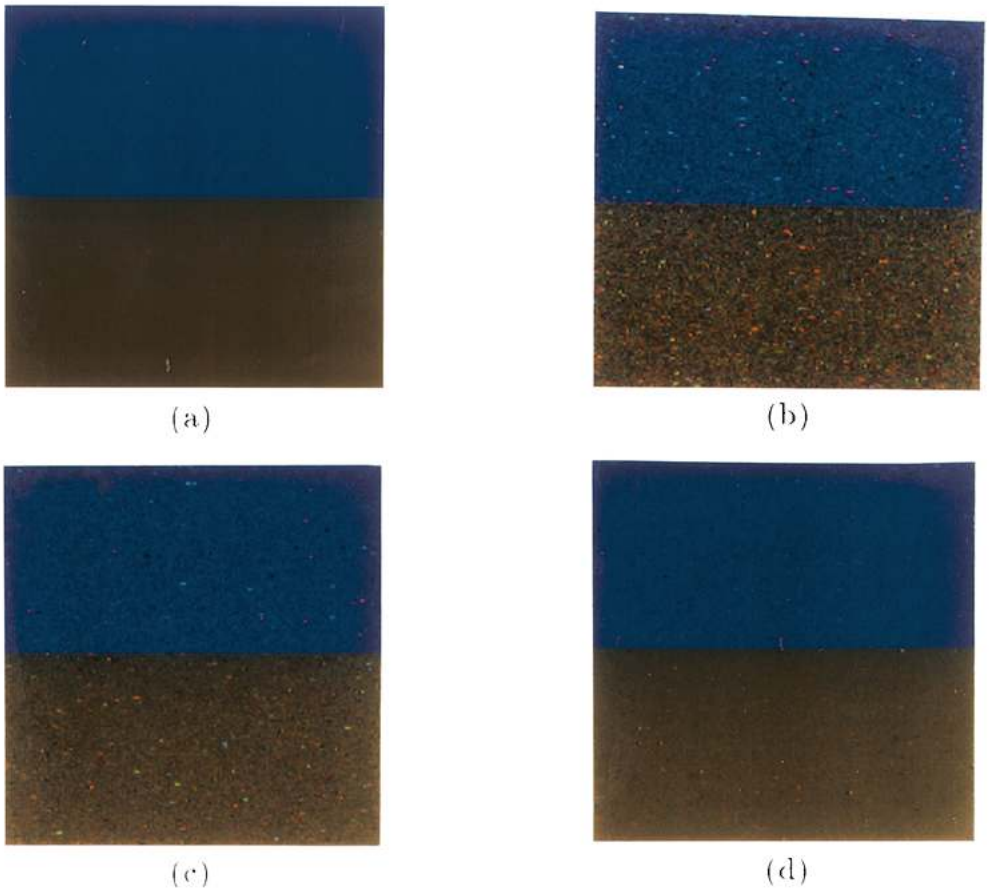


Fig. 6. Synthetic images of ideal color step edge. (a) Original image. (b) Gaussian noise added image of (a) with SNR being 8.6 dB. (c) Gaussian noise added image of (a) with SNR being 13.7 dB. (c) Gaussian noise added image of (a) with SNR being 21.5 dB.

Wu and Coll's BTC coder [7] to compress the test image. The reconstructed image and its expanded view by the proposed quaternion-moment-based BTC operator and [7] are displayed in Fig. 5, too. As we observe, the picture quality of the reconstructed image by the proposed operator is acceptable to human eyes and better than that of [7] especially in the area of chromatic boundaries. The compression ratio for two operators is the same with 4 b/pixel. Moreover, average peak signal-to-noise-ratio (APSNR), which is defined as follows to judge picture quality, for the proposed operator and [7] are 37 and 36 dB, respectively.

$$\mathrm{APSNR}(I_1, I_2, I_3) = 10 \log \frac{3 * N * 255^2}{\mathrm{TSE}(I_1, I_2, I_3)}$$

with TSE representing total square error between the original picture and reconstructed picture. It is also observed that the computation time of the proposed operator taken to compress the test image is only about 3 s using a Sun Ultra I workstation.

B. Subpixel Color Edge Detection

The advantage of gray moment-based edge operator proposed by [2] is that it can provide the subpixel accuracy. However, the edge operator of [2] may fail if it is directly used to find color edges in color images. Considering the synthetic image of ideal color step edge shown in Fig. 6(a), where (R, G, B) color values of upper blue part and lower yellow part are (0, 10, 190) and (110, 90, 0), respectively, and their intensity values are equal on both sides of edge, the operator of [2] would not detect the correct location of the color edge. Since two colors on both sides of edge are apparently different, the application of the proposed BQMP thresholding could localize the color edge.

