Parallel Computing Zernike Moments via Combined Algorithms

An-Wen Deng* & Chih-Ying Gwo**

*Assistant Professor, Department of Information Management, Chien Hsin University of Science and Technology, Taoyuan, TAIWAN. E-Mail: awdeng{at}uch{dot}edu{dot}tw

**Associate Professor, Department of Information Management, Chien Hsin University of Science and Technology, Taoyuan, TAIWAN. E-Mail: ericgwo{at}uch{dot}edu{dot}tw

***Abstract*—**The purpose of this study is to propose two kinds of thread-level parallelization computing Zernike moments. Our proposed parallel algorithms are based on the combination of the q-recursive method and the Prata's method with symmetry by a certain dihedral group. The experiments results show that our proposed method is fast and accurate. The synchronized parallelization is applicable to greater than 250 order Zernike moments calculation. The reductive parallelization is suitable for computing Zernike moments order between 10 and 250. For computing all Zernike moments up to order 500, it only requires 3.499 sec. on a quad-core personal computer for a 512×512 test image. By our proposed method, the normalized mean square error with less than 500 orders Zernike moments is 0.001435 for image reconstruction, whereas the error rate by q-recursion method is 0.001866.

***Keywords*—**Image Processing; Parallel Computing; Prata's Method; Q-Recursive Method; Zernike Moments.

***Abbreviations*—**Normalized Mean Square Error (NMSE).

I. INTRODUCTION

THIS paper is an extended version of our previous paper [Deng & Gwo, 16]. Zernike Moments are widely applied in the field of pattern recognition [Belkasim et al., 3], image recovery [Liao, 5; Singh & Walia, 10] and image retrieval [Singh & Pooja, 11; Chen et al., 12] due to their orthogonality and rotation invariance property. However, the computation of these moments is extremely time-consuming especially at high orders. The main purpose of this study is to propose parallel algorithms for computing high-order Zernike moments. Our proposed parallel algorithms are based on the combination of the q-recursive method [Chong et al., 6] and the Prata's method [Prata & Rusch, 4] with the dihedral group D_4 [Tom Dieck, 2] symmetry.

The Zernike polynomials are introduced by the optical physicist F. Zernike [1] who won the Noble prize in 1953. The Zernike polynomials build an orthogonal basis for the complex L^2-space over the unit disc D with respect to the Lebesgue measure. More explicitly, for a complex number $z = x + iy = re^{i\theta}$ (x, y) are real numbers, (r, θ) represent the polar coordinates and $i = \sqrt{-1}$ is the pure imaginary number), the Zernike polynomial consists of a radial part and an exponential part which is given by

$$V_{nm}(z) = R_{nm}(r)e^{im\theta} = R_{nm}(r)(\cos(m\theta) + i\sin(m\theta)) \quad (1)$$

where $R_{nm}(r)$ is the Zernike radial polynomial, the order n is a non-negative integer, the repetition m is an integer satisfying $n - |m|$ = an even number and $|m| \le n$. The radial polynomial $R_{nm}(r)$ can be expressed as

$$R_{nm}(r) = \sum_{\substack{k=|m| \\ k-|m|:even}} R_{nmk}\, r^k,$$

$$\text{where} \quad R_{nmk} = (-1)^{\frac{n-k}{2}} \frac{\left(\frac{n+k}{2}\right)!}{\left(\frac{n-k}{2}\right)!\left(\frac{k+|m|}{2}\right)!\left(\frac{k-|m|}{2}\right)!} \quad (2)$$

The Zernike moments Z_{nm} can be regarded as the inner product of $f(z)$ with the Zernike polynomials $V_{nm}(z)$. The Zernike moments Z_{nm} are defined as

$$Z_{nm} = \frac{n+1}{\pi} \iint_{z \in D} f(z)\overline{V_{nm}}(z) \frac{dz \wedge d\bar{z}}{-2i} \quad (3)$$

where $\overline{V_{nm}}(z)$ denotes the complex conjugation of $V_{nm}(z)$ and the area differential 2-form is given by:

$$\frac{dz \wedge d\bar{z}}{-2i} = dx \wedge dy = rdr \wedge d\theta \quad (4)$$

Let A denote the image set of size $N \times N$, and $A(Z)=\{(s,t) \in A \mid s, t \text{ are integers}\}$. The data of the image pixels can be regarded as in a two-dimensional table $P(s,t)$ for $(s,t) \in A(Z)$, and can be embedded into the unit disc D in the following way. The pixel (s,t) is projected via η onto the grid centered at

$$(x_s, y_t) = \eta(s,t) = \left(\frac{2s - N + 1}{N\sqrt{2}}, \frac{2t - N + 1}{N\sqrt{2}} \right) \tag{5}$$

This results in the corresponding image function $f(z) = f(x,y)$ over $A' = \eta(A) \subseteq D$ for which $f(x,y) = f(\eta(s,t)) = P(s,t)$. The usual way to approximate the Zernike moments is to use the `following discrete form of the Zernike moments:

$$\hat{Z}_{nm} = \frac{2(n+1)}{\pi N^2} \sum\sum_{(s,t)\in A(Z)} P(s,t) R_{nm}(r)(\cos(m\theta) - i\sin(m\theta)) \tag{6}$$

With the Zernike moments of an image $f(z)$, this image can be reconstructed by the following formula.

$$f(x,y) = \lim_{M\to\infty} \sum_{n=0}^{M} \sum_{\substack{m=-n \\ n-m=even}} Z_{nm} V_{nm}(z)$$

$$= \lim_{M\to\infty} \sum_{n=0}^{M} \left(Z_{n0} R_{n0}(r) + 2 \sum_{\substack{m>0 \\ n-m=even}} R_{nm}(r)(\mathrm{Re}(Z_{nm})\cos(m\theta) - \mathrm{Im}(Z_{nm})\sin(m\theta)) \right) \tag{7}$$

