

An Efficient Method for Computation of Legendre Moments

Huazhong Shu, Limin Luo, Xudong Bao, and Wenxue Yu

*Laboratory of Image Science and Technology, Department of Biomedical Engineering,
Southeast University, 210096 Nanjing, People's Republic of China*

and

Guoniu Han

IRMA, Université de Strasbourg, France

Received February 2, 1999; accepted January 27, 2000; published online May 24, 2000

The two-dimensional (2D) and three-dimensional (3D) orthogonal moments are useful tools for 2D and 3D object recognition and image analysis. However, the problem of computation of orthogonal moments has not been well solved because there exist few algorithms that can efficiently reduce the computational complexity. As is well known, the calculation of 2D and 3D orthogonal moments by a straightforward method requires a large number of additions and multiplications. In this paper, an efficient algorithm for computing 2D and 3D Legendre moments is presented. First, a new approach is developed for computing Legendre polynomials with one variable; the corresponding results are then used to calculate 1D Legendre moments. Second, we extend our method to calculating 2D Legendre moments, a more accurate approximation formula when an analog original image is digitized to its discrete form is also discussed, and the relationship between the usual approximation and the new approach is investigated. Finally, an efficient method for computing 3D Legendre moments is developed. As one can see, the proposed algorithm improves the computational efficiency significantly and can be implemented easily for high order of moments. © 2000 Academic Press

1. INTRODUCTION

Moment functions of image intensity values have been successfully used in object recognition [2–5], image analysis [6–8], object representation [9], edge detection [10–13], and texture analysis [14]. A survey of these applications can be found in Ref. [1]. Examples of moment-based feature descriptors include Cartesian geometrical moments, rotational

moments, orthogonal moments, and complex moments. Moments with an orthogonal basis set (e.g., Legendre and Zernike polynomials) can be used to represent the image with a minimum amount of information redundancy. These orthogonal moments and their inverse transforms have been used in the field of pattern representation, image analysis, and image reconstruction with some success [6, 8, 14, 15]. As is well known, the difficulty in the use of moments is due to their high computational complexity, especially when a higher order of moments is used. To solve this problem, many research works have been proposed to improve the accuracy and efficiency of moment calculations [17–25], but these methods mainly focus on 2D and 3D geometric moments. Orthogonal moments defined in terms of Legendre and Zernike polynomials have not been analyzed in detail from the point of view of reducing the number of computing operations. Recently, Mukundan and Ramakrishnan [25] used Green's theorem and then proposed a recursive algorithm for computing the 2D Legendre moments. Their method is efficient, but not accurate enough, since Mukundan and Ramakrishnan used a trapezoidal integration rule to approximate the integral function for Legendre moments. Liao and Pawlak [8] proposed a more accurate approximation formula for computing the 2D Legendre moments of a digital image when an analog original image was digitized. Then they used an alternative extended Simpson's rule to numerically calculate a double integral function for a higher order of Legendre moments in each pixel. These orthogonal moments have been successfully used to reconstruct some Chinese characters. The method proposed by Liao and Pawlak is accurate, but needs much more computation.

In this paper, we present a new method for calculating 2D and 3D Legendre moments that consists of transforming the Legendre moments as a combination of the geometric moments in which the coefficients can be easily derived. Then the existing algorithms for computing the geometric moments are applied to improve the computational efficiency in the calculation of the Legendre moments. The paper is organized as follows: We first describe a new approach for computing 1D Legendre moments; this is the object of Section 2. As one can see, the algorithm presented is very different from the known algorithms. In Section 3, we extend our method to calculating the 2D Legendre moments; the reduction of the number of additions and multiplications of the method proposed by Liao and Pawlak [8] will also be discussed in the same section. The generalization of the method to 3D objects is given in Section 4. Section 5 contains some discussions, and concluding remarks are given in Section 6.

2. CALCULATION OF 1D LEGENDRE MOMENTS

The 2D Legendre moment of order $p + q$ of an object with intensity function $f(x, y)$ is defined as [6]

$$L_{pq} = \frac{(2p+1)(2q+1)}{4} \int_{-1}^1 \int_{-1}^1 P_p(x)P_q(y)f(x, y) dx dy, \quad (1)$$

where the kernel function $P_p(x)$ denotes the p th-order Legendre polynomial and is given by

$$P_p(x) = \sum_{k=0}^p C_{pk}[(1-x)^k + (-1)^p(1+x)^k], \quad (2)$$

with

$$C_{pk} = \frac{(-1)^k}{2^{k+1}} \frac{(p+k)!}{(p-k)!(k!)^2}. \tag{3}$$

Since Legendre polynomials are orthogonal over the interval $[-1, 1]$ [26], a square image of $N \times N$ pixels with intensity function $f(i, j)$, with $1 \leq i, j \leq N$, must be scaled to be within the region $-1 \leq x, y \leq 1$. When an analog original image is digitized to its discrete form, the 2D Legendre moments L_{pq} defined by Eq. (1) is usually approximated by the formula

$$L_{pq} = \frac{(2p+1)(2q+1)}{(N-1)^2} \sum_{i=1}^N \sum_{j=1}^N P_p(x_i)P_q(y_j)f(x_i, y_j), \tag{4}$$

where $x_i = (2i - N - 1)/(N - 1)$ and $y_j = (2j - N - 1)/(N - 1)$, and for a binary image, $f(x_i, y_j)$ is given as

$$f(x_i, y_j) = \begin{cases} 1, & \text{if } (i, j) \text{ is in the original object,} \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

As indicated by Liao and Pawlak [8], Eq. (4) is not a very accurate approximation of Eq. (1). To improve the accuracy, they proposed to use the approximated form

$$\tilde{L}_{pq} = \frac{(2p+1)(2q+1)}{4} \sum_{i=1}^N \sum_{j=1}^N h_{pq}(x_i, y_j)f(x_i, y_j), \tag{6}$$

where

$$h_{pq}(x_i, y_j) = \int_{x_i-\Delta x/2}^{x_i+\Delta x/2} \int_{y_j-\Delta y/2}^{y_j+\Delta y/2} P_p(x)P_q(y) dx dy \tag{7}$$

with $\Delta x = x_i - x_{i-1} = 2/(N - 1)$ and $\Delta y = y_j - y_{j-1} = 2/(N - 1)$.

To evaluate the double integral $h_{pq}(x_i, y_j)$ defined by Eq. (7), an alternative extended Simpson’s rule was proposed by Liao and Pawlak. These values were then used to calculate the 2D Legendre moments \tilde{L}_{pq} defined by Eq. (6). Therefore, this method requires a large number of computing operations. One goal of this paper is to give an efficient approach for computing \tilde{L}_{pq} defined by Eq. (6). As one can see, \tilde{L}_{pq} can be expressed, with the help of a useful formula that will be given below, as a linear combination of L_{mn} defined by (4), with $0 \leq m \leq p, 0 \leq n \leq q$. For this reason, we consider first the computation of L_{pq} .

Equations (4) and (5) show that the main cost in Legendre moment computation is calculating $P_p(x_i)P_q(y_j)$; therefore this is where we first concentrate our attention. Here we discuss the following 1D Legendre moment with $f(x) = 1$ in a given interval and $f(x) = 0$ otherwise:

$$L_p = \frac{2p+1}{N-1} \sum_{i=1}^N P_p(x_i). \tag{8}$$

As is well known, a usual method for calculating $P_p(x_i)$ is to utilize the recursive relation

$$P_{l+1}(x) = \frac{2l+1}{l+1}xP_l(x) - \frac{l}{l+1}P_{l-1}(x), \quad \text{for } l \geq 1 \tag{9}$$

with $P_0(x) = 1$ and $P_1(x) = x$.

Using such a strategy, $O(N)$ additions and multiplications are needed for computing L_p . Another possible way is to express the Legendre moments, by substituting Eq. (2) into (1), as a function of the geometric moments, which can be calculated by means of efficient algorithms available. However, it is difficult to adopt such a strategy for computing a higher order of Legendre moments, due to the fact that the large value of coefficients appears in the expression, and more importantly, these coefficients are not easily derived. To overcome the disadvantages mentioned above, we present a new algorithm that consists of transforming the Legendre moments into a function of the geometric moments, whose coefficients are easily calculated. Then we utilize the existing algorithms for the geometric moments to reduce the computational complexity of Legendre moments. To do this, we will use the following basic results.

For two real numbers x and a , we want to express $P_p(x + a)$ in the separable form

$$P_p(x + a) = \sum_{k=0}^p \lambda_k^p(a) P_{p-k}(x). \tag{10}$$

As is easily seen, the key is the calculation of $\lambda_k^p(a)$, so we turn to it in the following.

From Eq. (2), we have

$$P_p(x + a) = \sum_{k=0}^p C_{pk} [(1 - a - x)^k + (-1)^p (1 + a + x)^k]. \tag{11}$$

Comparing the terms of the monomial x^p on the right-hand sides of Eqs. (10) and (11), it is easy to obtain

$$\lambda_0^p(a) = 1. \tag{12}$$

Similarly, we have

$$\lambda_1^p(a) = (2p - 1)a. \tag{13}$$

To obtain the expression of $\lambda_k^p(a)$ for any value of k with $k \leq p$, two cases need to be distinguished: k is odd and k is even.

2a. $k = 2m - 1$

THEOREM 1. For a given integer p , and for an odd number $k = 2m - 1$ less than or equal to p , $\lambda_{2m-1}^p(a)$ can be deduced by the recursive relation

$$\begin{aligned} \lambda_{2m-1}^p(a) &= \frac{(p - 2m + 1)!}{(2p - 4m + 2)!} \\ &\times \left\{ \frac{1}{2} \sum_{l=0}^{2m-2} \frac{(-1)^l}{2^{2m-l-1}} \frac{(2p-l)!}{l!(2m-l-1)!(p-1)!} [(1+a)^{2m-l-1} - (1-a)^{2m-l-1}] \right. \\ &\left. - \sum_{j=0}^{m-2} \lambda_{2j+1}^p(a) \sum_{l=2j+1}^{2m-1} \frac{(-1)^{l-1}}{2^{2m-l-1}} \frac{(2p-l-2j-1)!}{(l-2j-1)!(2m-l-1)!(p-l)!} \right\} \tag{14} \end{aligned}$$

Proof. It is easily seen that the coefficient of the monomial x^{p-2m+1} on the right-hand side of Eq. (11) can be expressed as

$$(-1)^p \left\{ \sum_{l=0}^{2m-2} C_{p,p-l} C_{p-l}^{2m-l-1} [(1+a)^{2m-l-1} - (1-a)^{2m-l-1}] \right\}, \quad (15)$$

where C_i^j is the combination number and is defined as $C_i^j = i!/(j!(i-j)!)$.

