Albert - Ludwigs Universität Freiburg
Institut für Informatik

Lehrstuhl für Mustererkennung und Bildverarbeitung

# Fast 3D Zernike Moments and - Invariants

Internal Report 5/97

Nikolaos Canterakis
ALU Freiburg
Institut für Informatik
79110 Freiburg, Germany
canterakis@informatik.uni-freiburg.de

December 1997

# Abstract

The aim of this report is threefold: First we generalize to 3D a long ago known fast algorithm for the computation of ordinary geometrical moments of 2D fields starting from what could be named cumulative moments. This is done by first reformulating the 2D algorithm in terms of matrix operations and subsequently extending the result straightforwardly to 3D.
Second, guided by the results of much research work done in the past on the performance of 2D moments and moment invariants in the presence of noise suggesting that by using orthogonal 2D Zernike rather than regular geometrical moments one gets many advantages regarding noise effects, information suppression at low radii and redundancy, we have worked out and introduce a complete set of 3D polynomials orthonormal within the unit sphere that exhibits a "form invariance" property under rotation like the 2D Zernike polynomials do. For that reason we call this set 3D Zernike polynomials. The role of the angular exponential function in the 2D Zernike polynomials set is now played by the spherical harmonics on the surface of the unit sphere. We show how to directly compute the associated 3D Zernike moments from the cumulative moments without the need to go through the geometrical moments, thus avoiding the enormous dynamic range required by the latter.
In the third part it is shown that a set of complete moment invariants for orthogonal transformations of 3D objects that we had previously derived may be rebuilt to yield the 3D Zernike moments of the object in its standard position.


# 1 Introduction

The purpose of this article is to put together all pieces concerning 3D Zernike moments to be introduced in this report, their fast computation and connection to the complete set of moment invariants using spherical harmonics [4] we had derived in [3].

In [5] a fast algorithm and its single chip implementation for computing the ordinary geometrical moments of a 2D image has been presented that achieved a saving of more than 5 orders of magnitude regarding the number of multiplications needed compared to the direct implementation if the task is to compute 16 moments of low order of a $512 \times 512$ image. Although the sequential implementation of the algorithm yielded remarkable results as well, we feel that this algorithm does not appear to have found in the vision literature the deserved attention. We argue that the reason might be the formulation of the algorithm in terms of filter theory and the given weight to the hardware implementation rather than to the algorithmic part. In order to describe the 3D fast moment computation algorithm we are interested in here, we first reformulate the 1D version of the algorithm above in terms of matrix theory and then proceed to its 2D and 3D generalization.

Besides, motivated by several results reported for example in [1], [6] and [8] indicating that 2D Zernike polynomials [2] and moments are superior to the ordinary geometrical moments regarding performance in the presence of noise, information suppression and redundancy, we introduce 3D Zernike polynomials and moments and discuss their fast computation with the intention to use them for 3D invariant object description.

# 2 The fast moment computation algorithm

## 2.1 The fast 1D moment computation algorithm

We start with the definition of the geometrical moments $\mu_p$ of a 1D discrete field $f(n)$:

$$\mu_p := \sum_{n=0}^{N-1} n^p f(n) = (\boldsymbol{n}^T)^p \boldsymbol{f}$$

where we have defined $\boldsymbol{n}^T := [0, 1, \cdots, (N-1)]$ and $(\boldsymbol{n}^T)^p := [0^p, 1^p, 2^p, \cdots, (N-1)^p]$. Here and elsewhere exponentiation of a vector to a scalar power or exponentiation of a scalar to a vectorial power is always meant componentwise and yields a vectorial result. By generalizing this notion, exponentiation of a row vector with an exponent being a column vector or vice versa will accordingly give as result a matrix with row dimension equal to the dimension of the row vector and with column dimension equal to the dimension of the column vector involved. Besides, we have denoted with $\boldsymbol{f}$ the vector made up from the N samples of $f(n)$ at $n = 0, 1, \cdots, N-1$ and adopted the convention $0^0 = 1$. Now we denote the vector containing all moments of $\boldsymbol{f}$ from zero-th to $P$-th order with $\boldsymbol{\mu}_P$ and obtain

$$\boldsymbol{\mu}_P = \begin{pmatrix} \mu_0 \\ \mu_1 \\ \vdots \\ \mu_P \end{pmatrix} = \begin{pmatrix} (\boldsymbol{n}^T)^0 \\ (\boldsymbol{n}^T)^1 \\ \vdots \\ (\boldsymbol{n}^T)^P \end{pmatrix} \cdot \boldsymbol{f} =: (\boldsymbol{n}^T)^{\boldsymbol{P}} \cdot \boldsymbol{f} \ .$$

The hereby arising $(P+1) \times N$ matrix $(\boldsymbol{n}^T)^{\boldsymbol{P}}$ is Van der Monde and reads explicitly

$$(\boldsymbol{n}^T)^{\boldsymbol{P}} = \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 0 & 1 & 2 & 3 & \cdots & N-1 \\ 0 & 1 & 4 & 9 & \cdots & (N-1)^2 \\ 0 & 1 & 8 & 27 & \cdots & (N-1)^3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & 2^P & 3^P & \cdots & (N-1)^P \end{bmatrix} .$$