To design a quaternion-moment-based operator for subpixel color edge detection, we have used the mechanism similar to that of [2]. The input image is first divided into a set of contiguous overlapping 4.5-unit circular pixel disks. The distance between the centers of every two neighboring disks is five pixels. In a disk, color value (R, G, B) of nth pixel with the coordinates (x_n, y_n) are denoted as $(R(x_n, y_n), G(x_n, y_n), B(x_n, y_n))$. The computation steps for the detection of color edge in a disk are summarized as follows.

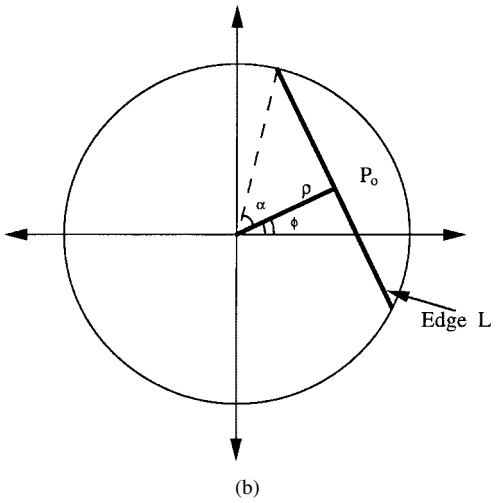
1) Compute the quaternion moments $\hat{m}_1$, $\hat{m}_2$, and $\hat{m}_3$ defined by (7) as a weighted sum of pixel quaternion values in the input disk.

$$\hat{m}_1 = \sum_{n=1}^{69} w_n \cdot \hat{q}(n)$$
$$\hat{m}_2 = \sum_{n=1}^{69} w_n \cdot \hat{q}(n) \cdot \hat{q}^*(n)$$
$$\hat{m}_3 = \sum_{n=1}^{69} w_n \cdot \hat{q}(n) \cdot \hat{q}^*(n) \cdot \hat{q}(n) \qquad (28)$$

1 2 3 4 5
6 7 8 9 10 11 12
13 14 15 16 17 18 19 20 21
22 23 24 25 26 27 28 29 30
31 32 33 34 35 36 37 38 39
40 41 42 43 44 45 46 47 48
49 50 51 52 53 54 55 56 57
58 59 60 61 62 63 64
65 66 67 68 69

W_j = 0.084670539 j = 1,5,13,21,49,57,65,69
= 0.013782918 j = 2,4,22,30,40,48,66,68
= 0.015573189 j = 3,31,39,67
= 0.013068037 j = 6,12,58,64
= 0.015719006 otherwise

(a)



(b)

Fig. 7. (a) Indexing and weighting associated with each element of a circular pixel disk. (b) Derivation of edge line equation.

where index n referring to pixels in the input disk, which contains 69 pixels, is shown in Fig. 7(a). w_n, the weighting associated with nth pixel, is also illustrated in Fig. 7(a).

2) Solve the quaternion moment-preserving equations (12) to obtain p_0, p_1, $\hat{z}_0$, $\hat{z}_1$.
3) Solve the following equations to obtain subpixel edge distance ρ

$$\alpha - \tfrac{1}{2}\sin(2\alpha) = \pi \cdot \min\{p_0, p_1\}$$
$$\rho = \cos(\alpha). \tag{29}$$

4) Solve the following equations to obtain edge orientation ϕ

$$\sin\phi = \frac{\overline{y}}{\sqrt{\overline{x}^2 + \overline{y}^2}}$$
$$\cos\phi = \frac{\overline{x}}{\sqrt{\overline{x}^2 + \overline{y}^2}} \tag{30}$$

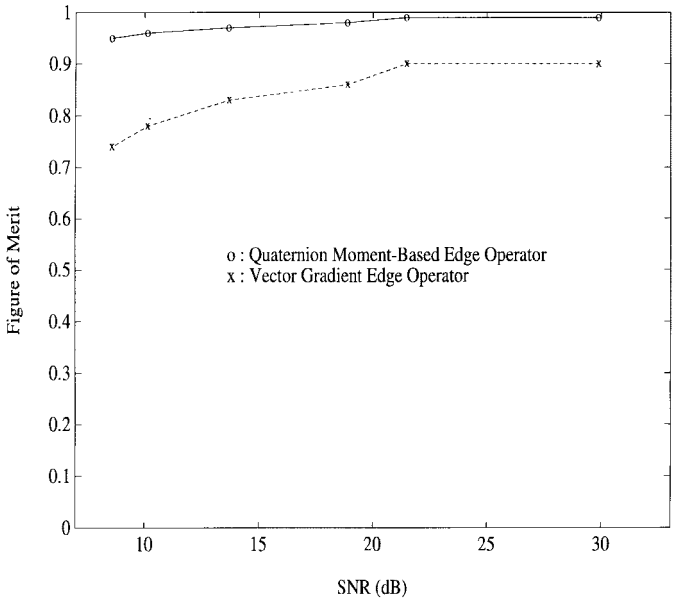


Fig. 8. Figure of merit (FOM) performance comparison of the quaternion moment-based edge operator and the vector gradient edge operator [22].

