New Algorithm for Constructing and Computing Scale Invariants of 3D Tchebichef Moments

Haiyong Wu, Jean-Louis Coatrieux, Huazhong Shu

Research Article

New Algorithm for Constructing and Computing Scale Invariants of 3D Tchebichef Moments

Haiyong Wu,[1] Jean Louis Coatrieux,[1,2,3,4] and Huazhong Shu[1,4]

[1] *Laboratory of Image Science and Technology, School of Computer Science and Engineering, Southeast University, Nanjing 210096, China*

[2] *INSERM, U1099, 35000 Rennes, France*

[3] *Laboratoire Traitement du Signal et de l'Image, Université de Rennes I, 35000 Rennes, France*

[4] *Centre de Recherche en Information Biomédicale Sino-Français (CRIBs), Nanjing 210096, China*

Correspondence should be addressed to Haiyong Wu; why@njxzc.edu.cn

Scale invariants of Tchebichef moments are usually achieved by a linear combination of corresponding invariants of geometric moments or via an iterative algorithm to eliminate the scale factor. According to the properties of Tchebichef polynomials, we propose a new approach to construct scale invariants of Tchebichef moments. An algorithm based on matrix multiplication is also provided to efficiently compute the 3D moments and invariants. Several experiments are carried out to validate the effectiveness of our descriptors and algorithm.

1. Introduction

Since the introduction of geometric moment invariants by Hu [1], moments and moment's functions have been extensively applied in pattern recognition [2, 3]. Due to the nonorthogonal kernel function of geometric moments, they suffer from high degree of information redundancy and are sensitive to noise, especially when higher order moments are concerned. Teague [4] introduced orthogonal continuous Legendre and Zernike moments, which can represent the image with minimal information redundancy and can be easily used for image reconstruction. The main drawback of the aforementioned moments is the discretization error, which accumulates with the increasing of moment order [5]. To resolve this problem, discrete orthogonal polynomials have been utilized to construct moments, such as Tchebichef [6], Krawtchouk [7], and Hahn [8]. Their basis functions exactly satisfy the orthogonality, which means they do not require any numerical approximation and spatial domain transformation. It makes them superior to the conventional continuous moments in terms of image representation capability.

Recently, the problem of moment invariance has been extensively investigated. For example, the invariants of Legendre moments have been achieved through image normalization method and indirect method. Chong et al. [9] proposed a direct method to construct the translation and scale Legendre moment invariants. Ong et al. [10] generalized this method to 3D directly. Following this way, Zhu et al. [11] derived translation and scale invariants of Tchebichef moments. However, Khalid pointed out that one weakness of this method is the high computational cost, especially when image size is large and higher order moments are concerned. Therefore, this method is suitable only for a set of binary images with small size [12]. Another difficulty of aforementioned scale invariants is how to determinate their parameters, which need to be selected carefully to keep a compromise between numerical stability and complexity.

Inspired by the method proposed by Zhang et al. [13], we propose in this paper an improved approach to construct 3D scale invariants of Tchebichef moments. Instead of normalizing by lower order moments, scale factors are eliminated by utilizing the orthogonality of coefficients. This method avoids enormous computing caused by iteration. Moreover,

we propose an algorithm based on matrix multiplication to efficiently compute the 3D moments and invariants.

The remaining of this paper is organized as follows: in Section 2, Tchebichef polynomials and derivation of corresponding scale invariants are described in detail. Section 3 discusses an efficient way for computing 3D moments and invariants. Experimental results for evaluating the performance of the proposed descriptors are given in Section 4. Finally, conclusions are provided in Section 5.

2. Improved Scale Invariants of 3D Tchebichef Moments

In this section, the falling factorial is introduced to build a mutual relationship between Tchebichef polynomials and power series. In order to separate the scale factor, Tchebichef polynomials need to be transformed into power series firstly. Then, these separated power series are expressed through Tchebichef polynomials. By this way, moments of scaled image can be expressed as a linear combination of the original moments. We use the Stirling numbers instead of tedious iterations to obtain scale invariants, because the recursive procedure is an inherent deficiency of descriptors reported in [9–11].

2.1. Some Properties of Tchebichef Polynomials. The squared-norm of scaled Tchebichef polynomials is defined as [6]

$$t_n(x) = \frac{(1-N)_n}{\sqrt{\rho(n,N)}} \sum_{k=0}^{n} \frac{(-n)_k(-x)_k(1+n)_k}{(k!)^2(1-N)_k},$$

$$n, x = 0, 1, 2, \ldots, N-1, \tag{1}$$

where $(a)_k$ is the Pochhammer symbol given by

$$(a)_k = a(a+1)(a+2)\cdots(a+k-1),$$

$$k \geq 1, \ (a)_0 = 1, \tag{2}$$

and $\rho(n, N)$ is the squared-norm defined by

$$\rho(n, N) = \frac{(N+n)!}{(2n+1)(N-n-1)!}. \tag{3}$$

The falling factorial $\langle a \rangle_k$ is defined as [14]

$$\langle a \rangle_k = (-1)^k(-a)_k = a(a-1)(a-2)\cdots(a-k+1),$$

$$k \geq 1, \langle a \rangle_0 = 1. \tag{4}$$

Using (4), (1) can be rewritten as

$$t_n(x) = \sum_{k=0}^{n} c_{nk} \langle x \rangle_k, \tag{5}$$

where

$$c_{nk} = \frac{1}{\sqrt{\rho(n,N)}} \frac{(n+k)!(1-N)_n}{(k!)^2(n-k)!(1-N)_k}. \tag{6}$$