Thus, the $p$-th row of $(\boldsymbol{n}^T)^{\boldsymbol{P}}$ reads $(\boldsymbol{n}^T)^p = [0^p, 1^p, \cdots, (N-1)^p]$ and the $n$-th column of $(\boldsymbol{n}^T)^{\boldsymbol{P}}$ reads $n^{\boldsymbol{P}} = (n^0, n^1, \cdots, n^P)^T$ . We observe that the $p$-th row of $(\boldsymbol{n}^T)^{\boldsymbol{P}}$ contains $N$ samples of the $p$-th degree monomial $n^p$ in the variable $n$ considered continuous. So, $(\boldsymbol{n}^T)^{\boldsymbol{P}}$ contains $N$ samples of each monomial in the variable $n$ up to $P$-th degree. It is well known that all monomials up to $P$-th degree form a basis of the $(P+1)$-dimensional vector space $V_P$ of polynomials up to $P$-th degree in one variable. Now, it will be shown that choosing a different basis for $V_P$ the complexity of computing $\boldsymbol{\mu}_P$ can be considerably reduced. In particular, almost all multiplications may be spared. Indeed, consider the basis $\boldsymbol{b}(\boldsymbol{n}^T)$ composed of binomial coefficients rather than monomials:

$$\boldsymbol{b}(\boldsymbol{n}^T) := \begin{bmatrix} \mathbf{1}^T \\ \binom{\boldsymbol{n}^T}{1} \\ \binom{\boldsymbol{n}^T+\mathbf{1}^T}{2} \\ \vdots \\ \binom{\boldsymbol{n}^T+\boldsymbol{P}^T-\mathbf{1}^T}{P} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & \cdots & 1 \\ 0 & 1 & 2 & 3 & 4 & \cdots & N-1 \\ 0 & 1 & 3 & 6 & 10 & \cdots & \binom{N}{2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & P+1 & \binom{P+2}{P} & \binom{P+3}{P} & \cdots & \binom{P+N-2}{P} \end{bmatrix} . \quad (1)$$

Here we have adopted the notations $\mathbf{1}^T = (1,1,1,\cdots,1)$ , $\boldsymbol{P}^T = (P,P,P,\cdots,P)$ and the componentwise formation of binomial coefficients $\binom{\boldsymbol{m}^T}{\nu} = \left[\binom{m_0}{\nu}, \binom{m_1}{\nu}, \binom{m_2}{\nu}, \cdots \binom{m_{N-1}}{\nu}\right]$ if $\boldsymbol{m}^T = (m_0, m_1, m_2, \cdots, m_{N-1})$. Thus, the $p$-th row of $\boldsymbol{b}(\boldsymbol{n}^T)$ contains $N$ samples of a polynomial in $n$ of $p$-th degree without $n^0$ coefficient, namely $\binom{p+n-1}{p} = \frac{(n+p-1)(n+p-2)\cdots n}{p!}$ for $n = 0,1,2,\cdots,N-1$ . Our aim is to use $\boldsymbol{b}(\boldsymbol{n}^T)$ as a new basis for $V_P$ and the next question must concern the transformation matrix $\boldsymbol{T}_P$ between the two systems of basis functions $(\boldsymbol{n}^T)^{\boldsymbol{p}}$ and $\boldsymbol{b}(\boldsymbol{n}^T)$. Later on it will be shown why this new basis $\boldsymbol{b}(\boldsymbol{n}^T)$ offers considerable computational savings.

### 2.1.1 Matrix elements of $\boldsymbol{T}_P$

We define $\boldsymbol{T}_P$ by $(\boldsymbol{n}^T)^{\boldsymbol{p}} = \boldsymbol{T}_P \cdot \boldsymbol{b}(\boldsymbol{n}^T)$. For a particular $n$ we thus have

$$\begin{pmatrix} 1 \\ n \\ n^2 \\ \vdots \\ n^P \end{pmatrix} = \boldsymbol{T}_P \cdot \begin{pmatrix} 1 \\ n \\ n(n+1)/2 \\ \vdots \\ \binom{P+n-1}{P} \end{pmatrix} .$$

Clearly, $\boldsymbol{T}_P$ must be lower triangular since $n^p$ may be linearly composed from $p$-th degree polynomials in $n$ and lower. It is then easy to see that $\boldsymbol{T}_P$ must be of the form

$$\boldsymbol{T}_P = \begin{bmatrix} 1 & & & & & \\ 0 & t_{11} & & & & \\ 0 & t_{21} & t_{22} & & & \\ 0 & t_{31} & t_{32} & t_{33} & & \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & t_{P1} & t_{P2} & t_{P3} & \cdots & t_{PP} \end{bmatrix}$$

with $t_{pq}$ independent of $n$, whence we have

$$\begin{pmatrix} 1 \\ n \\ n^2 \\ \vdots \\ n^{P-1} \end{pmatrix} = \begin{bmatrix} t_{11} & & & & \\ t_{21} & t_{22} & & & \\ t_{31} & t_{32} & t_{33} & & \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ t_{P1} & t_{P2} & t_{P3} & \cdots & t_{PP} \end{bmatrix} \cdot \begin{bmatrix} 1 \\ (n+1)/2 \\ (n+1)(n+2)/6 \\ \vdots \\ (n+1)(n+2)\cdots(P+n-1)/P! \end{bmatrix} .$$