where

$$\overline{x} = \frac{\sum_{n=1}^{69} x_n \cdot VN(x_n, y_n)}{\sum_{n=1}^{69} VN(x_n, y_n)}$$
$$\overline{y} = \frac{\sum_{n=1}^{69} y_n \cdot VN(x_n, y_n)}{\sum_{n=1}^{69} VN(x_n, y_n)} \tag{31}$$

and $VN(x_n, y_n)$, the vector norm of nth pixel, is denoted as

$$VN(x_n, y_n) = \sqrt{(R(x_n, y_n)^2 + G(x_n, y_n)^2 + B(x_n, y_n)^2)}. \tag{32}$$

5) Formulate the edge line equation

$$y \cdot \sin\phi + x \cdot \cos\phi = -\rho, \qquad p_0 \le p_1$$
$$y \cdot \sin\phi + x \cdot \cos\phi = \rho, \qquad p_0 > p_1 \tag{33}$$

which set pixels closest to the edge line to some predetermined intensity value (e.g., 255) resulting in a binary output image (edge map).

For an input pixel disk, the direction of detected color edge illustrated in Fig. 7(b) is perpendicular to the direction of the vector from the origin to the center of gravity of vector norm. In implementation, the proposed quaternion-moment-based edge operator is not employed to input disk when the sum of color component variances and $\min\{p_0, p_1\}$ are below threshold T_σ and threshold T_p, respectively. T_p used in the following experiments is 0.2. T_σ used in he following experiments is 800 except the experiment of Fig. 10, which we have chosen T_σ as 400 in order to have better performance. In

TABLE III
RESULT OF APPLYING THE QUATERNION MOMENT-BASED EDGE OPERATOR TO ONE INPUT CIRCULAR EDGE PATTERN OF FIG. 6(a)

2D Edge Pattern	(0, 10, 190) (110, 90, 0) (110, 90, 0)
Two Representatives Selected by the BQMP Thresholding	(109.9978, 89.9984, 0.0037) (0.0021, 10.0015, 189.9962)
Edge Line Equation	0.0000 X + 1.0000 Y = 0.03479

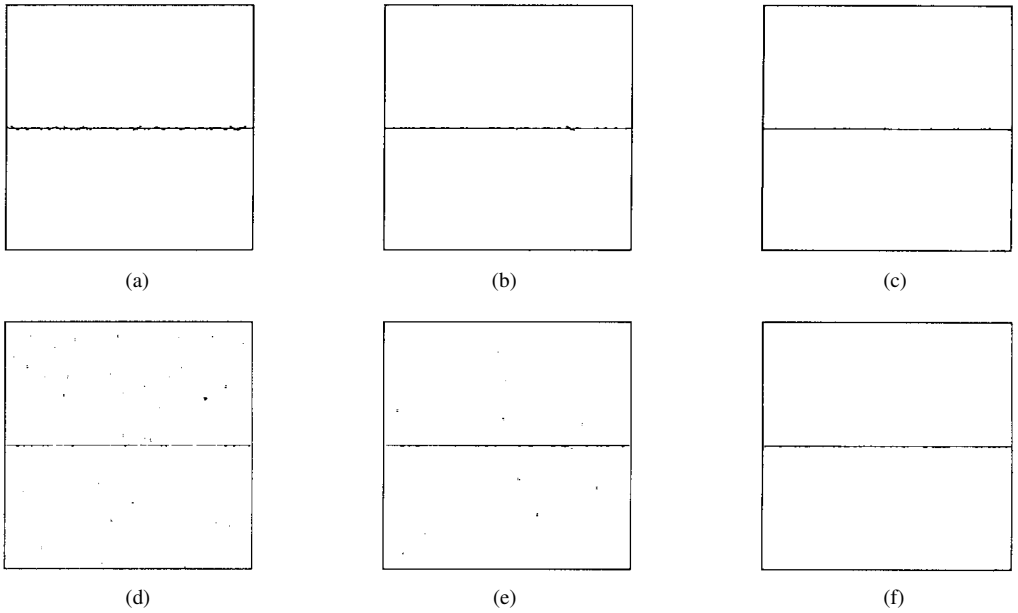


Fig. 9. Edge maps of Fig. 6(b)–(d). (a)–(c) Edge maps produced by the quaternion moment-based edge operator when SNR is equal to 8.6, 13.7, and 21.5 dB, respectively. (d)–(f) Edge maps produced by vector gradient edge operator [22] when SNR is equal to 8.6, 13.7, and 21.5 dB, respectively.