Let $U_M(x) = (t_0(x), t_1(x), \ldots, t_M(x))^T$ and $V_M(x) = (1, \langle x \rangle_1, \ldots, \langle x \rangle_M)^T$ be two column vectors, where the superscript **T** indicates the transposition. Using (5), we have

$$U_M(x) = C_M V_M(x), \tag{7}$$

where $C_M = (c_{nk}), 0 \leq k \leq n \leq M$, is a lower triangular matrix whose size is $(M+1) \times (M+1)$. The matrix C_M is invertible because its diagonal elements are not zero; therefore,

$$V_M(x) = C_M^{-1} U_M(x) = D_M U_M(x), \tag{8}$$

where $D_M = (d_{nk}), 0 \leq k \leq n \leq M$. D_M is a lower triangular matrix too, and its elements are given by [15]

$$d_{nk} = (-1)^{n+k} \frac{\sqrt{\rho(k,N)}(2k+1)(n!)^2(1-N)_n}{(n+k+1)!(n-k)!(1-N)_k}. \tag{9}$$

The falling factorial $\langle a \rangle_n$ and the power series x^n can be expanded mutually:

$$\langle x \rangle_n = \sum_{i=0}^{n} s_1(n,i) x^i, \qquad x^n = \sum_{i=0}^{n} s_2(n,i) \langle x \rangle_i, \tag{10}$$

where $s_1(n, i)$ is the Stirling number of the first kind, satisfying the following recurrence relations:

$$s_1(0,0) = 1, \quad s_1(0,i) = s_1(n,0) = 0, \quad n \geq 1, \ i \geq 1,$$

$$s_1(n,i) = s_1(n-1,i-1) - (n-1)s_1(n-1,i), \tag{11}$$

and $s_2(n, i)$ is the Stirling number of the second kind, satisfying

$$s_2(0,0) = 1, \quad s_2(0,i) = s_2(n,0) = 0, \quad n \geq 1, \ i \geq 1,$$

$$s_2(n,i) = s_2(n-1,i-1) + i s_2(n-1,i). \tag{12}$$

The relationship between Tchebichef polynomials and power series has been established via the falling factorial, which is the fundamental of our descriptors.

2.2. Scale Invariants of 1D Tchebichef Moments. The 1D Tchebichef moment of order p for an N-length signal $f(x)$ is defined as

$$T_p^f = \sum_{x=0}^{N-1} t_p(x) f(x). \tag{13}$$

Let $g(x)$ be a scaled version of $f(x)$ with a scale factor $1/\lambda$; that is, $g(x) = f(x/\lambda)$; then the Tchebichef moment of order p of $g(x)$ has the form of

$$T_p^g = \sum_{x=0}^{N-1} t_p(x) g(x) = \lambda \sum_{x=0}^{N-1} t_p(\lambda x) f(x). \tag{14}$$

From (5), (8), and (10), we can see that

$$t_n(\lambda x) = \sum_{k=0}^{n} \sum_{i=0}^{k} \sum_{j=0}^{i} \sum_{m=0}^{j} c_{nk} s_1(k,i) \lambda^i s_2(i,j) d_{jm} t_m(x). \tag{15}$$

Substituting (15) into (14), we have

$$T_p^g = \sum_{k=0}^{p} \sum_{i=0}^{k} \sum_{j=0}^{i} \sum_{m=0}^{j} c_{pk} s_1(k,i) \lambda^{i+1} s_2(i,j) d_{jm} T_m^f. \quad (16)$$

Equation (16) shows that the Tchebichef moment of the scaled signal can be expressed as a linear combination of the original ones. Based on this relationship, we derive the following theorem.

Theorem 1. *For a given integer p, let*

$$I_p^f = \sum_{k=0}^{p} \sum_{i=0}^{k} \sum_{j=0}^{i} \sum_{m=0}^{j} c_{pk} s_1(k,i) \Gamma_f^{-(i+1)} s_2(i,j) d_{jm} T_m^f, \quad (17)$$

with $\Gamma_f = T_0^f$. Then I_p^f is invariant to image scaling.

The proof is given in the appendix.