On the other hand, the coefficient of the monomial x^{p-2m+1} on the right-hand side of Eq. (10) is given as

$$2(-1)^{p-1} \sum_{j=0}^{m-1} \lambda_{2j+1}^p(a) \sum_{l=2j+1}^{2m-1} C_{p-2j-1,p-1} C_{p-l}^{2m-l-1}. \quad (16)$$

By equating (15) and (16), we obtain

$$\begin{aligned} \lambda_{2m-1}^p(a) C_{p-2m+1,p-2m+1} &= \frac{1}{2} \sum_{l=0}^{2m-2} C_{p,p-l} C_{p-l}^{2m-l-1} [(1-a)^{2m-l-1} - (1+a)^{2m-l-1}] \\ &\quad - \sum_{j=0}^{m-2} \lambda_{2j+1}^p(a) \sum_{l=2j+1}^{2m-1} C_{p-2j-1,p-1} C_{p-l}^{2m-l-1}. \end{aligned}$$

From Eq. (3), we have

$$\frac{C_{p,p-l} C_{p-l}^{2m-l-1}}{C_{p-2m+1,p-2m+1}} = \frac{(-1)^{l-1}}{2^{2m-l-1}} \frac{(2p-l)!}{l!(2m-l-1)!(p-l)!} \frac{(p-2m+1)!}{(2p-4m+2)!},$$

and

$$\frac{C_{p-2j-1,p-1} C_{p-l}^{2m-l-1}}{C_{p-2m+1,p-2m+1}} = \frac{(-1)^{l-1}}{2^{2m-l-1}} \frac{(2p-l-2j-1)!}{(l-2j-1)!(2m-l-1)!(p-l)!} \frac{(p-2m+1)!}{(2p-4m+2)!}.$$

So we achieve the proof of theorem 1 ■

We can easily deduce the following relations from Theorem 1

$$\lambda_1^p(a) = (2p-1)a, \quad (17)$$

$$\lambda_3^p(a) = \frac{a^3}{3!} (2p-1)(2p-3)(2p-5) + (2p-5)a, \quad (18)$$

$$\begin{aligned} \lambda_5^p(a) &= \frac{a^5}{5!} (2p-1)(2p-3)(2p-5)(2p-7)(2p-9) \\ &\quad + \frac{a^3}{2} (2p-3)(2p-5)(2p-9) + (2p-9)a. \end{aligned} \quad (19)$$

By examining the expressions (17), (18), and (19), it is natural to suppose that

$$\begin{aligned} \lambda_{2m-1}^p(a) &= f_0(p, m)a^{2m-1} + f_1(p, m)a^{2m-3} + \dots + f_i(p, m)a^{2m-2i-1} \\ &\quad + \dots + f_{m-2}(p, m)a^3 + f_{m-1}(p, m)a. \end{aligned} \quad (20)$$

The following theorem can then be used to calculate $f_i(p, m)$

THEOREM 2. For a given integer m , and for $0 \leq i \leq m-1$, we have

$$f_i(p, m) = \frac{(p-2m+1)!}{(2p-4m+2)!} \left\{ \sum_{l=0}^{2i} \frac{(-1)^l}{2^{2m-l-1}} \frac{(2p-l)!}{l!(2m-l-1)!(p-l)!} C_{2m-l-1}^{2i-l} \right. \\ \left. - \sum_{j=0}^{i-1} f_j(p, m-i+j) \sum_{l=2m-2i+2j-1}^{2m-1} \frac{(-1)^{l-1}}{2^{2m-l-1}} \right. \\ \left. \times \frac{(2p-l-2m+2i-2j+1)!}{(l-2m+2i-2j+1)!(2m-l-1)!(p-l)!} \right\}. \quad (21)$$

Proof. From Eq. (20), we have

$$\lambda_{2j+1}^p(a) = \sum_{s=0}^j f_s(p, j+1) a^{2j-2s+1}. \quad (22)$$

Substitution of (20) and (22) into (14) yields

$$\sum_{i=0}^{m-1} f_i(p, m) a^{2m-2i-1} \\ = \frac{(p-2m+1)!}{(2p-4m+2)!} \left\{ \frac{1}{2} \sum_{l=0}^{2m-2} \frac{(-1)^l}{2^{2m-l-1}} \frac{(2p-l)!}{l!(2m-l-1)!(p-l)!} \right. \\ \left. \times [(1+a)^{2m-l-1} - (1-a)^{2m-l-1}] - \sum_{j=0}^{m-2} \sum_{s=0}^j f_s(p, j+1) a^{2j-2s+1} \right. \\ \left. \times \sum_{l=2j+1}^{2m-1} \frac{(-1)^{l-1}}{2^{2m-l-1}} \frac{(2p-l-2j-1)!}{(l-2j-1)!(2m-l-1)!(p-l)!} \right\}.$$

The coefficient of the monomial $a^{2m-2i-1}$ on the right-hand side of this equation is

$$\frac{(p-2m+1)!}{(2p-4m+2)!} \left\{ \sum_{l=0}^{2i} \frac{(-1)^l}{2^{2m-l-1}} \frac{(2p-l)!}{l!(2m-l-1)!(p-l)!} C_{2m-l-1}^{2i-l} \right. \\ \left. - \sum_{j=m-i-1}^{m-2} f_{j-m+i+1}(p, j+1) \sum_{l=2j+1}^{2m-1} \frac{(-1)^{l-1}}{2^{2m-l-1}} \frac{(2p-l-2j-1)!}{(l-2j-1)!(2m-l-1)!(p-l)!} \right\}.$$

Using the change of variable $j = j' + m - i - 1$ in the last term of the above expression, we can deduce expression (21). ■

Theorem 2 shows that $f_i(p, m)$ can be calculated by the recursive relation (21). Using this relation, we have

$$f_0(p, m) = \frac{1}{(2m-1)!} \frac{1}{2^{2m-1}} \frac{(2p)!}{p!} \frac{(p-2m+1)!}{(2p-4m+2)!} \quad (23)$$

$$f_1(p, m) = \frac{2m-3}{(2m-3)!} \frac{1}{2^{2m-2}} \frac{(2p-2)!}{(p-1)!} \frac{(p-2m+1)!}{(2p-4m+2)!} \frac{1}{2p-4m+5} \quad (24)$$

$$f_2(p, m) = \frac{(2m - 4)(2m - 5)}{2!(2m - 5)!} \frac{1}{2^{2m-3}} \frac{(2p - 4)!}{(p - 2)!} \frac{(p - 2m + 1)!}{(2p - 4m + 2)!} \times \frac{1}{(2p - 4m + 7)(2p - 4m + 5)}. \tag{25}$$

Obviously, it is hard to make use of Theorem 2 directly to calculate $f_i(p, m)$ for the large value of i , because Eq. (21) requires a large number of operations. However, by observing the expressions (23), (24), and (25), we find that $f_i(p, m)$ given by Eq. (21) can be much simplified. That is, to calculate it, we have the following theorem:

THEOREM 3. *The function $f_i(p, m)$, for $0 \leq i \leq m - 1$, defined by Eq. (21) can be simplified as*

$$f_i(p, m) = \frac{C_{2m-i-2}^i}{(2m - 2i - 1)!} \frac{1}{2^{2m-i-1}} \frac{(2p - 2i)!}{(p - i)!} \frac{(p - 2m + 1)!}{(2p - 4m + 2)!} \frac{1}{\prod_{j=0}^{i-1} (2p - 4m + 2j + 5)}, \tag{26}$$

where C_{2m-i-2}^i is a combination number.

It is easily seen that expression (26) is equivalent to

$$f_i(p, m) = \frac{1}{2^{2m-2i-2}} \frac{2p - 4m + 3}{i!(2m - 2i - 1)!} \frac{(2m - i - 2)!}{(2m - 2i - 2)!} \frac{(2p - 2i)!}{(p - i)!} \frac{(p - 2m + i + 2)!}{(2p - 4m + 2i + 4)!} \tag{27}$$

To prove Theorem 3, we should verify that $f_i(p, m)$ given by (26) or (27) satisfies expression (21). For this purpose, we first introduce the notations

$$A = \frac{(p - 2m + 1)!}{(2p - 4m + 2)!} \tag{28}$$

$$B_l = \frac{(-1)^l}{2^{2m-l-1}} \frac{(2p - l)!}{l!(2i - l)!(2m - 2i - 1)!(p - l)!} \tag{29}$$

$$D_{j,l} = \frac{(-1)^{l-1}}{2^{2m-l-1}} \frac{(2p - l - 2m + 2i - 2j + 1)!}{(l - 2m + 2i - 2j + 1)!(2m - l - 1)!(p - l)!}. \tag{30}$$

Using the above notations, Eq. (21) can be written as

$$f_i(p, m) = A \left[\sum_{l=0}^{2i} B_l - \sum_{j=0}^{i-1} f_j(p, m - i + j) \sum_{l=2m-2i+2j-1}^{2m-1} D_{j,l} \right]. \tag{31}$$

Then we give the following lemmas:

LEMMA 1. *If $f_i(p, m)$ satisfies the relations*

$$B_{2t} = \sum_{j=0}^t f_j(p, m - i + j) D_{j,2m-2i+2t-1}, \quad t = 0, 1, \dots, i - 1, \tag{32}$$

$$B_{2t+1} = \sum_{j=0}^t f_j(p, m - i + j) D_{j,2m-2i+2t}, \quad t = 0, 1, \dots, i - 1 \tag{33}$$

$$f_i(p, m) = A \left[B_{2i} - \sum_{j=0}^{i-1} f_j(p, m - i + j) D_{j, 2m-1} \right], \tag{34}$$

then $f_i(p, m)$ satisfies Eq. (21) or Eq. (31).

