

# HIGH ORDER MULTI-DIMENSIONAL MOMENT GENERATING ALGORITHM AND THE EFFICIENT COMPUTATION OF ZERNIKE MOMENTS

Mohammed Sadiq Abdul-Hameed

Electronics & Communications Eng. Dept., Baghdad University

Jadiria, P.O.Box 47316, Baghdad, IRAQ.

## ABSTRACT

The optimum computation of multi-dimensional (multi-D) image moments is presented in this paper. The developed algorithm is designed for the general case, specifically, to an arbitrary moment order  $R$  and to dimension  $d$ . The properties of the algorithm makes it best suited for obtaining the well known 2-D Zernike moments when they are computed through their relation to ordinary moments. Computational complexity model shows that the proposed algorithm takes only  $(NR+N)(N+R+1)$  additions with a negligible amount of multiplications, when an  $N$ -sized image is used to generate 2-D ordinary moments up to the order  $R$ . While the speed improvement of obtaining Zernike moments is of the order  $O(R)$  with respect to direct computation through Zernike polynomials. The regular structure of the processing elements and the minimum no. of operations of the algorithm makes it best suited for hardware and software implementations.

## I INTRODUCTION

The methods of Zernike moments have been presented as the notion of the general theory of image moments [1]. The most simple and basic form of moments is known as ordinary moments, this was first introduced by Hu [2] and since then they were used in pattern recognition problems. Other forms of moments also exist, i.e., pseudo Zernike [3], Legendre [1], complex [4] etc.. Zernike moments were proven to be the most powerful among all moments in image representation and the optimum encoding of essential features of an image [1],[3],[5]. Also the simple construction of invariants (features that are invariant to translation, rotation, and scaling) is another characteristics [1]. Most of these works, as well as others [6]-[8], show the necessity of moments theory in pattern recognition problems with a rapid increase of Zernike moments which requires the generation of high order moments. In [7], high order moments were used in robot sensing techniques, while [8] discusses the 3-D moment invariants and their extensions.

The direct computation of Zernike moments (through the Zernike polynomials) suffers from huge arithmetic operations which makes it impractical for real-time implementation or large statistical data tests. Fortunately, the possibility of the efficient computation of Zernike moments from ordinary's were investigated lately [9], a review of the proposed computation scheme is demonstrated in Sec.II.

Many works have addressed the computational theory of moments both in the sequential and the parallel mode [10]-[18] in a case that most of them concern the computation up to the third order (that is considered sufficient for obtaining the so called moment invariants). But the computation of Zernike moments into arbitrarily high orders requires an efficient high

order moments algorithm or else the computation process through ordinary moments will not be as efficient as expected and may have a delay that exceeds that of the direct computation through the Zernike polynomials. The reason beyond this, is that the number of ordinary moments required to generate the Zernike moments is more than the latest them self which means simply more operations and the existing fast algorithms may not solve the problem on the above terms.

One of the most powerful moments generators is through the method of digital filters [10] where the no. of operations is the minimum compared to all existing fast algorithms. To see how digital filters are cascaded to produce moments see [10]. So, the extension of the most optimum moment algorithm into high orders will be the topic of Sec.III. The complexity of computing model for testing the speed of the proposed algorithm linked to the Zernike moments is presented in Sec.IV. conclusions are given in Sec.V.

## II EFFICIENT COMPUTATION OF ZERNIKE MOMENTS

As stated by Teague [1] that, Zernike moments are related to ordinary moments if they are calculated inside the unit disk. A systematic approach is presented here to handle the computation of Zernike moments through the well known ordinary moments which is defined on the whole array domain (not only inside the unit disk). Then using the proper normalization it will be as if it is computed inside the unit disk. With this major move, the computation of Zernike moments can be performed (directly) through the fast moment algorithms which is designed specially for ordinary moments.