Then the reconstructed image function can be expressed as

$$\hat{f}(x,y) = \sum_{n\le M} \sum_{m} Z_{nm} V_{nm}(x,y) \tag{8}$$

for a non-negative integer M. In comparison, the difference between the two images can be estimated by the normalized mean square error (*NMSE*), expressed in Equation (9). This measurement will be used to show the performance of stable computation among different methods.

$$\varepsilon^2 = \frac{\iint_D \left| f(x,y) - \hat{f}(x,y) \right|^2 dxdy}{\iint_D \left| f(x,y) \right|^2 dxdy} \tag{9}$$

For a considerable image, say 512x512 pixels or 1024x1024 pixels, to gain an acceptable reconstructed image, an order of 200 or higher Zernike moments would be required. The amount of computing for these images is enormous. Therefore, the finite groups symmetry method is useful in quickening the process [Li & Boyd, 14; Deng et al., 17]. For example the groups of reflections and rotations such as dihedral groups

$$D_4 = \{\tau_\theta, \iota_\theta \mid \theta = 0, \pi/2, \pi, 3\pi/2\},$$
$$V_4 = \{id, \iota_{\pi/2}, \iota_0, \tau_{3\pi/2}\} \tag{10}$$

Where denotes the rotation θ counter-clockwise and the reflection about the line $\theta/2$. To further explain the formulaic details, let the subset $H(Z)$ in $A(Z)$ be given by

$$H(Z) = \left\{ (s,t) \in A(Z) \mid \frac{N}{2} \le s,t \le N-1, t \le s \right\} \tag{11}$$

Then the Equation (6) can be modified with Equation (10) as the following equation:

$$\hat{z}_{nm} = \frac{2(n+1)}{\pi N^2} \left(\begin{array}{l} \sum\sum_{\substack{(s,t)\in H(Z) \\ s\ne t}} \sum_{\sigma\in D_4} P(\sigma(s,t)) R_{nm}(r) \begin{pmatrix} \cos(m\sigma(\theta)) - \\ i\sin(m\sigma(\theta)) \end{pmatrix} \\ + \sum\sum_{\substack{(s,t)\in H(Z) \\ s=t}} \sum_{\sigma\in V_4} P(\sigma(s,t)) R_{nm}(r) \begin{pmatrix} \cos(m\sigma(\pi/4)) \\ -i\sin(m\sigma(\pi/4)) \end{pmatrix} \end{array} \right) \tag{12}$$

Theoretically with this algorithm, the symmetry operated by the group D_4 or V_4 can be sped up to 8 or 4 times respectively. Besides the acceleration through symmetry by finite groups, there are other ways to hasten Zernike moments computation algorithms. Parallelization is one of those solutions. In this paper, we will focus on the thread-level parallelism for the multi-core computers.

II. PROPOSED ALGORITHMS

2.1. Q-Recursive Method and Modified Prata's Method

Our proposed method is based on the combination of q-recursive method and the Prata's method. Let us recall the both methods in this part. The q-recursive method is introduced by Chong et al., [6] which is based the following recurrence:

$$R_{nm}(r) = K_1 R_{n,m+4}(r) + \left(K_2 + \frac{K_3}{r^2} \right) R_{n,m+2},$$
$$m = n-4, n-6, \ldots, 1 \ (or \ \ 0) \tag{13}$$

with the initial conditions.

$$R_{nn}(r) = r^n, n \ge 0 \ \text{ and}$$
$$R_{n,n-2}(r) = nr^n - (n-1)r^{n-2}, \ n \ge 2 \tag{14}$$

Moreover, the coefficients satisfy

$$K_1 = \frac{(m+4)(m+3)}{2} - (m+4)K_2 + \frac{K_3(n+m+6)(n-m-4)}{8}$$
$$K_2 = \frac{K_3(n+m+4)(n-m-2)}{4(m+3)} + (m+2) \tag{15}$$
$$K_3 = \frac{-4(m+2)(m+1)}{(n+m+2)(n-m)}$$

for $n \ge 4$, $m \le n-4$. The time complexity of q-recursive method is $O(N^2 M^2)$. For a fixed order n and a fixed r, computing all Zernike radial polynomial $R_{nm}(r)$ for all repetition m takes a time complexity of $O(M)$.

The other method, namely the Prata's method, originally introduced in [Prata & Rusch, 4], uses the recurrence given by

$$R_{nm}(r) = rC_1 R_{n-1,|m-1|}(r) + C_2 R_{n-2,m}(r) \tag{16}$$

where the constants can computed as follows:

$$C_1 = \frac{2n}{m+n}, \ C_2 = \frac{m-n}{m+n} = 1 - C_1 \ \text{ for } \ m = n, n-2, \ldots, 1 \ (or \ \ 0) \tag{17}$$

A modified version of Prata's method computes the radical polynomials R_{n0} by using the recurrence:

$$R_{n0}(r) = 2rR_{n-1,1}(r) - R_{n-2,0}(r) \tag{18}$$

which needs a time complexity of $O(N^2 M^2)$.

