



Eötvös Loránd University  
Faculty of Informatics  
Department of Numerical Analysis

# **Robust quaternion Zernike moments and their applications in color image analysis and recognition**

TDK thesis

*Supervisor:*  
Zsolt Németh  
Assistant Professor

*Author:*  
Gergely Nagy  
Computer Science MSc  
2<sup>nd</sup> year

Budapest, 2020

## Abstract

Image moments and moment invariants are widely used in applications for pattern matching, image recognition, or feature extraction.

Most of the classical moments are defined for grayscale images, however, extending these techniques to multichannel ones is a generally unresolved problem. Conventionally, for color images either RGB decomposition or grayscale conversion was used.

More recently, the algebra of quaternions has been used to extend the grayscale methods to color images. Quaternion Zernike moments have been introduced as an extension of Zernike moments [1]. These Zernike moments are defined by the Zernike functions, which are a system of orthogonal functions over the unit disk, possessing certain inherent invariance properties.

The method for discretizing quaternion Zernike moments in works [1] and [2] introduces uniformly distributed points over the unit disk. This method does not achieve discrete orthogonality, which is an important property of the discretization of methods defined by orthogonal functions.

In this thesis, we propose a novel method for the discretization of quaternion Zernike moments over the unit disk: a system of points is defined on the unit disk, over which the quaternion Zernike functions are discrete orthogonal. This improves the robustness of computing the moments by decreasing the error introduced by discretization.

The new method is compared to the original one. In the case of image reconstruction, we find that the novel method decreases the error of reconstruction significantly.

The recognition capabilities for rotated, scaled and translated images with varying levels of noise are also studied. We find that with respect to Gaussian noise the new method achieves significantly better rates of recognition, even for high noise values.

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Contributions . . . . .	4
1.2	Structure of the thesis . . . . .	4
<b>2</b>	<b>Background</b>	<b>6</b>
2.1	Image moments for grayscale images . . . . .	6
2.2	Image moments for multichannel, color images . . . . .	10
2.3	Discretization . . . . .	18
2.4	Applications . . . . .	22
<b>3</b>	<b>Proposed method for discretization</b>	<b>24</b>
<b>4</b>	<b>Implementation</b>	<b>28</b>
4.1	Programming language and libraries used . . . . .	28
4.2	Calculating moments and moment invariants . . . . .	28
4.3	New image transformation . . . . .	33
<b>5</b>	<b>Tests and comparison with the original method</b>	<b>39</b>

5.1	Test images . . . . .	39
5.2	Invariance test . . . . .	43
5.3	Image reconstruction . . . . .	46
5.4	Image recognition . . . . .	50
5.5	Template matching . . . . .	57
5.6	Comparisons with QRHFsMs . . . . .	59
<b>6</b>	<b>Conclusion</b>	<b>60</b>
6.1	Future possibilities . . . . .	61
	<b>Bibliography</b>	<b>62</b>

# Chapter 1

## Introduction

Algorithms relying on moment invariants are widely used in applications for pattern matching [3], image recognition [4], or to extract useful features from images [5].

Over the past few decades, many different kinds of moments were defined and tested for single-channel, grayscale images. However, extending these techniques to multichannel color images is an important and generally unresolved problem with numerous potential applications. Conventionally, for color images either RGB decomposition or grayscale conversion was used in order to utilize the methods defined for single-channel images. This often leads to loss of information, for example in the case of grayscale conversion (where the average of the color channels is taken) some important color information can be lost.

More recently, the algebra of quaternions has been used to extend the single-channel methods to color images. For example, quaternion Fourier-Mellin moments have been introduced as an extension of the conventional Fourier-Mellin moments [6], as well as the quaternion Zernike moments as an extension of the conventional Zernike moments [1].

The Zernike functions are a system of orthogonal functions defined over the unit disk by Nobel laureate F. Zernike [7]. Using these functions as a basis for series expansions proved to be useful because of certain inherent invariance properties. Zernike moments, and by extension quaternion Zernike moments are defined by these functions.

Considering a digital image as a discrete sampling of an image function defined over a continuous domain, the need arises to discretize the computation of these moments. One important aspect of the discretization is to preserve the orthogonality over the discrete system in order to avoid redundancy and achieve high robustness with respect to noise.

The conventional method for discretizing quaternion Zernike moments (used by Chen et al. [2]) consists of uniformly distributed points over the unit disk. This method does not achieve discrete orthogonality thus reducing the robustness of the moments.

Our goal was to create a system over which the quaternion extension of the Zernike functions is discrete orthogonal and thus improve the robustness of quaternion Zernike moments and to decrease the error introduced by discretization.

## 1.1 Contributions

In this thesis a new method is proposed for the discretization of quaternion Zernike moments over the unit disk. A point system is constructed on the unit disk, over which the Zernike functions extended to quaternions are discrete orthogonal.

This new method is compared to the original method of computing quaternion Zernike moment via uniform sampling, used by Chen et al. [2], as well as to another recent state-of-the-art method using discrete orthogonal moments based on harmonic trigonometric functions [8]. For the tests, image sets from the Columbia Object Image Library [9] and the Amsterdam Library of Object Images [10] were used.

Besides the theoretical invariance properties of QZMIs, these are also verified empirically. The respective image reconstruction capabilities are compared and we find that the proposed method decreases the error of reconstruction significantly.

The methods are also applied to the recognition of rotated, scaled and translated (RST transformed) images with varying levels of either Gaussian or salt-and-pepper noise. We find that with respect to Gaussian noise the new method achieves significantly better rates of recognition, even for images with high noise values. For salt-and-pepper noise no significant difference can be found between the capabilities of the methods. Additionally, we also show that by decreasing the number of points used for discretization, the new method is able to achieve similar results as the original method with high number of points, but the computational need to obtain these results is much lower using the new method.

## 1.2 Structure of the thesis

This section serves as an overview of the structure of this thesis and contains a short summary of each chapter.

Chapter 1 serves as an introduction, where the motivation for this work is described and the contributions featured in this thesis are presented.

Chapter 2 provides the background for our work. The core concepts, such as (quaternion) Zernike moments and invariants are described in detail. A summary of previous methods for the discretization of quaternion Zernike moments is also given. Finally, some applications relying on Zernike moments are described.

The method we propose for the discretization of Zernike moments is presented in Chapter 3. The construction of a discrete orthogonal point system and the proof of discrete orthogonality is also given.

The different methods and technologies used for the computation of the invariants is shown in Chapter 4. The computation of the proposed new system is also described.

Chapter 5 contains the description of all the tests conducted on the methods, such as image reconstruction or recognition. The results of these tests are evaluated and the original and proposed methods are compared.

Finally, Chapter 6 summarizes the work and results presented in this thesis. Future development possibilities are also presented.

# Chapter 2

## Background

This chapter contains a summary of the concepts, definitions and results this thesis is based on. These include the definition and relevance of image moments (for both grayscale and color images), with special emphasis on Zernike moments. Moment invariants with respect to image rotation, translation and scaling are reviewed. Moreover, the algebra of quaternions is also introduced as a tool to generalize grayscale image moments to color images. Finally some examples of the state-of-the-art applications in image analysis are presented.

### 2.1 Image moments for grayscale images

In general, image moments are certain descriptive values calculated using the pixel intensities of an image.

Traditionally, these image moments are defined for grayscale images, where pixel values are described by a single (gray) channel. A grayscale image can be thought of as a discrete sampling of a real valued, two-dimensional function  $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ , where the function values at a point  $(x, y) \in \mathbb{R}^2$  describe the pixel intensity at that point [11].