To investigate the computation scheme and the possible improvements, it can be shown easily that Zernike moments can be modified to be given as follows:

$$A_{RL} = \frac{(R+1)}{\pi} \sum_{k=L}^R \sum_{j=0}^S \sum_{\ell=0}^L (-i)^{\ell} \begin{bmatrix} S \\ j \end{bmatrix} \begin{bmatrix} L \\ \ell \end{bmatrix} B_{RKL} \times Z_{k-2j-L+\ell, 2j+L-\ell} \quad \dots(1)$$

where  $i = \sqrt{-1}$ ,  $S=(k-L)/2$ ,  $A_{RL}$  is the Zernike moment of order  $R$ ,  $B_{RLK}$  coefficients are given in [3]. Also,  $L$  takes now only positive integer values subject to the condition  $(R-L)$  is even and  $L \leq R$ . This have been accomplished through making use of the complex conjugate property which is  $A_{RL} = (A_{R,-L})^*$  that helps in getting the -ve values of  $L$ . In addition,  $Z_{pq}$  values are given by:

$$Z_{pq} = \frac{G_{pq}}{(D-1)^{(p+q)}} \quad \dots(2)$$

where

$$G_{pq} = \sum_{i=0}^p \sum_{j=0}^q \left[ \prod_{i=0}^p \prod_{j=0}^q \right] (-D)^{p+q-(i+j)} M_{ij} \quad \dots(3)$$

where  $D=(N+1)/2$ . As it can be seen, ordinary moments calculated inside the unit disk are functions to ordinary moments  $M_{ij}$ . This brings us to a great interest since the fast moments algorithms [10]-[18] can be designed to calculate ordinary moments with the maximum speed and efficiency. Note that  $M_{ij}$  are ordinary moments defined as:

$$M_{ij} = \sum_{n=0}^N \sum_{m=0}^N f(n,m) n^i m^j \quad \dots(4)$$

here the image  $f$  must be defined inside the circle  $x^2 + y^2 \leq N/2$ .

The major resource of computation redundancy is through the success of performing integer additions if computation is intended up to the third order. Unfortunately, extending the computation to the general order of Zernike moments (specifically to orders more than three.) requires some review to develop a fast ordinary moments algorithm to higher orders. This is because, most of the fast moment algorithms are designed for low orders (not more than three) knowing that not all these fast algorithms may result an efficient computation of high order Zernike moments where one must use (floating point) FLP arithmetics. All what is needed right now is the computation of ordinary moments given in Eq.(4) on the condition that the complexity of computing is at its minimum rate so that and after all, the whole operation complexity of Zernike moments from the above is less than that of the direct computation, see [1] for details on direct computation. The optimum computation of Eq.(4) is the topic of the next section towards a very general fast algorithm extended to any given order and expandable to higher dimensions as well.

### III. HIGH ORDER MULTI-D MOMENT GENERATING ALGORITHM

According to what had been discussed by Hatamian [10], the 1-D moment of order  $p$  of the sequence  $f(n)$  and length  $N+1$  can be defined as

$$\mu_p^N = \sum_{n=0}^N f(n) (N-n)^p \quad \dots(5)$$

obviously the above moment is computed with respect to the reference point  $N$  and the notation in Hatamian's work was  $\{\mu_N^p\}$ , we have slightly change it to  $\{\mu_p^N\}$  to keep up with the papers notations. And using a filter that can be realized by cascading a number of lower order filters which have a transfer function of the form:

$$\hat{H}_p(Z) = \frac{1}{(Z-1)^{p+1}} \quad \dots(6)$$

He solved the impulse response problem of the above filter up to the 3rd order to prove that the output of this filter is a linear combination of ordinary moments [10]. The most important features of this computation scheme is the minimum no. of operations required to compute moments compared to other fast algorithms, see for instance [11]. However, if Zernike moments is the computation issue, orders more than three are needed and the algorithm does not fit in its mean shape.

Within the framework of this paper, we will try to extend the moment computation by digital filters to high orders (a general form up to the order  $p$ ). The first step towards this general algorithm is finding the impulse response of the transfer function given in Eq.(6), which we found it to have the form:

$$\hat{h}_p(n) = u[n-(p+1)] \prod_{r=1}^p \{1 + [n-(p+1)]b_r\} \quad \dots(7)$$

where  $b_r = 1/r$  on the interval  $[1,p]$  and  $b_r = 1$  elsewhere, moreover, and by expanding the above binomial product the impulse response may now take the form

$$\hat{h}_p(n) = u[n-(p+1)] \sum_{r=0}^p B_r^p [n-(p+1)]^r \quad \dots(8)$$

where the calculation of  $B_r^p$  are given in Appendix.A. In fact, we are of most interest in the output of the filter, this output in response to  $f(n)$  is given by:

$$y_p(n) = \sum_{r=0}^p B_r^p \sum_{s=0}^{n-(p+1)} f(s) [n-(p+1)-s]^r \quad \dots(9)$$