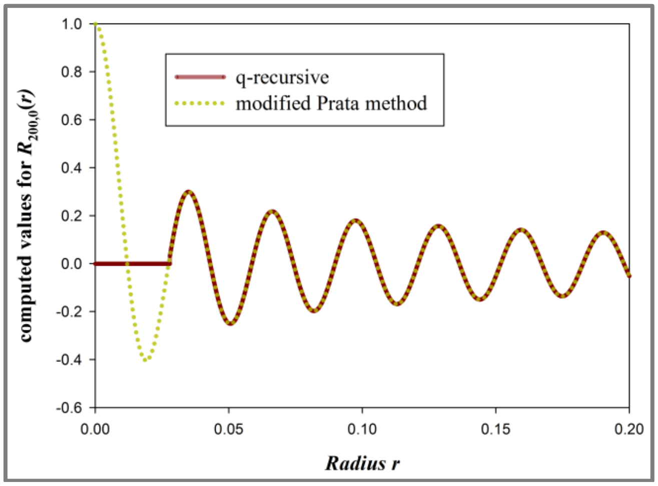


Figure 1: The plots of radial polynomial $R_{200,0}(r)$ using the q-recursive method and the modified Prata's method

It is to mention that using any of the two methods to compute Zernike moments for high order can lead to a noticeable error in the reconstructed image. Due to our experiments, the reconstruction from the Zernike moments computed by the modified Prata's method is inacceptable for the maximal order greater than 95. The q-recursive method obtains the computed values for the Zernike radial polynomial $R_{nm}(r)$=0 for $n \geq n_r, |m| \leq 106$ (n_r is some positive number which is dependent on n). For high Zernike moment order, the q-recursive computation method will lose information. This phenomenon is observed by [Papakostas et al., 9]. Due to the phenomena seen in Figure 1 on interval [0, 0.2] where a discrepancy appeared between the two methods, we sought to combine these two algorithms in the next section.

2.2. *Proposed Method in Serial Version*

For the maximal Zernike moment $M \geq 250$, the reconstruction from the computed Zernike moments using q-recursive method has distinct errors. Although the computed Zernike radial polynomial $R_{nm}(r)$ by using the modified Prata's method are not correct for $n \geq 95$ and large m, the computed values for $R_{nm}(r)$ are usable for small repetition m and has better estimate for small r than by using q-recursive method when $n \leq 270$. Our main idea is to combine both these methods to obtain algorithms with better performance.

Algorithm A:

Let Z[n][m] denote the element of the array storing the values for Zernike moments Z_{nm}; let R_nm denote the memory storing the temporary computed value for the Zernike radial polynomial $R_{nm}(r)$.

1. Compute all constants C_1, C_2 as in Equation (17) for modified Prata's method, and compute all constants K_1, K_2, K_3 as in Equation (15) for q-recursive method for all n, m where $n \leq M$.
2. For each (s,t) in A(Z)
   - x=2*s-N+1
   - y=2*t-N+1
   - r= $\sqrt{x*x+y*y}$
   - 2a. Using linear recurrence in [Deng et al., 15] to compute $\cos m\theta$, $\sin m\theta$, r^k for m, k=0,1,...,M
   - 2b. Compute Z[0][0]=Z[0][0]+2/π *N*N*f(x,y)
   - 2c. For n=1 to 3 step 1
     - Using q-recursive method to compute Z[n][m]=Z[n][m]+2/π*N*N*f(x,y)*R_nm*($\cos m\theta - i*\sin m\theta$) for m=n, n-2,..., 1(or 0)
     - Next n
   - 2d. For n=4 to M step 1
     - 2d1. Using q-recursive method to compute Z[n][m]=Z[n][m]+2/π *N*N*f(x,y)*R_nm*($\cos m\theta - i*\sin m\theta$) for m=n, n-2,..., 7(or 6)
     - 2d2. If n is even then
       - a=4
       - Else
       - a=5
       - End if
     - 2d3. Using Prata's method to compute Z[n][m]=Z[n][m]+2/π*N*N*f(x,y)* R_nm*($\cos m\theta - i*\sin m\theta$) for m=a, a-2,..,1(or 0)
     - Next n

End for

Algorithm A takes a time complexity of $O(N^2M^2)$. The total memory need has a complexity of $O(N^2+M^2)$, where the array Z[0:M][0:M] storing the computational values for Zernike moments contributes a complexity of $O(M^2)$, the array storing the values $f(x,y)$ for each pixel in a gray-level image contributes a complexity of $O(N^2)$, and other arrays and variables storing the temporary values for cos mθ, sin mθ, r^k and $R_{nm}(r)$ contribute a complexity of $O(M)$.

Algorithm A can be sped up by using D_4 symmetry. Such a modification can speed up to 8 times which we denote by Algorithm A+.

2.3. *Thread-Level Parallelism using Synchronization*

The control variables s, t in the outer loop computing Zernike moments have dependencies in the index numbers n, m for the array storing the values for Zernike moments Z_{nm}. A division of this array into disjoint subsets can be done, but not indexed by the loop control variables s, t in outer loop. A careless use of parallel for-loop will definitely cause race conditions: either two threads write to the same memory location simultaneously or one thread reads and another thread writes to the same memory location.

To avoid race conditions, synchronization is a common solution [Mattson et al., 7]. But a heavy synchronization results in performance impediment. Some parallel program with heavy synchronization is even slower than the serial version. The parallel algorithm, denoted as Algorithm B, is a slight modification for Algorithm A in which the foremost outer for-loop is replace by the parallel-for loop; moreover, the light-weight synchronization is used within the parallel-for loop only when updating the values for Z[n][m].

Since the extra memory need for a thread takes a complexity of $O(M)$, a P multithreads with an image of pixels $N \times N$ and maximal Zernike moment order M require a memory complexity of $O(PM+N^2)$ in Algorithm B. For high maximal Zernike moment order M and large number P, parallelism brings a mild increase on extra memory need. On the other hand, the synchronization demands an increased amount of time. For low maximal Zernike moment order M, a heavy slowdown of performance can be observed. Such way of parallelism can be applied to Algorithm A with D speedup, which we denote by Algorithm B+.

