Available online at www.pelagiaresearchlibrary.com

Pelagia Research Library

Advances in Applied Science Research, 2012, 3 (1):583-590

Pelagia Research Library
ISSN: 0976-8610
CODEN (USA): AASRFC

Study of Zernike moments using analytical Zernike polynomials

Sundus Y. Hasan

Physics dep., Educational College for Women, Kufa University, Iraq

---

ABSTRACT

In this research, the analytical Zernike polynomials written in Cartesian coordinates and in polar coordinates were used to calculate the Zernike moments instead of the numerical complex Zernike polynomials used before. The results show that there is no difference between them except in time calculating where this method takes more time.

INTRODUCTION

Moments are widely used in pattern recognition, image processing, computer vision and multi-resolution analysis. Teague[1] proposed Zernike moments based on the basis set of orthogonal Zernike polynomials. It is well known that a discrete image function can be reconstructed by Zernike moments[2]. Particularly, the Zernike moments have been shown to be rotation invariance and noise robust. A relatively small set of Zernike moments can characterize the global shape of a pattern effectively[3]. The low order moments represent the global shape of a pattern and the higher order the detail.

When the Zernike moments of the image are suitable for a large number of terms, then the reconstruction of the input image function can be achieved with high accuracy. The reconstruction depends on the number of Zernike polynomials involved[4].

In 1997 Simon X. Liao and Miroslaw Pawlak carried out the geometrical error and numerical error analyses for the Zernike moment computing [5]. And in the same year R. Mukundan presents a fast algorithm for the computation of Zernike moments of a binary image. The Zernike moment integrals are evaluated along the object boundary points using a discrete version of the Green's theorem [6]. In 2003 Heloise Hse and A. Richard Newton present an on-line method for hand sketched symbols using Zernike moments descriptors to represent symbols[7]. In this year also A. Padilla-Vivanco et. al. reconstructed discrete image functions using the complex Zernike polynomials. They compare the reconstruction of the input image function in two cases. When the input image is mapped inside or outside a unit circle for several expansion orders[4].

So, in all of these researches, the researchers were used the complex Zernike moments in the processing. In this research, the analytical forms of circular Zernike polynomials that were written in the form of Cartesian and polar coordinates have been used.

Zernike moments [8]

Zernike moments use the complex Zernike polynomials as the moment basis set. The 2D Zernike moments, Z_{nm}, of order n with repetition m, are defined in polar coordinates (r, θ) inside the unit circle as [9]

$$Z_{nm} = \frac{n+1}{\pi}\int_0^1\int_0^{2\pi} R_{nm}(r)e^{-jm\theta} f(r,\theta)rdrd\theta, \qquad 0 \le |m| \le n, \quad n-|m| \text{ is even.}$$

where $R_{nm}(r)$ is the nth order of Zernike radial polynomial given by

$$R_{nm}(r) = \sum_{k=0}^{(n-|m|)/2} (-1)^k \frac{(n-k)!}{k!\left[(n-2k+|m|)/2\right]!\left[(n-2k-|m|)/2\right]!} r^{n-2k}$$

Like the rotational moments and the complex moments, the magnitude of the Zernike moments is invariant under image rotation transformation. The image can be reconstructed using a set of moments through order M as

$$f(r,\theta) \approx \sum_{n=0}^{M}\sum_{m} Z_{nm}R_{nm}(r)e^{jm\theta}.$$

In this research, Z_{nm} does not take that form but a form of a summation of a number of polynomials taken from reference (9) were adopted.

RESULTS AND DISCUSSION

In this paper, a code written by Christian Wolf [10] in matlab program has been depended (see appendix I) which is written to calculate complex Zernike polynomials and complex Zernike moments and also a code to reconstruct image. In this research, this code was changed to be suitable to calculate Zernike moments and then to reconstruct image using analytical Zernike polynomials instead of the complex one. A group of Zernike polynomials (65 polynomials or n=10) were feed to the function Zernike_bf (appendix I, b) twice. First the form written in polar coordinates was used as in appendix II,a, and then the form written in Cartesian coordinates was used as in appendix II,b.

The image chosen in this research is that used by Christian Wolf (see figure (1)).

Fig. (1) Image used for analysis[5].

Fig (2) shows the original image (to the upper right) and reconstructed image using Christian Wolf code with different values of n[5].