2.3. Scale Invariants of 3D Tchebichef Moments. The $(p + q + r)$th Tchebichef moment of a 3D image $f(x, y, z)$ is defined by

$$T_{pqr}^f = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \sum_{z=0}^{N-1} t_p(x) t_q(y) t_r(z) f(x,y,z). \quad (18)$$

Let us assume that the original image f is scaled with factors $1/\lambda$, $1/\mu$, and $1/\xi$, along x-direction, y-direction, and z-direction, respectively. That is, $g(x, y, z) = f(x/\lambda, y/\mu, z/\xi)$. The $(p + q + r)$th moments of scaled image g are given by

$$T_{pqr}^g = \lambda \mu \zeta \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \sum_{z=0}^{N-1} t_p(\lambda x) t_q(\mu y) t_r(\zeta z) f(x,y,z). \quad (19)$$

Similarly to 1D case, we can rewrite (19) as

$$T_{pqr}^g = \sum_{k=0}^{p} \sum_{i=0}^{k} \sum_{j=0}^{i} \sum_{m=0}^{j} c_{pk} s_1(k,i) \lambda^{i+1} s_2(i,j) d_{jm}$$
$$\times \sum_{n=0}^{q} \sum_{s=0}^{n} \sum_{w=0}^{b} \sum_{v=0}^{w} c_{qn} s_1(n,b) \mu^{b+1} s_2(b,w) d_{wv}$$
$$\times \sum_{e=0}^{r} \sum_{h=0}^{e} \sum_{a=0}^{h} \sum_{l=0}^{a} c_{re} s_1(e,h) \zeta^{h+1} s_2(h,a) d_{al} T_{mvl}^f. \quad (20)$$

To construct the 3D moment invariants, we need the following lemma.

Lemma 2. *Let*

$$\Gamma_f = \frac{T_{100}^f}{T_{000}^f} - c_{10} d_{00}, \qquad \Theta_f = \frac{T_{010}^f}{T_{000}^f} - c_{10} d_{00},$$
$$\Lambda_f = \frac{T_{001}^f}{T_{000}^f} - c_{10} d_{00}. \quad (21)$$

Then, one has $\Gamma_g = \lambda \Gamma_f$, $\Theta_g = \mu \Theta_f$, and $\Lambda_g = \zeta \Lambda_f$.

Proof. Using (6), (9), and (19), we have

$$\Gamma_g = \frac{T_{100}^g}{T_{000}^g} = \frac{c_{10} d_{00} T_{000}^f + \lambda c_{11} d_{10} T_{000}^f + \lambda T_{100}^f}{T_{000}^f}$$
$$= c_{10} d_{00} + \lambda \left(\frac{T_{100}^f}{T_{000}^f} + c_{11} d_{10} \right). \quad (22)$$

Taking $c_{10} d_{00} + c_{11} d_{10} = 0$ into account, we have $\Gamma_g = \lambda \Gamma_f$. The other two relationships can be demonstrated in a similar way. □

Using (20) and Lemma 2, we can construct the scale invariants of 3D Tchebichef moments, which are described in following theorem.

Theorem 3. *For given integers p, q, r, let*

$$I_{pqr}^f = \sum_{k=0}^{p} \sum_{i=0}^{k} \sum_{j=0}^{i} \sum_{m=0}^{j} c_{pk} s_1(k,i) \Gamma_f^{-(i+1)} s_2(i,j) d_{jm}$$
$$\times \sum_{n=0}^{q} \sum_{b=0}^{n} \sum_{u=0}^{b} \sum_{v=0}^{u} c_{qn} s_1(n,b) \Theta_f^{-(b+1)} s_2(b,u) d_{uv}$$
$$\times \sum_{e=0}^{r} \sum_{h=0}^{e} \sum_{a=0}^{h} \sum_{l=0}^{a} c_{re} s_1(e,h) \Lambda_f^{-(h+1)} s_2(h,a) d_{al} T_{mvl}^f, \quad (23)$$

where Γ_f, Θ_f, and Λ_f are defined in Lemma 2. Then, I_{pqr}^f is invariant to image scale.

The proof is similar to that of Theorem 1 since the kernel function is separable, and so it is omitted here.

3. Computing 3D Moments and Invariants

To achieve the scale invariance of 3D moments, (23) implies that it needs to compute a 12-level nested loop. Since the structure of (23) is symmetric, we introduce a new algorithm based on matrix multiplication to reduce the number of loops.

3.1. Computing 3D Moments. Let us begin with discussion of the 1D case. Let $\mathbf{M} = (\mathbf{M}_0, \mathbf{M}_1, \dots, \mathbf{M}_p)^{\mathbf{T}}$ be a column vector constituted by the $(0 \sim p)$th moments of $f(x)$. From (5), it is well known that

$$\mathbf{M} = \begin{bmatrix} M_0 \\ M_1 \\ \dots \\ M_p \end{bmatrix}$$
$$= \begin{bmatrix} t_0(1) & t_0(2) & \cdots & t_0(N-1) \\ t_1(1) & t_1(2) & \cdots & t_1(N-1) \\ \dots & & & \\ t_p(1) & t_p(2) & \cdots & t_p(N-1) \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ \dots \\ f(N-1) \end{bmatrix}. \quad (24)$$

Yap et al. [7] proposed a matrix form of moments for a 2D image

$$M = K_1 A K_2^T, \qquad (25)$$

where M is an $m \times n$ matrix of Tchebichef moments, $K_1 = \{t_i(x)\}_{i=0, x=0}^{i=m-1, x=N-1}$, $K_2 = \{t_j(y)\}_{j=0, y=0}^{j=n-1, y=N-1}$, and $A = \{f(x, y)\}_{i, j=0}^{i, j=N-1}$.

Exchanging the order of summation, we rewrite (18) as

$$T_{pqr}^f = \sum_{z=0}^{N-1} t_r(z) \left\{ \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} t_p(x) t_q(y) f(x, y, z) \right\}. \quad (26)$$

By combining (24) with (25), an effective algorithm to calculate (26) can be derived as follows.