2.4. Thread-Level Parallelism using Reduction Method

Other parallel version of Algorithm A is so constructed that each thread has its own private copy lists storing the values of Zernike moment Z_{nm}. We call such method as reduction method. By having the values for Z_{nm} update only for thread private copies and parallel for-loop to sum up the values for all thread private lists together, neither race conditions nor false sharing would occur. It means that the reconstruction from Zernike moments computed by parallel version of an algorithm is identical with the computation by its serial version.

Algorithm C:

Let z[p][0:M][0:M] denote the array storing the values for Zernike moments Z_{nm} for thread number p; let R_{nm} denote the memory storing the temporary computed value for the Zernike radial polynomial $R_{nm}(r)$.

1. Parallel compute all constants C_1, C_2 as in Equation (17) for modified Prata's method, and parallel compute all constants K_1, K_2, K_3 as in Equation (15) for q-recursive method for all n, m where $n \le M$.
2. Parallel for each (s,t) in A(Z) proceeding P threads with thread number p=0, 1, 2,..., P-1

    p=get thread number of running thread
    x=2*s-N+1
    y=2*t-N+1
    r= $\sqrt{x*x+y*y}$

    2a. Using linear recurrence in [Deng et al., 15] to compute $\cos m\theta,\ \sin m\theta,\ r^k$ for m, k=0,1,...,M

    2b. Compute
        z[p][0][0]=z[p][0][0]+2/π *N*N*f(x,y)
      For n=1 to 3 step 1
        Using q-recursive method to compute
        z[p][n][m]=z[p][n][m]+2/π*N*N*f(x,y)*R_nm*($\cos m\theta - i*\sin m\theta$)
            for m=n, n-2,..., 1(or 0)
      Next n

    2d. For n=4 to M step 1
      2d1. Using q-recursive method to compute
          z[p][n][m]=z[p][n][m]+2/π *N*N*f(x,y)*R_nm*($\cos m\theta - i*\sin m\theta$)
            for m=n, n-2,..., 7(or 6)
      2d2. If n is even then
            a=4
          Else
            a=5
          End if
      2d3. Using Prata's method to compute
          z[p][n][m]=z[p][n][m]+2/π*N*N*f(x,y)*R_nm*($\cos m\theta - i*\sin m\theta$)
            for m=a, a-2,..., 1(or 0)
        Next n

End Parallel for
Parallel for p=0 to P-1 in P threads
    For n=0 to M step 1
      For m=n to 0 step -2
            Z[n][m]= Z[n][m]+z[p][n][m]
      Next m
    Next n
End Parallel for

The memory need for any kind of parallel algorithms can be computed as follows:

$$(\text{total memory}) = (\text{shared memory}) + P \times (\text{private memory for a thread}) \tag{19}$$

So, for P multithreads, an image of pixels $N \times N$ and maximal Zernike moment order M, the total memory for Algorithm C has a complexity of $O(PM^2+N^2)$. An observation that extra memory need is large for high maximal Zernike moment order M and large number P is made. Due to this extra memory need, the parallel version can be even slower than the serial version.

Such way of parallelism can be applied to Algorithm A that has been sped up by using D_4 symmetry, which we denote by Algorithm C+.

III. EXPERIMENTAL RESULTS

3.1. Setting for Experiments

We implement our algorithms into C/C++ code with parallel library openMP 4.0 [Chapman et al., 8; Liao et al., 13] which supports the thread-level parallelism and has a plenty of thread synchronization mechanisms. Furthermore, the source code is complied with the TMD-GCC 4.9.2 64-bit C/C++ release compiler with optimization O2 and fast-math. The automatic parallelization is disabled. Most of computations for real numbers use the 64-bit double-precision float format. The computer is installed with OS win7 with 8 GB of RAM and an Intel i7- 4790 quad-core 3.6 GHz processor which supports eight multithreads. The test images are all in size of 512×512 pixels as shown in Figure 2. For experimental needs, some test images are also converted into 1024×1024 pixels size.

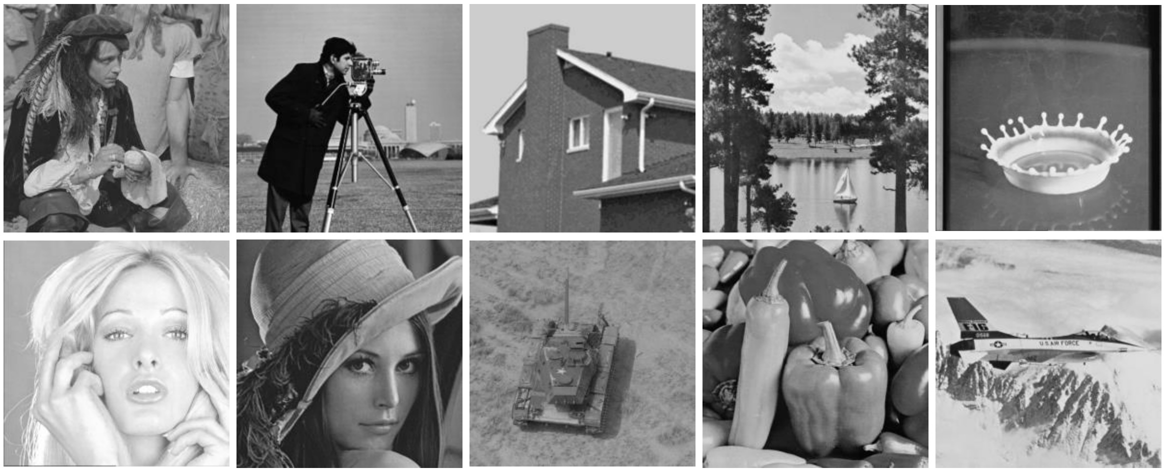


Figure 2: Ten standard test images are: Pirate, cameraman, house, lake, splash, Tiffany, Lena, tank, peppers and fighter F-16

3.2. *Error Analysis*

We test three different methods: the proposed method (Algorithm A with D_4 speedup, i.e. Algorithm A+), q-recursive method and the modified Prata method, for their reconstruction from computed Zernike moments up to maximal order *M*. The error rate for the reconstructed images at different maximal order between 0 and 500 are shown in Figure 4. All these three different recursive have the same reconstruction error rate for low maximal Zernike moment order $M \le 50$.