and evaluating the output at  $n = N+(p+1)$  yields

$$y_p(N+p+1) = \sum_{r=0}^p B_r^p \sum_{s=0}^N f(s) (N-s)^r \quad \dots(10)$$

or simply

$$y_p = \sum_{r=0}^p B_r^p \mu_r^N \quad \dots(11)$$

as can be seen, the output  $y_p$  is a linear combination to ordinary moments  $\mu_r^N$ , (with  $r=0$  to  $p$ ). The problem now is finding  $\mu_r^N$ , in terms of  $y_p$ , this can be accomplished easily, hence, we may write Eq.(11) in the following inverse form

$$\mu_p^N = \sum_{r=0}^p C_r^p y_r \quad \dots(12)$$

where  $C_r^p$  are the coefficients obtained from Eq.(11) through the matrix inverse of the  $B$  coefficients. And for the 2-D moment of order  $(p,q)$ , it can be shown easily that, see Appendix.B:

$$\mu_{pq}^{NN} = \sum_{r=0}^p \sum_{s=0}^q C_{rs}^{pq} y_{rs} \quad \dots(13)$$

where  $C_{rs}^{pq} = C_r^p C_s^q$ , clearly it is very easy to expand it again to a third dimension (3-D moments) or more. It must be noted that in Hatamian's work [10], the above equation was in a vector form that has no general rule for generating moments up to high orders (the maximum is order 3). Again Eq.(13) can be put in the array form, if one wishes, which might support a multiplication redundancy though it will not affect computation since it does not depend on  $N$  (the image size). The 2-D moment computation by digital filters is shown in Fig.1 where we call blocks in the first row (horizontally connected blocks or filters) by row filters while the vertically connected blocks will be named as column filters.

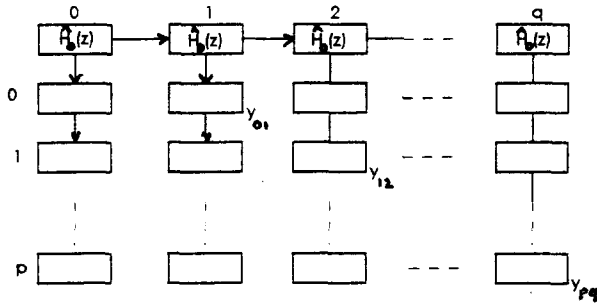


Fig.1 The suggested 2-D digital filter for generating a linear combination of image moments up to the order (p,q). Each block in the array is a single-pole filter similar to the one below in Fig. 2.

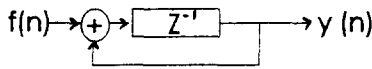


Fig.2 A single-pole digital filter for generating zero-order moment. This is equivalent to an accumulator (adopted from [10]).

The next step is expressing moments  $\{M_{ij}\}$  defined in Eq.(4) in terms of  $\{\mu_{ij}^{NN}\}$ , this can be done by changing the D factor in Eq.(3) which afterwards takes the form:

$$G_{pq} = \sum_{i=0}^p \sum_{j=0}^q \left[ \prod_{i=0}^p \prod_{j=0}^q \right] (D-N)^{p+q-(i+j)} \mu_{ij}^{NN} \quad \dots(14)$$

where  $\mu_{ij}^{NN}$ , now can be calculated directly from Eq.(13), thus Eq.(14) must be used instead of Eq.(3).

#### IV. COMPLEXITY OF COMPUTING

The performance of the proposed algorithms will be discussed in this section. The no. of arithmetic operations (additions and multiplications) will be used as the measure of the computational complexity for both Zernike and ordinary moments. To clarify each case, the complexity model will be classified into the categories as given next.

##### 1. Complexity of Computing Ordinary Moments via Digital Filters

Up to the (p,q)th-order (assuming  $R=p=q$ ) the no. of additions required to generate moments from digital filters (i.e., generating  $y_{pq}$ ) is given by:

$$NA_d = (R+1)N^2 + (R+1)^2 N \quad \dots(15)$$

where  $NA_d$  is the no. of additions required to generate moments  $\{y_{pq}\}$  from digital filters, N is the image size. The above no. of additions contributes mainly to the computation time due to its dependency to N. Furthermore, the no. of operations required to get  $\mu$  out of  $y$  through performing Eq.(13) is found as:

$$\lambda(R) = 2\rho(R) + 2R - 1 \quad \dots(16)$$

where  $\rho(s) = s(s+1)/2$ , and  $\lambda(R)$  is the no. of additions. Also an equal amount (to  $\lambda(R)$ ) of multiplications is required.