Using this approach, the regular (geometric) image moments  $M_{ij}$  can be defined as

$$M_{ij} = \sum_x \sum_y x^i y^j f(x, y), \quad (2.1)$$

where  $(x, y)$  are the discrete pixel coordinates. These geometric moments can be used

to calculate the centroid of a grayscale image as

$$\{\bar{x}, \bar{y}\} = \left\{ \frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}} \right\}. \quad (2.2)$$

In more general terms, image moments with order  $p$  and repetition  $q$  can be defined using a set of basis functions  $\{H_{pq}\}$ . Now considering  $f$  as a continuous function, moments are defined as

$$\iint_A H_{pq}(x, y) f(x, y) dx dy, \quad (2.3)$$

where  $A$  is the region of the plane, which contains the image (the domain of  $f$ ).

By choosing the set of basis functions appropriately, many further kinds of image moments can be defined. Each of these moments has applications that utilize certain special properties of the basis functions. Some examples of these moments are Fourier-Mellin moments [6], Chebyshev-Fourier moments [12], radial harmonic Fourier moments [13], and Zernike moments [14]. The latter is described in detail later in this section.

## Invariance properties

Certain image moments, or some (e.g. linear) combinations of moments of different orders can be invariant with respect to image transformations, such as rotation, scaling, and translation (RST transformations).

**Rotation invariance.** To define what is meant by rotation invariance, it is useful to consider the image function  $f$  over polar variables  $r$  and  $\theta$ . Then the image  $f_o$ , rotated by some degree  $\alpha$  can be written as  $f_o(r, \theta) = f(r, \theta - \alpha)$ . A moment invariant achieves rotation invariance, when the moments of  $f$  and  $f_o$  are the same for all possible values of  $\alpha$ .

**Scaling invariance.** The scaled image  $f_s$  can be defined as  $f_s(r, \theta) = f(r/\lambda, \theta)$  for some scaling factor  $\lambda > 0$ . A moment invariant is said to achieve scaling invariance, when the moments for  $f$  and  $f_s$  are identical. In practice, this means that scaling the image, such that the scaled image is still within the region of the plane over which the moments are defined, does not change the values of the moment invariants.

**Translation invariance.** To define translation invariance, it is useful to consider the image in Cartesian coordinates. The image translated by  $(x_0, y_0)$  is defined as  $f_t(x, y) =$

$f(x - x_0, y - y_0)$ . Again, similarly to the other types of invariance, a moment invariant is said to be translation invariant if its values for  $f$  and  $f_t$  are the same. Practically, this means that however the image is translated on the plane, as long as it is within the region of the plane over which the moments are defined, the moment invariant values do not change.

**RST invariance.** A moment invariant is called RST invariant if it satisfies the criteria for all three previous invariances at the same time.

Translation invariance is generally considered the easiest to achieve, as translating the image such that the centroid falls on the origin neutralizes any prior translation. Also, certain types of image moments maintain natural invariances to translations when evaluated over the periodic extension of image data. The other two kinds of invariance require some more special properties of the basis function set or some specific construction and combination of the moments.

The exact construction of RST invariant moments for Zernike moments is presented in detail in Section 2.2

These invariance properties are widely utilized in applications for pattern matching and image recognition [3, 15, 16]. In particular, moment invariants can be used in medical applications, such as solving the Pathological Brain Detection problem [17].

## Zernike moments

As mentioned previously, Zernike moments are image moments, defined by choosing the set of basis functions as the Zernike functions. This set of functions (defined later in this section) proved to be a suitable basis for series expansion in numerous optical applications (see for example [18–20]), and moment invariants could easily be constructed, especially for rotation invariance.

Zernike functions, first introduced by F. Zernike [7] in 1934, are a set of complex functions defined on the unit disk. Since the domain of these functions is the unit disk, it is useful to define them in terms of polar coordinates. The Zernike functions  $V_{n,m}$  ( $n \in \mathbb{N}$ ,  $m \in \mathbb{Z}$ ,  $n \geq |m|$  and  $n - |m|$  is even) are defined as

$$V_{n,m}(r, \theta) = R_{n,m}(r)e^{im\theta} \quad (r \in [0, 1], \theta \in [0, 2\pi]), \quad (2.4)$$

where  $R_{n,m}$  is the radial part and  $e^{im\theta}$  is the azimuthal part of the function. The radial

part is defined as the following radial polynomials (of degree  $n$ )

$$R_{n,m}(r) = \sum_{k=0}^{\frac{n-|m|}{2}} \frac{(-1)^k (n-k)!}{k! \left(\frac{n+|m|}{2} - k\right)! \left(\frac{n-|m|}{2} - k\right)!} r^{n-2k}. \quad (2.5)$$

The set of Zernike functions is orthogonal over the unit disk with respect to the weight  $r$  [14], so for indices where  $n - |m|$  and  $n' - |m'|$  are both even we get

$$\int_0^{2\pi} \int_0^1 V_{n,m}(r, \theta) V_{n',m'}^*(r, \theta) r dr d\theta = \frac{\pi}{n+1} \delta_{n,n'} \delta_{m,m'}, \quad (2.6)$$

where  $\delta_{a,b}$  denotes the Kronecker delta function, and  $V_{n,m}^*(r, \theta)$  is the complex conjugate of the Zernike functions, meaning

$$V_{n,m}^*(r, \theta) = R_{n,m}(r) e^{-im\theta}.$$

It is important to note that the radial polynomials  $R_{n,m}$  are the same for both  $m$  and  $-m$ , so  $R_{n,m}(r) = R_{n,-m}(r)$ . These radial polynomials also satisfy an orthogonality relation over  $[0, 1]$  with respect to the weight  $r$  [21], that is, for a fixed  $m$  we have

$$\int_0^1 R_{n,|m|}(r) R_{n',|m|}(r) r dr = \frac{1}{2n+2} \delta_{n,n'}. \quad (2.7)$$

Another important and useful property of these radial polynomials is that they can be expressed in terms of the shifted Jacobi polynomials. Let  $P_k^{(\alpha, \beta)}(x)$  denote the  $k$ -th degree classical Jacobi polynomials [22]. The Jacobi polynomials are the set of polynomials defined over  $[-1, 1]$ , which are orthogonal with respect to the weight  $(1-x)^\alpha (1+x)^\beta$ . By shifting the argument of these polynomials, we can get polynomials which are defined over  $[0, 1]$ , the domain of the radial polynomials  $R_{n,m}$ . Expressing these radial polynomials in terms of the shifted Jacobi polynomials we get

$$R_{n,m}(r) = r^{|m|} P_{\frac{n-|m|}{2}}^{(0,|m|)}(2r^2 - 1). \quad (2.8)$$

Let  $f(r, \theta)$  be a grayscale, continuous image function (given in polar coordinates), whose domain is the unit circle. Using the previously defined set of Zernike functions, the Zernike moment of order  $n$ , repetition  $m$  of  $f(r, \theta)$  can be defined as

$$Z_{n,m}(f) = \frac{n+1}{\pi} \int_0^1 \int_0^{2\pi} f(r, \theta) V_{n,m}^*(r, \theta) r dr d\theta. \quad (2.9)$$

By utilizing the orthogonality of the Zernike functions, the original image function  $f(r, \theta)$  can be approximately reconstructed from a finite number of Zernike moments up to degree  $M$ :

$$f(r, \theta) \approx \sum_{n=0}^M \sum_{m=-n}^n Z_{n,m}(f) V_{n,m}(r, \theta). \quad (2.10)$$

**Invariance properties of Zernike moments.** Translation invariance (as described previously) can be achieved by translating the image such that the centroid of the image falls on the origin. This method is suitable to use with Zernike moments as well.