Step 1. Computing N 2D images along z-direction by (25), we can get temporary matrices (see Figure 8), where $M'_{ij}(k)$ denotes the element of ith row and jth column in the kth plane of z-direction.

Step 2. Along z-direction, we rearrange the temporary matrices obtained in Step 1 and get a $(p + 1)$ matrix with size of $N \times (q + 1)$. Then the required 3D moments are achieved after they are premultiplied by the kernel function matrices; that is,

$$
\begin{bmatrix} M_{000} & \cdots & M_{0q0} \\ M_{001} & \cdots & M_{0q1} \\ \cdots & & \\ M_{00r} & \cdots & M_{0qr} \end{bmatrix} = \begin{bmatrix} t_0(0) & t_0(1) & \cdots & t_0(N-1) \\ t_1(0) & t_1(1) & \cdots & t_1(N-1) \\ \cdots & & \\ t_r(0) & t_r(1) & \cdots & t_r(N-1) \end{bmatrix}
$$

$$
\times \begin{bmatrix} M'_{00}(0) & \cdots & M'_{0q}(0) \\ M'_{00}(1) & \cdots & M'_{0q}(1) \\ \cdots & & \\ M'_{00}(N-1) & \cdots & M'_{0q}(N-1) \end{bmatrix},
$$

$$
\begin{bmatrix} M_{100} & \cdots & M_{1q0} \\ M_{101} & \cdots & M_{1q1} \\ \cdots & & \\ M_{10r} & \cdots & M_{1qr} \end{bmatrix} = \begin{bmatrix} t_0(0) & t_0(1) & \cdots & t_0(N-1) \\ t_1(0) & t_1(1) & \cdots & t_1(N-1) \\ \cdots & & \\ t_r(0) & t_r(1) & \cdots & t_r(N-1) \end{bmatrix}
$$

$$
\times \begin{bmatrix} M'_{10}(0) & \cdots & M'_{1q}(0) \\ M'_{10}(1) & \cdots & M'_{1q}(1) \\ \cdots & & \\ M'_{10}(N-1) & \cdots & M'_{1q}(N-1) \end{bmatrix},
$$

$$\vdots ;$$

$$
\begin{bmatrix} M_{p00} & \cdots & M_{pq0} \\ M_{p01} & \cdots & M_{pq1} \\ \cdots & & \\ M_{p0r} & \cdots & M_{pqr} \end{bmatrix} = \begin{bmatrix} t_0(0) & t_0(1) & \cdots & t_0(N-1) \\ t_1(0) & t_1(1) & \cdots & t_1(N-1) \\ \cdots & & \\ t_r(0) & t_r(1) & \cdots & t_r(N-1) \end{bmatrix}
$$

$$
\times \begin{bmatrix} M'_{p0}(0) & \cdots & M'_{pq}(0) \\ M'_{p0}(1) & \cdots & M'_{pq}(1) \\ \cdots & & \\ M'_{p0}(N-1) & \cdots & M'_{pq}(N-1) \end{bmatrix}.
$$

$$(27)$$

The matrix representation is usually considered very effective in software packages such as MATLAB. But in Section 4, our experiment shows that it performs well in C++ too.

3.2. Computing 3D Invariants. Equation (23) has a similar structure to the definition of 3D moments. Exchanging the order of summation, we can rewrite (23) as

$$I_{pqr}^f = \sum_{m=0}^{p} \sum_{v=0}^{q} \sum_{l=0}^{r} \theta_1(p, m) \theta_2(q, v) \theta_3(r, l) T_{mvl}^f, \qquad (28)$$

where

$$\theta_1(p, m) = \sum_{k=m}^{p} \sum_{i=m}^{k} \sum_{j=m}^{i} c_{pk} s_1(k, i) \Gamma_f^{-(i+1)} s_2(i, j) d_{jm}, \qquad (29a)$$

$$\theta_2(q, v) = \sum_{n=v}^{q} \sum_{b=v}^{n} \sum_{u=v}^{b} c_{qn} s_1(n, b) \Theta_f^{-(b+1)} s_2(b, u) d_{uv}, \quad (29b)$$

$$\theta_3(r, l) = \sum_{e=l}^{r} \sum_{h=l}^{e} \sum_{a=l}^{h} c_{re} s_1(e, h) \Lambda_f^{-(h+1)} s_2(h, a) d_{al}. \qquad (29c)$$

Therefore, the algorithm presented in the previous subsection can be applied to compute the invariants. Let us denote matrices $\{c_{ij}\}_{i, j=0}^{i, j=p}$, $\{s_1(i, j)\}_{i, j=0}^{i, j=p}$, diag $(\tau_f^1, \tau_f^2, \ldots, \tau_f^{p+1})$, $\{s_2(i, j)\}_{i, j=0}^{i, j=p}$, and $\{d_{ij}\}_{i, j=0}^{i, j=p}$ by $\mathbf{C}$, $\mathbf{S_1}$, $\mathbf{\Delta}$, $\mathbf{S_2}$, and $\mathbf{D}$, respectively. They are all lower triangular matrices, and (29a) can be evaluated by a matrix way; that is, $\Xi = C * S_1 * \Delta * S_2 * D$, where Ξ denotes the matrix $\{\theta_1(i, j)\}_{i, j=0}^{i, j=p}$. The same procedure will be repeated in computing (29b) and (29c).