Table III, the empirical results are shown when the proposed operator is applied to one circular input color edge pattern of Fig. 6(a). As we see, the proposed color edge operator correctly locates the color edge in subpixel accuracy. To simulate the performance of the proposed operator under the noisy environment, we add different portion of Gaussian noise to ideal step-edge color image of Fig. 6(a). Fig. 8 shows the simulation results where the performance measure is Pratt's figure of merit (FOM) [21]:

$$\mathrm{FOM} = \frac{1}{\max\{N_I, N_D\}} \sum_{n=1}^{N_D} \frac{1}{1+\beta(d_n)^2}$$

with N_I, N_D are the number of ideal edge points and the number of detected points, respectively. β is a calibration constant with value 1/14 in our working examples, and d_n is the deviation from the ideal edge for the nth detected edge point. And the SNR is defined as

$$\mathrm{SNR} = 10\log \frac{\sigma_{Ri}^2 + \sigma_{Gi}^2 + \sigma_{Bi}^2}{\sigma_{Rn}^2 + \sigma_{Gn}^2 + \sigma_{Bn}^2}$$

where σ_{Ri}^2, σ_{Gi}^2, and σ_{Bi}^2 are color component variances of input original image. σ_{Rn}^2, σ_{Gn}^2, and σ_{Bn}^2 are color component variances of noise-added image. In Fig. 8, the performance of a color edge operator, called the *vector gradient edge operator* [22], is also illustrated. The resultant edge map of vector

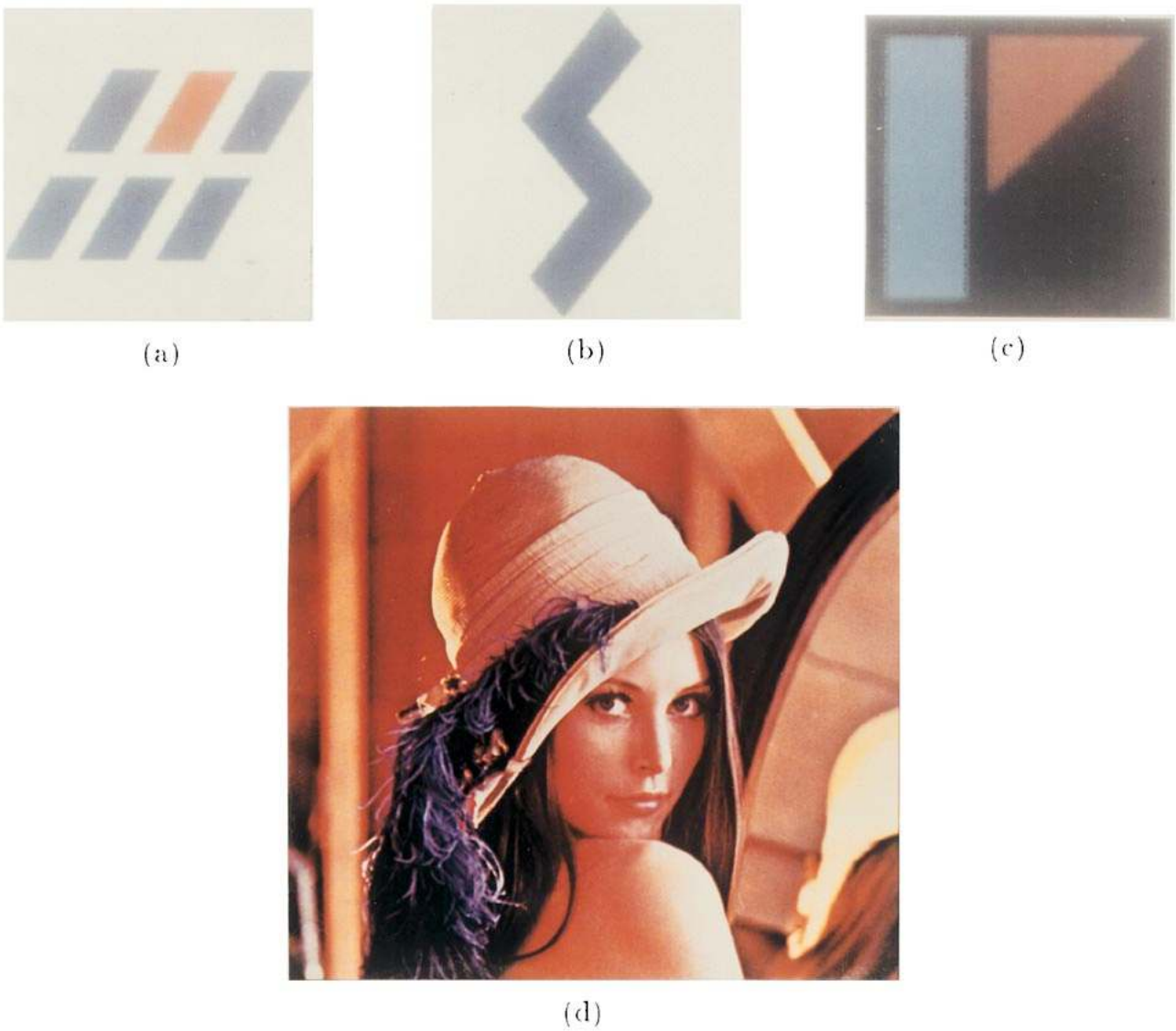


Fig. 10. Original test images. (a)–(c) Trademark images. (d) Lena image.