To calculate rotation invariant features from Zernike moments, first consider the moments of an image rotated by some degree  $\alpha$ , defined using polar coordinates:  $f_r(r, \theta) = f(r, \theta - \alpha)$ . In this case it can be shown that

$$Z_{n,m}(f_r) = Z_{n,m}(f) e^{-im\alpha}.$$

This means that the modulus Zernike moments is the same regardless of how the image is rotated, thus  $|Z_{n,m}(f)|$  are rotation invariant features [14].

Scaling invariance, as well as an extended rotation invariance is discussed in detail in Section 2.2, for quaternion Zernike moments.

## 2.2 Image moments for multichannel, color images

The image moments in Section 2.1 are only defined for single-channel, grayscale images. Extending these signal and image processing techniques efficiently to multichannel color images is an important and generally unresolved problem. This section introduces some conventional approaches to generalize the grayscale methods to RGB images. Later, the algebra of quaternions is also presented, and using this algebra, quaternion moments are introduced, specifically quaternion Zernike moments and their invariant properties.

### Conventional techniques

Conventional approaches rely on direct utilization of grayscale methods in some form. One such way is to use grayscale conversion to convert the RGB image to a single-channel grayscale image and use the grayscale method on this converted image. The drawback of this method is that the color information contained in multiple channels is lost. Thus, for example, this method cannot distinguish between a red and a blue object.

The other notable conventional method utilizes RGB-decomposition, whereby grayscale (single-channel) techniques can be used on each of the individual color channels [23]. This approach does not lose color information, but any relationship between the color channels cannot be utilized by the method.

## Algebra of quaternions

More recently, the algebra of quaternions was employed to generalize various single-channel moments, so that color images can be analysed holistically. In order to introduce this generalization, we first introduce the algebra of quaternions and how color images can be represented as quaternion valued functions.

Quaternions were defined by Hamilton [24] as a generalization of the complex numbers. The set of quaternions is denoted by  $\mathbb{H}$ . A quaternion,  $q \in \mathbb{H}$ , consists of four components:

$$q = a + bi + cj + dk,$$

where  $a, b, c$  and  $d$  are real numbers, and  $\mathbf{i}, \mathbf{j}, \mathbf{k}$  are the imaginary units, defined according to the following rules:

$$\begin{aligned} \mathbf{i}^2 &= \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1, \\ \mathbf{ij} &= -\mathbf{ji} = \mathbf{k}, \quad \mathbf{jk} = -\mathbf{kj} = \mathbf{i}, \quad \mathbf{ki} = -\mathbf{ik} = \mathbf{j}. \end{aligned}$$

Therefore, the set of quaternions  $\mathbb{H}$  is an *algebra*. It is important to note, that the multiplication of quaternions is not commutative. The conjugate and modulus of  $q$  are respectively defined by

$$\begin{aligned} q^* &= a - bi - cj - dk, \\ |q| &= \sqrt{a^2 + b^2 + c^2 + d^2}. \end{aligned}$$

A quaternion is called pure quaternion, when  $a = 0$ , i.e. the real part of the quaternion is zero.  $q$  is called a unit quaternion if  $|q| = 1$ .

**Representing color images with quaternions.** The 3-dimensional RGB color space can be thought of in terms of pure quaternions, where the color  $(r, g, b)$ , consisting of the three components  $r, g, b \in [0, 1]$ , is uniquely described by the pure quaternion  $r\mathbf{i} + g\mathbf{j} + b\mathbf{k}$ .

In practice, quaternions are useful for describing three-dimensional rotations, thus rotations in the RGB color space can be easily described by representing the colors in terms of pure quaternions.

Ell and Sangwine [25] utilized quaternions to represent a color image,  $f : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ , as follows:

$$f(x, y) = \mathbf{i}f_R(x, y) + \mathbf{j}f_G(x, y) + \mathbf{k}f_B(x, y),$$

where functions  $f_R, f_G, f_B : \mathbb{R}^2 \rightarrow \mathbb{R}$  represent the red, green and blue components of the  $(x, y)$  pixel, respectively. This way, a color image can be thought of as a pure quaternion valued function and the algebra of quaternions can be employed to define image moments in a similar manner as in the case of grayscale images.

## Quaternion moments

Similar to the general formula for defining the moments of grayscale images in (2.3), quaternion moments can be defined by a set of quaternion valued basis functions and a quaternion valued, continuous image function. Most commonly, the set of complex valued basis functions is generalized such that we get a set of quaternion valued basis functions.

For example, Guo and Zhu [6] introduced quaternion Fourier-Mellin moments (QFMMs) which are an extension of the conventional Fourier-Mellin moments. Similarly, the same quaternion techniques were applied successfully to other function systems (e.g. [12, 26]), yielding similar results.

## Quaternion Zernike moments

Chen et al. [1, 2] proposed the quaternion Zernike moments (QZMs), an extension of the conventional Zernike moments to color images using quaternions. Generally, this method overperforms other similar approaches in color image recognition, due to the natural invariances of Zernike functions.

Quaternion Zernike moments, just like regular Zernike moments, are defined over the complex unit disk

$$\mathbb{D} = \left\{ z = re^{i\theta} \in \mathbb{C} : r \in [0, 1], \theta \in [0, 2\pi) \right\}.$$

In order to define quaternion Zernike moments, first we have to define the quaternion-valued generalization of the complex-valued Zernike functions defined in (2.4). Since the radial part of the Zernike functions is a real-valued function of  $r$ , only the azimuthal component needs to be generalized. Thus the quaternion-valued Zernike functions can be defined as

$$\phi_{n,m}(r, \theta) = R_{n,m}(r)e^{-\mu m\theta},$$

where  $\mu$  is an arbitrary unit pure quaternion. The usual choice for  $\mu$  is the value  $\frac{i+j+k}{\sqrt{3}}$ , because this is the unit pure quaternion representing the color gray in the RGB color space.

Similarly to the complex-valued Zernike functions, the generalized Zernike functions also satisfy the following orthogonality relation with respect to the weight  $r$ :

$$\frac{n+1}{\pi} \int_0^1 \int_0^{2\pi} \phi_{n,m}(r, \theta) \phi_{n',m'}^*(r, \theta) r d\theta dr = \delta_{n,n'} \delta_{m,m'}. \quad (2.11)$$

Let  $f(r, \theta)$  be a pure quaternion valued RGB image function with a continuous domain, defined in polar coordinates over the unit disk  $\mathbb{D}$ . As described previously, each color component of the image function corresponds to one of the imaginary units.

The definition of QZMs is similar to the definition of Zernike moments in (2.9), but instead of the complex-valued  $V_{n,m}$  functions we use the generalized  $\phi_{n,m}$  functions. Since the multiplication of quaternion is not commutative, both right-side and left-side quaternion Zernike moments can be defined, depending on which side we multiply  $f(r, \theta)$  from. The right-side QZM of order  $n$  and repetition  $m$  is defined as

$$Z_{n,m}^R(f) = \frac{n+1}{\pi} \int_0^1 \int_0^{2\pi} R_{n,m}(r) f(r, \theta) e^{-\mu m \theta} r dr d\theta, \quad (2.12)$$

$n \geq |m|$  and  $n - |m|$  is even,

while the left-side QZM for the same indices is defined as

$$Z_{n,m}^L(f) = \frac{n+1}{\pi} \int_0^1 \int_0^{2\pi} R_{n,m}(r) e^{-\mu m \theta} f(r, \theta) r dr d\theta.$$

Analogously to the grayscale case in (2.10), the image function  $f(r, \theta)$  can be approximated by using either right-side or left-side QZMs up to a finite  $M$  degree.

$$f(r, \theta) \approx \sum_{n=0}^M \sum_{m=-n}^n Z_{n,m}^R(f) R_{n,m}(r) e^{\mu m \theta} \quad (2.13)$$

$$f(r, \theta) \approx \sum_{n=0}^M \sum_{m=-n}^n e^{\mu m \theta} Z_{n,m}^L(f) R_{n,m}(r)$$

It is important to note that a left-side QZM can be expressed in terms of the right-side QZM of the same order and repetition using the following relation:

$$Z_{n,m}^L(f) = -(Z_{n,m}^R(f))^*$$

Because of this, in this thesis only the right-side QZMs will be studied.