When the computation of 2-D moments is devoted for high orders, the integer accumulator is not suited due to the very high word length that is increased as one moves towards high and high orders. On the other hand, the use of scaling and truncation must be accompanied with a well planned correction algorithm since computation is performed recursively. Therefore, computations are performed through an FLP arithmetics to prevent overflow of data as one deals with large size of data and high order.

##### 2. Complexity of Computing Zernike Moments

The computation of Zernike moments up to a certain order requires the generation of ordinary moments up to the same order, but the no. of ordinary moments is more than that of Zernike moments and is given by:

$$N_o = \rho(R+1) \quad \dots(17)$$

comparing the above to the no. of Zernike moments (given later in Eq.(19)) we can see that the additional no. of moments (a factor of two) makes the computation of Zernike moments through ordinary's a critical issue unless the ordinary moment generation is very efficient. For instance, it can be shown easily that a generalization of the recursive moment algorithm given in [11] towards high orders calculations would not be more efficient and faster than direct computation through Zernike polynomials. Nevertheless, the complexity model discussed below shows the success of the algorithm developed in this paper.

##### A. Zernike moments through ordinary moments

According to the previous discussions Zernike moments can be computed from ordinary moments that are computed by digital filters, the total no. of operations is given as

$$NO_z = [(NR+N)(N+R+1) + \lambda(R)]_{\text{adds}} + [\lambda(R)]_{\text{mults}} + \xi(R) \quad \dots(18a)$$

where  $NO_z$  is the total no. of operations required to obtain Zernike moments from ordinary's, and  $\xi(R)$  is the no. of operations required to transform ordinary moments to Zernike moments by performing Eq.(1), with simple enumeration it is found that:

$$\xi(R) = \left[ \frac{1}{6} R(R+1)(R+5) \right]_{\text{adds}} + \left[ \frac{1}{6} (R+1)((R+1)^2 - 7) \right]_{\text{mults}} \quad \dots(18b)$$

Obviously, the computation of Eq.(18b) is of the order of  $O(R^3)$  and for large N or  $N \gg R$ , which is the case mostly, the computation of Eq.(1) can be considered to be of the  $O(RN^2 + R^2N)$ . In fact the computation of Eq.(1) can be reduced much further through matrix factorization especially for the case of small N and where  $N \leq R$ .

##### B. Direct computation through Zernike polynomials

If computation is up to the order R, then the total no. of Zernike moments is  $\rho(R+1)$ , and making use from the

complex conjugate property of obtaining Zernike moments, this number can be reduced to half the mentioned, so

$$N_z = \left\lceil \frac{1}{2} \rho(R+1) \right\rceil \quad \dots(19)$$

where  $N_z$  is the no. of Zernike moments to be calculated, the notation  $\lceil x \rceil$  is the least integer greater or equal to  $x$ . The total no. of operations would be given by:

$$NZ_z = [N_z [2G(N)]]_{\text{adds}} + [(N_z - R) [2G(N)]]_{\text{mults}} \quad \dots(20)$$

where  $G(N) = \left\lceil \pi \left( \frac{N}{2} \right)^2 \right\rceil$  due to the circular image used in the computation,  $NZ_z$  is the total no. of operations required to calculate the Zernike moments from the Zernike polynomials for simplicity it can be expressed as

$$NZ_z = \left[ \rho(R+1) \pi \left( \frac{N}{2} \right)^2 \right]_{\text{adds}} + \left[ [\rho(R+1) - 2R] \pi \left( \frac{N}{2} \right)^2 \right]_{\text{mults}} \quad \dots(21)$$

obviously the computation complexity is approximately of the order of  $O(R^2 N^2 + RN^2)$ . Assuming that the time complexity of multiplication is the same as that of addition, all operations (in the direct method) must be performed through FLP arithmetic due to the coefficients of Zernike polynomials.