For the determination of $t_{pq}$ we therefore must solve the equations

$$t_{p1} + t_{p2} \cdot \frac{n+1}{2} + t_{p3} \cdot \frac{(n+1)(n+2)}{6} + \cdots + t_{pp} \cdot \frac{(n+1)(n+2)\cdots(n+p-1)}{p!} = n^{p-1} \quad ;$$

$p = 1, 2, \cdots, P$ .

Although we consider only integers $n$ in the range $[0, \cdots, (N-1)]$ the polynomial equations above must of course hold for *any* $n$. Upon setting for $n$ the values $n = -1, n = -2, \cdots$ we obtain recursively from above

$$\begin{aligned} t_{p1} &= (-1)^{p-1} \\ t_{p2} &= (-1)^p 2(2^{p-1} - 1) \\ t_{p3} &= (-1)^{p-1} 3(3^{p-1} - 2 \cdot 2^{p-1} + 1) \\ t_{p4} &= (-1)^p 4(4^{p-1} - 3 \cdot 3^{p-1} + 3 \cdot 2^{p-1} - 1) \end{aligned}$$

and so on, and it is not hard to prove by induction

$$t_{pq} = (-1)^p \cdot \sum_{\nu=1}^{q} (-1)^\nu \binom{q}{\nu} \nu^p \quad .$$

Thus, the coefficients $t_{pq}$ and hence the matrix $\boldsymbol{T}_P$ too are known constants that can be computed beforehand for any $p$, $q$ and $P$.

### 2.1.2 Analysis of the 1D algorithm

As we will now see, since $\boldsymbol{\mu}_P = (\boldsymbol{n}^T)^{\boldsymbol{P}} \cdot \boldsymbol{f} = \boldsymbol{T}_P \cdot \boldsymbol{b}(\boldsymbol{n}^T) \cdot \boldsymbol{f}$, all multiplications needed to compute $\boldsymbol{\mu}_P$ have been moved to the multiplication by the $(P+1) \times (P+1)$ lower triangular matrix $\boldsymbol{T}_P$ whereas the product $\boldsymbol{b}(\boldsymbol{n}^T) \cdot \boldsymbol{f}$ requires only additions. To show the latter we introduce the $N$-vectors $\boldsymbol{e}_1^T := (1, 0, 0, \cdots, 0)$ and $\boldsymbol{e}_N^T := (0, 0, \cdots, 0, 1)$ as well as the $N \times N$ matrices

$$
\boldsymbol{C}_N := \begin{bmatrix} 1 & & & & & \\ 0 & 1 & & & & \\ 0 & 1 & 1 & & & \\ 0 & 1 & 1 & 1 & & \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 1 & 1 & 1 & \cdots & 1 \end{bmatrix} \text{ and } \boldsymbol{V}_N := \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 1 & 0 & 0 & \cdots & 0 \end{bmatrix} .
$$

We now form powers of $\boldsymbol{C}_N$ and observe

$$
\boldsymbol{C}_N^{p+1} = \begin{bmatrix} 1 & & & & & \\ 0 & 1 & & & & \\ 0 & p+1 & 1 & & & \\ 0 & \binom{p+2}{p} & p+1 & 1 & & \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \binom{p+N-2}{p} & \binom{p+N-3}{p} & \cdots & p+1 & 1 \end{bmatrix} .
$$

Comparing with (1) we see that with the exception of the zero-th row the $p$-th row of $\boldsymbol{b}(\boldsymbol{n}^T)$ may be obtained from the last row of $\boldsymbol{C}_N^{p+1}$ after some trivial reordering. To be specific, we have

$$
\boldsymbol{b}(\boldsymbol{n}^T) = \underbrace{\begin{pmatrix} \boldsymbol{e}_1^T + \boldsymbol{e}_N^T \boldsymbol{C}_N \\ \boldsymbol{e}_N^T \boldsymbol{C}_N^2 \\ \boldsymbol{e}_N^T \boldsymbol{C}_N^3 \\ \vdots \\ \boldsymbol{e}_N^T \boldsymbol{C}_N^{P+1} \end{pmatrix}}_{=:\boldsymbol{B}_N} \cdot \boldsymbol{V}_N
$$

and hence

$$
\boldsymbol{\mu}_P = \boldsymbol{T}_P \boldsymbol{B}_N \boldsymbol{V}_N \boldsymbol{f} \ .
$$

Since multiplication by $\boldsymbol{C}_N$ is essentially equivalent with building the series of partial sums we may define the product $\boldsymbol{B}_N \boldsymbol{V}_N \boldsymbol{f} =: \boldsymbol{\kappa}_P$ and call $\boldsymbol{\kappa}_P$ the "cumulative" moments of $\boldsymbol{f}$ up to order $P$. The connection to the ordinary geometrical moments $\boldsymbol{\mu}_P$ is then given by

$$
\boldsymbol{\mu}_P = \boldsymbol{T}_P \boldsymbol{\kappa}_P \ .
$$

An important point to note here is that $\boldsymbol{\kappa}_P$ requires typically a much less numerical dynamic range than $\boldsymbol{\mu}_P$. This observation will lead us later to formulate the 3D Zernike moments to be developed in this report directly in terms of cumulative moments, thus avoiding to a large extend the numerical problems associated with using geometrical moments, especially of high order.