## Quaternion Zernike moment invariants

Chen et al. [1, 2] proposed the constructions described in this section in order to create rotation, scaling and translation invariant moment invariants using the quaternion Zernike moments.

**Rotation invariance.** In order to achieve rotation invariance consider the image rotated by some degree  $\alpha$ :  $f_\circ(r, \theta) = f(r, \theta - \alpha)$ . Chen et al. [2] proved that for the rotated image  $Z_{n,m}^R(f_\circ) = Z_{n,m}^R(f)e^{-\mu m \theta}$  and  $Z_{n,m}^L(f_\circ) = e^{-\mu m \theta} Z_{n,m}^L(f)$ .

Similarly to the complex-valued case, the modulus  $|Z_{n,m}^R(f)|$  can be used to achieve rotation invariance [1], but this provides only one real-valued invariant, whereas the following construction can provide quaternion-valued invariants.

Based on the properties of the QZMs of a rotated image, it follows that the quaternion defined as

$$\Phi_{n,k}^m(f) = Z_{n,m}^R(f)Z_{k,-m}^L(f) = -Z_{n,m}^R(f)(Z_{k,m}^R(f))^* \quad (2.14)$$

is invariant under rotation, meaning that  $\Phi_{n,k}^m(f) = \Phi_{n,k}^m(f_\circ)$ . From now on in this thesis, this invariant will be referred to as a quaternion Zernike moment rotation invariant (QZMRI).

**Translation invariance.** Analogously to the construction of translation invariant Zernike moments in the grayscale case, now translation invariance can be achieved by setting the origin of the coordinate system as the centroid of the image.

For grayscale images the centroid can be defined as in (2.2). For RGB images, we need the common centroid of all three color channels. This can be calculated using zero-order and first-order geometric moments for each individual color channel, as described by Suk and Flusser [23]:

$$\{\bar{x}, \bar{y}\} = \left\{ \frac{M_{10}^R + M_{10}^G + M_{10}^B}{M_{00}^R + M_{00}^G + M_{00}^B}, \frac{M_{01}^R + M_{01}^G + M_{01}^B}{M_{00}^R + M_{00}^G + M_{00}^B} \right\}, \quad (2.15)$$

where  $M_{ij}^R, M_{ij}^G, M_{ij}^B$  denote the  $M_{ij}$  geometric moments (defined in (2.1)) for the red, green, and blue color channels respectively.

Consider a translated image function in Cartesian coordinates:  $f_t(x, y) = f(x - x_0, y - y_0)$ .

If the origin of the coordinate system is placed on the common centroid of the color channels  $(\bar{x}, \bar{y})$ , then the QZMs calculated in this coordinate system will be invariant to translation. This further translated image is denoted by  $\bar{f}_t(x, y) = f_t(x - \bar{x}, y - \bar{y})$ . Note that the common centroid of  $f_t$  and  $f$  are different, thus these two images need to be further translated by different amounts.

By considering the QZMs of both  $\bar{f}_t$  and  $\bar{f}$ , we get

$$Z_{n,m}^R(\bar{f}_t) = Z_{n,m}^R(\bar{f}).$$

This means that first translating the image so that the common centroid falls on the origin of the coordinate system and only then calculating the QZMs yields translation invariance. Let  $\bar{Z}_{n,m}^R(f) = Z_{n,m}^R(\bar{f})$  denote these translation invariant QZMs.

**Scaling invariance.** For non-negative integers  $m$  and  $l$ , Chen et al. [2] constructed scaling invariants utilizing the symmetric property of the radial polynomials with respect to  $m$  and an alternate form of the QZMs.

Let  $f_s(r, \theta) = f(r/\lambda, \theta)$  denote an image scaled with some  $\lambda > 0$  scaling parameter. It was shown that the QZMs of the scaled image can be expressed as a linear combination of the QZMs of the original image in the following way:

$$\begin{aligned} c_{m,l}^{t,k} &= (-1)^{l-k} \frac{(m+2l+1)(m+k+l)!}{(l-k)!(k-t)!(m+k+t+1)!} \\ Z_{m+2l,m}^R(f_s) &= \sum_{t=0}^l \sum_{k=t}^l \lambda^{m+2k+2} c_{m,l}^{t,k} Z_{m+2t,m}^R(f). \end{aligned}$$

Utilizing the formula above, the following scaling invariants have been constructed:

$$L_{m+2l,m}^R(f) = \sum_{t=0}^l \sum_{k=t}^l \left( \sqrt{|Z_{0,0}^R(f)|} \right)^{-(m+2k+2)} c_{m,l}^{t,k} Z_{m+2t,m}^R(f).$$

These scaling invariants satisfy  $L_{m+2l,m}^R(f_s) = L_{m+2l,m}^R(f)$ , meaning that their values are the same regardless of how the original image was scaled.

**Combined RST invariants.** Similarly to the construction of the rotation invariants  $\Phi_{n,k}^m$ , the previously defined scaling invariants can also be used to construct  $\Psi_{n,k}^m(f) = L_{n,m}^R(f)(L_{k,m}^R(f))^*$ , which is invariant to both rotation and scaling.

In order to achieve translation invariance as well, throughout the construction of the scaling invariants  $L_{n,m}^R$  the translation invariant QZMs  $(\bar{Z}_{n,m}^R(f))$  can be used, thus defining the  $\bar{L}_{n,m}^R$  combined translation and scaling invariants. This means that before constructing the scaling invariants, the image has to be translated so that the combined centroid falls on the origin.

Finally, to achieve combined RST invariance, the combined translation and scaling invariants can be used to construct  $\bar{\Psi}_{n,k}^m(f) = \bar{L}_{n,m}^R(f)(\bar{L}_{k,m}^R(f))^*$ , which is invariant to rotation, scaling, and translation. These  $\bar{\Psi}_{n,k}^m(f)$  values are called quaternion Zernike moment invariants (QZMI).

### Quaternion radial harmonic Fourier moments

Another set of quaternion-valued moments are the quaternion radial harmonic Fourier moments (QRHFs) [8, 27, 28]. We introduce and describe these moments, as in Chapter 5 we present comparisons between the capabilities of QRHFs and QZMIs.

Similarly to Zernike moments, these Fourier moments are also defined for images whose domain is the unit disk  $\mathbb{D}$ . The basis functions for QRHFs are defined as

$$H_{n,m}(r, \theta) = R_n(r)e^{\mu m \theta},$$

where  $\mu$  is a unit pure quaternion, usually  $\mu = \frac{i+j+k}{\sqrt{3}}$ , and  $R_n(r)$  are the radial kernel functions defined as

$$R_n(r) = \begin{cases} \frac{1}{\sqrt{r}}, & n = 0 \\ \sqrt{\frac{2}{r}} \cos(\pi nr), & n \text{ is even} \\ \sqrt{\frac{2}{r}} \sin(\pi(n+1)r), & n \text{ is odd} \end{cases}$$

The functions  $H_{n,m}(r, \theta)$  also satisfy the following orthogonality relation:

$$\int_0^{2\pi} \int_0^1 H_{n,m}(r, \theta) H_{n',m'}^*(r, \theta) r dr d\theta = 2\pi \delta_{n,n'} \delta_{m,m'}.$$

The definition of the QRHFs is similar to the definition of QZMs, both right-side and

left-side moments can be defined as

$$\begin{aligned} F_{n,m}^R(f) &= \frac{1}{2\pi} \int_0^{2\pi} \int_0^1 f(r, \theta) H_{n,m}^*(r, \theta) r dr d\theta \\ F_{n,m}^L(f) &= \frac{1}{2\pi} \int_0^{2\pi} \int_0^1 H_{n,m}^*(r, \theta) f(r, \theta) r dr d\theta, \end{aligned} \quad (2.16)$$

respectively, where  $n \in \mathbb{N}$  and  $m \in \mathbb{Z}$ . Left-side moments can be expressed in terms of right-side moments, so from now on we only study right-side QRHFs.