Fig. (2) image reconstruction computed by Christian Wolf with n=10,20,30, 40, 45[5].

Figure (3) shows the reconstructed image using the analytical Zernike polynomials (in its two forms) compared with that of complex Zernike polynomials with n=4,6,8, and 10. Where the maximum order (n) we got for analytical Zernike polynomials is 10 (i.e. 65 polynomials). It is obvious that there is no difference between them which is a clue that the program is correct.

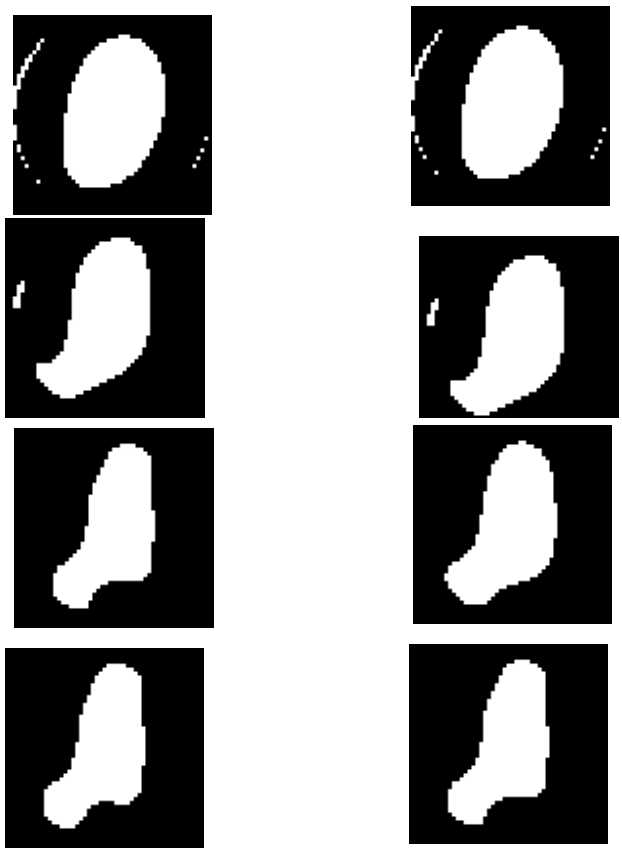

Fig. 3 Acomparison of reconstructing image using complex Zernike moments (left) and analytical polar and Cartesian Zernike moments (right).

number	n	m	Analytical Zernike moments	Analytical Zernike moments	Complex Zernike moment
1	0	0	359.00	359.00	359.00
2	1	-1	-14.70	-14.70	-17.33 - 10.40i
3	1	1	-24.51	-24.51	-17.33 + 10.40i
4	2	-2	-65.31	-65.31	-58.21 - 46.18i
5	2	0	-400.30	-400.30	-400.30
6	2	2	-82.32	-82.32	-58.21 + 46.18i
7	3	-3	-19.80	-19.80	-10.54 - 14.00i
8	3	-1	-3.27	-3.27	062.69 - 2.31i
9	3	1	88.66	88.66	62.69 + 2.31i
10	3	3	-14.91	-14.91	-10.54 + 14.00i
11	4	-4	41.52	41.52	-06.30 + 29.36i
12	4	-2	88.84	88.84	111.19 + 62.82i
13	4	0	241.70	241.70	241.70
14	4	2	157.25	157.25	111.19 - 62.82i
15	4	4	-8.91	-8.91	-6.30 - 29.36i

Table (1) shows the values of first ten moments which appears in complex form when using the complex form of Zernike polynomials (Christian code) and in real form when an analytical form (Cartesian and polar form) is used. And it is clear that the results are the same. Note when Zernike polynomial is r- dependent only (i.e. does not depend on θ) the value of moment using analytical Zernike equal to that used complex Zernike (look at the first and the fifth and the thirteenth values in table (1)).

This method of computing moments using the analytical forms of Zernike polynomials is important because it is a way to calculate moments for object using apertures other than circle which there is only an analytical form of Zernike polynomials and could not write its complex form.