4. Experimental Results

In this section, we first evaluate the efficiency of our computing algorithm, which is compared with geometric moment invariants [16], Legendre invariants [10], and Zhu's invariants [11]. Then we illustrate the performance of the proposed descriptors based on a 3D MRI image. Classification abilities of four invariants are tested on character sets finally.

A 3D silhouette image shown in Figure 1 is composed of 128 binary images with size of 128×128. It is used to evaluate the computational speed of the algorithm described in Section 3. Figure 2 shows the computational time required to calculate the invariants of order up to 18. It should be noted that the program is implemented in C++ on a PC with Intel Core 2 Duo P8400 2.4 GHz CPU, 4 GB RAM. Figure 2 implies that our descriptors require less computation time when the order of invariants increases, because we apply the proposed algorithm for computing the Tchebichef moments as well as the invariants. The geometric moments and corresponding invariants are calculated directly through nested loops. For Zhu's invariants and Legendre invariants, we utilize our method to obtain their 3D moments firstly, and then nested loops are used to compute moment invariants.
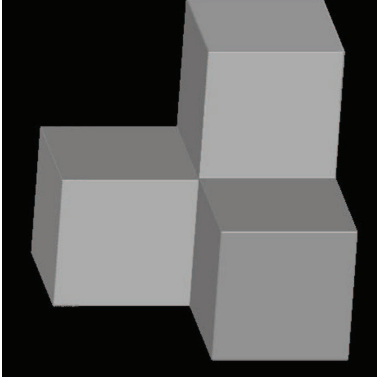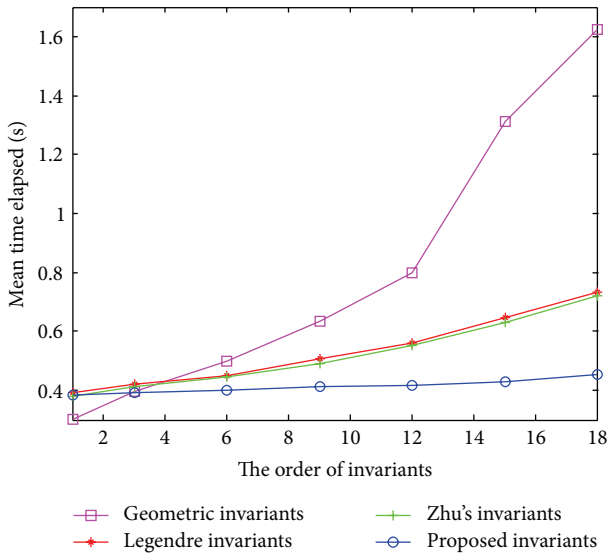
FIGURE 1: A 3D silhouette.

FIGURE 3: A 3D MRI head image.

FIGURE 2: Comparison of mean time elapsed (s).

FIGURE 4: Standard deviation of lower order invariants for a non-uniformly scaled 3D MRI image.

Since there are no nested loops in our algorithm, the required time increases little with the order growing.

The Volume Library [16] contains regular volume data mainly coming from CT or MRI scanners. Figure 3 shows a 3D MRI head image selected from this library. We resize this head image with the factors of 0.5, 1, 1.5, and 2 along x-axis, y-axis, and z-axis, respectively, which form a test set consisting of 64 images with different sizes. In order to measure "invariant" performance of the descriptors, we also adopt the deviation σ/μ, which was proposed by Chong et al. [9]. Here σ and μ denote the standard deviation and mean of invariants with the same order, respectively. Figure 4 illustrates the standard deviation of the geometric invariants, Legendre invariants, Zhu's invariants, and the proposed descriptors with the same order. It shows that the geometric invariants have the worst performance. Since high order of geometric moments has large variation in the dynamic range of values [6], this leads to unstable relative errors. Our descriptors have a slightly lower relative error than Zhu's invariants and Legendre invariants.
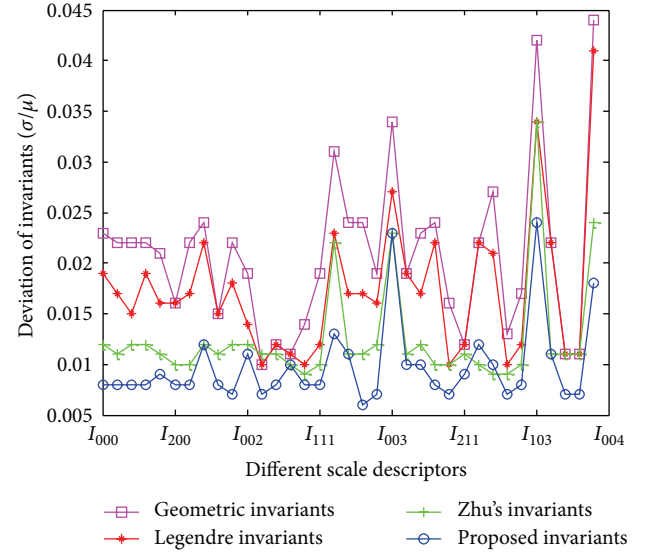
Moments of higher orders are generally considered more sensitive to image noise. To test the robustness of our invariants for degraded images, we add Gaussian noise (with mean $\mu = 0$ and variance σ varying from 0.01 to 0.3) and salt-and-pepper noise (with different noise densities from 0.01 to 0.3) to Figure 1, respectively. We rearrange the scale invariants of an image following the scheme through the path $(0, 0, 0) \rightarrow (1, 0, 0) \rightarrow (0, 1, 0) \rightarrow (0, 0, 1) \rightarrow (2, 0, 0) \rightarrow (1, 1, 0) \rightarrow (1, 0, 1) \rightarrow (0, 2, 0) \rightarrow (0, 1, 1) \rightarrow (0, 0, 2) \rightarrow \cdots \rightarrow (0, 0, m)$ and so on, to construct an invariant vector $I_f(m)$, where m is the maximum order of moment invariants. In this experiment, $m = 4$ is chosen; that is, the length of $I_f(m)$ equals 35.