Similarly to the previous cases, the original image can be approximated using only a finite number of moments:

$$f(r, \theta) \approx \sum_{n=0}^N \sum_{m=-M}^M F_{n,m}^R(f) H_{n,m}(r, \theta).$$

**Invariance properties.** Translation invariance can be achieved by translating the image before calculating the moments such that the common centroid of the image falls on the origin. This construction is exactly the same as in the case of the quaternion Zernike moments.

Liu et al. [8] show that for a rotated image  $f_r(r, \theta) = f(r, \theta - \alpha)$  the following holds:

$$F_{n,m}^R(f_r) = F_{n,m}^R(f) e^{\mu m \theta},$$

which means that the modulus of a QRHFM is invariant to image rotation, so  $|F_{n,m}^R(f)|$  can be used as rotation invariant features.

Liu et al. [8] do not present a construction which provides scaling invariance, they only show a method of rescaling the images such that they fit inside the unit circle. This method is only suitable, when the region of interest on the image is known in advance (or the entire image only consists of the object that has to be analysed).

Because of this lack of general scaling invariance, comparisons later on focus mainly on the rotation invariance and image reconstruction capabilities of the QZM and QRHFM methods.

## Other function systems

Over the past 8 years, a similar approach for generalizing image moments to multichannel images was utilized numerous times in the literature. These methods use basis functions which are similar in construction to the Zernike functions, but the radial polynomials

$R_{n,m}$  are substituted with by other, well-known orthogonal systems over the radial interval  $[0, 1]$ . Further examples for this include the QFMMs [6], the QPETs, QPCTs and QPSTS [29], the QCMs [30], the QBFMs [26], the QG-CHFMs and QG-PJFMs [31] and the QSBFMs [32].

## 2.3 Discretization

The previously defined moments are defined for image functions, which have a continuous domain. Specifically both quaternion Zernike moments and quaternion radial harmonic Fourier moments are defined for functions, whose domain is the unit disk. In practice, digital images are usually defined in terms of discrete image coordinates, with the coordinates ranging from 0 to  $N - 1$  (the number of pixels along each axis). Thus the need arises to discretize these moments and transform the digital image to polar coordinates inside the unit disk.

### Transforming a digital image onto the unit disk

There are two "natural" ways to linearly transform a square image from image coordinates to polar coordinates inside the unit circle, using only translation and scaling [33].

The first method is to transform the entire image inside the unit circle, as shown on Figure 2.1. This way all of the pixels will be used for calculating the (quaternion) moments, but some areas of the unit disk will remain empty, no pixels fall on those areas.

The other method is to transform the image such that the unit disk becomes the inscribed circle of the square image, as shown on Figure 2.2. This method fills the entire unit disk with pixels, but some pixels fall outside the unit circle and thus will not be used for the calculation of (quaternion) moments.

The polar coordinates  $(r, \theta)$  corresponding to the image coordinates  $(x, y)$  can be calculated by the following formulas, which is the mapping proposed by Chong et al. [34]:

$$r = \sqrt{(c_1 x + c_2)^2 + (c_1 y + c_2)^2},$$

$$\theta = \tan^{-1} \left( \frac{c_1 y + c_2}{c_1 x + c_2} \right),$$

where  $c_1, c_2 \in \mathbb{R}$  depend on which of the previously mentioned transformations is used. A  $\lambda \in \mathbb{R}$  scaling factor is also defined for each transformation. For the first one (Figure 2.1):  $c_1 = \frac{\sqrt{2}}{N-1}$ ,  $c_2 = -\frac{1}{\sqrt{2}}$ , and  $\lambda = \frac{2}{\pi}$ , while for the other transformation (Figure 2.2):

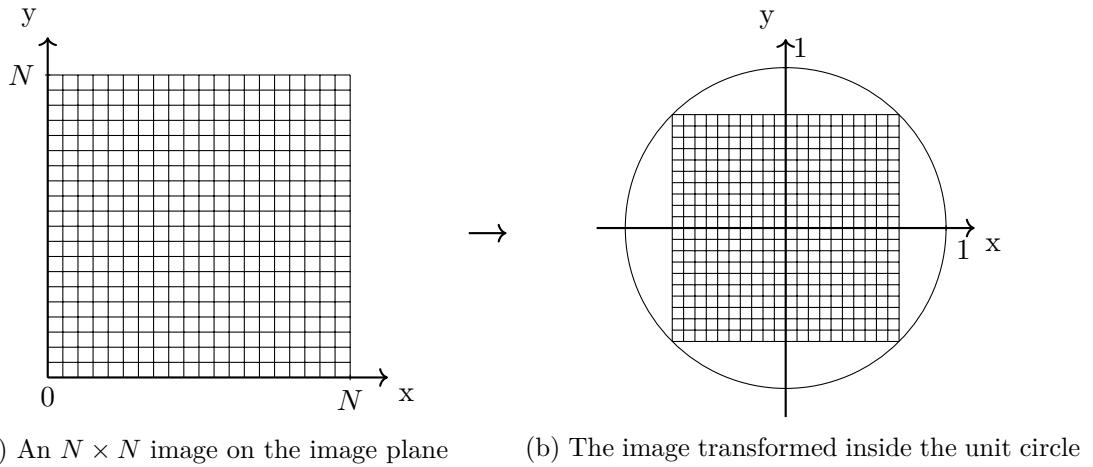


Figure 2.1: The image before and after applying the transformation inside the unit disk

$c_1 = \frac{2}{N-1}$ ,  $c_2 = -1$  and  $\lambda = 1$ , where  $N$  is the number of pixels along each axis of the image.

The system of points defined by performing either one of these transformations for the entire image can be used as the basis points for the discretization, or the pixel values at these points can be used to approximate the values of the image function at other points on the unit disk using interpolation techniques.

### Discretization of QZMs

In general, using the polar form of the transformed coordinates, the continuous integral of (2.12) can be replaced by discrete approximations of form

$$Z_{n,m}^R(f) \approx \lambda_{n,m} \sum_{k=1}^{N_1} \sum_{j=1}^{N_2} f(r_k, \theta_j) \phi_{n,m}(r_k, \theta_j) w(r_k, \theta_j), \quad (2.17)$$

with some discretization weight function  $w$  and normalization constants  $\lambda_{n,m}$ . In this discretization formula, the system of points  $(r_k, \theta_j)$  does not necessarily coincide with the transformed lattice of the pixels, so  $f(r_k, \theta_j)$  is some approximation of the image function, calculated from the discrete pixel values.