Now, to assess the overall complexity we summarize:

- multiplication by $V_N$ is a trivial reordering

- multiplication by $\boldsymbol{e}_1^T$ or $\boldsymbol{e}_N^T$ means simply the selection of the zero-th or the last element respectively

- multiplication by $\boldsymbol{C}_N$ requires only $N-2$ additions and hence multiplication by $B_N$ or equivalently computation of $\boldsymbol{\kappa}_P$ requires $(P+1)(N-2)$ additions only

- multiplication by $\boldsymbol{T}_P$ requires $P(P+1)/2$ multiplications and $(P-1)P/2$ additions as can be easily seen from the form of $\boldsymbol{T}_P$ .

Thus, the overall complexity is:

- Number of multiplications $\#M = P(P+1)/2$
- Number of additions $\#A = (P+1)(N+2) + (P-1)P/2$ .

We notice in particular that the number of multiplications does not depend on the amount of the data, $N$. The figures above must be compared to the direct algorithm $(\boldsymbol{n}^T)^{\boldsymbol{P}} \cdot \boldsymbol{f}$ requiring $P(N-2)$ multiplications and $(P+1)(N-2)$ additions. Since in a typical application we will have $P << N$ we see that the number of multiplications has been reduced from $O(N)$ to $O(1)$ and the number of additions remained approximately unchanged and is $O(N)$.

The achieved savings are indeed impressive in all cases where one multiplication is much more expensive than one addition. However, in cases where one multiplication is almost as expensive as one addition the described algorithm achieves a rather modest saving of about 50%.

In the next section we extend the results above to 2D and 3D discrete fields.

## 2.2 The 2D and 3D fast moment computation algorithm

To simplify notation in the sequel we will assume quadratic fields in the 2D and cubical fields in the 3D case. However, only minor modifications are required in the more general rectangular and cuboidal cases respectively.

The definition of the geometrical moments $m_{pq}$ of a 2D discrete field $f(n,l)$ is

$$m_{pq} = \sum_{n=0}^{N-1} \sum_{l=0}^{N-1} n^p l^q f(n,l) \ .$$

Since this is a separable mapping we see that if we denote with $\boldsymbol{F}$ the matrix made up by the samples of $f(n,l)$ at

$$(n,l) = \begin{matrix} (0,0) & \cdots & (0,N-1) \\ (1,0) & \cdots & (1,N-1) \\ \vdots & \vdots & \vdots \\ (N-1,0) & \cdots & (N-1,N-1) \end{matrix}$$

and set $\boldsymbol{F} = (\boldsymbol{f}_0, \boldsymbol{f}_1, \cdots, \boldsymbol{f}_{N-1})$ we obtain

$$m_{pq} = \sum_{n=0}^{N-1} n^p \sum_{l=0}^{N-1} l^q f(n,l) = \sum_{l=0}^{N-1} (\boldsymbol{n}^T)^p \boldsymbol{f}_l l^q = (\boldsymbol{n}^T)^p \boldsymbol{F} \boldsymbol{n}^q$$

and hence

$$\boldsymbol{m}_{PP} := \begin{pmatrix} m_{00} & m_{01} & \cdots & m_{0P} \\ m_{10} & m_{11} & \cdots & m_{1P} \\ \vdots & \vdots & \vdots & \vdots \\ m_{P0} & m_{P1} & \cdots & m_{PP} \end{pmatrix} = (\boldsymbol{n}^T)^{\boldsymbol{p}} \boldsymbol{F} \boldsymbol{n}^{\boldsymbol{p}^T} \ .$$

Since $(\boldsymbol{n}^T)^{\boldsymbol{p}}$ is $(P+1) \times N$ we see from the last expression that the complexity of the direct 2D algorithm is $(N+P+1)$ times the complexity of the 1D algorithm . Now, the fast 2D algorithm is obtained by using the factored form of $(\boldsymbol{n}^T)^{\boldsymbol{p}} = \boldsymbol{T}_P \boldsymbol{B}_N \boldsymbol{V}_N$ :

$$\boldsymbol{m}_{PP} = \boldsymbol{T}_P (\boldsymbol{B}_N \boldsymbol{V}_N \boldsymbol{F} \boldsymbol{V}_N \boldsymbol{B}_N^T) \boldsymbol{T}_P^T \ .$$