LEMMA 2. For a function $F(t, j)$ defined by

$$F(t, j) = \frac{2(2p - 2H - 4j - 1)(p - 2t)!}{H!(2t + 1)(2t + 2)(p - H - 2t - 1)!} \frac{C_p^j C_{2p-2j}^{2p-2t}}{C_p^{H+j} C_{2p-2H-2j}^{2t+2}}, \tag{35}$$

where $t, p,$ and H are positive integers, then we have

$$\sum_{j=0}^t F(t, j) = 1. \tag{36}$$

LEMMA 3. For a function $F(t, j)$ defined by

$$F(t, j) = \frac{2(2p - 2H - 4j - 1)(p - 2t - 1)!}{H!(2t + 2)(2t + 3)(p - H - 2t - 2)!} \frac{C_p^j C_{2p-2j}^{2p-2t-1}}{C_p^{H+j} C_{2p-2H-2j}^{2t+3}}, \tag{37}$$

where $t, p,$ and H are positive integers, we have

$$\sum_{j=0}^t F(t, j) = 1. \tag{38}$$

The proofs of Lemmas 1–3 and Theorem 3 are deferred to Appendix A.

2b. $k = 2m$

In this case, we can obtain the corresponding results in a way similar to that for $k = 2m - 1$. So we give the following theorems without proof.

THEOREM 4. For a given integer $p,$ and for any even number $k = 2m$ less than or equal to $p,$ $\lambda_{2m}^p(a)$ can be deduced by the following recursive relation:

$$\begin{aligned} &\lambda_{2m}^p(a) \\ &= C_{2p-2m}^{2m} + \frac{(p - 2m)!}{(2p - 4m)!} \left\{ \frac{1}{2} \sum_{l=0}^{2m-1} \frac{(-1)^l}{2^{2m-l}} \frac{(2p - l)!}{l!(2m - l)!(p - l)!} [(1 + a)^{2m-l} + (1 - a)^{2m-l}] \right. \\ &\quad \left. - \sum_{j=0}^{m-1} \lambda_{2j}^p(a) \sum_{l=2j}^{2m} \frac{(-1)^l}{2^{2m-l}} \frac{(2p - l - 2j)!}{(l - 2j)!(2m - l)!(p - l)!} \right\}. \end{aligned} \tag{39}$$

Then we decompose $\lambda_{2m}^p(a)$ for $m \geq 1$ in the form

$$\lambda_{2m}^p(a) = g_0(p, m)a^{2m} + g_1(p, m)a^{2m-2} + \dots + g_i(p, m)a^{2m-2i} + \dots + g_{m-1}(p, m)a^2. \tag{40}$$

THEOREM 5. For a given m , and for $0 \leq i \leq m - 1$, we have

$$g_i(p, m) = \frac{(p - 2m)!}{(2p - 4m)!} \left\{ \sum_{l=0}^{2i} \frac{(-1)^l}{2^{2m-l}} \frac{(2p - l)!}{l!(2m - l)!(p - l)!} C_{2m-l}^{2i-l} - \sum_{j=0}^{i-1} g_j(p, m - i + j) \right. \\ \left. \times \sum_{l=2m-2i+2j}^{2m} \frac{(-1)^l}{2^{2m-l}} \frac{(2p - l - 2m + 2i - 2j)!}{(l - 2m + 2i - 2j)!(2m - l)!(p - l)!} \right\}. \quad (41)$$

THEOREM 6. Expression (41) can be simplified as

$$g_i(p, m) = \frac{C_{2m-i-1}^i}{(2m - 2i)!} \frac{1}{2^{2m-i}} \frac{(2p - 2i)!}{(p - i)!} \frac{(p - 2m)!}{(2p - 4m)!} \frac{1}{\prod_{l=0}^{i-1} (2p - 4m + 2l + 3)}. \quad (42)$$

Based on Theorems 3 and 6, we now propose our method to calculate the 1D Legendre moment defined by (8).

Using Eq. (10), the relation $P_{p-k}(-1) = (-1)^{p-k}$, and the notation $x_i = -1 + a_i$ with $a_i = 2(i - 1)/(N - 1)$, we have

$$L_p = \frac{2p + 1}{N - 1} \sum_{i=1}^N P_p(-1 + a_i) = \frac{2p + 1}{N - 1} \sum_{i=1}^N \sum_{k=0}^p \lambda_k^p(a_i) P_{p-k}(-1) \\ = \frac{2p + 1}{N - 1} \sum_{i=1}^N \sum_{k=0}^p (-1)^{p-k} \lambda_k^p(a_i) = \frac{2p + 1}{N - 1} \sum_{i=1}^N U_p \lambda^p(a_i), \quad (43)$$

where

$$U_p = [(-1)^p, (-1)^{p-1}, (-1)^p, (-1)^{p-1}, \dots, -1, 1]$$

and

$$\lambda^p(a_i) = [\lambda_0^p(a_i), \lambda_1^p(a_i), \dots, \lambda_p^p(a_i)]^T.$$

Here T indicates transposition, and U_p and $\lambda^p(a_i)$ are both $(p + 1)$ -dimensional vectors. By using the notation

$$\lambda^p(a_i) = M_p A_p(a_i), \quad (44)$$

where $A_p(a_i) = [1, a_i, a_i^2, \dots, a_i^p]^T$, M_p is a $(p + 1) \times (p + 1)$ triangle matrix whose elements depend only on p , and the definition of the matrix M_p , for $0 \leq l \leq k \leq p$, is given as

$$M_p(k, l) = \begin{cases} f_{m-j}(p, m), & \text{if } k = 2m - 1 \text{ and } l = 2j - 1, j = 1, 2, \dots, m \\ g_{m-j}(p, m), & \text{if } k = 2m \text{ and } l = 2j, j = 1, 2, \dots, m \\ 0, & \text{otherwise.} \end{cases} \quad (45)$$

Then we have

$$L_p = \frac{2p + 1}{N - 1} U_p M_p \left(\sum_{i=1}^N A_p(a_i) \right) = \frac{2p + 1}{N - 1} U_p M_p D_p G_p, \quad (46)$$

where D_p is a $(p+1) \times (p+1)$ diagonal matrix whose diagonal elements are $D_p(k, k) = (2/(N-1))^k$ for $k = 0, 1, \dots, p$, and

$$G_p = \left[\sum_{i=0}^{N-1} 1, \sum_{i=0}^{N-1} i, \sum_{i=0}^{N-1} i^2, \dots, \sum_{i=0}^{N-1} i^p \right]^T$$

is a $(p+1)$ -dimensional vector whose elements are 1D geometric moments of order up to p .

So our strategy for computing the 1D Legendre moments of order up to p can be described as follows:

1. Use the Pascal Triangle Transform proposed by Li and Shen [21] to calculate 1D geometric moments of order up to p , so the vector G_p can be obtained without multiplication.
2. Construct the matrix M_k with $k \leq p$ by Eq. (45). The 1D Legendre moments of order up to p can then be calculated by using Eq. (46).

Note that using the method proposed above, only the multiplication is needed in the construction of matrix M_k , and the product of M_k and G_k . Therefore, the number of multiplications depends no longer on N , but only on p . It should also be pointed out that expressions (26) and (42) seem to be complex, but in fact, their final results can easily be derived for any given p and m . A list of $\lambda_k^p(a)$, for $k \leq 20 \leq p$, is given in Appendix B.

3. CALCULATION OF 2D LEGENDRE MOMENTS

Based on the result obtained in Section 2, we are now in a position to express the 2D Legendre moments \tilde{L}_{pq} defined by Eq. (6) as a function of L_{pq} defined by Eq. (4).

By using the relation

$$\int P_p(x) dx = [P_{p+1}(x) - P_{p-1}(x)]/(2p+1),$$

we can deduce from Eq. (7) that

$$\begin{aligned} & (2p+1)(2q+1)h_{pq}(x_i, y_j) \\ &= [P_{p+1}(x_i + \Delta x/2) - P_{p+1}(x_i - \Delta x/2) - P_{p-1}(x_i + \Delta x/2) + P_{p-1}(x_i - \Delta x/2)] \\ & \cdot [P_{q+1}(y_j + \Delta y/2) - P_{q+1}(y_j - \Delta y/2) - P_{q-1}(y_j + \Delta y/2) + P_{q-1}(y_j - \Delta y/2)]. \end{aligned} \quad (47)$$

With the help of Eq. (10), we have

$$\begin{aligned} & P_{p+1}(x_i + \Delta x/2) - P_{p+1}(x_i - \Delta x/2) - P_{p-1}(x_i + \Delta x/2) + P_{p-1}(x_i - \Delta x/2) \\ &= \sum_{k=0}^{p+1} \lambda_k^{p+1}(\Delta x/2) P_{p+1-k}(x_i) - \sum_{k=0}^{p+1} \lambda_k^{p+1}(-\Delta x/2) P_{p+1-k}(x_i) \\ & \quad - \sum_{k=0}^{p-1} \lambda_k^{p-1}(\Delta x/2) P_{p-1-k}(x_i) + \sum_{k=0}^{p-1} \lambda_k^{p-1}(-\Delta x/2) P_{p-1-k}(x_i) \\ &= 2 \left[\sum_{k=1}^{p+1} \lambda_k^{p+1}(\Delta x/2) P_{p+1-k}(x_i) - \sum_{k=1}^{p-1} \lambda_k^{p-1}(\Delta x/2) P_{p-1-k}(x_i) \right], \end{aligned} \quad (48)$$

where $\sum_{k=1}^{p+1}$ stands for summation with respect to the odd value of k varied from 1 to $p+1$. Similarly, we have

$$\begin{aligned} & P_{q+1}(y_j + \Delta y/2) - P_{q+1}(y_j - \Delta y/2) - P_{q-1}(y_j + \Delta y/2) + P_{q-1}(y_j - \Delta y/2) \\ &= 2 \left[\sum_{l=1}^{q+1} \lambda_l^{q+1} (\Delta y/2) P_{q+1-l}(y_j) - \sum_{l=1}^{q-1} \lambda_l^{q-1} (\Delta y/2) P_{q-1-l}(y_j) \right]. \end{aligned} \quad (49)$$