gradient edge operator is obtained by selecting those points whose gradient magnitudes are above a given threshold. We have set the threshold as the value greater than 90% of gradient magnitudes for this and following experiments. In Fig. 8, it is observed that vector gradient edge operator has good performance for low noise level, although the performance of the proposed operator is slightly better. However, as the SNR decreases, the superiority of the proposed operator becomes clear. Fig. 9 contains the edge maps of two operators when input noisy images are Fig. 6(b)–(d) with SNR equal to 8.6, 13.7, and 21.5 dB, respectively.

To test real color images, we have first chosen three RGB trademark images with 128 $\times$ 128 size, which are coded at 8 b/pixel/component and illustrated in Fig. 10(a)–(c). As we see, these images are mainly composed of objects with straight lines. Fig. 11 also displays the results of the application of our edge operator and [22]. Since [22] is a difference operator that responds to changes in color levels, we notice that the detected edge lines of [22] is sensitive to the effect of noise and thicker than those of our operator. On the other hand, since the proposed operator utilizes a linear line equation to estimate the edge, we observe that it does not perform well at object corners. To overcome the problem of locating the corner points, the approach such as [23] or [24] can be adopted, and is still under the study by the authors. Concerning the computation time, the proposed operator took about 1 s using a Sun Ultra I workstation to process a test image.

Furthermore, we have tested the RGB image Lena with 512 $\times$ 512 size, which is coded at 8 b/pixel/component and shown in Fig. 10(d). The result of the proposed operator shown in Fig. 11(g) is compared with that of edge operator of [22] shown in Fig. 11(h). Although the proposed operator is expected to fail at corners, a visual evaluation gives the impression that the proposed operator performs better. The computation time of the proposed operator to process this test image is about 13 s using a Sun Ultra I workstation.