The relative error of the invariant vector corresponding to the original image f and the degraded image $\widetilde{f}$ is defined as

$$e\left(f, \widetilde{f}\right) = \frac{\left\|I_f - I_{\widetilde{f}}\right\|}{\left\|I_f\right\|}, \tag{30}$$

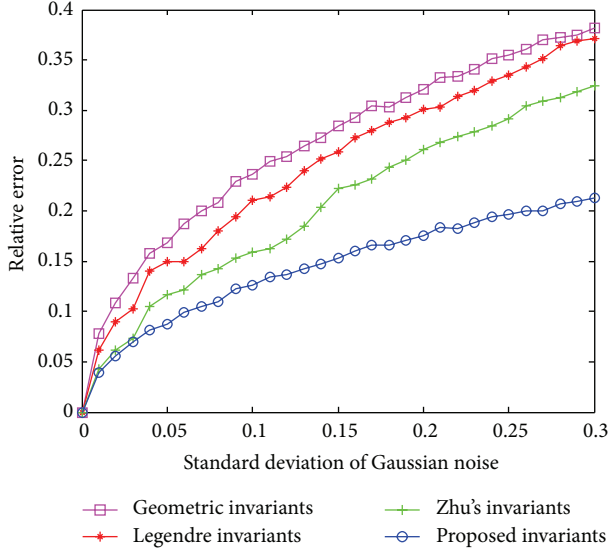where $\|\cdot\|$ is the Euclidean norm.

FIGURE 5: Relative errors of four descriptors with respect to additive Gaussian noise.
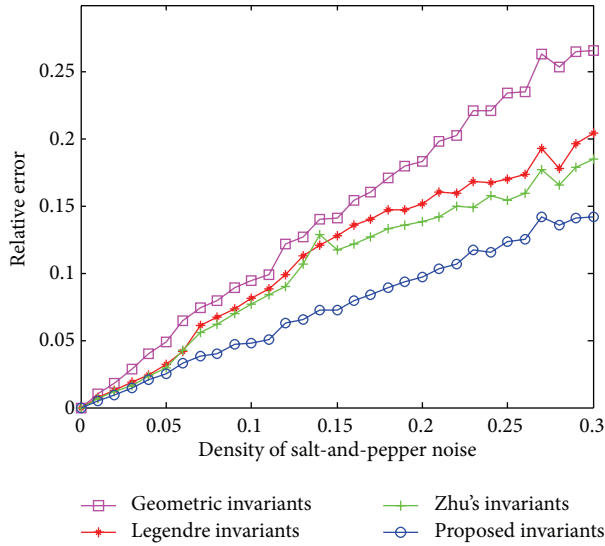

FIGURE 6: Relative error of four descriptors with respect to additive salt-and-pepper noise.

Relative errors caused by Gaussian and salt-and-pepper noise are depicted in Figures 5 and 6, respectively. We can observe that the relative error increases when increasing the noise level, and the proposed descriptors are more robust to noise than the other three competitors.

In the last experiment, we test the classification ability of our descriptors in both noise-free and noisy conditions. A classifier is required to identify the class of an unknown input object. We utilize the second and third order of invariants to form a feature vector. Therefore, the length of the vector equals 16. During the classification, the feature of an unknown object is compared with the training feature

FIGURE 7: Original alphanumeric characters as a training set for invariant character recognition.

TABLE 1: Classification results.