Substitution of Eqs. (48) and (49) into Eq. (6) yields

$$\begin{aligned} \tilde{L}_{pq} &= \sum_{k=1}^{p+1} \sum_{l=1}^{q+1} \frac{4\lambda_k^{p+1} (\Delta x/2) \lambda_l^{q+1} (\Delta y/2)}{(2p-2k+3)(2q-2l+3)\Delta x \Delta y} L_{p+1-k, q+1-l} \\ &\quad - \sum_{k=1}^{p+1} \sum_{l=1}^{q-1} \frac{4\lambda_k^{p+1} (\Delta x/2) \lambda_l^{q-1} (\Delta y/2)}{(2p-2k+3)(2q-2l-1)\Delta x \Delta y} L_{p+1-k, q-1-l} \\ &\quad - \sum_{k=1}^{p-1} \sum_{l=1}^{q+1} \frac{4\lambda_k^{p-1} (\Delta x/2) \lambda_l^{q+1} (\Delta y/2)}{(2p-2k-1)(2q-2l+3)\Delta x \Delta y} L_{p-1-k, q+1-l} \\ &\quad + \sum_{k=1}^{p-1} \sum_{l=1}^{q-1} \frac{4\lambda_k^{p-1} (\Delta x/2) \lambda_l^{q-1} (\Delta y/2)}{(2p-2k-1)(2q-2l-1)\Delta x \Delta y} L_{p-1-k, q-1-l}. \end{aligned} \quad (50)$$

Note that Eq. (50) is of the form of the convolution operator. In fact, by introducing the sequences

$$B_{kl}^{pq} = \begin{cases} \frac{4\lambda_k^p (\Delta x/2) \lambda_l^q (\Delta y/2)}{\Delta x \Delta y}, & \text{for all odd values of } (k, l) \\ 0, & \text{otherwise} \end{cases} \quad (51)$$

$$E_{kl} = \frac{L_{kl}}{(2p+1)(2q+1)} \quad (52)$$

Eq. (50) can be written

$$\tilde{L}_{pq} = B_{kl}^{p+1, q+1} * E_{kl} - B_{kl}^{p+1, q-1} * E_{kl} - B_{kl}^{p-1, q+1} * E_{kl} + B_{kl}^{p-1, q-1} * E_{kl}, \quad (53)$$

where $B_{kl}^{pq} * E_{kl}$ denotes the 2D $p * q$ -point discrete convolution operator.

Equation (53) shows that the 2D Legendre moments \tilde{L}_{pq} defined by Eq. (6) can be deduced from the values of L_{mn} with $0 \leq m \leq p$ and $0 \leq n \leq q$. For this reason, we turn to the fast computation of L_{pq} in the following. First, we give a list of relationships between \tilde{L}_{pq} and L_{pq} for $p+q \leq 3$.

$$\begin{aligned} \tilde{L}_{00} &= L_{00} \\ \tilde{L}_{10} &= L_{10}, \quad \tilde{L}_{01} = L_{01} \\ \tilde{L}_{20} &= L_{20} + \frac{5(\Delta x)^2}{8} L_{00}, \quad \tilde{L}_{11} = L_{11}, \quad \tilde{L}_{02} = L_{02} + \frac{5(\Delta y)^2}{8} L_{00} \\ \tilde{L}_{30} &= L_{30} + \frac{35(\Delta x)^2}{24} L_{10}, \quad \tilde{L}_{21} = L_{21} + \frac{5(\Delta x)^2}{8} L_{01} \\ \tilde{L}_{12} &= L_{12} + \frac{5(\Delta y)^2}{8} L_{10}, \quad \tilde{L}_{03} = L_{03} + \frac{35(\Delta y)^2}{24} L_{01} \end{aligned}$$

The first three leading terms in Eq. (50) are

$$\begin{aligned}\tilde{L}_{pq} = & L_{pq} + \frac{(2p+1)(2p-1)(\Delta x)^2}{24} L_{p-2,q} \\ & + \frac{(2q+1)(2q-1)(\Delta y)^2}{24} L_{p,q-2} + O((\Delta x \Delta y)^2).\end{aligned}\quad (54)$$

The above equation shows that the difference between \tilde{L}_{pq} and L_{pq} could be important when the moments of higher order are concerned. Note that the method discussed above is an extension of research by Liao and Pawlak [8]. In fact, they adopted the same strategy to calculate the 2D geometric moments \tilde{m}_{pq} based on the values of m_{pq} , where \tilde{m}_{pq} and m_{pq} are defined as

$$\begin{aligned}m_{pq} &= \frac{4}{(N-1)^2} \sum_{i=1}^N \sum_{j=1}^N x_i^p y_j^q f(x_i, y_j), \\ \tilde{m}_{pq} &= \sum_{i=1}^N \sum_{j=1}^N f(x_i, y_j) \int_{x_i-\Delta x/2}^{x_i+\Delta x/2} \int_{y_j-\Delta y/2}^{y_j+\Delta y/2} x^p y^q dx dy.\end{aligned}$$

We now describe the algorithm for computing the 2D Legendre moments L_{pq} defined by Eq. (4). To do this, a new approach based on Eq. (10) is used to calculate $P_p(x_i)P_q(y_j)$. Denoting $x_i = -1 + a_i$, $y_j = -1 + b_j$, with $a_i = 2(i-1)/(N-1)$, $b_j = 2(j-1)/(N-1)$, we have

$$P_p(x_i)P_q(y_j) = P_p(-1 + a_i)P_q(-1 + b_j) = \sum_{l=0}^p \sum_{m=0}^q P_{p-l}(-1)P_{q-m}(-1)\lambda_l^p(a_i)\lambda_m^q(b_j).\quad (55)$$

Introducing the notation

$$h_u(p, [l/2]) = \begin{cases} g_u(p, l/2), & \text{if } l \text{ is an even number} \\ f_u(p, (l+1)/2), & \text{otherwise,} \end{cases}\quad (56)$$

where $[x]$ denotes the integer part of x , for $0 \leq l \leq p$, we have

$$\lambda_l^p(a_i) = \sum_{u=0}^{[(l-1)/2]} h_u(p, [l/2]) a_i^{l-2u}.\quad (57)$$

Similarly, we have

$$\lambda_m^q(b_j) = \sum_{v=0}^{[(m-1)/2]} h_v(q, [m/2]) b_j^{m-2v}.\quad (58)$$

Substitution of Eqs. (57) and (58) into (55) yields

$$P_p(x_i)P_q(y_j) = \sum_{l=0}^p \sum_m^q (-1)^{p+q-l-m} F_{lm}(a_i, b_j),\quad (59)$$

where

$$F_{lm}(a_i, b_j) = \sum_{u=0}^{[(l-1)/2]} \sum_{v=0}^{[(m-1)/2]} h_u(p, [l/2])h_v(q, [m/2])a_i^{l-2u}b_j^{m-2v}. \quad (60)$$

We now calculate the 2D Legendre moments L_{pq} . We deduce from Eqs. (59) and (60) that

$$L_{pq} = \frac{(2p+1)(2q+1)}{(N-1)^2} \sum_{l=0}^p \sum_{m=0}^q (-1)^{p+q-l-m} \sum_{i=1}^N \sum_{j=1}^N F_{lm}(a_i, b_j) f(x_i, y_j). \quad (61)$$

Equation (61) can be written

$$L_{pq} = \frac{(2p+1)(2q+1)}{4} \sum_{l=0}^p \sum_{m=0}^q \sum_{u=0}^{[(l-1)/2]} \sum_{v=0}^{[(m-1)/2]} H_{uv}(p, q, l, m) G_{l-2u, m-2v}, \quad (62)$$

where

$$H_{uv}(p, q, l, m) = (-1)^{p+q-l-m} h_u(p, [l/2])h_v(q, [m/2]) \quad (63)$$

and G_{ab} denoting the 2D $(a+b)$ th order geometric moment except for a constant multiplicative factor, is defined as

$$G_{ab} = \left(\frac{2}{N-1} \right)^{a+b+2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} i^a j^b f(x_i, y_j). \quad (64)$$

Based on the above discussion, the algorithm for computing the 2D Legendre moments L_{pq} and \tilde{L}_{pq} can be described as follows.

1. The geometric moments G_{ab} of order up to M are calculated by use of a fast method proposed by Yang and Albreghsten [23]. For an image of size $N \times N$, this method needs only $O(N)$ additions and multiplications, respectively.

2. The coefficients $h_u(p, [l/2])$, with $0 \leq p \leq M$, $0 \leq l \leq p$, and $0 \leq u \leq [(l-1)/2]$, are calculated using Eqs. (26) and (42), in which $M(M-1)(2M-1)/6$ multiplications and $M(M-1)(2M-1)/6$ additions are needed, respectively. The values of $H_{uv}(p, q, l, m)$ can then be deduced from Eq. (63).

3. Using Eq. (62) to calculate 2D Legendre moments L_{pq} , with $0 \leq p+q \leq M$. When all the values of G_{ab} and $h_u(p, [l/2])$ have been obtained, this step needs approximately $2M^3(M+1)^3/72^2$ additions and $4M^3(M+1)^3/72^2$ multiplications, respectively.

4. The 2D Legendre moments \tilde{L}_{pq} of order up to M can be calculated with the help of Eq. (53) when the values of L_{pq} , with $0 \leq p+q \leq M$, have been obtained.