The reconstructed image for the modified Prata method becomes inaccurate for the medium maximal Zernike order *n*>95. Both of the proposed recursive method in Algorithm A+ and the q-recursive method have the similar reconstruction error for the medium maximal Zernike order between 95 and 250. For high maximal order 250<*M*, the proposed method, i.e. Algorithm A, has better performance, whereas the reconstruction from the computed Zernike moments using q-recursive method has noticeable image disturbance in the form of concentric circles. At maximal order 500, the error rate (*NMSE*) of reconstruction for the 512×512 pixels test image `Lena' from the computed Zernike moments using Algorithm A+ is 0.001435, whereas the error rate for the image reconstruction by q-recursion method is 0.001866. Those reconstructed images are shown in Figure 3.

(a)

(b)

(c)

Figure 3: The test image (a) 'Lena' of 512×512 pixels for experiments and its reconstruction images from computed values for Zernike moments up to order 500 (b) using proposed method Algorithm A+ and (c) using q-recursive method

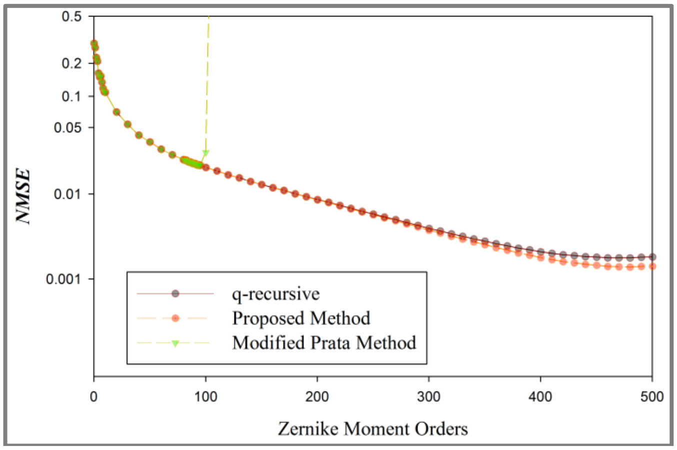


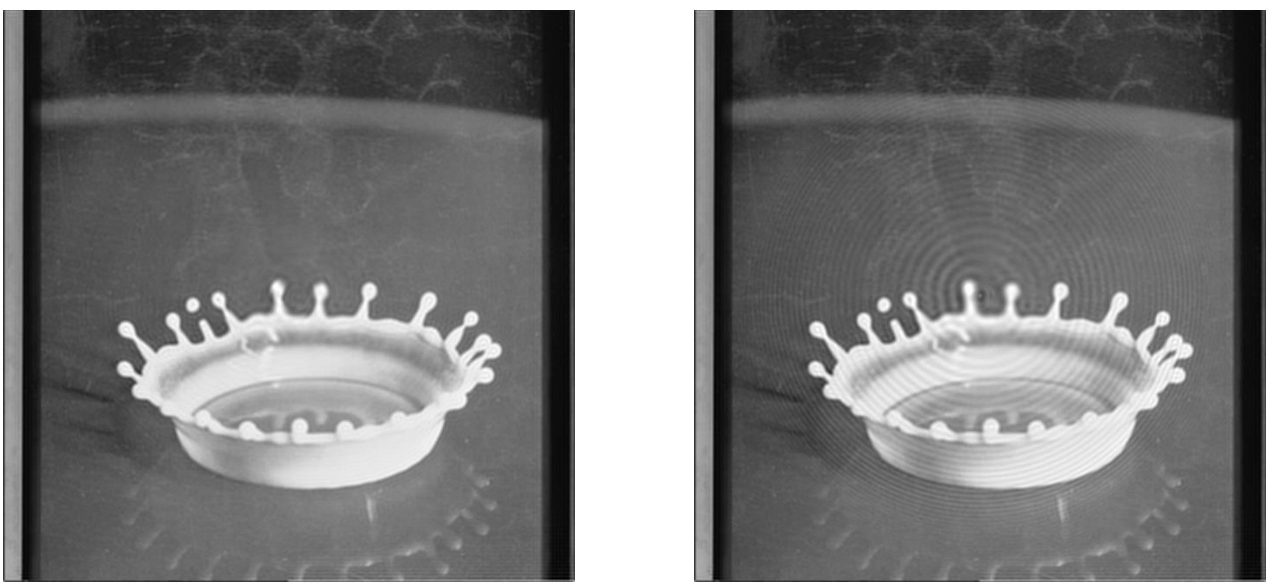
Figure 4: The error rate for the reconstructed images using different recursive methods

The NMSE for those ten test images using the proposed method and the q-recursive method at maximal order 500 are all listed in Table 1.