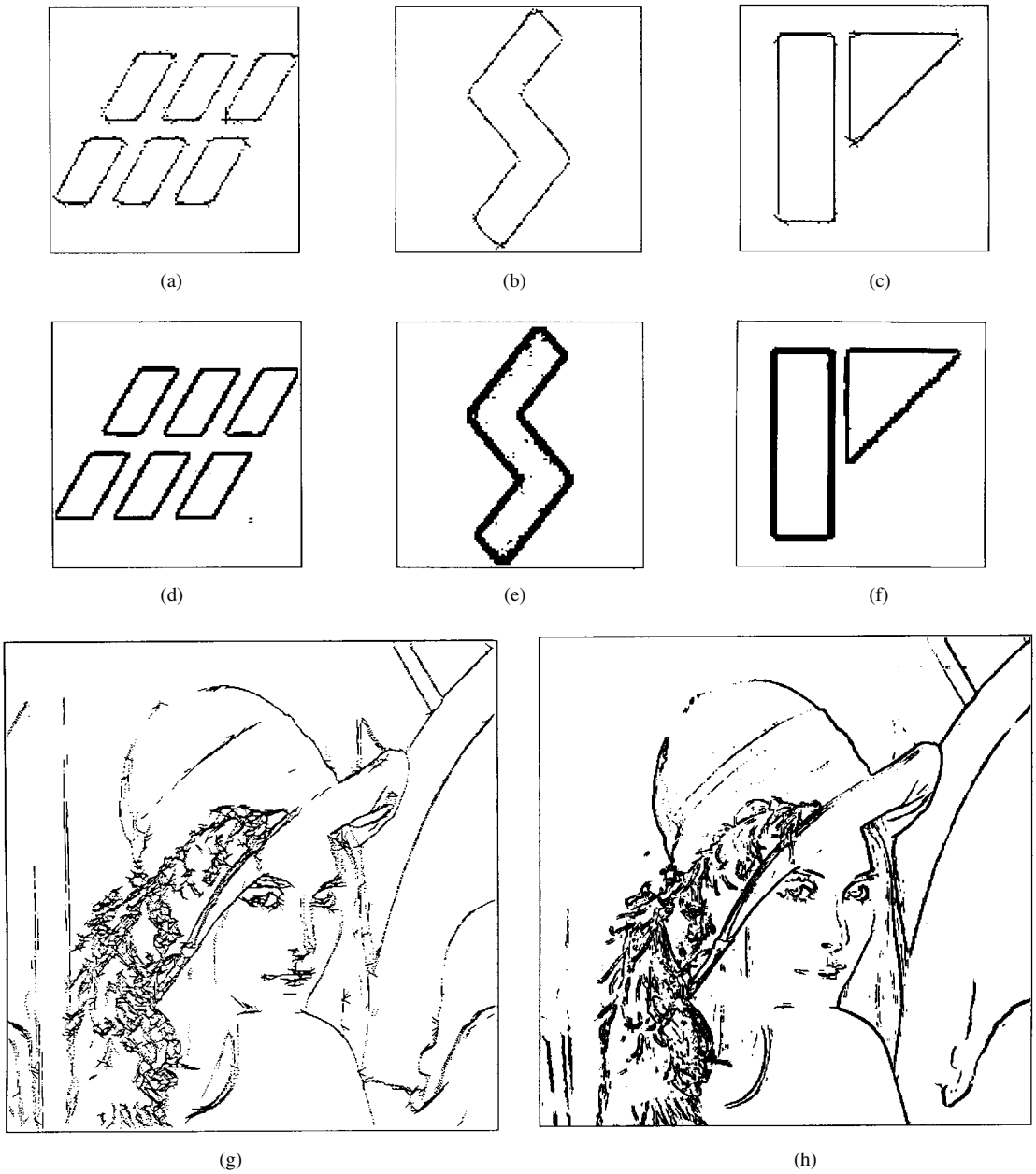


Fig. 11. Associated edge maps of test images of Fig. 10. (a)–(c) Edge maps of trademark images produced by the quaternion moment-based edge operator. (d)–(f) Edge maps of trademark images produced by vector gradient edge operator [22]. (g) Edge map of image Lena produced by the quaternion moment-based edge operator. (h) edge map of image Lena produced by vector gradient edge operator [22].

C. Multiclass Clustering of Color Data

Although the BQMP thresholding is primarily a two-class clustering operator, it can be extended to do multiclass clustering by the binary decision tree approach. The proposed operator will try to split a data set Q until the predetermined number of clusters, M, is reached. The resultant partitioning of Q will exhibit the structure of binary tree. Each node of the tree represents a subset of Q, and the children of any node split the members of the parent node into two sets. The method used for splitting nodes is the BQMP thresholding. Whether a node can be split further or not is indicated by the number of members in its child sets. If any of child sets is empty, then this node will be declared as unsplitable. For those splitable nodes, the criterion to determine which node should be split in the next stage is the variance of the node. The operation steps of this quaternion moment-based operator can be described as follows.

1) Input the data set Q
2) Do the following $M-1$ times or when no nodes are splitable:
   a) find a splitable node such that its variance is maximum;
   b) use the BQMP thresholding to form two new nodes.
3) Assign the membership of clusters formed by step 2 to the data points in Q. Then, choose the centroid of each cluster as a representative color.

One application of multiclass clustering by the BQMP thresholding is color quantization [25], which tries to cluster the input color pixels into M class and extract a repre-