Note that Liao and Pawlak [8] adopted an alternative Simpson's formula to calculate the double integral $h_{pq}(x_i, y_j)$ defined by Eq. (7), in which different integration rules, with $I = 3, 8, 13, 18,$ and 23 , were employed. Here I denotes the number of points within a given pixel required to evaluate $h_{pq}(x_i, y_j)$. These values are then used to calculate the 2D Legendre moments \tilde{L}_{pq} . It is obvious that the method used by Liao and Pawlak needs much more computation than the algorithm proposed in this paper. To illustrate the efficiency and the accuracy of our method compared with the methods described by Mukundan and Ramakrishnan [25], and by Liao and Pawlak [26] for computing the 2D Legendre moments,

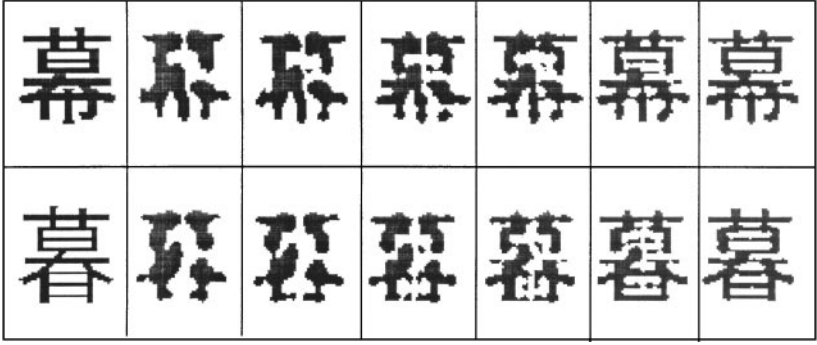


FIG. 1. The two original Chinese characters and their patterns reconstructed using Mukundan and Ramakrishnan's method [25].

we use the three methods to reconstruct some Chinese characters with the help of the inverse transform

$$f(x_i, y_j) \approx \sum_{p=0}^M \sum_{q=0}^p \tilde{L}_{p-q,q} P_{p-q}(x_i) P_q(y_j). \quad (65)$$

Figure 1 shows the two original Chinese characters and their reconstructed pattern from the higher order of Legendre moments by using Mukundan and Ramakrishnan's method. The first column displays the two original characters. The second column to the seventh column illustrate the reconstructed patterns with order of up to 20, 24, 28, 32, 36, and 40, respectively. The results obtained with Eq. (6) using Liao and Pawlak's method and the method proposed in this paper are depicted in Figs. 2 and 3, respectively. Note that in Figs. 2 and 3, the orders of the Legendre moments used in Eq. (65) are the same as those used in Fig. 1. Table 1 shows the CPU times needed to calculate the Legendre moments of order up to 20 and 40 for the three methods mentioned above, respectively. Note also that the programs have been implemented on an AMD K6/300 personal computer.

4. CALCULATION OF 3D LEGENDRE MOMENTS

In this section, we describe briefly an efficient method for computing 3D Legendre moment using the results obtained in Section 2. Similarly to the 2D Legendre moments, the

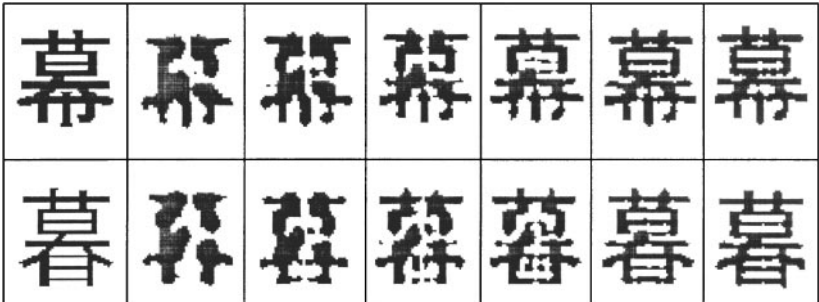


FIG. 2. The two original Chinese characters and their patterns reconstructed using Liao and Pawlak's method [8].

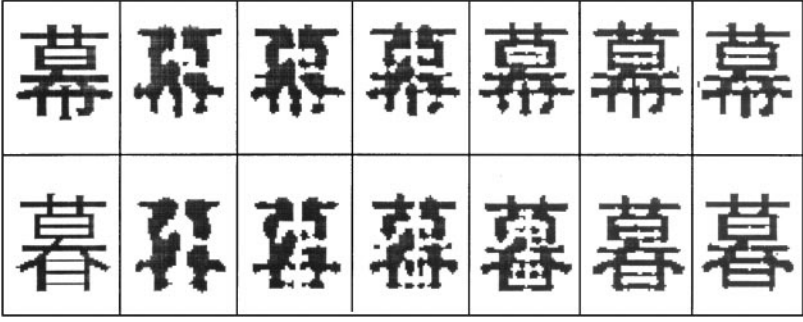


FIG. 3. The two original Chinese characters and their patterns reconstructed using the method presented in this paper.

3D Legendre moment L_{pqr} of order $p + q + r$ of a 3D object is defined as

$$L_{pqr} = \frac{(2p + 1)(2q + 1)(2r + 1)}{8} \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 P_p(x)P_q(y)P_r(z)f(x, y, z) dx dy dz. \quad (66)$$

When an analog original image $f(x, y, z)$ is digitized into its discrete version, Eq. (66) is usually approximated by

$$L_{pqr} = \frac{(2p + 1)(2q + 1)(2r + 1)}{(N - 1)^3} \sum_{i=1}^N \sum_{j=1}^N \sum_{k=1}^N P_p(x_i)P_q(y_j)P_r(z_k)f(x_i, y_j, z_k). \quad (67)$$

More accurately, we can approximate L_{pqr} by

$$\tilde{L}_{pqr} = \frac{(2p + 1)(2q + 1)(2r + 1)}{8} \sum_{i=1}^N \sum_{j=1}^N \sum_{k=1}^N h_{pqr}(x_i, y_j, z_k)f(x_i, y_j, z_k), \quad (68)$$

where

$$h_{pqr}(x_i, y_j, z_k) = \int_{x_i - \Delta x/2}^{x_i + \Delta x/2} \int_{y_j - \Delta y/2}^{y_j + \Delta y/2} \int_{z_k - \Delta z/2}^{z_k + \Delta z/2} P_p(x)P_q(y)P_r(z) dx dy dz. \quad (69)$$

Like 2D Legendre moments \tilde{L}_{pq} , 3D Legendre moments \tilde{L}_{pqr} can also be expressed as a linear combination of L_{klm} defined by Eq. (67), with $0 \leq k \leq p, 0 \leq l \leq q, 0 \leq m \leq r$. Therefore, we discuss here only the efficient computation of L_{pqr} defined by Eq. (67).

Introducing the notation $x_i = -1 + a_i, y_j = -1 + b_j,$ and $z_k = -1 + c_k,$ with $a_i = 2(i - 1)/(N - 1), b_j = 2(j - 1)/(N - 1),$ and $c_k = 2(k - 1)/(N - 1),$ respectively, and

TABLE 1
The CPU Times Needed to Calculate the Legendre Moments of Different Orders with $N = 100$

Maximum order	Mukundan and Ramakrishnan's method [25]	Liao and Pawlak's method [8]	Our method
$M = 20$	0.05 s	5.15 s	0.8 s
$M = 30$	0.06 s	10.5 s	1.4 s
$M = 40$	0.11 s	19 s	2.7 s

using Eq. (10), we obtain

$$\begin{aligned}
 P_p(x_i)P_q(y_j)P_r(z_k) &= P_p(-1 + a_i)P_q(-1 + b_j)P_r(-1 + c_k) \\
 &= \sum_{l=0}^p \sum_{m=0}^q \sum_{n=0}^r P_{p-l}(-1)P_{q-m}(-1)P_{r-n}(-1) \lambda_l^p(a_i) \lambda_m^q(b_j) \lambda_n^r(c_k).
 \end{aligned} \tag{70}$$

Using the function $h_u(p, [l/2])$ defined by Eq. (56), we have

$$P_p(x_i)P_q(y_j)P_r(z_k) = \sum_{l=0}^p \sum_{m=0}^q \sum_{n=0}^r (-1)^{p+q+r-l-m-n} F_{lmn}(a_i, b_j, c_k), \tag{71}$$

where

$$\begin{aligned}
 &F_{lmn}(a_i, b_j, c_k) \\
 &= \sum_{u=0}^{[(l-1)/2]} \sum_{v=0}^{[(m-1)/2]} \sum_{w=0}^{[(n-1)/2]} h_u(p, [l/2])h_v(q, [m/2])h_w(r, [n/2])a_i^{l-2u}b_j^{m-2v}c_k^{n-2w}.
 \end{aligned} \tag{72}$$

We now turn to calculate 3D Legendre moments L_{pqr} . Since $(2p + 1)(2q + 1)(2r + 1)(N - 1)^3$ appearing in Eq. (67) is a normalized factor, we neglect it in the following.

From Eqs. (71) and (72), and by exchanging the order of the summation, we deduce

$$L_{pqr} = \sum_{l=0}^p \sum_{m=0}^q \sum_{n=0}^r (-1)^{p+q+r-l-m-n} \sum_{i=1}^N \sum_{j=1}^N \sum_{k=1}^N F_{lmn}(a_i, b_j, c_k) f(x_i, y_j, z_k). \tag{73}$$

Equation (73) can be written as

$$L_{pqr} = \sum_{l=0}^p \sum_{m=0}^q \sum_{n=0}^r \sum_{u=0}^{[(l-1)/2]} \sum_{v=0}^{[(m-1)/2]} \sum_{w=0}^{[(n-1)/2]} H_{uvw}(p, q, r, l, m, n)G_{l-2u, m-2v, n-2w}, \tag{74}$$

where

$$H_{uvw}(p, q, r, l, m, n) = (-1)^{p+q+r-l-m-n} h_u(p, [l/2])h_v(q, [m/2])h_w(r, [n/2]) \tag{75}$$

and G_{abc} , denoting the geometric moment of order $(a + b + c)$ except for a constant multiplicative factor, is defined by

$$G_{abc} = \left(\frac{2}{N - 1} \right)^{a+b+c} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} i^a j^b k^c f(x_i, y_j, z_k). \tag{76}$$

From Eqs. (74), (75), and (76), we now describe our strategy to calculate 3D Legendre moments of order up to M .

1. The geometric moments G_{abc} , with $0 \leq a, b, c \leq M$, are calculated by use of the fast method proposed by Yang *et al.* [24]. Using such a method, this step needs only $O(N^2)$ operations to calculate G_{abc} of order up to M .

2. The coefficients $h_u(p, [l/2])$, with $0 \leq p \leq M$, $0 \leq l \leq p$, and $0 \leq u \leq [(l-1)/2]$, are calculated with Eqs. (26) and (42). The values of $H_{uvw}(p, q, r, l, m, n)$ can then be deduced from Eq. (75).