Using either one of the previously defined "natural" transformations, a point system on the unit disk can be obtained. The approximation of the image function  $f$  at these points is not necessary as each point can have the corresponding pixel value assigned to it. Using this point system in (2.17), the following formula can be used to approximate

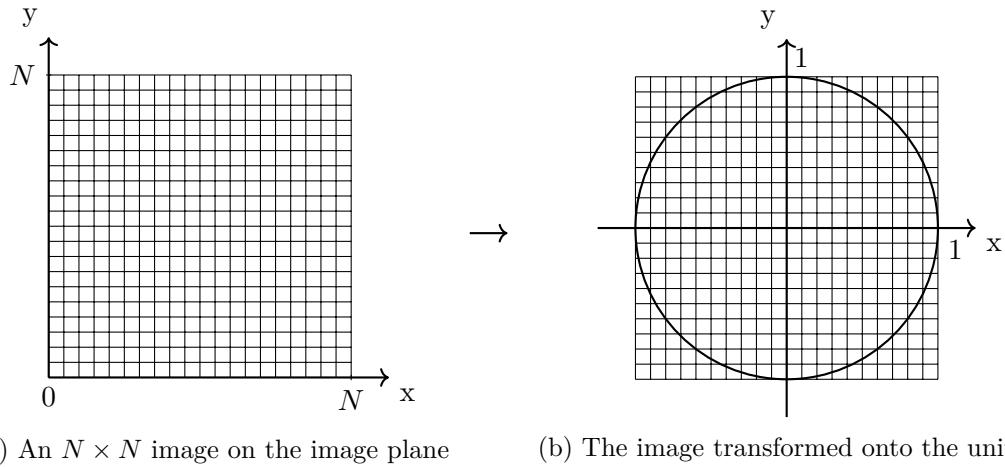


Figure 2.2: The image before and after applying the transformation onto the unit disk

the QZMs of the original image function:

$$Z_{n,m}^R(f) \approx \lambda \frac{n+1}{(N-1)^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} R_{n,m}(r_{x,y}, \theta_{x,y}) f(x, y) e^{-\mu m \theta_{x,y}},$$

where the RGB image  $f \in \mathbb{R}^2 \rightarrow \mathbb{H}$  is now defined in image coordinates,  $(r_{x,y}, \theta_{x,y})$  are the polar coordinates belonging to the image coordinates  $(x, y)$ , and  $\lambda$  is the previously defined scaling factor belonging to the linear image transformation.

In the rest of this thesis, for comparison purposes, the transformation of the image inside the unit circle (Figure 2.1) will be used as Chen et al. [2] used this discretization to present their results.

**Discrete orthogonality.** Even though the (quaternion-valued) Zernike functions are a set of orthogonal functions on the complex unit disk, neither of the previously defined point systems provide discrete orthogonality, meaning that the discretization of the orthogonality relations in (2.6) and (2.11) over these point systems is not satisfied.

This lack of discrete orthogonality results in poor performance of the methods, especially under noisy conditions. It also makes the moment representation of an image redundant, hence introducing unwanted errors during image reconstruction.

A method for the discretization of the quaternion-valued Zernike functions, providing discrete orthogonality is proposed in Chapter 3.

## Other discretization techniques

The "natural" method of discretization (as described in the case of QZMs) proved to be susceptible to inaccuracies of both geometric and numeric nature when applied to grayscale images. Liao and Pawlak described these issues in the case of the conventional Zernike moments [35, 36]. These problems are inherited by the quaternion generalizations as well.

Later, Xin et al. [37] proposed a more accurate computational technique, where the unit disk  $\mathbb{D}$  is partitioned into polar sectors of approximately equal areas. The original pixels are transformed to the centroids of these sectors via cubic interpolation. The weights for the discretization can be computed by integrating the  $\phi_{n,m}$  basis functions over the respective polar sectors.

The same idea, adapted to quaternion-type moments for any aforementioned radial system could substantially improve the performance of tasks like image reconstruction, as well as recognition after RST transformations and addition of different kinds of noise. The reason for this is the improved accuracy gained for the computation of respective moments and invariants. Examples of this technique applied for color images include the papers of Hosny and Darwish for Legendre [38] and Chebyshev [39] radial systems.

A significant drawback of this approach is that the cubic interpolation used for coordinate transformation is irreversible. This means that in order to reconstruct the original image some other transformation has to be used. This additional transformation can further introduce error during image reconstruction. To circumvent this issue, in previous works [35–39], the error of reconstruction was measured for the transformed image and its reconstructions on the unit disk  $\mathbb{D}$  only.

Another, although minor issue is the natural smoothing provided by the cubic interpolation, which unintentionally improves the obtained results for experiments on noisy images by filtering some of the noise. This is a sometimes unwanted side effect, which can distort measurements of the performance of the moments themselves. To ensure fair comparisons with the previously introduced "natural" transformations, linear interpolation should be used, ensuring invertible coordinate transformations, and also excluding unwanted side effects.

Finally, the results of Wang et al. [28] and Liu et al. [8] for radial harmonic Fourier moments (as described previously) are some of the recent advances in the study of quaternion moments, introducing a novel coordinate transformation, which improves accuracy for this specific function system while maintaining fast computation times, without utilizing cubic interpolation at all. These improvements stem from the discrete orthogonality provided by the novel coordinate transformation, with respect to the specific basis functions.

## Discretization of QRHFs

Liu et al. [8] defined a point system for the discretization of quaternion radial harmonic Fourier moments, which consists of equally spaced radii and uniformly distributed points along these radii:

$$(r_u, \theta_v) = \left( \frac{u}{N}, \frac{2\pi v}{N} \right), \quad u = 0, 1, \dots, N-1, \quad v = 0, 1, \dots, N-1,$$

where  $N$  is the number of pixels along each axis in the original image. The value of the image function  $f$  can be approximated at these points by assigning the value of the closest pixel to each point:

$$f(r_u, \theta_v) \approx f' \left( \left\lfloor r_u \frac{N}{2} \sin \theta_v \right\rfloor + \frac{N}{2}, \left\lfloor r_u \frac{N}{2} \cos \theta_v \right\rfloor + \frac{N}{2} \right),$$

where  $f'$  denotes the original image, given in image coordinates.

Using this system of points for the discretization of the integrals in (2.16), we get

$$F_{n,m}^R(f) \approx \frac{4}{2\pi N^2} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} f(r_u, \theta_v) H_{n,m}^*(r_u, \theta_v).$$

The basis functions  $H_{n,m}$  satisfy the discrete orthogonality relation with this method of discretization, thus no additional error is introduced by the discretization.

## 2.4 Applications

In this section, we list some practical applications of image moments and moment invariants, with special attention to Zernike moments. Zernike moments and moment invariants are a useful tool in optical and medical applications. Specifically, Zernike moments can be used to represent and capture optical aberrations [18–20].

Furthermore, applications in pattern recognition have shown that in noisy environments Zernike moment yield the best results [4]. Castro-Ortega et al. [40] have used Zernike moment invariants to extract descriptions of hand vein patterns.

Another field of application is image watermarking, where a watermark that is resistant against transformation- and noise-based attacks may be inserted directly into the moment decomposition of an image [41–43].

Finally, one potentially powerful application is to use Zernike moments in conjunction with neural networks. The robustness of neural networks with respect to noise is a known shortcoming of current systems [44, 45]. Zernike moments can be used as some components of the feature vector input of neural networks, thus providing their high robustness to noise during forward propagation [5, 46].

## Chapter 3

# Proposed method for discretization

In this chapter we describe our proposed approach to the discretization of continuous quaternion Zernike moments. Our aim is to ensure simple and reversible computability, but achieve considerable reduction in potential computational inaccuracies at the same time. For this, we define a system of sampling points  $(r_k, \theta_j)$ , over which the integral discretization (2.17) maintains advantageous theoretical properties.

We emphasize that for the system of Zernike functions, there is no known system of points providing a "natural" discretization and discrete orthogonality. The idea behind our approach is motivated by the work of Margit Pap and Ferenc Schipp [21], which is in fact the first example of a point system over which the discrete complex-valued Zernike functions are orthogonal. On the other hand, it is not "natural" in the sense that the number of nodes needed to achieve this is somewhat greater (in fact, almost twice as much) than what the order of functions would suggest.

