The Zernike Polynomials

Joshua Cook

10 December 2014

A program in Python to generate, evaluate, and visualize Zernike polynomials, a family of orthogonal polynomials over the unit disk, $D = \{(\rho, \theta) | 0 \leq \rho \leq 1, 0 \leq \theta \leq 2\pi\}$ and discussions and proofs of some of their properties. Zernike polynomials are used to describe aberrations in a lens (e.g., the cornea).

Description

Even and odd Zernike Polynomials are defined respectively as

$$Z_n^m(\rho, \theta) = R_m^n(\rho)\cos(m\theta) \quad \text{and} \quad Z_n^{-m}(\rho, \theta) = R_m^n(\rho)\sin(m\theta) \tag{1}$$

where $n, m \in \mathbb{Z}, n \geq m \geq 0$, θ is the *azimuthal angle*, ρ the radial distance, and $R_n^m(\rho)$, the radial polynomials given by

$$R_n^m(\rho) = \sum_{k=0}^{\frac{n-m}{2}} \frac{(-1)^k\,(n-k)!}{k!\,\left(\frac{n+m}{2}-k\right)!\,\left(\frac{n-m}{2}-k\right)!}\,\rho^{n-2\,k} \tag{2}$$

Rewriting the ratios of factorials in the radial part as products of Binomial coefficient shows that the coefficients are integer numbers:

$$R_n^m(\rho) = \sum_{k=0}^{\frac{n-m}{2}} (-1)^k \binom{n-k}{k} \binom{n-2k}{\frac{n-m}{2}-k} \rho^{n-2k} \tag{3}$$

They satisfy $|Z_n^m(\rho, \theta)| \leq 1|$ for all $\rho \in (0, 1)$ and all θ.

Properties of the Zernike Polynomials

Generating the Polynomials

Toward a Recursive Method, we can find a recursive formula to compute Zernike radial polynomials[1]. The following recurrence relationship has been previously proposed[2]

$$R_n^m(\rho) = \frac{1}{K_1}\left[(K_2\rho^2 + K_3)R_{n\ 2}^{m}(\rho) + K_4 R_{n\ 4}^{m}(\rho)\right] \tag{4}$$

for

$$n = m + 4, m + 6, \ldots$$

[1] Barmak Honarvar Shakibaei and Raveendran Paramesran. Recursive formula to compute zernike radial polynomials. *Optics letters*, 38(14): 2487–2489, 2013

[2] Eric C Kintner. On the mathematical properties of the zernike polynomials. *Journal of Modern Optics*, 23(8):679–680, 1976

where the coefficients are given by[3]

$$K_1 = \frac{(n+m)(n-m)(n-2)}{2}$$
$$K_2 = 2n(n-1)(n-2)$$
$$K_3 = -m^2(n-1) - n(n-1)(n-2)$$
$$K_4 = -\frac{n(n+m-2)(n-m-2)}{2}$$

[3] Augustus JEM Janssen and Peter Dirksen. Computing zernike polynomials of arbitrary degree using the discrete fourier transform. *Journal of the European Optical Society-Rapid publications*, 2, 2007

To initialize the method we specify cases in which $n = m$ and $n - m = 2$:[4]

[4] Chee-Way Chong, P Raveendran, and R Mukundan. A comparative analysis of algorithms for fast computation of zernike moments. *Pattern Recognition*, 36(3):731–742, 2003

$$R_m^m(\rho) = \rho^m \tag{5}$$

$$R_{m+2}^m(\rho) = (m+2)\rho^{m+2} - (m+1)r^m \tag{6}$$

Taking a three-term recurrence relation previously derived[5]

[5] Aluizio Prata Jr and WVT Rusch. Algorithm for computation of zernike polynomials expansion coefficients. *Applied Optics*, 28(4):749–754, 1989

$$R_n^m(\rho) = \rho L_1 R_{n-1}^{|m-1|}(\rho) + L_2 R_{n-2}^m(\rho) \tag{7}$$

where the coefficients L_1 and L_2 are

$$L_1 = \frac{2n}{m+n}$$
$$L_2 = \frac{m-n}{m+n}$$

We then start with $R_0^0 = 1$ and can generate all the polynomials.

a Recurrence Relation

Janssen and Dirksen have shown the following integral representation of radial polynomials[6]

[6] Augustus JEM Janssen and Peter Dirksen. Computing zernike polynomials of arbitrary degree using the discrete fourier transform. *Journal of the European Optical Society-Rapid publications*, 2, 2007

$$R_n^m(\rho) = \frac{1}{2\pi}\int_{k=0}^{2\pi} U_n \rho \cos\theta \cos\theta \tag{8}$$

where U_n is the Chebyshev polynomial of the second kind and of degree n that satisfies the following[7] recurrence relation for $n \geq 2$

[7] Barmak Honarvar Shakibaei and Raveendran Paramesran. Recursive formula to compute zernike radial polynomials. *Optics letters*, 38(14): 2487–2489, 2013

$$U_n(x) = 2xU_{n-1}(x) - U_{n-2}(x) \tag{9}$$

with $U_0(x) = 1$ and $U_1(x) = 2x$.