Table 1: The NMSE for test images using the proposed method Algorithm A+ and the q-recursive method at maximal order 500

	NMSE (10 ^{3})									
Algorithms	**Camera Man**	**House**	**Tiffany**	**F-16 fighter**	**Lena**	**Tank**	**Pirate**	**Splash**	**Pepper**	**Lake**
Proposed Method	2.537	1.811	1.397	2.220	1.435	3.301	4.957	1.872	3.486	3.272
Q-Recursive Method	2.822	2.064	1.733	2.492	1.866	3.509	5.234	2.516	3.923	3.666

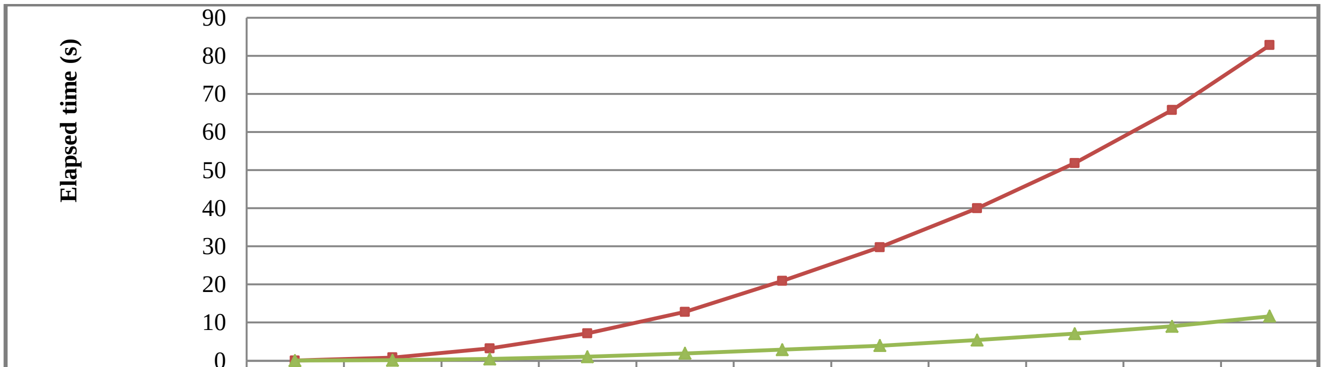
When those algorithms are applied to the enlarged test image 'Splash' of 1024×1024 pixels, one can better see the different performances in a variety of algorithms. The reconstructed image using proposed method, i.e. Algorithm A+, is almost identical with the original one, whereas the reconstruction from the computed Zernike moments using q-recursive method has ripples around the image center. At maximal order 500, the error rate (*NMSE*) of reconstruction from the computed Zernike moments using Algorithm A is 0.001274, whereas the error rate for the image reconstruction by q-recursion method is 0.001923. Those reconstructed images are shown in Figure.



(proposed method) $\varepsilon^2 = 0.001274$ (q-recursive method) $\varepsilon^2 = 0.001923$

Figure 5: The reconstructed images from 1024×1024 'Splash' using the proposed method and the q-recursive method and their *NMSE*

3.3. *Speed Analysis*



	10	50	100	150	200	250	300	350	400	450	500
q-recursive	0.047	0.827	3.23	7.176	12.807	20.92	29.734	39.983	51.823	65.754	82.742
Algorithm A+	0.0139999	0.111	0.456	1.032	1.931	2.866	3.945	5.409	7.071	9.003	11.663

Figure 6: Elapsed time for Algorithm A+ and q-recursive method applied to the test image 'Lena' of 512×512 pixels

In this part, we present the speed performance for both of the parallel algorithms using synchronization, i.e. Algorithm B+, and the parallel algorithms using reduction method, i.e. Algorithm C+. The numerical information is taken from the experiments on the test image 'Lena' of 512×512 pixels. The elapsed time among those ten test images varies insignificantly.

By comparing our proposed method in serial version, i.e. Algorithm A+, with the widely used algorithm q-recursive method, the results show that our proposed method is faster than q-recursive method. The numerical data is shown in Figure 6.

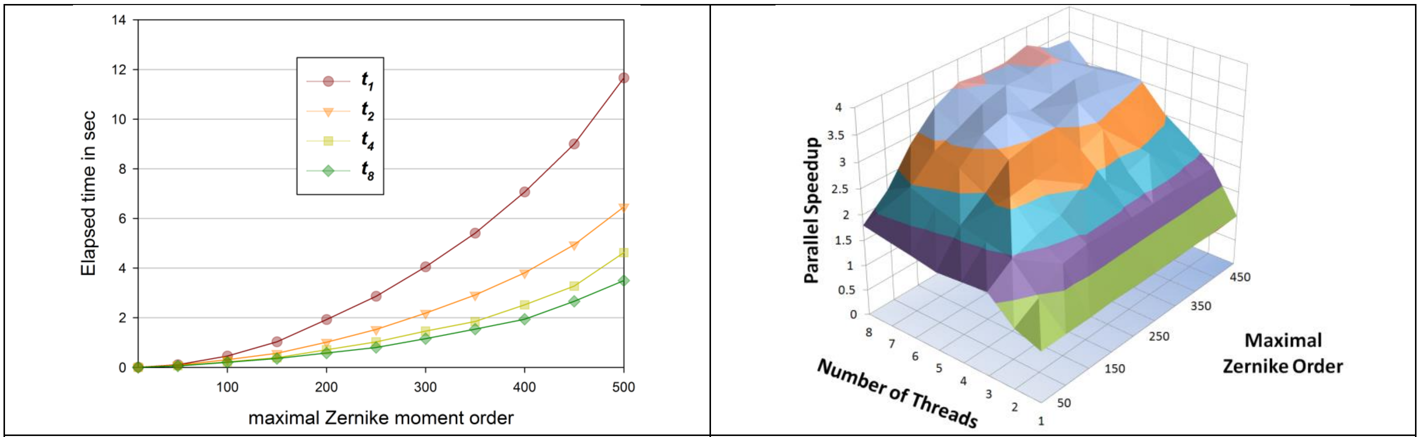
We test the speed performance for the parallel algorithm given by Algorithm B+ up to maximal Zernike moment as shown in Table 2. t_0 denotes the elapsed time for Algorithm A and t_1 denotes the elapsed time for Algorithm A with D_4 speedup, i.e. Algorithm A+ while t_i denotes the elapsed time for Algorithm B+ using i multithreads.