Formula (2.8) helps us understand the reason why a natural discretization is so difficult to find for Zernike functions: even though the azimuthal components could be discretized easily, it is not the case for the radial terms which are shifted Jacobi polynomials of *varying* parameters.

Nonetheless, it is the same formula that provides an idea how to succeed: we combine the natural discretization of the azimuthal components with an orthogonality relation ensured by the Gaussian quadrature formula for the radial polynomials.

For this, we shall define the suitable quadrature nodes and coefficients as the roots of Legendre polynomials and the corresponding Christoffel numbers. Fix a positive integer

$N$ . Let us denote by  $\rho_{k,N}$ ,  $k = 1, \dots, N$  the roots of the  $N$ -th order Legendre polynomial [22]. Notating the fundamental polynomials of Lagrange interpolation with respect to the roots  $\rho_{k,N}$  as  $\ell_{k,N}$ , we define the constants

$$\mathcal{A}_{k,N} = \int_{-1}^1 \ell_{k,N}(x) dx, \quad (k = 1, \dots, N).$$

Then, the system of sampling points  $X_N$  over the unit disk  $\mathbb{D}$  is defined in polar form by

$$X_N \ni (r_{k,N}, \theta_{j,N}) = \left( \sqrt{\frac{1 + \rho_{k,N}}{2}}, \frac{2\pi j}{4N} \right), \quad (k = 1, \dots, N, j = 1, \dots, 4N),$$

and the respective weight values and constants are

$$w(r_{k,N}, \theta_{j,N}) = \frac{\mathcal{A}_{k,N}}{8N}, \quad \lambda_{n,m} = n + 1, \quad (3.1)$$

so the generated integral approximation based on (2.17) is

$$\frac{1}{\pi} \int_0^1 \int_0^{2\pi} f(r, \theta) d\theta dr \approx \int_{X_N} f = \sum_{k=1}^N \sum_{j=1}^{4N} f(r_{k,N}, \theta_{j,N}) \frac{\mathcal{A}_{k,N}}{8N}.$$

We remark that the values  $\mathcal{A}_{k,N}$  are somewhat close to  $\frac{1}{N}$ , so the weighting is close to being uniform.

What is appealing in this choice of  $X_N$  is the fact that the orthogonality of quaternion Zernike functions (2.11) is preserved under changing to discrete integration over this set of points, i.e.

**Theorem 1.** *Suppose that for  $n, n' \in \mathbb{N}$  naturals and  $m, m' \in \mathbb{Z}$  integers we have*

$$\frac{n + n'}{2} + \min(|m|, |m'|) < 2N.$$

*Then*

$$(n + 1) \int_{X_N} \phi_{n,m} \phi_{n',m'}^* = \delta_{n,n'} \delta_{m,m'}.$$

**Proof.** Let us consider the radial orthogonality relation (2.7) expressed with the Jacobi polynomials of (2.8), i.e.

$$\begin{aligned} \frac{1}{2n+2} \delta_{n,n'} &= \int_0^1 R_{n,|m|}(r) R_{n',|m|}(r) r dr = \\ &\int_0^1 r^{2|m|} P_{\frac{n-|m|}{2}}^{(0,|m|)}(2r^2 - 1) P_{\frac{n'-|m|}{2}}^{(0,|m|)}(2r^2 - 1) r dr, \end{aligned}$$

and apply a change of variable  $u = 2r^2 - 1$ , to obtain

$$\frac{1}{2n+2}\delta_{n,n'} = \frac{1}{4} \int_{-1}^1 \left(\frac{1+u}{2}\right)^{2|m|} P_{\frac{n-|m|}{2}}^{(0,|m|)}(u) P_{\frac{n'-|m|}{2}}^{(0,|m|)}(u) du.$$

Notice that here the integrand is a polynomial of degree  $\frac{n+n'}{2} + |m| < 2N$ , so the exact value of the integral is equal to the value of the Gauss-Legendre quadrature (see e.g. [22]), i.e.

$$\frac{1}{2n+2}\delta_{n,n'} = \frac{1}{4} \sum_{k=1}^N \mathcal{A}_{k,N} \left(\frac{1+\rho_{k,N}}{2}\right)^{2|m|} P_{\frac{n-|m|}{2}}^{(0,|m|)}(\rho_{k,N}) P_{\frac{n'-|m|}{2}}^{(0,|m|)}(\rho_{k,N}).$$

Rewriting this in terms of  $r_{k,N} = \sqrt{\frac{1+\rho_{k,N}}{2}}$  gives a discrete orthogonality relation for the radial polynomials, as

$$\frac{1}{2n+2}\delta_{n,n'} = \frac{1}{4} \sum_{k=1}^N \mathcal{A}_{k,N} R_{n,m}(r_{k,N}) R_{n',m}(r_{k,N}). \quad (3.2)$$

Now we can proceed with proving the statement.

$$\begin{aligned} \int_{X_N} \phi_{n,m} \phi_{n',m'}^* &= \sum_{k=1}^N \sum_{j=1}^{4N} \phi_{n,m}(r_{k,N}, \theta_{j,N}) \phi_{n',m'}^*(r_{k,N}, \theta_{j,N}) \frac{\mathcal{A}_{k,N}}{8N} = \\ &= \frac{1}{8N} \left( \sum_{j=1}^{4N} e^{-\mu m \theta_{j,N}} e^{\mu m' \theta_{j,N}} \right) \left( \sum_{k=1}^N \mathcal{A}_{k,N} R_{n,m}(r_{k,N}) R_{n',m'}(r_{k,N}) \right), \end{aligned}$$

and since quaternions  $-\mu m \theta_{j,N}$  and  $\mu m' \theta_{j,N}$  commute, we have

$$\sum_{j=1}^{4N} e^{-\mu(m-m')\frac{2\pi j}{4N}} = 4N \delta_{m,m'}$$

for the first sum, so if  $m \neq m'$ , the discrete integral equals to 0 and we are done.

Suppose  $m = m'$ , now we are left with

$$\int_{X_N} \phi_{n,m} \phi_{n',m}^* = \frac{1}{2} \sum_{k=1}^N \mathcal{A}_{k,N} R_{n,m}(r_{k,N}) R_{n',m}(r_{k,N}) = \frac{1}{n+1} \delta_{n,n'},$$

where we used (3.2) for the last equation as we have  $\frac{n+n'}{2} + |m| < 2N$ .  $\square$

Theoretically, the discrete orthogonality relation of Theorem 1 means that the exact values of right and left moments can be computed, provided we can measure the function

values over  $X_N$ , for a sufficiently large  $N$ : let us consider an arbitrary linear combination of right moments (2.12) in the form

$$f \approx f_N(r, \theta) = \sum_{n+m < 2N} Z_{n,m}^R(f) \phi_{n,m}^*(r, \theta),$$

then the exact value of the moment  $Z_{n,m}^R(f)$ , using the previous result, is

$$Z_{n,m}^R(f) = \int_{X_N} f_N \phi_{n,m},$$

and the same can be said for the left moments. This proves that for system  $X_N$ , the discretization error is zero, a property that no other previously used quaternion moment method possessed. Besides this, only numerical roundoffs and the moment order threshold  $f \approx f_N$  generates computational inaccuracies.

# Chapter 4

## Implementation

This chapter presents the tools and methods used for implementing a program for calculating the QZMIs of an image using both the conventional and the novel method of discretization.

### 4.1 Programming language and libraries used

The implementation was created using the Python programming language [47] and relies on the Numpy [48] and Numba [49] libraries to achieve efficient and fast computation of the moment invariants, as well as the Python Imaging Library (PIL, Pillow) [50] for image manipulation. Numpy provided a way to efficiently work with arrays and matrices, as well as the quaternion package, which supports the quaternion data type.