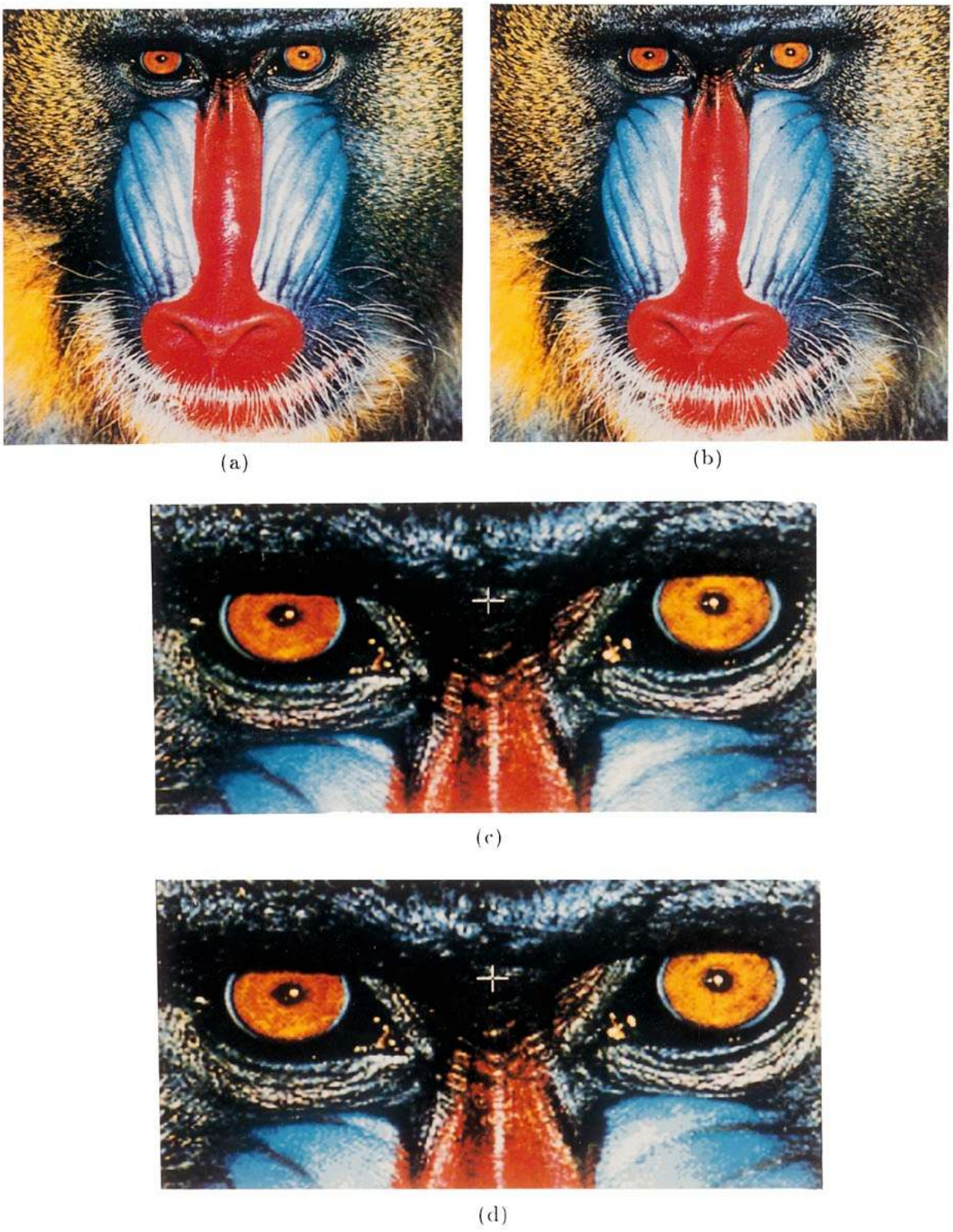


Fig. 12. Original and quantized images of baboon. (a) Original image. (b) Quantized image by the quaternion moment-based operator. (c) Expanded view of (a). (d) Expanded view of (b).

sentative color from each class. The representative color of each segmented pixel class is chosen as the centroid of color pixels of each class. To reveal the efficiency of the proposed multiclass clustering operator, we have chosen in the following experiment a test color image, named baboon, whose original and expanded view are shown in Fig. 12(a) and (c), respectively. Baboon is an RGB image with 512 $\times$ 512 size coded at 8 b/pixel/component. That means this image

utilizing 24 b or one of the 16 million colors to specify the color of each pixel. We desire to extract 256 colors from the color data of baboon. In this way, the number of clustered class is 256, or $M = 256$. The color quantization result and its expanded view by the proposed operator are shown in the Fig. 12(b) and (d). The APSNR for color quantized image is 30.2 dB. As we observe, the picture quality of color quantized image by the proposed operator is acceptable and satisfactory as compared with the original test image.

V. Conclusions

In this paper, we propose a binary thresholding method, called the BQMP thresholding, which is based on the quaternion-moment-preserving principle. Thresholding methods of traditional 1-D gray level moment-based operators can be considered as a special case of the proposed thresholding method. Through preserving first three quaternion moments, an analytic solution can be obtained. The computational complexity of the proposed algorithm is of order N, the data size. The performance analysis is given to show the behavior of the proposed thresholding method. For normal-distribution test data sets, the classified results of the proposed thresholding method is close to that of the optimum Bayes classifier. Quaternion-moment-based operators for applications of color image processing, such as color image compression, subpixel color edge detection, and multiclass clustering of color data are then presented by using the BQMP thresholding. The reconstructed images of the proposed operators for color image compression and color quantization are acceptable to human eyes. Furthermore, experimental results of color edge detection showed that the proposed operator can perform as a subpixel color edge operator.

Acknowledgment

The authors would like to thank the anonymous reviewers who made many useful comments. Their help is gratefully appreciated.