### 3. Speed Improvement

A roughly estimated value for the speed improvement can be interpreted from dividing the estimated order of computations as follows

$$SpIm = \frac{R^2 N^2 + RN^2}{RN^2 + R^2 N} \quad \dots(22)$$

or it can be reduced to

$$SpIm = \frac{RN + N}{R + N} \quad (23)$$

where  $SpIm$  is the speed improvement factor of obtaining Zernike moments through the developed algorithm with respect to direct computation. Obviously this factor for, large  $N$ , can be of the order of  $O(R)$ . Thus, if computation is up to the 20th order, the suggested algorithm in this paper is twenty times faster than the direct approach, though it can be fastened much more since low order computations can be performed through integer accumulators. In fact, obtaining high order ordinary moments through integer additions are possible and we are investigating this problem in the meantime.

## V. CONCLUSIONS

The generalization of moments computation via digital filters have been developed successfully to high orders and multi-D's. The no. of operations required to generate moments is the optimum among all existing fast algorithms. The new generalization of the algorithm has been used in the efficient computation of Zernike moments which takes heavy computations in the direct approach. This computation scheme has become possible by making use of the relation between Zernike and ordinary moments keeping in mind that the efficient computation are not possible without the use of the optimum computation through digital filters. Of most

importance, however, is that multiplications can be enormously reduced in the new calculation of Zernike moments compared to the direct approach.

High order moments have a very high word length (the length increases with the moment order) therefore all arithmetics, must be performed via FLP operations. In fact, a well planned correction algorithm for the scaling or shifting of moments data (calculated by the digital filter) or the use of high word lengths adders for hardware implementations can greatly fasten computations through integer additions. It is shown also that, the expandability of the algorithm to high dimensions is straightforward and unusually simple, a perfect solution to the problem of finding the impulse response of the transfer function  $\hat{H}_p = 1/(Z-1)^{p+1}$  exist. The high regular structure of the algorithm makes it best suited for hardware implementation and software as well.

## REFERENCES

- [1] M. R. Teague, "Image Analysis Via The General Theory of Moments," J. Opt. Soc. Amer., vol. 70, pp. 920-930, Aug. 1980.
- [2] M. K. Hu., "Visual Pattern Recognition by Moment Invariants," IRE. Trans. Inform. Theory, IT-8, pp. 179-187, 1962.
- [3] C. H. Teh and R. T. Chin, "On Image Analysis by the Methods of Moments," IEEE PAMI, vol. 10, No. 4, pp. 496-512, July 1988.
- [4] Y. Abu-Mustafa, "Image Normalization by Complex Moments," IEEE PAMI, vol. 7, No. 1, pp. 46-55, Jan. 1985.
- [5] A. Khotanzad and Y. Hau Hong, "Invariant Image Recognition by Zernike Moments," IEEE PAMI, vol. 12, No. 5, pp. 489-497, May 1990.
- [6] M. S. Abdul-Hameed and S. M. Ali, "Template Matching in Translated, Rotated, and Scaled Images," ICECS'95, Amman, Jordan, Dec. 1995.
- [7] V. Markandey and R. J. P. deFigueiredo, "Robot Sensing Techniques Based on High-Dimensional Moment Invariants and Tensors," IEEE Trans. Robot. Automa., vol. 8, No. 2, pp. 186-195, Apr. 1992.
- [8] F. A. Sadjadi and E. L. Hall, "Three-Dimensional Moment Invariants," IEEE PAMI, vol. 2, pp. 127-136, 1980.
- [9] M. S. Abdul-Hameed, "On the Fast and Efficient Computation of Zernike Moment Invariants," to be published.
- [10] M. Hatamian, "A Real-Time Two-Dimensional Moment Generating Algorithm and its Single Chip Implementation," IEEE ASSP, vol. ASSP-4, pp. 546-553, June 1986.
- [11] K. Chen, "Efficient Parallel Algorithms for the Computation of Two-Dimensional Image Moments," Pattern Recognition, vol. 23, no. 12, pp. 109-119, 1990.
- [12] S. Wan-Chi, "Efficient Computation of Moments for Pattern Recognition," Proc. IEEE Pacific Rim Conf. Communications, Computers and Signal Processing, Victoria, BC, USA, 1991.
- [13] M. F. Zakaria et al., "Fast Algorithm For the Computation of Moment Invariants" Pattern Recognition, vol. 20, pp. 639-643, 1987.
- [14] M. S. Abdul-Hameed, "Parallel Algorithms for the Efficient Computation of Large Size Two-Dimensional Image Moments," to be published.