Clearly, the degree of the radial polynomials is equal to the degree of the Chebshev polynomials of the second kind.

Furthermore, the azimuthal order equals the frequency of the cosine function.

Therefore, simply set $x = \rho\cos\theta$ and multiply by $\cos(m\theta)$ to get

$$U_n(\rho\cos\theta)\cos(m\theta) = 2\rho\cos\theta U_{n-1}(\rho\cos\theta)\cos(m\theta) - U_{n-2}(\rho\cos\theta)\cos(m\theta)$$
$$U_n(\rho\cos\theta)\cos(m\theta) = \rho\Big[\cos(|m-1|\theta) + \cos((m+1)\theta)\Big]U_{n-1}(\rho\cos\theta)cos(m\theta) - U_{n-2}(\rho\cos\theta)\cos(m\theta)$$
$$\int_0^{2\pi} U_n(\rho\cos\theta)\cos(m\theta)\,d\theta =$$
$$\int_0^{2\pi}\left(\rho\Big[\cos(|m-1|\theta) + \cos((m+1)\theta)\Big]U_{n-1}(\rho\cos\theta)cos(m\theta) - U_{n-2}(\rho\cos\theta)\cos(m\theta)\right)d\theta$$

Using the integral representation of radial polynomials, we have the following recurrence relation

$$R_n^m(\rho) = \rho\left[R_{n-1}^{|m-1|}(\rho) + R_{n-1}^{m+1}(\rho)\right] - R_{n-2}^m(\rho) \tag{10}$$

or

$$R_n^m(\rho) + R_{n-2}^m(\rho) = \rho\left[R_{n-1}^{|m-1|}(\rho) + R_{n-1}^{m+1}(\rho)\right] \tag{11}$$

Orthogonality

Two functions F_1 and F_2 are orthogonal over a unit cirle if:

$$\int_0^1\int_0^{2\pi} F_1F_2\rho d\theta d\rho = 0$$

For axisymmetric functions with no θ variance, this reduces to

$$2\pi\int_0^1 F_1F_2\rho d\rho = 0$$

Consider $F_1 = \rho^2$ and $F_2 = \rho^4$.

$$2\pi\int_0^1 \rho^2\rho^4\rho d\rho = \frac{\pi}{4} \neq 0$$

Consider $F_3 = 2\rho^2 - 1$ and $F_4 = 6\rho^4 - 6\rho^2 + 1$.

$$2\pi\int_0^1 (2\rho^2-2)(6\rho^4-6\rho^2+1)\rho d\rho = 0$$

These are Zernike polynomials. In fact, the Zernike polynomials are created by subtracting lower order polynomials to create this orthogonality relationship.

We seek:

$$\int_0^1 \int_0^{2\pi} R_i(\rho)\Theta_i(\theta)R_j(\rho)\Theta_j(\theta)\rho d\theta d\rho = \delta_j = \begin{cases} C_j & i = j \\ 0 & i \neq j \end{cases}$$

$$\int_0^1 R_i R_j \rho d\rho \int_0^{2\pi} \Theta_i \Theta_j d\theta = \delta_j$$

For a 2π-periodic function, we can chose an orthogonal basis (in fact, orthonormal) in the trigonometric functions $\sin n\theta$ and $\cos n\theta$. Therefore, our problem is reduced to finding an orthogonal radial basis.

$$\int_0^1 R_i R_j \rho d\rho = \delta_j$$

The Program

THE PROGRAM CONSISTS of two files, a View file (`menu.py`) and a Controller file (`notebook.py`).

Controller File

`notebook.py` consists of two classes `Zernike_Polynomial` and `Polynomial_Notebook`. The first is an object representing a specific polynomial, where as the second is an object containing multiple polynomial instances.

Instantiaton of a Notebook Each `Polynomial_Notebook` has several notebook level variables and lists. These include:

`self.polynomials` A list containing polynomials created.

`self.density` The density to be used for the plotting of polynomials.

`self.rho` A numpy array containing possible radial values.

`self.theta` A numpy array containing possible angular values.

`self.Rho` A meshgrid object associated with radial values.

`self.Theta` A meshgrid object associated with angular values.

Instantiation of a Polynomial Calling `new_polynomial(m,n)` initializes a new instance of a Zernike polynomial with the passed m and n values. It does so by calling the following methods from the `Zernike_Polynomial` class.

```
def __init__(self, n,m):
    '''Initialize a polynomial with m, n values.'''
    self.creation_date = datetime.date.today()
    self.m = m
    self.n = n
    self.id = last_id
    self.coeffs = np.zeros(self.n+1)

    global last_id
    last_id += 1
```

Figure 1: The polynomial initialization method.