The total speedup factor is given by $s_0=t_0/t_i$ and the parallel speedup factor by $p_i=t_1/t_i$. Some experiment results are shown in Figure . For the low maximal order $M \le 100$, some parallel factor p_i is small. In this case, parallelism causes a heavy synchronization; in fact, the use of Algorithm A with speedup by D_4 is already very fast, parallelism using synchronization is not recommended in this situation. The most significant total speedup factor s_0 is between 7.45 and 7.76 for maximal Zernike moment order $M \ge 200$, which is near our theoretical prediction of 8 times faster for D_4 speedup.

Table 2: Elapsed time in sec. for Algorithm A, Algorithm A+ and Algorithm B+, and the associated speedup factors

max. order	t_0	t_1	s_0	t_2	p_2	s_2	t_4	p_4	s_4	t_8	p_8	s_8
200	14.385	1.931	7.45	1.019	1.89	14.12	0.713	2.71	20.18	0.582	3.32	24.72
250	21.755	2.866	7.59	1.53	1.87	14.22	1.028	2.79	21.16	0.804	3.56	27.06
300	30.629	3.945	7.76	2.115	1.87	14.48	1.514	2.61	20.23	1.085	3.64	28.23
350	41.275	5.409	7.63	2.920	1.85	14.14	1.85	2.92	22.31	1.542	3.51	26.77
400	53.776	7.071	7.61	3.811	1.86	14.11	2.523	2.80	21.31	1.942	3.64	27.69
450	68.934	9.003	7.66	4.947	1.82	13.93	3.277	2.75	21.04	2.664	3.38	25.88
500	88.428	11.663	7.58	6.471	1.80	13.67	4.625	2.52	19.12	3.499	3.33	25.27
												Unit in time: second

For maximal order $M \ge 200$, the parallel speedup factor p_8 lies between 3.32 and 3.64; and this is an acceptable parallelism performance for a quad-core computer. This concludes that the parallelism using synchronization method is suitable for high maximal Zernike moment order. A geometrical view of parallel speedup in our implementation of Algorithm B+ is shown in Figure 6 (b) which can be represented as a surface where $s_P=f(M,P)$, M is the maximal order and P is the number of threads in use.



(a) (b)

Figure 7: (a) the elapsed time for Algorithm A+ and Algorithm B+ with number of threads p=2, 4 and 8 (b) The surface given by the parallel factor function $s_P = f(M, P)$ using Algorithm B+

We also apply Algorithm C+ to the 512×512 pixels image 'Lena'. Some numerical results are listed in Table 3 and Figure 7. For maximal order between 50 and 250, the parallel speedup factor p_8 lies between 3.17 and 3.70; and the parallel speedup factor p_4 lies between 2.56 and 3.26. This concludes the parallelism using reduction method is suitable for medium maximal Zernike moment order. For high-order Zernike moments, experimental results show that the elapsed time for parallelism using reduction method is even larger than that of serial version. In this case, the large extra memory is needed which can result in computation hindrance. In such situation, parallelism using reduction method is not recommended.

Table 3: Elapsed time in sec. for Algorithm A, Algorithm A+ and Algorithm C+, and the associated speedup factors

max. order	t_0	t_1	s_0	t_2	p_2	s_2	t_4	p_4	s_4	t_8	p_8	s_8
10	0.045	0.014	3.21	0.008	1.75	5.63	0.005	2.80	9.00	0.005	2.80	9.00
50	0.764	0.111	6.88	0.059	1.88	12.95	0.034	3.26	22.47	0.035	3.17	21.83
100	3.23	0.456	7.08	0.244	1.87	13.24	0.178	2.56	18.15	0.137	3.33	23.58
150	7.488	1.032	7.26	0.608	1.70	12.32	0.366	2.82	20.46	0.303	3.41	24.71
200	14.385	1.931	7.45	1.015	1.90	14.17	0.632	3.06	22.76	0.522	3.70	27.56
250	21.755	2.866	7.59	1.544	1.86	14.09	0.959	2.99	22.69	0.812	3.53	26.79
300	30.629	3.945	7.76	2.224	1.77	13.77	1.462	2.70	20.95	1.647	2.40	18.60

Unit in time: second

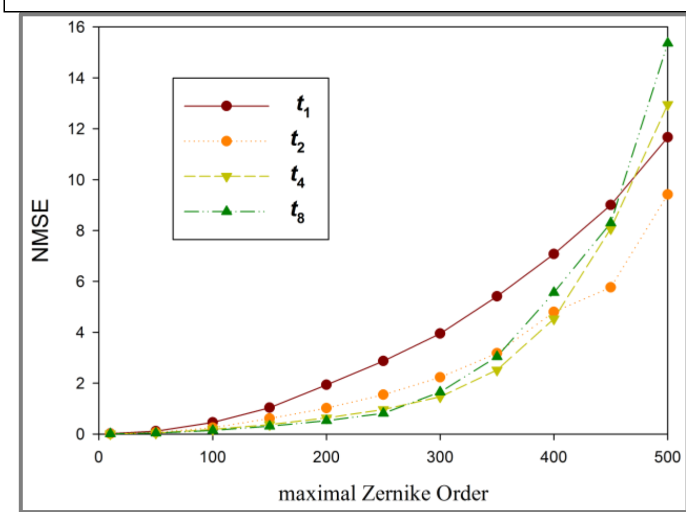


Figure 8: The elapsed time for Algorithm A+ and Algorithm C+ with number of threads p=2, 4 and 8

IV. Conclusion

The proposed method can yield accurate numerical results, especially for high-order moments. The experimental results show that the proposed parallel method using synchronization takes 3.499 seconds to compute the top 500-order Zernike moments of an image with 512×512 pixels. With the use of eight multithreads to compute high-order Zernike moments, the parallel speedup factor lies between 3.32 and 3.64, a good parallelism performance for a quad-core computer.