References

[1] E. J. Delp and O. R. Mitchell, "Image compression using block truncation coding," *IEEE Trans. Commun.*, vol. COMM-27, pp. 1335–1341, Sept. 1979.
[2] A. J. Tabatabai and O. R. Mitchell, "Edge location to subpixel values in digital imagery," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-6, pp. 188–201, 1984.
[3] W. Tsai, "Moment preserving thresholding: A new approach," *Comput. Vis., Graph., Image Process.*, 1985, pp. 377–393.
[4] A. K. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1991.
[5] P. K. Sahoo, S. Soltani, and A. K. C. Wong, "A survey of thresholding techniques," *Comput. Vis., Graph., Image Process.*, 1988, pp. 233–260.
[6] M. D. Lema and O. R. Mitchell, "Absolute moment block truncation coding and application to color image," *IEEE Trans. Commun.*, vol. COMM-32, pp. 1148–1157, Oct. 1984.
[7] Y. Wu and D. C. Coll, "Single bit-map block truncation coding of color images," *IEEE J. Select. Areas Commun.*, vol. 10, pp. 952–959, June 1992.
[8] S. C. Pei and C. M. Cheng, "A fast two-class classifier for 2D data using complex-moment-preserving principle," *Pattern Recognit.*, vol. 29, pp. 519–531, Mar. 1996.
[9] C. K. Yang, T. C. Wu, J. C. Lin, and W. Tsai, "Color image sharpening by moment-preserving techniques," *Signal Process.*, vol. 45, pp. 397–403, 1995.
[10] G. Healey, "Segmenting images using normalized color," *IEEE Trans. Syst., Man, Cybern.*, vol. 22, pp. 64–73, 1992.
[11] R. Machuca and K. Phillips, "Applications of vector fields to image processing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-5, pp. 318–329, 1983.
[12] J. B. Fraleigh, *A First Course in Abstract Algebra*. Reading, MA: Addison-Wesley, 1982.
[13] C. K. Papadopoulos and C. L. Nikias, "Parameter estimation of exponentially damped sinusoids using high order statistics," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, pp. 1424–1435, Aug. 1990.
[14] S. L. Marple, *Digital Spectral Analysis with Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1987.
[15] J. C. Lin and W. Tsai, "Feature-preserving clustering of 2D data for two-class problems using analytic formulas: An automatic and fast approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 5, pp. 554–560, May 1994.
[16] A. M. Mood, F. A. Graybill, and D. C. Bose, *Introduction to the Theory of Statistics*. New York: McGraw Hill, 1974.
[17] M. Law and W. D. Kelton, *Simulation Modeling and Analysis*. New York: McGraw Hill, 1991.
[18] H. L. Van trees, *Detection, Estimation, and Modulation Theory*. New York: Wiley, 1968.
[19] A. Papoulis, *Probability, Random Variables and Stochastic Processes*. New York: McGraw Hill, 1984.
[20] *International Mathematical and Statistical Library*, IMSL, Houston, TX, 1989.
[21] W. Pratt, *Digital Image Processing*. New York: Wiley, 1991.
[22] H. C. Lee and D. R. Cok, "Detecting boundaries in a vector field," *IEEE Trans. Signal Processing*, vol. 39, pp. 1181–1194, 1991.
[23] L. H. Chen and W. Tsai, "Moment-preserving curve detection," *IEEE Trans. Syst., Man, Cybern.*, vol. 18, pp. 148–158, 1988.
[24] S. T. Liu and W. Tsai, "Moment-preserving corner detection," *Pattern Recognit.*, vol. 23, pp. 441–460, May 1990.
[25] P. Heckbert, "Color image quantization for frame buffer display," *Comput. Graph.*, vol. 16, pp. 297–307, July 1982.

Soo-Chang Pei (M'86–SM'89) was born in Soo-Auo, Taiwan, R.O.C., in 1949. He received the B.S.E.E. degrees from National Taiwan University (NTU), Taipei, in 1970, and the M.S.E.E. and Ph.D. degrees from University of California, Santa Barbara (UCSB), in 1972 and 1975, respectively.

He was an engineering officer in the Chinese Navy Shipyard from 1970 to 1971. From 1971 to 1975, he was a Research Assistant at UCSB. He was Professor and Chairman in the Electrical Engineering Department, Tatung Institute of Technology, from 1981 to 1983. He is currently Professor in the Electrical Engineering Department, NTU. His research interests include digital signal processing, image processing, optical information processing, and laser holography.

Dr. Pei is member of Eta Kappa Nu and the Optical Society of Ameria.

Ching-Min Cheng was born in Taipei, Taiwan, R.O.C., in 1959. He received the B.S.E.E. degree from National College of Marine Science and Technology, Keelung, Taiwan, in 1982, the M.S.E.E. degree from the University of California, San Diego, in 1986, and the Ph.D. degree in electrical engineering from National Taiwan University, Taipei, in 1996.

From 1983 to 1984, he was an Engineering Officer in the Chinese Airforce Antiaircraft Corps. From 1986 to 1989, he served as a Patent Examiner in the National Bureau of Standards. Since September 1989, he has been with the Telecommunication Laboratories, Ministry of Communications, Taiwan, as a Research Engineer. His research interests include digital signal processing, video compression, and multimedia communication.