We observe that the part in parentheses of the expression above is $(P+1) \times (P+1)$ and may be computed by applying $(N+P+1)$ times multiplication by $\boldsymbol{B}_N$ on an $N$-vector. Thus, it requires $(P+1)(N-2)(N+P+1)$ additions only. It corresponds to the 2D cumulative moments $\boldsymbol{k}_{PP}$ of $\boldsymbol{F}$ . We define:

$$\boldsymbol{k}_{PP} := \boldsymbol{B}_N \boldsymbol{V}_N \boldsymbol{F} \boldsymbol{V}_N \boldsymbol{B}_N^T$$

and obtain

$$\boldsymbol{m}_{PP} = \boldsymbol{T}_P \boldsymbol{k}_{PP} \boldsymbol{T}_P^T \ . \tag{2}$$

Thus, $\boldsymbol{T}_P$ must be applied $2(P+1)$ times and this requires $P(P+1)^2$ multiplications and $P(P^2-1)$ additions yielding the following overall complexity for the fast 2D algorithm:

- number of multiplications $\#M = P(P+1)^2$
- number of additions $\#A = (P+1)(N-2)(N+P+1) + P(P^2-1)$ .

We observe that the $\#M$ of the fast 2D algorithm still does not depend on $N$ whereas the direct one requires $O(N^2)$ multiplications. This is the reason for obtaining impressive results under the assumptions made in Sec.(2.1.2.).

Concluding the exposition of the 2D moment computation algorithm we note that a moment *vector* $\boldsymbol{m}_Q$ containing all 2D moments $m_{pq}$ with the same order $p+q=Q$ arranged lexicographically may be traced from $\boldsymbol{m}_{PP}$ above along a diagonal line lying in the upper left part of $\boldsymbol{m}_{PP}$ if $Q \leq P$. This operation can be conveniently written with the aid of a one-zero sparse matrix $\boldsymbol{D}_Q$ selecting the moment values from vec($\boldsymbol{m}_{PP}$) which is a vector obtained from $\boldsymbol{m}_{PP}$ by concatenating its rows. With respect to (2) we then have

$$\mathrm{vec}(\boldsymbol{m}_{PP}) = (\boldsymbol{T}_P \otimes \boldsymbol{T}_P)\mathrm{vec}(\boldsymbol{k}_{PP})$$

where $\otimes$ denotes Kronecker product and

$$\boldsymbol{m}_Q = \boldsymbol{D}_Q \mathrm{vec}(\boldsymbol{m}_{PP}) = \boldsymbol{D}_Q(\boldsymbol{T}_P \otimes \boldsymbol{T}_P)\mathrm{vec}(\boldsymbol{k}_{PP}) \ .$$

Straightforward generalization of the concepts above to 3D leads to the 3D geometrical moments $M_{pqr}$ of a 3D field $f(n,l,m)$ given by

$$M_{pqr} = \sum_{n=0}^{N-1} n^p \sum_{l=0}^{N-1} l^q \sum_{m=0}^{N-1} m^r f(n,l,m) \ .$$

The three-dimensional field $f(n,l,m)$ may be conceived as a series of 2D layers $\boldsymbol{F}_m$, each with constant index $m$. We then have from above

$$M_{pqr} = \sum_{m=0}^{N-1} (\boldsymbol{n}^T)^p \boldsymbol{F}_m \boldsymbol{n}^q m^r \ .$$

The field containing the moments $M_{pqr}$ can also be conceived as a series of matrices $(\boldsymbol{M}_{PP})_r$ for $0 \leq r \leq P$ and this results in the expression

$$(\boldsymbol{M}_{PP})_r = \sum_{m=0}^{N-1} (\boldsymbol{n}^T)^{\boldsymbol{p}} \boldsymbol{F}_m \boldsymbol{n}^{\boldsymbol{p}^T} m^r \ .$$

Finally, the result of the fast 3D moment computation algorithm may be perhaps best written in terms of Kronecker products in the form

$$\mathrm{vec}\left(\mathrm{vec}((\boldsymbol{M}_{PP})_0), \mathrm{vec}((\boldsymbol{M}_{PP})_1), \cdots, \mathrm{vec}((\boldsymbol{M}_{PP})_P)\right) =: \mathrm{vec}(\boldsymbol{M}_{PPP}) = (\boldsymbol{T}_P \otimes \boldsymbol{T}_P \otimes \boldsymbol{T}_P)\mathrm{vec}(\boldsymbol{k}_{PPP}) \quad (3)$$

still requiring $O(1)$ multiplications in contrast to $O(N^3)$ multiplications of the direct algorithm.

# 3 3D Zernike polynomials and - moments

In this section we derive a set of polynomials in the three components $x, y$ and $z$ of $\boldsymbol{x} \in I\!R^3$ which is orthonormal and complete in the unit sphere. Besides, it exhibits a certain "form invariance" with respect to 3D rotations much like the well known 2D Zernike polynomials do in the plane. The motivation is to take advantage of the many useful properties the 2D Zernike polynomials and the associated 2D Zernike moments are known to enjoy, especially when compared to the ordinary geometrical moments. These properties are among others: Noise insensitivity, no information suppression at low radii and no redundancy, and are naturally expected to be valid also in the 3D case. Although we will not apply these polynomials in the present report, we would like to derive them in order to have some standard reference in the future.