`zernike_Rcoeffs()` Uses the binomial relationship defined in eq. 3 to generate the coefficients of the indicial degree.

`zernike_even()` Generates a mesh-grid, `Z_even`, representing the multi-variate function $Z_n^m(\rho, \theta)$ on the domain $[0, 1] \times [0, 2\pi]$ using the coefficients generated by `zernike_Rcoeffs()`.

`zernike_odd()` Generates a mesh-grid, `Z_odd`, representing the multi-variate function $Z_n^{-m}(\rho, \theta)$ on the domain $[0, 1] \times [0, 2\pi]$ using the coefficients generated by `zernike_Rcoeffs()`.

```
def new_polynomial(self,m,n):
    new_poly = Zernike_Polynomial(m,n)
    new_poly.zernike_Rcoeffs()
    new_poly.zernike_even(self.Rho,self.Theta,self.density)
    new_poly.zernike_odd(self.Rho,self.Theta,self.density)
    self.polynomials.append(new_poly)
```

Figure 2: Method creating a new polynomial instance.

Preparation of Plots Plots are generated using the `matplotlib` commands `pcolormesh` and `plot_surface`. The meshgrid objects `notebook.Rho`, `notebook.Theta`, `poly.Z_even`, and `poly.Z_odd` are used. `pcolormesh` uses a polar projection. `plot_surface` uses a cartesian projection in three dimensions requiring the conversion of the meshgrid of cartesian coordinates.

View File

`menu.py` imports both classes from `notebook.py` and adds another `Menu`. Instantiating the `Menu` also instantiates a new `Polynomial_Notebook`.

The methods included as part of the `Menu` are designed to manipulate this notebook. Running the menu from the command line provides the user with the following menu:

```
Notebook Menu
1. Configure Notebook
2. Add Polynomial
3. Plot Polynomial
4. Compare 2-D and 3-D plots
5. Quit
```

Configure Notebook Allows the user the change the set density of plots generated.

Add Polynomial Allows the user to add a polynomial to the notebook.

Plot Polynomial Generates and displays a plot of the even and odd variants of a particular polynomial.

Compare 2-D and 3-D plots Generates and displays a plot of an even or odd polynomial in two- and three-dimensions of a particular polynomial.

Quit Quits the program.

A sample notebook After several polynomials have been added, we may see the following polynomials in the notebook after running the `Show all Polynomials` command:

```
1: created: 2014-12-06 n,m: 0,0 coeffs: [ 1.]
2: created: 2014-12-06 n,m: 1,1 coeffs: [ 0.  1.]
3: created: 2014-12-06 n,m: 2,0 coeffs: [-1.  0.  2.]
4: created: 2014-12-06 n,m: 2,2 coeffs: [ 0.  0.  1.]
5: created: 2014-12-06 n,m: 3,1 coeffs: [ 0. -2.  0.  3.]
6: created: 2014-12-06 n,m: 3,3 coeffs: [ 0.  0.  0.  1.]
7: created: 2014-12-06 n,m: 4,0 coeffs: [ 1.  0. -6.  0.  6.]
8: created: 2014-12-06 n,m: 4,2 coeffs: [ 0.  0. -3.  0.  4.]
9: created: 2014-12-06 n,m: 4,4 coeffs: [ 0.  0.  0.  0.  1.]
```

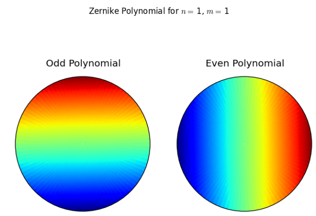

Figure 3: Even and Odd Plots, $n = 1$, $m = 1$

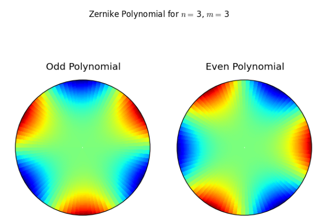

Figure 4: Even and Odd Plots, $n = 3$, $m = 3$

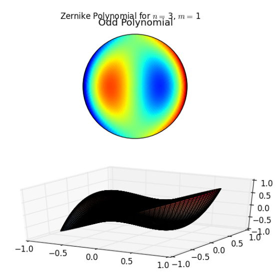

Figure 5: 2-D and 3-D plots, Odd polynomial, $n = 3$, $n = 1$

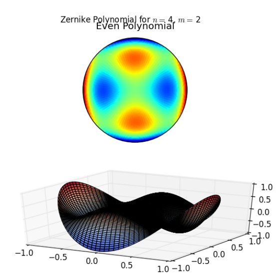

Figure 6: 2-D and 3-D plots, Even polynomial, $n = 4$, $n = 2$

The first 15 Zernike Polynomials