3. Using Eq. (74) to calculate 3D Legendre moments L_{pqr} of order up to M . When all G_{abc} and $h_u(p, [l/2])$ are calculated, the computation L_{pqr} , with $0 \leq p+q+r \leq M$, needs approximately $(M^2(M+1)/180)^3$ additions and $3(M^2(M+1)/180)^3$ multiplications, respectively.

Obviously, the number of computing operations of the method using the algorithm described above for calculating the 3D Legendre moments L_{pqr} defined by Eq. (67) is less than that of the direct method since this latter requires $O(M^3N^3)$ additions and multiplications, respectively.

5. DISCUSSION

Since Hu [3] introduced the moment invariants, moment functions of image intensity values have been widely used in the field of image analysis and pattern recognition. During the past two decades, many works have been done, discussing the application and the efficient computation of the moment functions; most of them focus on Cartesian geometric moments. As is well known, orthogonal moments defined in terms of Legendre and Zernike polynomials can be used to represent an image with minimum redundancy. Furthermore, these orthogonal moments and their inverse transforms are very useful tools for image analysis, image reconstruction, and feature representation. However, the direct computation of orthogonal moments requires a large amount of additions and multiplications, especially where higher order moments are concerned; for this reason, the application of these moments is limited.

Mukundan and Ramakrishnan [25] proposed to use Green's theorem to transform the double integral into a boundary integral; then they just used the contour information to evaluate the 2D Legendre moments. Their method is efficient, but not accurate, since they adopted a trapezoidal integration rule to approximate the integral function for Legendre moments. Recently, Liao and Pawlak [8] proposed a more accurate approximation formula for computing the 2D Legendre moments of a digital image; an alternative extended Simpson's rule was then used to numerically calculate a double integral function for higher order Legendre moments for each pixel. The method proposed by Liao and Pawlak can be used to reduce the discretization error, and it has been used successfully for image reconstruction; however, their method requires a large number of computing operations.

In this paper, a simple and efficient algorithm for computing 2D and 3D Legendre moments has been proposed. The main contribution of the paper is to establish the relationship indicated by Eq. (10). Using this equation, we can express the 2D Legendre moments \tilde{L}_{pq} defined by (6) as a linear combination of L_{mn} given by (4). Therefore, to obtain the value of \tilde{L}_{pq} , it suffices to calculate the 2D Legendre L_{mn} with $0 \leq m \leq p$ and $0 \leq n \leq q$; the value of \tilde{L}_{pq} can then be deduced from Eq. (53). Note that to reduce the computation time, the relationship (53) can be found by the 2D fast Fourier transform for large values of p and q . With the help of Eq. (10), we still can transform the 2D Legendre moments L_{pq} into a function of the geometric moments of the same order and lower, whose coefficients appearing in the expression are easy to calculate. Since many efficient algorithms are available to compute the geometric moments, our method tremendously decreases the number

of computing operations compared with that of Liao and Pawlak, and the accuracy of our method is similar to theirs. The above discussion shows that Eq. (10) is very useful; the other application of this formula will be given in the near future.

As is well known, the usual way to express the Legendre moments as functions of the geometric moments consists of substituting Eqs. (2) and (3) into (4). However, during the programming, we found that it was difficult to obtain accurate coefficient values for higher order Legendre polynomials. For example, for $p = 30$, the numeric result is not the same as the theoretic value, and the difference becomes important with increasing order. In contrast, the accurate coefficient values of $\lambda_k^p(a)$ defined by Eqs. (20) and (40) can be obtained without difficulty for large value of p ; we have tested it up to 50. So our algorithm deserves to be implemented much more.

6. CONCLUSION

In this paper, we have discussed the problems of accuracy and efficiency in 2D and 3D Legendre moment computing. Based on the improved moment computing techniques, some Chinese characters have been used as the test images in the image reconstruction procedure. The comparison of the algorithm with known algorithms shows that our method is efficient and accurate; therefore, it can be useful in image reconstruction and feature representation.

APPENDIX A

Proof of Lemma 1. The last term on the right-hand side of Eq. (31) can be written as

$$\begin{aligned} & \sum_{j=0}^{i-1} f_j(p, m-i+j) \sum_{l=2m-2i+2j-1}^{2m-1} D_{j,l} \\ &= \sum_{j=0}^{i-1} f_j(p, m-i+j) \sum_{l=0}^{2i-2j} D_{j,l+2m-2i+2j-1} \\ &= \sum_{j=0}^{i-1} f_j(p, m-i+j) \left[\sum_{l=0}^{i-j-1} (D_{j,2l+2m-2i+2j-1} + D_{j,2l+2m-2i+2j}) + D_{j,2m-1} \right]. \quad (A1) \end{aligned}$$

Furthermore,

$$\begin{aligned} & \sum_{j=0}^{i-1} f_j(p, m-i+j) \sum_{l=0}^{i-j-1} D_{j,2l+2m-2i+2j-1} \\ &= \sum_{j=0}^{i-1} f_j(p, m-i+j) \sum_{l=0}^{i-j-1} D_{j,2m-2l-3} \quad \text{using the change of variable} \\ & \quad \quad \quad l := i-j-1-l \\ &= \sum_{l=0}^{i-1} \sum_{j=0}^{i-l-1} f_j(p, m-i+j) D_{j,2m-2l-3} \\ &= \sum_{l=0}^{i-1} \sum_{j=0}^l f_j(p, m-i+j) D_{j,2m-2i+2l-1} \quad \text{using the change of variable} \\ & \quad \quad \quad l := i-1-l \\ &= \sum_{l=0}^{i-1} B_{2l} \quad \text{by use of the assumption (32)}. \quad (A2) \end{aligned}$$

Similarly, by using the assumption (33), we can deduce

$$\sum_{j=0}^{i-1} f_j(p, m - i + j) \sum_{l=0}^{i-j-1} D_{j,2l+2m-2i+2j} = \sum_{l=0}^{i-1} B_{2l+1}. \tag{A3}$$

From Eqs. (A2) and (A3), the right-hand side of Eq. (31) can be written as

$$\begin{aligned} & A \left[\sum_{l=0}^{2i} B_l - \sum_{j=0}^{i-1} f_j(p, m - i + j) \sum_{l=2m-2i+2j-1}^{2m-1} D_{j,l} \right] \\ &= A \left[B_{2i} - \sum_{j=0}^{i-1} f_j(p, m - i + j) D_{j,2m-1} \right] \\ &= f_i(p, m) \end{aligned} \tag{by use of the assumption (34).}$$

The proof is now complete. ■

Proof of Lemma 2. Because $C_p^j = 0$ for $j < 0$ and $C_{2p-2j}^{2p-2t} = 0$ for $j > t$, we deduce from (35) that $F(t, j) = 0$ for $j < 0$, and $F(t, j) = 0$ for $j > t$. Therefore

$$\sum_{j=0}^t F(t, j) = \sum_{j=-\infty}^{\infty} F(t, j). \tag{A4}$$

Introducing the functions

$$\begin{aligned} R(t, j) &= 4p - 1 - pj - j + 2jH + 2j^2 - 8t^3 - 4p^2t - 4pj t - 3p^2 \\ &\quad + 4j^2t - 2jt - 2H - 6t - 2pj^2 + 2p^2j + 12pt^2 + 4ptH \\ &\quad + 14pt - 6Ht^2 - 7tH + 3pH - 12t^2 - 2pjH + 4jtH \end{aligned}$$

and

$$G(t, j) = -\frac{j(2p - 2j + 1)R(t, j)}{(p - 2t - 1)(p - 2t)(2p - 2H - 4j - 1)(t - j + 1)(2t - 2j + 1)} F(t, j).$$

After a fair amount of mathematical manipulations, we obtain

$$F(t, j) - F(t + 1, j) = G(t, j + 1) - G(t, j).$$

Using the relations $G(t, t + 2) = G(t, -1) = 0$ (because $F(t, t + 2) = F(t, -1) = 0$), we have

$$\begin{aligned} \sum_{j=-1}^{t+1} F(t, j) - \sum_{j=-1}^{t+1} F(t + 1, j) &= \sum_{j=-1}^{t+1} G(t, j + 1) - \sum_{j=-1}^{t+1} G(t, j) \\ &= G(t, t + 2) - G(t, -1) = 0. \end{aligned}$$

We deduce from (A4)

$$\sum_{j=-\infty}^{\infty} F(t + 1, j) = \sum_{j=-\infty}^{\infty} F(t, j). \tag{A5}$$

Equation (A5) shows that $\sum_{j=-\infty}^{\infty} F(t, j)$ is independent on t . Then, by setting $t=0$, we obtain

$$\sum_{j=-\infty}^{\infty} F(t, j) = \sum_{j=-\infty}^{\infty} F(0, j) = F(0, 0) = 1,$$

which completes the proof of Lemma 2. ■

Proof of Lemma 3. The proof of Lemma 3 is similar to that of Lemma 2 by introducing the following functions

$$\begin{aligned} R(t, j) = & 14p - 8 - 3pj - 2j + 4jH + 4j^2 - 8t^3 - 4p^2t - 4pjt - 5p^2 \\ & + 4j^2t - 2jt - 7H - 24t - 2pj^2 + 2p^2j + 12pt^2 + 4ptH \\ & + 26pt - 6Ht^2 - 13tH + 5pH - 24t^2 - 2pjH + 4jtH \end{aligned}$$

and

$$G(t, j) = \frac{j(2p - 2j + 1)R(t, j)}{(p - 2t - 2)(p - 2t - 1)(2p - 2H - 4j - 1)(t - j + 1)(2t - 2j + 3)} F(t, j).$$

■

Proof of Theorem 3. Based on Lemma 1, it is sufficient to prove that the function $f_i(p, m)$ defined by Eq. (26) or Eq. (27) satisfies (32), (33), and (34). For this purpose, we first prove that $f_i(p, m)$ defined by Eq. (27) satisfies (32). Recall that Eq. (32) is written as

$$B_{2t} = \sum_{j=0}^t f_j(p, m - i + j) D_{j, 2m-2i+2t-1}, \quad t = 0, 1, \dots, i - 1, i. \tag{A6}$$