	Noise-free	1%	2%	3%	4%
Geometric invariants	97.16	87.07	81.25	68.04	50.85
Legendre invariants	97.87	89.06	83.95	74.15	56.96
Zhu's invariants	98.30	90.91	85.94	76.28	57.24
Proposed invariants	98.72	92.61	89.25	80.97	61.36

of a particular class. The Euclidean distance is frequently utilized as the classification measure, which is defined by

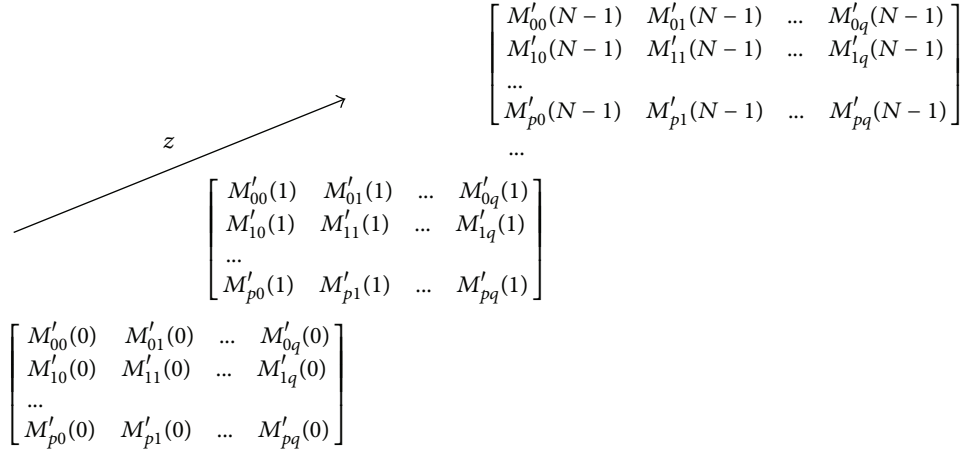$$d\left(V_s, V_t^{(k)}\right) = \left\| I_s - I_t \right\|, \tag{31}$$

where V_s and $V_t^{(k)}$ are feature vectors of the unknown sample and the k class, respectively. We define the classification rule such that an unknown input object will belong to the nearest class. The average classification accuracy is defined as

$$\eta = \frac{\text{Number of correctly classified images}}{\text{The total number of images}} \times 100\%. \tag{32}$$

An original set of alphanumeric characters with size of $64 \times 64 \times 64$ shown in **Figure 7** is used in this experiment. The reason for such a choice is that the elements in subsets $\{6, b, d\}$, $\{9, p, q\}$, and $\{a, c, o, 0, e\}$ may be confused due to the similarity. Every element is scaled with the factors $\{0.5, 1, 1.5,$ and $2\}$ along x-, y-, and z-axes, respectively, forming a testing set including 11 classes and 704 images. Additive salt-and-pepper noise with different noise densities 0.01, 0.02, 0.03, and 0.04 is added to the test set. The feature vectors based on the proposed invariants, geometric invariants, Legendre invariants, and Zhu's invariants are used to classify these images. The comparison result is listed in **Table 1**. We can see that there are few differences among the four descriptors in noise-free case. However, with the increase of noise density, our method is more robust than the other three.

5. Conclusions

We have presented a new method to derive the scale invariance of 3D Tchebichef moments. To reduce the computation time, we have proposed an efficient algorithm based on matrix multiplication for computing both 3D moments and 3D invariants. Experimental results show that our method has better classification ability, it is more robust to noise than the existing moment-based methods, and it is very effective.

$$\begin{bmatrix} M'_{00}(N-1) & M'_{01}(N-1) & ... & M'_{0q}(N-1) \\ M'_{10}(N-1) & M'_{11}(N-1) & ... & M'_{1q}(N-1) \\ ... \\ M'_{p0}(N-1) & M'_{p1}(N-1) & ... & M'_{pq}(N-1) \end{bmatrix}$$

...

$$\begin{bmatrix} M'_{00}(1) & M'_{01}(1) & ... & M'_{0q}(1) \\ M'_{10}(1) & M'_{11}(1) & ... & M'_{1q}(1) \\ ... \\ M'_{p0}(1) & M'_{p1}(1) & ... & M'_{pq}(1) \end{bmatrix}$$

$$\begin{bmatrix} M'_{00}(0) & M'_{01}(0) & ... & M'_{0q}(0) \\ M'_{10}(0) & M'_{11}(0) & ... & M'_{1q}(0) \\ ... \\ M'_{p0}(0) & M'_{p1}(0) & ... & M'_{pq}(0) \end{bmatrix}$$

FIGURE 8

Appendix

Proof of Theorem 1

Taking $\Gamma_g = \lambda \Gamma_f$ into account, we have

$$
I_P^g = \sum_{k=0}^{p} \sum_{i=0}^{k} \sum_{j=0}^{i} \sum_{m=0}^{j} c_{pk} s_1(k,i) \Gamma_g^{-(i+1)} s_2(i,j) d_{jm} T_m^g
$$

$$
= \sum_{k=0}^{p} \sum_{i=0}^{k} \sum_{j=0}^{i} \sum_{m=0}^{j} c_{pk} s_1(k,i) \lambda^{-(i+1)} \Gamma_f^{-(i+1)} s_2(i,j) d_{jm}
$$

$$
\times \sum_{n=0}^{m} \sum_{a=0}^{n} \sum_{b=0}^{a} \sum_{e=0}^{b} c_{mn} s_1(n,a) \lambda^{a+1} s_2(a,b) d_{bc} T_e^f. \tag{A.1}
$$

Exchanging the order of summation, we can rewrite (A.1) as

$$
I_P^g = \sum_{k=0}^{p} \sum_{i=0}^{k} c_{pk} s_1(k,i) \Gamma_f^{-(i+1)}
$$

$$
\times \sum_{j=0}^{i} \sum_{n=0}^{j} \sum_{a=0}^{n} \sum_{b=0}^{a} \sum_{e=0}^{b} s_2(i,j) s_1(n,a) \lambda^{a-i} s_2(a,b) d_{be} T_e^f
$$

$$
\times \sum_{m=n}^{j} d_{jm} c_{mn}. \tag{A.2}
$$

Since

$$
\sum_{m=n}^{j} d_{jm} c_{mn} = \delta_{nj}, \tag{A.3}
$$

where δ_{nj} is the Kronecker symbol.