For medium maximal Zernike order, between 50 and 250, the parallelism by reduction method can be aptly used. The parallel speedup factor lies between 3.17 and 3.70 with eight multithreads, and the parallel speedup factor nears 3 just by using four multithreads.

At maximal order 500, the normalized mean square error is 0.001435 by using Algorithm A on the 512×512 pixels test image 'Lena', whereas the error rate for the image reconstruction by q-recursion method is 0.001866. When computing high-order Zernike moments, the proposed method outperforms other compared methods in terms of accuracy.

References

[1] F. Zernike (1934), "Beugungstheorie des Schneidenverfahrens und seiner verbesserten Form, der Phasenkontrastmethode", *Physica*, Vol. 1, No. 8, Pp. 689–704.

[2] T. Tom Dieck (1987), "Transformation Groups", *de Gruyter Studies in Mathematics*, 8.

[3] S.O. Belkasim, M. Shridhar & M. Ahmadi (1989), "Shape Recognition using Zernike Moments Invariants", *Proceedings of 23rd Annual Asilomar Conference on Signals Systems and Computers*.

[4] A. Prata & W.V.T. Rusch (1989), "Algorithm for Computation of Zernike Polynomials Expansion Coefficients", *Applied Optics*, Vol. 28, Pp. 749–754.

[5] S. Liao (1993), "Image Analysis by Moments". (Ph.D.), *The University of Manitoba*, Canada.

[6] C.W. Chong, P. Raveendran & R. Mukundan (2003), "A Comparative Analysis of Algorithms for Fast Computation of Zernike Moments", *Pattern Recognition*, Vol. 36, No. 3, Pp. 731–742.

[7] T.G. Mattson, B.A. Sanders & B.L. Massingill (2004), "Patterns for Parallel Programming", *Addison–Wesley Professional*.

[8] B. Chapman, G. Jost & R.v.d. Pas (2008), "Using OpenMP: Portable Shared Memory Parallel Programming", *MIT Press*.

[9] G.A. Papakostas, Y.S. Boutalis, D.A. Karras & B.G. Mertzios (2007), "A New Class of Zernike Moments for Computer Vision Applications", *Information Sciences*, Vol. 177, Pp. 2802–2819.

[10] C. Singh & E. Walia (2010), "Fast and Numerically Stable Methods for the Computation of Zernike Moments", *Pattern Recognition*, Vol. 43, Pp. 2497–2506.

[11] C. Singh & Pooja (2011), "Improving Image Retrieval using Combined Features of Hough Transform and Zernike Moments", *Optics and Lasers in Engineering*, Vol. 49, Pp. 1384–1396.

[12] B.J. Chen, H.Z. Shu, H. Zhang, G. Chen, C. Toumoulin, J.L. Dillenseger & L.M. Luo (2012), "Quaternion Zernike Moments and their Invariants for Color Image Analysis and Object Recognition", *Signal Processing*, Vol. 92, No. 2, Pp. 308–318.

[13] C. Liao, Y. Yan, B.R.D. Supinski, D.J. Quinlan & B. Chapman (2013), "Early Experiences with the OpenMP Accelerator Model", *OpenMP in the Era of Low Power Devices and Accelerators*, 8122.

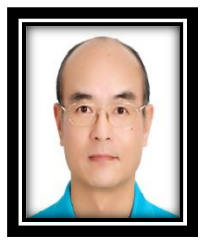
[14] S. Li & J.P. Boyd (2014), "Symmetrizing Grids, Radial Basis Functions, and Chebyshev and Zernike Polynomials for the Symmetry Group; Interpolation within a Squircle, Part I", *Journal of Computational Physics*, Vol. 258, Pp. 931–947.

[15] A.W. Deng, C.H. Wei & C.Y. Gwo (2015), "Algorithms for Computing Zernike Moments and Image Reconstruction in Parallel Process", *International Conference on Information Science and Control Engineering*, Shanghai, China.

[16] A.W. Deng & C.Y. Gwo (2016), "Parallel Algorithms for Computing Zernike Moments", *International Congress on Engineering and Information (ICEAI 2016)*, Osaka, Japan.

[17] A.W. Deng, C.H. Wei & C.Y. Gwo (2016), "Stable and Fast Computation of High-Order Zernike Moments using a Recursive Method", *Pattern Recognition*, Vol. 56, Pp. 16–25.

An-Wen Deng received his Bachelor degree from the National Central University, Taiwan, M.S. degree in Mathematics from National Tsing Hua University, Taiwan, and Ph.D. in Mathematics from University of Göttingen, Germany. He is currently an assistant professor of Department of Information Management at Chien Hsin University of Science and Technology, Taiwan. His current research interests include number theory, cryptography, parallel computation and pattern recognition.



Chih-Ying Gwo is currently an associate professor of the Department of Information Management at Chien Hsin University of Science and Technology, Taiwan. He received his Bachelor degree from the National Tsing Hua University, Taiwan, M.S. degree in computer engineering from Syracuse University, New York, USA, and Ph.D. in Electrical Engineering from University of Southern California, USA. His current research interests include online character recognition, pattern recognition, content-based image retrieval, and medical image processing and analysis.