From (29), (30), and (27), we have

$$B_{2t} = \frac{(-1)^{2t}}{2^{2m-2t-1}} \frac{(2p - 2t)!}{(2t)!(2i - 2t)!(2m - 2i - 1)!(p - 2t)!} \tag{A7}$$

$$D_{j, 2m-2i+2t-1} = \frac{(-1)^{2m-2i+2t-2}}{2^{2i-2t}} \frac{(2p - 4m + 4i - 2t - 2j + 2)!}{(2t - 2j)!(2i - 2t)!(p - 2m + 2i - 2t + 1)!} \tag{A8}$$

and

$$\begin{aligned} f_j(p, m - i + j) = & \frac{1}{2^{2m-2i-2}} \frac{(2p - 4m + 4i - 4j + 3)(2m - 2i + j - 2)!}{j!(2m - 2i - 1)!(2m - 2i - 2)!} \\ & \times \frac{(2p - 2j)!}{(p - j)!} \frac{(p - 2m + 2i - j + 2)!}{(2p - 4m + 4i - 2j + 4)!}. \end{aligned} \tag{A9}$$

Using (A7), (A8), and (A9), we deduce that (A6) is equivalent to the equation

$$\frac{(2p - 2t)!}{(2t)!(p - 2t)!} = \sum_{j=0}^t E(t, j), \tag{A10}$$

where

$$E(t, j) = 2(2p - 4m + 4i - 4j + 3) \frac{(2p - 4m + 4i - 2t - 2j + 2)!}{j!(2t - 2j)!(p - 2m + 2i - 2t + 1)!} \\ \times \frac{(2m - 2i + j - 2)! (2p - 2j)! (p - 2m + 2i - j + 2)!}{(2m - 2i - 2)! (p - j)! (2p - 4m + 4i - 2j + 4)!}. \quad (\text{A11})$$

By using the change of variable $H = 2m - 2i - 2$, we obtain

$$E(t, j) = 2(2p - 2H - 4j - 1) \frac{(2p - 2H - 2t - 2j - 2)!}{j!(2t - 2j)!(p - H - 2t - 1)!} \\ \times \frac{(H + j)! (2p - 2j)! (p - H - j)!}{H! (p - j)! (2p - 2H - 2j)!}. \quad (\text{A12})$$

Note that H is a positive integer for $i = 0, 1, 2, \dots, m - 1$. Setting

$$F(t, j) = \frac{(2t)!(p - 2t)!}{(2p - 2t)!} E(t, j) \quad (\text{A13})$$

and substituting (A12) into (A13) yields

$$F(t, j) = \frac{2(2p - 2H - 4j - 1)(p - 2t)!}{H!(p - H - 2t - 1)!(2t + 1)(2t + 2)} \frac{C_p^j C_{2p-2j}^{2p-2t}}{C_p^{H+j} C_{2p-2H-2j}^{2t+2}}.$$

By using Lemma 2, we have

$$\sum_{j=0}^t F(t, j) = 1.$$

So Eq. (A10) is demonstrated, and the proof of (A6), or that of Eq. (32) is achieved.

Equation (33) can be proven in a similar way by using Lemma 3. To prove (34), it suffices to use Eq. (32) for $t = i$, we have

$$A \left[B_{2i} - \sum_{j=0}^{i-1} f_j(p, m - i + j) D_{j, 2m-1} \right] \\ = A \left[\sum_{j=0}^i f_j(p, m - i + j) D_{j, 2m-1} - \sum_{j=0}^{i-1} f_j(p, m - i + j) D_{j, 2m-1} \right] \\ = A \cdot f_i(p, m) \cdot D_{i, 2m-1}.$$

From (30), we have

$$D_{i, 2m-1} = \frac{(2p - 4m + 2)!}{(p - 2m + 1)!} = \frac{1}{A}.$$

The proof of Eq. (34) is now complete. ■

APPENDIX B

$$\lambda_0^p(a) = 1$$

$$\lambda_1^p(a) = (2p - 1)a$$

$$\lambda_2^p(a) = \frac{1}{2!}(2p - 1)(2p - 3)a^2$$

$$\lambda_3^p(a) = \frac{1}{3!}(2p - 1)(2p - 3)(2p - 5)a^3 + (2p - 5)a$$

$$\lambda_4^p(a) = \frac{1}{4!}(2p - 1)(2p - 3)(2p - 5)(2p - 7)a^4 + (2p - 3)(2p - 7)a^2$$

$$\lambda_5^p(a) = \frac{1}{5!}(2p - 1)(2p - 3)(2p - 5)(2p - 7)(2p - 9)a^5 \\ + \frac{1}{2}(2p - 3)(2p - 5)(2p - 9)a^3 + (2p - 9)a$$

$$\lambda_6^p(a) = \frac{1}{6!}(2p - 1)(2p - 3)(2p - 5)(2p - 7)(2p - 9)(2p - 11)a^6 \\ + \frac{1}{3!}(2p - 3)(2p - 5)(2p - 7)(2p - 11)a^4 + \frac{3}{2}(2p - 5)(2p - 11)a^2$$

$$\lambda_7^p(a) = \frac{1}{7!}(2p - 1)(2p - 3)(2p - 5)(2p - 7)(2p - 9)(2p - 11)(2p - 13)a^7 \\ + \frac{1}{4!}(2p - 3)(2p - 5)(2p - 7)(2p - 9)(2p - 13)a^5 \\ + (2p - 5)(2p - 7)(2p - 13)a^3 + (2p - 13)a$$

$$\lambda_8^p(a) = \frac{1}{8!}(2p - 1)(2p - 3)(2p - 5)(2p - 7)(2p - 9)(2p - 11)(2p - 13)(2p - 15)a^8 \\ + \frac{1}{5!}(2p - 3)(2p - 5)(2p - 7)(2p - 9)(2p - 11)(2p - 15)a^6 \\ + \frac{5}{12}(2p - 5)(2p - 7)(2p - 9)(2p - 15)a^4 + 2(2p - 7)(2p - 15)a^2$$

$$\lambda_9^p(a) = \frac{1}{9!} \prod_{i=1}^9 (2p - 2i + 1)a^9 + \frac{1}{6!} \prod_{i=2}^7 (2p - 2i + 1)(2p - 17)a^7 \\ + \frac{1}{8} \prod_{i=3}^6 (2p - 2i + 1)(2p - 17)a^5 + \frac{5}{3}(2p - 7)(2p - 9)(2p - 17)a^3 \\ + (2p - 17)a$$

$$\lambda_{10}^p(a) = \frac{1}{10!} \prod_{i=1}^{10} (2p - 2i + 1)a^{10} + \frac{1}{7!} \prod_{i=2}^8 (2p - 2i + 1)(2p - 19)a^8 \\ + \frac{7}{2 \cdot 5!} \prod_{i=3}^7 (2p - 2i + 1)(2p - 19)a^6 + \frac{5}{6} \prod_{i=4}^6 (2p - 2i + 1)(2p - 19)a^4 \\ + \frac{5}{2}(2p - 9)(2p - 19)a$$

$$\begin{aligned}\lambda_{11}^p(a) &= \frac{1}{11!} \prod_{i=1}^{11} (2p-2i+1)a^{11} + \frac{1}{8!} \prod_{i=2}^9 (2p-2i+1)(2p-21)a^9 \\ &+ \frac{1}{180} \prod_{i=3}^8 (2p-2i+1)(2p-21)a^7 + \frac{7}{24} \prod_{i=4}^7 (2p-2i+1)(2p-21)a^5 \\ &+ \frac{5}{2}(2p-9)(2p-11)(2p-21)a^3 + (2p-21)a\end{aligned}$$

$$\begin{aligned}\lambda_{12}^p(a) &= \frac{1}{12!} \prod_{i=1}^{12} (2p-2i+1)a^{12} + \frac{1}{9!} \prod_{i=2}^{10} (2p-2i+1)(2p-23)a^{10} \\ &+ \frac{9}{2 \cdot 7!} \prod_{i=3}^9 (2p-2i+1)(2p-23)a^8 + \frac{7}{90} \prod_{i=4}^8 (2p-2i+1)(2p-23)a^6 \\ &+ \frac{35}{24} \prod_{i=5}^7 (2p-2i+1)(2p-23)a^4 + 3(2p-11)(2p-23)a^2\end{aligned}$$

$$\begin{aligned}\lambda_{13}^p(a) &= \frac{1}{13!} \prod_{i=1}^{13} (2p-2i+1)a^{13} + \frac{1}{10!} \prod_{i=2}^{11} (2p-2i+1)(2p-25)a^{11} \\ &+ \frac{5}{8!} \prod_{i=3}^{10} (2p-2i+1)(2p-25)a^9 + \frac{1}{60} \prod_{i=4}^9 (2p-2i+1)(2p-25)a^7 \\ &+ \frac{7}{12} \prod_{i=5}^8 (2p-2i+1)(2p-25)a^5 + \frac{7}{2}(2p-11)(2p-13)(2p-25)a^3 \\ &+ (2p-25)a\end{aligned}$$

$$\begin{aligned}\lambda_{14}^p(a) &= \frac{1}{14!} \prod_{i=1}^{14} (2p-2i+1)a^{14} + \frac{1}{11!} \prod_{i=2}^{12} (2p-2i+1)(2p-27)a^{12} \\ &+ \frac{11}{2 \cdot 9!} \prod_{i=3}^{11} (2p-2i+1)(2p-27)a^{10} + \frac{15}{7!} \prod_{i=4}^{10} (2p-2i+1)(2p-27)a^8 \\ &+ \frac{7}{40} \prod_{i=5}^9 (2p-2i+1)(2p-27)a^6 + \frac{7}{3} \prod_{i=6}^8 (2p-2i+1)(2p-27)a^4 \\ &+ \frac{7}{2}(2p-13)(2p-27)a^2\end{aligned}$$