Substitution of (A.3) into (A.2) yields

$$
I_P^g = \sum_{k=0}^{p} \sum_{i=0}^{k} c_{pk} s_1(k,i) \Gamma_f^{-(i+1)}
$$

$$
\times \sum_{j=0}^{i} \sum_{a=0}^{j} \sum_{b=0}^{a} \sum_{e=0}^{b} s_2(i,j) s_1(j,a) \lambda^{a-i} s_2(a,b) d_{be} T_e^f. \tag{A.4}
$$

Again, exchanging the order of a and j, we have

$$
I_P^g = \sum_{k=0}^{p} \sum_{i=0}^{k} c_{pk} s_1(k,i) \Gamma_f^{-(i+1)}
$$

$$
\times \sum_{a=0}^{i} \lambda^{a-i} \sum_{j=a}^{i} s_2(i,j) s_1(j,a) \sum_{b=0}^{a} \sum_{e=0}^{b} s_2(a,b) d_{be} T_e^f. \tag{A.5}
$$

Using $\sum_{j=a}^{i} s_2(i,j) s_1(j,a) = \delta_{ai}$, we obtain

$$
I_P^g = \sum_{k=0}^{p} \sum_{i=0}^{k} c_{pk} s_1(k,i) \Gamma_f^{-(i+1)} \sum_{b=0}^{i} \sum_{e=0}^{b} s_2(i,b) d_{be} T_e^f = I_p^f. \tag{A.6}
$$

Conflict of Interest

The authors declare that they have no financial and personal relationships with other people or organizations that can inappropriately influence their work. There is no professional or other personal interest of any nature or kind in any product, service, and/or company that could be construed as influencing the position presented in or the review of this paper.

References

[1] M. K. Hu, "Visual pattern recognition by moment invariants," *IRE Transactions on Information Theory*, vol. 8, no. 2, pp. 179–187, 1962.

[2] F. L. Alt, "Digital pattern recognition by moments," *Journal of the ACM*, vol. 9, no. 2, pp. 240–258, 1962.

[3] S. A. Dudani, K. J. Breeding, and R. B. McGhee, "Aircraft identification by moment invariants," *IEEE Transactions on Computers*, vol. 26, no. 1, pp. 39–46, 1977.

[4] M. R. Teague, "Image analysis via the general theory of moments," *Journal of the Optical Society of America*, vol. 70, no. 8, pp. 920–930, 1980.

[5] S. X. Liao and M. Pawlak, "On the accuracy of Zernike moments for image analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1358–1364, 1998.

[6] R. Mukundan, S. H. Ong, and P. A. Lee, "Image analysis by Tchebichef moments," *IEEE Transactions on Image Processing*, vol. 10, no. 9, pp. 1357–1364, 2001.

[7] P.-T. Yap, R. Paramesran, and S.-H. Ong, "Image analysis by Krawtchouk moments," *IEEE Transactions on Image Processing*, vol. 12, no. 11, pp. 1367–1377, 2003.

[8] P. T. Yap, R. Paramesran, and S. H. Ong, "Image analysis using Hahn moments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 11, pp. 2057–2062, 2007.

[9] C. W. Chong, P. Raveendran, and R. Mukundan, "Translation and scale invariants of Legendre moments," *Pattern Recognition*, vol. 37, no. 1, pp. 119–129, 2004.

[10] L. Y. Ong, C. W. Chong, and R. Besar, "Scale invariants of three-dimensional legendre moments," in *Proceedings of the 18th International Conference on Pattern Recognition (ICPR '06)*, pp. 141–144, August 2006.

[11] H. Zhu, H. Shu, T. Xia, L. Luo, and J. Louis Coatrieux, "Translation and scale invariants of Tchebichef moments," *Pattern Recognition*, vol. 40, no. 9, pp. 2530–2542, 2007.

[12] K. M. Hosny, "Refined translation and scale Legendre moment invariants," *Pattern Recognition Letters*, vol. 31, no. 7, pp. 533–538, 2010.

[13] H. Zhang, H. Shu, G. Coatrieux et al., "Affine Legendre moment invariants for image watermarking robust to geometric distortions," *IEEE Transactions on Image Processing*, vol. 20, no. 8, pp. 2189–2199, 2011.

[14] L. Comtet, *Advanced Combinatorics: The Art of Finite and Infinite Expansions*, D. Reidel Publishing, Dordrecht, The Netherlands, 1974.

[15] H. Shu, H. Zhang, B. Chen, P. Haigron, and L. Luo, "Fast computation of Tchebichef moments for binary and grayscale images," *IEEE Transactions on Image Processing*, vol. 19, no. 12, pp. 3171–3180, 2010.

[16] http://www9.informatik.uni-erlangen.de/External/vollib/.