CONCLUSION

- Zernike moments can be calculated using analytical forms of Zernike polynomials instead of complex Zernike polynomials with no difference in results till n=10.
- Because the previous obtained number of polynomials in analytical form is limited (here n=10), the reconstructed image by complex method is better because it can take more n values (in figure 2 n reaches 45).
- The time calculating moments and reconstructing image is longer in analytical method.
- Analytical Zernike polynomials are available for many forms of apertures but the complex form is not, so Zernike moments can be calculated using this method and many features of the image can be calculated (which is a project for afuture work).

REFERENCES

[1] M. R. Teague, *J. Opt. Soc. Am.* **1980**, *70*, 920 .
[2] C. Chong, P. Raveendran and R. Mukundan, *Pattern Rec*. **2003**, 36, 731 .
[3] W. Kim and Y. Kim, Signal Processing: Image Communication, **2000**, 16, 95 .
[4] A. Padilla-Vivanco A. Martinez-Ramirez and F. Granados-Agustin, SPIE USE, **2003**, 1, 5237
[5] Simon X.Liao and Miroslaw Pawlak IEEE, **1997**, 700 .
[6] R. Mukundan, National Conference on Research and Development in Computer Science and its Applications, 27-29 Nov. **1997,** School of Computer Sciences, Universiti Sains Malaysia, Penang, Malaysia
[7] Heloise Hse and A. Richard, **2003**, Barkeley, CA 94720, U.S.A..
[8] Huazhong Shu1, Limin Luo1, Jean Louis Coatrieux , *IEEE Engineering in Medicine and Biology Magazine* **2007**, 26, 70
[9] "Periodic Table of the Disc Polynomials of Zernike " Advanced Medical Optics/WaveFront Sciences · 14820 Central Ave. SE · Albuquerque, NM 87123-3905· http://www.wavefrontsciences.com
[10] Christian Wolf, http://liris.cnrs.fr/christian.wolf/index.html.

Appendix I: Christian Wolf mathlab code and the code for finding complex Zernike polynomials. Please return to the site http://liris.chrs.fr/christian.wolf) to get the rest codes, i.e. complex Zernike moment code and reconstructing binary image.[10]

a) Base program

```
% ********************************************************************
% Christian Wolf, http://liris.chrs.fr/christian.wolf
global DEB0 DEB1;
DEB0=[];
DEB1=[];
imsize=51;
load runningman.mat
size(runningman)
m=runningman(11:11+imsize-1,11:11+imsize-1);
size(m)
m=centersquare(m,imsize);
m=uint8(m);
whos
zs1=zernike_bf(imsize,4,1);
```

```
zs0=zernike_bf(imsize,4,0);
v0=zernike_mom(m,zs0);
v1=zernike_mom(m,zs1);
rec0=zernike_rec(v0,imsize,zs0);
rec1=zernike_rec(v1,imsize,zs1);
fprintf ('Difference: %f\n',sum(sum(rec0-rec1)));
```

b) Function for calculating complex Zernike polynomials.

```
function ZBFSTR=zernike_bf(SZ,ORDER,WITHNEG)
if nargin<3
WITHNEG=0;
end
limitfastcomp=50;
F=factorial(0:ORDER);
pq=zernike_orderlist(ORDER, WITHNEG);
len=size(pq,1);
szh=SZ/2;
pqind=-1*ones(1+2*ORDER,1+2*ORDER);
src=1+ORDER+pq;
pqind(sub2ind(size(pqind),src(:,1),src(:,2)))=(1:len)';
Rmns=zeros(1+2*ORDER,1+2*ORDER+1,1+2*ORDER);
for flat=1:min(len,limitfastcomp);
m=pq(flat,1);
n=pq(flat,2);
mpnh=floor((m+abs(n))/2);
mmnh=floor((m-abs(n))/2);
for s=0:mmnh
Rmns(1+ORDER+m,1+ORDER+n,1+s)=(((-1)^s)*F(1+m-s))/...
(F(1+s)*F(1+mpnh-s)*F(1+mmnh-s));
end
end
for flat=limitfastcomp+1:len
m=pq(flat,1);
n=pq(flat,2);
mpnh=floor((m+abs(n))/2);
mmnh=floor((m-abs(n))/2);
for s=0:mmnh
Rmns(1+ORDER+m,1+ORDER+n,1+s)=((-1)^s)*robust_fact_quot(...
1:m-s,...
[ 1:s 1:mpnh-s 1:mmnh-s ]);
end
end
ZBF=zeros(SZ,SZ,len);
for y=1:SZ
for x=1:SZ
% ---- Take cartesian coordinates with the origin in the
% ---- middle of the image and transform into polar
rho=sqrt((szh-x)^2+(szh-y)^2);
theta=atan2(szh-y,szh-x);
if rho>szh
continue
end
rho=rho/szh;
if theta<0
theta=theta+2*pi;
end
```

```
for flat=1:len
m=pq(flat,1);
n=pq(flat,2);
R=0;
for s=0:(m-abs(n))/2
R=R+Rmns(1+ORDER+m,1+ORDER+n,1+s)*(rho^(m-2*s));
end
ZBF(y,x,flat) = R*exp(n*theta*1j);
end
end
end
% ---- Package the whole thing into a structure
ZBFSTR=struct;
ZBFSTR.maxorder=ORDER;
ZBFSTR.withneg=WITHNEG;
ZBFSTR.orders=pq;
ZBFSTR.index=pqind;
ZBFSTR.bf=ZBF;
% ********************************************************************
```