$$\begin{aligned}\lambda_{15}^p(a) &= \frac{1}{15!} \prod_{i=1}^{15} (2p-2i+1)a^{15} + \frac{1}{12!} \prod_{i=2}^{13} (2p-2i+1)(2p-29)a^{13} \\ &+ \frac{6}{10!} \prod_{i=3}^{12} (2p-2i+1)(2p-29)a^{11} + \frac{55}{3 \cdot 8!} \prod_{i=4}^{11} (2p-2i+1)(2p-29)a^9 \\ &+ \frac{1}{24} \prod_{i=5}^{10} (2p-2i+1)(2p-29)a^7 + \frac{21}{20} \prod_{i=6}^9 (2p-2i+1)(2p-29)a^5 \\ &+ \frac{14}{3}(2p-13)(2p-15)(2p-29)a^3 + (2p-29)a\end{aligned}$$

$$\begin{aligned}\lambda_{16}^p(a) &= \frac{1}{16!} \prod_{i=1}^{16} (2p-2i+1)a^{16} + \frac{1}{13!} \prod_{i=2}^{14} (2p-2i+1)(2p-31)a^{14} \\ &+ \frac{13}{2 \cdot 11!} \prod_{i=3}^{13} (2p-2i+1)(2p-31)a^{12} + \frac{22}{9!} \prod_{i=4}^{12} (2p-2i+1)(2p-31)a^{10} \\ &+ \frac{165}{4 \cdot 7!} \prod_{i=5}^{11} (2p-2i+1)(2p-31)a^8 + \frac{7}{20} \prod_{i=6}^{10} (2p-2i+1)(2p-31)a^6 \\ &+ \frac{7}{2} \prod_{i=7}^9 (2p-2i+1)(2p-31)a^4 + 4(2p-15)(2p-31)a^2\end{aligned}$$

$$\begin{aligned}\lambda_{17}^p(a) &= \frac{1}{17!} \prod_{i=1}^{17} (2p-2i+1)a^{17} + \frac{1}{14!} \prod_{i=2}^{15} (2p-2i+1)(2p-33)a^{15} \\ &+ \frac{7}{12!} \prod_{i=3}^{14} (2p-2i+1)(2p-33)a^{13} + \frac{26}{10!} \prod_{i=4}^{13} (2p-2i+1)(2p-33)a^{11} \\ &+ \frac{55}{8!} \prod_{i=5}^{12} (2p-2i+1)(2p-33)a^9 + \frac{11}{5!} \prod_{i=6}^{11} (2p-2i+1)(2p-33)a^7 \\ &+ \frac{7}{4} \prod_{i=7}^{10} (2p-2i+1)(2p-33)a^5 + 6(2p-15)(2p-17)(2p-33)a^3 \\ &+ (2p-33)a\end{aligned}$$

$$\begin{aligned}\lambda_{18}^p(a) &= \frac{1}{18!} \prod_{i=1}^{18} (2p-2i+1)a^{18} + \frac{1}{15!} \prod_{i=2}^{16} (2p-2i+1)(2p-35)a^{16} \\ &+ \frac{15}{2 \cdot 13!} \prod_{i=3}^{15} (2p-2i+1)(2p-35)a^{14} + \frac{91}{3 \cdot 11!} \prod_{i=4}^{14} (2p-2i+1)(2p-35)a^{12} \\ &+ \frac{143}{2 \cdot 9!} \prod_{i=5}^{13} (2p-2i+1)(2p-35)a^{10} + \frac{99}{7!} \prod_{i=6}^{12} (2p-2i+1)(2p-35)a^8 \\ &+ \frac{77}{5!} \prod_{i=7}^{11} (2p-2i+1)(2p-35)a^6 + 5 \prod_{i=8}^{10} (2p-2i+1)(2p-35)a^4 \\ &+ \frac{9}{2}(2p-17)(2p-35)a^2\end{aligned}$$

$$\begin{aligned}\lambda_{19}^p(a) &= \frac{1}{19!} \prod_{i=1}^{19} (2p-2i+1)a^{19} + \frac{1}{16!} \prod_{i=2}^{17} (2p-2i+1)(2p-37)a^{17} \\ &+ \frac{8}{14!} \prod_{i=3}^{16} (2p-2i+1)(2p-37)a^{15} + \frac{35}{12!} \prod_{i=4}^{15} (2p-2i+1)(2p-37)a^{13} \\ &+ \frac{91}{10!} \prod_{i=5}^{14} (2p-2i+1)(2p-37)a^{11} + \frac{143}{8!} \prod_{i=6}^{13} (2p-2i+1)(2p-37)a^9\end{aligned}$$

$$\begin{aligned}
& + \frac{11}{60} \prod_{i=7}^{12} (2p-2i+1)(2p-37)a^7 + \frac{11}{4} \prod_{i=8}^{11} (2p-2i+1)(2p-37)a^5 \\
& + \frac{15}{2} (2p-17)(2p-19)(2p-37)a^3 + (2p-37)a \\
\lambda_{20}^p(a) = & \frac{1}{20!} \prod_{i=1}^{20} (2p-2i+1)a^{20} + \frac{1}{17!} \prod_{i=2}^{18} (2p-2i+1)(2p-39)a^{18} \\
& + \frac{17}{2 \cdot 15!} \prod_{i=3}^{17} (2p-2i+1)(2p-39)a^{16} + \frac{40}{13!} \prod_{i=4}^{16} (2p-2i+1)(2p-39)a^{14} \\
& + \frac{455}{4 \cdot 11!} \prod_{i=5}^{15} (2p-2i+1)(2p-39)a^{12} + \frac{2002}{10!} \prod_{i=6}^{14} (2p-2i+1)(2p-39)a^{10} \\
& + \frac{429}{2 \cdot 7!} \prod_{i=7}^{13} (2p-2i+1)(2p-39)a^8 + \frac{11}{10} \prod_{i=8}^{12} (2p-2i+1)(2p-39)a^6 \\
& + \frac{55}{8} \prod_{i=9}^{11} (2p-2i+1)(2p-39)a^4 + 5(2p-19)(2p-39)a^2.
\end{aligned}$$

ACKNOWLEDGMENTS

We thank the anonymous referees for their helpful comments and suggestions. Thanks also to J. D. Zhou for her contribution to the programming of the experimental results.

REFERENCES

1. R. J. Prokop and A. P. Reeves, A survey of moment based techniques for unoccluded object representation, *CVGIP: Graphical Models Image Process.* **54**, 1992, 438–460.
2. C. H. Lo and H. S. Don, 3D moment forms: Their construction and application to object identification and positioning, *IEEE Trans. Pattern Anal. Mach. Intell.* **11**, 1989, 1053–1064.
3. M. K. Hu, Visual pattern recognition by moments invariants, *IRE Trans. Inform. Theory* **8**, 1962, 179–187.
4. S. A. Dudani, K. J. Breeding, and R. B. McGhee, Aircraft identification by moment invariants, *IEEE Trans. Comput.* **26**, 1977, 39–47.
5. J. Flusser and T. Suk, Pattern Recognition by affine moment invariants, *Pattern Recognition* **26**, 1993, 167–174.
6. M. R. Teague, Image analysis via the general theory of moments, *J. Opt. Soc. Am.* **70**, 1980, 920–930.
7. C. H. Teh and R. T. Chin, On image analysis by the method of moments, *IEEE Trans. Pattern Anal. Mach. Intell.* **10**, 1988, 485–513.
8. S. X. Liao and M. Pawlak, On image analysis by moments, *IEEE Trans. Pattern Anal. Mach. Intell.* **18**, 1996, 254–266.
9. R. C. Papademetriou, Reconstructing with moments, in *Proceedings, 11th International Conference, Pattern Recognition, Volume C, 1992*, pp. 476–480.
10. A. P. Reeves, M. L. Akey and O. R. Mitchell, A moment based two-dimensional edge operator, in *Proceedings, IEEE Conference, Computer Vision and Pattern Recognition, 1983*, pp. 312–317.
11. S. Ghosal and R. Mehrotra, Orthogonal moment operators for subpixel edge detection, *Pattern Recognition* **26**, 1993, 295–306.

12. L. M. Luo, C. Hamitouche, J. L. Dilenseger, and J. L. Coatrieux, A moment-based three-dimensional edge operator, *IEEE Trans. Biomed. Eng.* **40**, 1993, 693–703.
13. L. M. Luo, X. H. Xie, and X. D. Bao, A modified moment-based edge operator for rectangular pixel image, *IEEE Trans. Circuits Systems Video Technol.* **4**, 1994, 552–554.
14. M. Tuceryan, Moment-based texture segmentation, *Pattern Recognition Lett.* **15**, 1994, 115–123.
15. A. Khotanzad and Y. H. Hong, Rotation invariant image recognition using features selected via a systematic method, *Pattern Recognition* **23**, 1990, 1089–1101.
16. A. Khotanzad, Invariant image recognition by Zernike moments, *IEEE Trans. Pattern Anal. Mach. Intell.* **12**, 1990, 489–497.
17. M. F. Zakaria, L. J. Vroomen, P. J. A. Zsombor-Murray, and J. M. H. M. Van Kessel, Fast algorithm for the computation of moment invariants, *Pattern Recognition* **20**, 1987, 639–643.
18. X. Y. Jiang and H. Bunke, Simple and fast computation of moments, *Pattern Recognition* **24**, 1991, 801–806.
19. B. C. Li and J. Shen, Fast computation of moment invariant, *Pattern Recognition* **24**, 1991, 807–813.
20. B. C. Li, The moment calculation of polyhedra, *Pattern Recognition* **26**, 1993, 1229–1233.
21. B. C. Li and J. Shen, Pascal triangle transform approach to the calculation of 3D moments, *CVGIP: Graphical Models Image Process.* **54**, 1992, 301–307.
22. W. Philips, A new fast algorithm for moment computation, *Pattern Recognition* **26**, 1993, 1619–1621.
23. L. Yang and F. Albrechtsen, Fast and exact computation of Cartesian geometric moments using discrete Green's theorem, *Pattern Recognition* **29**, 1996, 1061–1073.
24. L. Yang, F. Albrechtsen, and T. Taxt, Fast computation of three-dimensional geometric moments using a discrete divergence theorem and a generalization to higher dimensions, *CVGIP: Graphical Models Image Process.* **59**, 1997, 97–108.
25. R. Mukundan and K. R. Ramakrishnan, Fast computation of Legendre and Zernike moments, *Pattern Recognition* **28**, 1995, 1433–1442.
26. G. Sansone, *Orthogonal Functions*, Dover, New York, 1991.