If $\boldsymbol{x} = |\boldsymbol{x}|\boldsymbol{\xi} = r\boldsymbol{\xi} = r(\sin\vartheta\sin\phi, \sin\vartheta\cos\phi, \cos\phi)^T$ with $|\boldsymbol{x}| = r$ and $|\boldsymbol{\xi}| = 1$ we demand for a three-fold indexed member $Z_{nl}^m(\boldsymbol{x})$ of the 3D Zernike polynomials to be of the form:

$$Z_{nl}^m(\boldsymbol{x}) = R_{nl}(r) \cdot Y_l^m(\boldsymbol{\xi}) \ ,$$

where $Y_l^m(\boldsymbol{\xi})$ are complex valued spherical harmonics orthonormal on the surface of the unit sphere with $m$ ranging from $-l$ to $l$, $n-l$ is an even nonnegative integer, $n-l =: 2k$, and $R_{nl}(r)$ is the real factor depending on the radius $r$ we want to calculate so that the $Z_{nl}^m(\boldsymbol{x})$ become a set of polynomials orthonormal within the unit sphere. If we collect all $2l+1$ spherical harmonics $Y_l^m(\boldsymbol{\xi})$ with the same index $l$ in a vector $\boldsymbol{Y}_l(\boldsymbol{\xi}) = (Y_l^l(\boldsymbol{\xi}), Y_l^{l-1}(\boldsymbol{\xi}), Y_l^{l-2}(\boldsymbol{\xi}), \cdots, Y_l^{-l}(\boldsymbol{\xi}))^T$ and all 3D Zernike polynomials $Z_{nl}^m(\boldsymbol{x})$ with the same indices $n$ and $l$ in a vector $\boldsymbol{Z}_{nl}(\boldsymbol{x}) = (Z_{nl}^l(\boldsymbol{x}), Z_{nl}^{l-1}(\boldsymbol{x}), Z_{nl}^{l-2}(\boldsymbol{x}), \cdots, Z_{nl}^{-l}(\boldsymbol{x}))^T$, then with $\boldsymbol{P}$ a 3D rotation matrix we will have

$$\boldsymbol{Z}_{nl}(\boldsymbol{P}\boldsymbol{x}) = R_{nl}(r)\boldsymbol{Y}_l(\boldsymbol{P}\boldsymbol{\xi}) = R_{nl}(r)\boldsymbol{o}_l(\boldsymbol{P})\boldsymbol{Y}_l(\boldsymbol{\xi}) = \boldsymbol{o}_l(\boldsymbol{P})\boldsymbol{Z}_{nl}(\boldsymbol{x})$$

where $\boldsymbol{o}_l(\boldsymbol{P})$ is the $l$-th representation of the group $SO(3)$ being unitary [3]. The relation above is the advertised "form invariance" under rotation exhibited by the polynomials $Z_{nl}^m(\boldsymbol{x})$.

In what follows we will determine the factor $R_{nl}(r)$ . Since we know that $r^l Y_l^m(\boldsymbol{\xi}) =: e_l^m(\boldsymbol{x})$ is a homogeneous polynomial of $l$-th order in the components of $\boldsymbol{x}$, it follows from $Z_{nl}^m(\boldsymbol{x}) = \frac{R_{nl}(r)}{r^l} \cdot r^l Y_l^m(\boldsymbol{\xi}) = \frac{R_{nl}(r)}{r^l} \cdot e_l^m(\boldsymbol{x})$ that $\frac{R_{nl}(r)}{r^l}$ must be an $(n-l)/2 = k$-th degree polynomial in $r^2$ in order for $Z_{nl}^m(\boldsymbol{x})$ to be an $n$-th degree polynomial in the components of $\boldsymbol{x}$. We denote $\frac{R_{nl}(r)}{r^l}$ with $Q_{kl}(r^2)$ and obtain from the orthonormality condition of the 3D Zernike polynomials in the interior of the unit sphere

$$\frac{3}{4\pi}\int_0^1\int_0^{2\pi}\int_0^{\pi} Z_{nl}^m(\boldsymbol{x}) Z_{n'l'}^{m'}(\boldsymbol{x})^* \cdot r^2 \sin\vartheta \, d\vartheta d\phi dr = \delta_{nn'}\delta_{ll'}\delta^{mm'}$$

the condition for the polynomials $Q_{kl}$:

$$\frac{3}{2}\int_0^1 Q_{kl}(t) Q_{k'l}(t) t^{l+1/2} \, dt = \delta_{kk'} \ .$$

Thus, the $k$-th degree polynomials $Q_{kl}(t)$ may be obtained by orthogonalizing the monomials $1, t, t^2, \cdots$ with respect to the weighting factor $t^{l+1/2}$ for every $l$. Upon setting

$$Q_{kl}(t) = \sum_{\nu=0}^{k} q_{kl}^{\nu} t^{\nu}$$

we reduce the problem to the determination of the coefficients $q_{kl}^{\nu}$ for which the following equation results:

$$3 \sum_{\mu=0}^{k'} q_{k'l}^{\mu} \sum_{\nu=0}^{k} \frac{q_{kl}^{\nu}}{2(\mu+\nu+l)+3} = \delta_{kk'} \quad .$$