Appendix II

a) Function for calculating values of the analytical Zernike polynomials in polar form.

```
function ZBFSTR=zernike1_bf(SZ,ORDER,WITHNEG)
syms r q
G11=[1 r*sin(q) r*cos(q) r^2*sin(2*q)  2*r^2-1  r^2*cos(2*q)  r^3*sin(3*q)
(3*r^3-2*r)*sin(q)  (3*r^3-2*r)*cos(q)...
r^3*cos(3*q)  r^4*sin(4*q)...% 11
(4*r^4-3*r^2)*sin(2*q)  6*r^4-6*r^2 + 1  (4*r^4-3*r^2)*cos(2*q) r^4*cos(4*q)
r^5*sin(5*q) (5*r^5-4*r^3)*sin(3*q) ...
(10*r^5-12*r^3 + 3*r)*sin(q)  (10*r^5-12*r^3 + 3*r)*cos(q)...%19
(5*r^5-4*r^3)*cos(3*q)  r^5*cos(5*q)  r^6*sin(6*q)  (6*r^6-5*r^4)*sin(4*q)
(15*r^6-20*r^4 + 6*r^2)*sin(2*q) 20*r^6-30*r^4 + 12*r^2-1 ...
(15*r^6-20*r^4 + 6*r^2)*cos(2*q) (6*r^6-5*r^4)*cos(4*q) r^6*cos(6*q) ... %6
r^7*sin(7*q)  (7*r^7-6*r^5)*sin(5*q)  (21*r^7-30*r^5  +  10*r^3)*sin(3*q)
(35*r^7-60*r^5 + 30*r^3-4*r)*sin(q)...
(35*r^7-60*r^5  +  30*r^3-4*r)*cos(q)  (21*r^7-30*r^5+10*r^3)*cos(3*q)...%
CHANGE 2 TO 21
(7*r^7-6*r^5)*cos(5*q) r^7*cos(7*q) ...% 7
r^8*sin(8*q) ...
(8*r^8-7*r^6)*sin(6*q) (28*r^8-42*r^6 + 15*r^4)*sin(4*q)...
(56*r^8-105*r^6+60*r^4-10*r^2)*sin(2*q) ...
70*r^8-140*r^6+90*r^4-20*r^2+1 ...
(56*r^8-105*r^6+60*r^4-10*r^2)*cos(2*q) ...
(28*r^8-42*r^6+  15*r^4)*cos(4*q)  (8*r^8-7*r^6)*cos(6*q)  r^8*cos(8*q)
r^9*sin(9*q) (9*r^9-8*r^7)*sin(7*q) (36*r^9-56*r^7+21*r^5)*sin(5*q)...
(84*r^9-168*r^7+105*r^5-20*r^3)*sin(3*q)  (126*r^9-280*r^7+210*r^5  -
60*r^3+5*r)*sin(q)  (126*r^9-280*r^7 + 210*r^5-60*r^3+5*r)*cos(q)...
(84*r^9-168*r^7 + 105*r^5-20*r^3)*cos(3*q) (36*r^9-56*r^7 + 21*r^5)*cos(5*q)
(9*r^9-8*r^7)*cos(7*q)  r^9*cos(9*q)  r^10*sin(10*q)  (10*r^10-9*r^8)*sin(8*q)
(45*r^10-72*r^8 + 28*r^6)*sin(6*q)...
(120*r^10-252*r^8 + 168*r^6-35*r^4)*sin(4*q)...
(210*r^10-504*r^8 + 420*r^6-140*r^4 + 15*r^2)*sin(2*q)...
252*r^10-630*r^8 + 560*r^6-210*r^4 + 30*r^2-1 ...
(210*r^10-504*r^8 + 420*r^6-140*r^4 + 15*r^2)*cos(2*q)...
(120*r^10-252*r^8 + 168*r^6-35*r^4)*cos(4*q) ...
(45*r^10-72*r^8 + 28*r^6)*cos(6*q) (10*r^10-9*r^8)*cos(8*q)...
r^10*cos(10*q)];
```

```
GG=[1 2 2 6^.5 3^.5 6^.5 8^.5 8^.5 8^.5 8^.5 10^.5 10^.5 5^.5 ...
10^.5 10^.5 12^.5 12^.5 12^.5 12^.5 12^.5 12^.5 14^.5 14^.5 ...
14^.5 7^.5 14^.5 14^.5 14^.5 4 4 4 4 4 4 4 12^.5 12^.5 12^.5 12^.5...
3 12^.5 12^.5 12^.5 12^.5 20^.5 20^.5 20^.5 20^.5 20^.5 20^.5 20^.5 ...
20^.5 20^.5 20^.5 22^.5 22^.5 22^.5 22^.5 22^.5 11^.5 ...
22^.5 22^.5 22^.5 22^.5 22^.5];
ss=size(G11);
ss1=size(GG);
G11=G11;
pq=zernike_orderlist(ORDER, WITHNEG);
pq
len=size(pq,1);
szh=SZ/2;
src=1+ORDER+pq    ;
Rmns=zeros(1+2*ORDER,1+2*ORDER+1,1+2*ORDER);

ZBF=zeros(SZ,SZ,len);
for y=1:SZ
for x=1:SZ
r=sqrt((szh-x)^2+(szh-y)^2);
q=atan2(szh-y,szh-x);
if r>szh
continue
end
r=r/szh;
ii=0;
for flat=1:len
m=pq(flat,1);
n=pq(flat,2);

ii=ii+1;
R=eval (G11(ii));
ZBF(y,x,flat) = R;
end
end
end
ssss=size(ZBF)
ZBFSTR=struct;
ZBFSTR.maxorder=ORDER;
ZBFSTR.withneg=WITHNEG;
ZBFSTR.orders=pq;
ZBFSTR.bf=ZBF;
```

b) Function for calculating values of the analytical Zernike polynomials in Cartizian form (here function G11 must be continued from the table in reference [9]).

```
function ZBFSTR=zernike1_bf(SZ,ORDER,WITHNEG)
syms x y  q
r=(x^2+y^2)^.5
G11=[1 x y 2*x*y  2*r^2-1  -x^2+y^2 ...
-x^3+3*x*y^2 -2*x+3*x^3+3*x*y^2  -2*y+3*y^3+3*x^2*y ...
y^3-3*x^2*y …]
GG=[1 2 2 6^.5 3^.5 8^.5 8^.5 8^.5 8^.5 8^.5]
ss=size(G11);
ss1=size(GG);
pq=zernike_orderlist(ORDER, WITHNEG);
len=size(pq,1);
```

```
szh=SZ/2;
src=1+ORDER+pq    ;
Rmns=zeros(1+2*ORDER,1+2*ORDER+1,1+2*ORDER);
ZBF=zeros(SZ,SZ,len);
for yy=1:SZ
for xx=1:SZ
x=szh-xx;
y=szh-yy;
if (x^2+y^2)^.5>szh
continue
end
x=x/szh;
y=y/szh;
ii=0;
for flat=1:len
m=pq(flat,1);
n=pq(flat,2);
ii=ii+1;
R=eval (G11(ii));
ZBF(yy,xx,flat) = R;
end
end
end
ssss=size(ZBF)
ZBFSTR=struct;
ZBFSTR.maxorder=ORDER;
ZBFSTR.withneg=WITHNEG;
ZBFSTR.orders=pq;
ZBFSTR.bf=ZBF;
```