We suppose without restricting generality $k' \leq k$ and consider the equations above for all $k'$ starting from 0 to $k$. This gives the following system of equations:

$$\underbrace{\begin{pmatrix} \frac{1}{2l+3} & \frac{1}{2l+5} & \frac{1}{2l+7} & \cdots & \frac{1}{2l+2k+1} \\ \frac{1}{2l+5} & \frac{1}{2l+7} & \frac{1}{2l+9} & \cdots & \frac{1}{2l+2k+3} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{1}{2l+2k+1} & \frac{1}{2l+2k+3} & \frac{1}{2l+5} & \cdots & \frac{1}{2l+4k-1} \end{pmatrix}}_{=:\boldsymbol{A}_{kl}} \underbrace{\begin{pmatrix} q_{kl}^{0} \\ q_{kl}^{1} \\ \vdots \\ q_{kl}^{k-1} \end{pmatrix}}_{=:\boldsymbol{q}_{kl}} = -q_{kl}^{k} \underbrace{\begin{pmatrix} \frac{1}{2l+2k+3} \\ \frac{1}{2l+2k+5} \\ \vdots \\ \frac{1}{2l+4k+1} \end{pmatrix}}_{=:\boldsymbol{a}_{kl}} \quad \text{and}$$

$$q_{kl}^{k} \cdot \sum_{\nu=0}^{k} \frac{q_{kl}^{\nu}}{2(\nu+l+k)+3} = \frac{1}{3}$$

which may be written compactly with the notations introduced above:

$$\boldsymbol{A}_{kl} \boldsymbol{q}_{kl} = -q_{kl}^{k} \cdot \boldsymbol{a}_{kl} \qquad \text{and}$$

$$q_{kl}^{k} \cdot \left( \boldsymbol{a}_{kl}^{T} \boldsymbol{q}_{kl} + \frac{q_{kl}^{k}}{2l+4k+3} \right) = \frac{1}{3}$$

and gives the solution

$$\boldsymbol{q}_{kl} = -q_{kl}^{k} \boldsymbol{A}_{kl}^{-1} \boldsymbol{a}_{kl} \qquad \text{with}$$

$$q_{kl}^{k} = \frac{1}{\sqrt{3\left[\frac{1}{2l+4k+3} - \boldsymbol{a}_{kl}^{T} \boldsymbol{A}_{kl}^{-1} \boldsymbol{a}_{kl}\right]}} \quad .$$

Omitting the proof we also give the formula for the coefficients $q_{kl}^{\nu}$ in explicit form for all $0 \leq k,\ l$ and $0 \leq \nu \leq k$:

$$q_{kl}^{\nu} = \frac{(-1)^k}{2^{2k}} \sqrt{\frac{2l+4k+3}{3}} \binom{2k}{k} (-1)^{\nu} \frac{\binom{k}{\nu} \binom{2(k+l+\nu)+1}{2k}}{\binom{k+l+\nu}{k}} \quad .$$

This result may be proved by induction and verified for any specific case testing for orthonormality with respect to the weighting function $t^{l+1/2}$. Now with known coefficients $q_{kl}^{\nu}$ the 3D Zernike polynomials are obtained in the form

$$Z_{nl}^{m}(\boldsymbol{x}) = \sum_{\nu=0}^{k} q_{kl}^{\nu} |\boldsymbol{x}|^{2\nu} e_l^m(\boldsymbol{x})$$

where $e_l^m(\boldsymbol{x}) = |\boldsymbol{x}|^l Y_l^m(\boldsymbol{x})$ is an $l$-th order homogeneous polynomial in $\boldsymbol{x}$ that has been explicitly derived in [3] and can be written with known vector of coefficients $\boldsymbol{k}_l^m$ in the following form:

$$e_l^m(\boldsymbol{x}) = (\boldsymbol{k}_l^m)^T [\boldsymbol{x}]^l \quad .$$

In the expression above we have used the definition

$$[\boldsymbol{x}]^l = (x^l, x^{l-1}y, \cdots, z^l)^T \quad .$$

Thus, $[\boldsymbol{x}]^l$ is an $(l+1)(l+2)/2$ - dimensional vector containing all monomials of $l$-th order in the three variables $x$, $y$ and $z$ arranged lexicographically. With the additional definition

$$|\boldsymbol{x}|^{n-l}[\boldsymbol{x}]^l = |\boldsymbol{x}|^{2k}[\boldsymbol{x}]^l =: \boldsymbol{S}_{ln}[\boldsymbol{x}]^n$$

for a one-zero sparse matrix $\boldsymbol{S}$ selecting from the vector $[\boldsymbol{x}]^n$ containing all monomials of $l$-th order in $\boldsymbol{x}$, the lower dimensinal vector $|\boldsymbol{x}|^l[\boldsymbol{x}]^{n-l}$ we are now in a position to give explicitly the connection between the 3D Zernike moments

$$\Omega_{nl}^m := \frac{3}{4\pi} \int\limits_{|\boldsymbol{x}|\leq 1} f(\boldsymbol{x}) Z_{nl}^m(\boldsymbol{x})^* \, d\boldsymbol{x}$$

and the geometrical moments of a 3D object $f(\boldsymbol{x})$ given by the vectors

$$\boldsymbol{M}_l = \int\limits_{|\boldsymbol{x}|\leq 1} f(\boldsymbol{x})[\boldsymbol{x}]^l \, d\boldsymbol{x} \quad :$$

$$\Omega_{nl}^m = \frac{3}{4\pi} (\boldsymbol{k}_l^m)^{*T} \sum_{\nu=0}^{k} q_{kl}^{\nu} \boldsymbol{S}_{l,l+2\nu} \boldsymbol{M}_{l+2\nu} \quad . \tag{4}$$

We know from the Weierstraß approximation theorem that any continuous function with compact region of support may be uniformly approximated by polynomials. Since we have derived a complete set of orthonormal polynomials $Z_{nl}^m(\boldsymbol{x})$ in the unit sphere and used them to build the 3D Zernike moments $\Omega_{nl}^m$, the latter are essentially generalized Fourier coefficients of $f(\boldsymbol{x})$ and if we suppose $f(\boldsymbol{x})$ continuous with region of support the interior of the unit sphere we also get a Fourier series expansion for $f(\boldsymbol{x})$:

$$f(\boldsymbol{x}) = \sum_{n=0}^{\infty} \sum_{k=0}^{\lfloor n/2 \rfloor} \sum_{\nu=0}^{k} q_{kl}^{\nu} |\boldsymbol{x}|^{2\nu} \sum_{m=-l}^{l} \Omega_{nl}^{m} e_{l}^{m}(\boldsymbol{x}) \quad \text{with} \quad l = n - 2k \ .$$

This formula may be used to approximately reconstruct the shape of a general 3D object from a small number of moments of low degree simply by considering only $0 \leq n \leq N$ for some low $N$.

Now returning to the fast computation of the 3D Zernike moments we only have to introduce like in the 2D case a one-zero sparse matrix $\boldsymbol{D}_Q$ that selects from $\text{vec}(\boldsymbol{M}_{PPP})$ the vector $\boldsymbol{M}_Q$ consisting of all 3D moments of order $Q$ arranged lexicographically by defining

$$\boldsymbol{M}_Q = \boldsymbol{D}_Q \text{vec}(\boldsymbol{M}_{PPP}) \ .$$

Combining with (3) and (4) we finally get

$$\Omega_{nl}^{m} = \frac{3}{4\pi} (\boldsymbol{k}_l^m)^{*T} \sum_{\nu=0}^{k} q_{kl}^{\nu} \boldsymbol{S}_{l,l+2\nu} \boldsymbol{D}_{l+2\nu} (\boldsymbol{T}_P \otimes \boldsymbol{T}_P \otimes \boldsymbol{T}_P) \text{vec}(\boldsymbol{k}_{PPP})$$

and we see that we can absorb the constant matrices $\boldsymbol{T}_P$ in the coefficients $q_{kl}^{\nu}$ thus getting the 3D Zernike moments $\Omega_{nl}^{m}$ directly from the cumulative moments $\boldsymbol{k}_{PPP}$ without the need to first compute geometrical moments. As already pointed out, this results in increased numerical precision.

Concluding, we note that since the complete moment invariants for orthogonal transformations of 3D objects we had derived in [3] were based on spherical harmonics and were nothing more than spherical moments of the object in its standard position, it is plain that this set of invariants is readily rebuilt to give the 3D Zernike moments of the object in the standard position. In fact, only the factor $R_{nl}$ has to be taken in addition into consideration. The expansion formula then computes from the invariants the object in the standard position.

# References


[1] Y.S. Abu-Mostafa and D. Psaltis "Recognitive aspects of moment invariants" *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 698-706, November 1984.

[2] M. Born and E. Wolf *Principles of Optics* New York: Pergamon, 1975.

[3] N. Canterakis " Complete Moment Invariants and Pose Determination for Orthogonal Transformations of 3D Objects" *In: B. Jähne, P. Geißler, H. Haußecker, and F. Hering (eds.): Mustererkennung 1996 (18. DAGM Symposium)* Informatik aktuell, Springer 1996, pp.339-350.

[4] H. Dym, H.P. McKean *Fourier Series and Integrals* Academic Press, 1972.

[5] M. Hatamian, " A Real-Time Two-Dimensional Moment Generating Algorithm and its Single Chip Implementation" *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. ASSP-34, pp.546-553, June 1986.

[6] A.Khotanzad and Y.H. Hong "Invariant image recognition by Zernike moments" *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 12, no. 5, pp. 489-497, 1990.

[7] C.-H. Lo, and H.-S. Don, "3D Moment forms: Their construction and application to object identification and positioning" *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 11, no. 10, pp. 1053-1064, October 1989.

[8] C.-H. Teh and R.T. Chin "On Image Analysis by the Method of Moments" *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 10, pp. 496-513, July 1988.