Using the just-in-time compilation (JIT) capabilities of Numba (i.e. the `@jit` annotation), the computationally heavy parts of the implementation could be made almost as fast and efficient as native code. The disadvantage of using JIT is that it limits the available data types and functions, so in the implementation the use of `@jit` is kept only to the critical, computationally intensive functions.

### 4.2 Calculating moments and moment invariants

To obtain the quaternion Zernike moment invariants (QZMIs) of an image as described in Section 2.2, the quaternion Zernike moments (QZMs) must be calculated first. Chen et al. [2] showed that instead of calculating the QZMs directly by using the algebra of

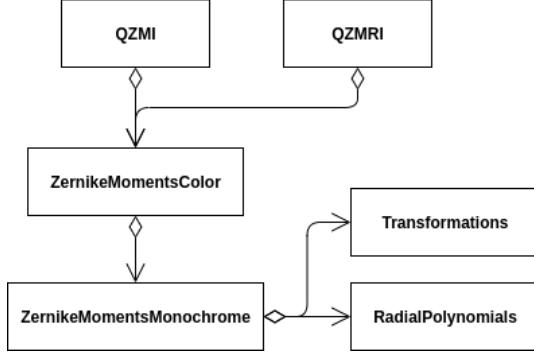


Figure 4.1: The relationships between the classes used to calculate QZMIs and QZMRIs

quaternions, it is possible to calculate the real and the imaginary parts of the quaternion-valued QZMs individually by using some linear combination of the real and imaginary parts of the complex-valued, single channel Zernike moments. This means that the single channel Zernike moments have to be calculated for all three of the RGB color channels. Furthermore, the calculation of the Zernike moments requires the computation of the radial polynomials  $R_{n,m}$ , introduced in (2.5).

To calculate the required values, the following four classes were created, each relying on the next one to perform the computation.

- **QZMI**, for calculating the quaternion Zernike moment invariants
- **ZernikeMomentsColor**, for calculating the quaternion-valued Zernike moments
- **ZernikeMomentsMonochrome**, for calculating the complex-valued, single channel Zernike moments
- **RadialPolynomials**, for calculating the values of the radial polynomials at a given point

The relationships between these classes are shown on Figure 4.1, as well as the **QZMRI** class, which calculates QZMs, which provide only rotation invariance, needed for some test cases. A description of the algorithms used in these classes and the data stored by them is given below.

Since the work presented in this thesis involves changing the way an image is transformed from image coordinates to polar coordinates inside the unit circle, the classes were made modular with respect to the transformation used. This makes it easy to create and test a new image transformation function with the interface expected by the calculating classes.

## Radial polynomials

To calculate the value of all  $R_{n,m}$  radial polynomials up to some maximal degree  $P$  at a point  $r \in [0, 1]$ , the modified Kintner's method was used, as described in [33]. This recursive algorithm computes the value of  $R_{n,m}(r)$  for all  $0 \leq |m| \leq n \leq P$ , ( $n - |m|$  is even) with complexity  $\mathcal{O}(P^2)$ . Kintner's method is ideal for the precomputation of all radial polynomial values up to a maximum degree.

The modified Kintner's method utilized the recurrence relation

$$R_{n,m}(r) = \frac{(K_2 r^2 + K_3) R_{n-2,m}(r) + K_4 R_{n-4,m}(r)}{K_1},$$

where  $n \neq m$  and  $n - m \neq 2$ , and the  $K_i$ ,  $i = 1, 2, 3, 4$  coefficients are defined as

$$\begin{aligned} K_1 &= \frac{(n+m)(n-m)(n-2)}{2} \\ K_2 &= 2n(n-1)(n-2) \\ K_3 &= -m^2(n-1) - n(n-1)(n-2) \\ K_4 &= \frac{-n(n+m-2)(n-m-2)}{2}. \end{aligned}$$

In the cases where  $n = m$  or  $n - m = 2$  the following formulas can be used:

$$\begin{aligned} R_{n,n}(r) &= r^n \\ R_{n,n-2}(r) &= nR_{n,n}(r) - (n-1)R_{n-2,n-2}(r). \end{aligned}$$

Since this routine is called many times during the calculation of Zernike moments, just-in-time compilation was used to further increase efficiency. Figure 4.2 shows the JIT-enabled function.

## Complex Zernike moments

The `ZernikeMomentsMonochrome` class calculates the conventional Zernike moments of degree at most  $P$  of a square  $N \times N$ , single channel (grayscale) image. The algorithm is based directly on the discretized definition of the Zernike moments. For the original

```

1 @jit(void(float64, int32, float64[:, :], nopython=True)
2 def calculateRadialPolynomials(r, P, values):
3     values[0, 0] = 1
4     values[1, 1] = r
5     for n in range(2, P + 1):
6         h = n * (n - 1) * (n - 2)
7         K2 = 2 * h
8         values[n, n] = (r ** n)
9         values[n, n-2] = n * values[n, n] - (n-1) * values[n-2, n-2]
10        for m in range(n-4, -1, -2):
11            K1 = (n + m) * (n - m) * (n - 2) / 2
12            K3 = (-1) * m * m * (n - 1) - h
13            K4 = (-1) * n * (n + m - 2) * (n - m - 2) / 2
14            r2 = r ** 2
15            values[n, m] = ((K2 * r2 + K3) * values[n-2, m] + K4 * values[n-4, m]) / K1

```

Figure 4.2: Function for calculating radial polynomial values

linear transformation of the image onto the unit disk, this gives:

$$\begin{aligned}
Z_{n,m}(f) &= \lambda \frac{(n+1)}{(N-1)^2} \sum_{x=1}^N \sum_{y=1}^N f(x,y) V_{n,m}^*(r_{x,y}, \theta_{x,y}) \\
&= \lambda \frac{(n+1)}{(N-1)^2} \sum_{x=1}^N \sum_{y=1}^N f(x,y) R_{n,m}(r_{x,y}) e^{-im\theta_{x,y}} \\
&= \lambda \frac{(n+1)}{(N-1)^2} \sum_{x=1}^N \sum_{y=1}^N f(x,y) R_{n,m}(r_{x,y}) (\cos(m\theta_{x,y}) - i \sin(m\theta_{x,y})),
\end{aligned}$$

where  $0 \leq |m| \leq n \leq P$ ,  $n - |m|$  is even,  $(r_{x,y}, \theta_{x,y})$  are the polar form of the  $(x, y)$  coordinates transformed to the unit disk,  $\lambda$  is the scaling parameter also given by the transformation (as described in Section 2.3) and  $f$  is the real-valued, grayscale image.

By precomputing the sine and cosine values for all possible combinations of the values of  $m$  and  $\theta_{x,y}$ , as well as precomputing the values of the radial polynomials, this formula gives an efficient way to calculate the real and imaginary parts of the Zernike moments separately. This way only primitive data types have to be used during the computation, hence it can be made more efficient using JIT.

Also, there is no need to calculate the Zernike moments for  $m < 0$  values, as the  $Z_{n,m}(f) = Z_{n,-m}(f)^*$  identity can be used later to obtain the values for  $m < 0$ .

## Quaternion Zernike moments

The class `ZernikeMomentsColor` calculates the quaternion Zernike moments of an RGB image. First, the conventional Zernike moments for each of the three color channels are calculated, then the relationship between QZMs and Zernike moments is applied to construct the quaternions [2]:

$$\begin{aligned} Z_{n,m}^R(f) = & -\frac{1}{\sqrt{3}} (\operatorname{Im}(Z_{n,m}(f_R)) + \operatorname{Im}(Z_{n,m}(f_G)) + \operatorname{Im}(Z_{n,m}(f_B))) \\ & + \left[ \operatorname{Re}(Z_{n,m}(f_R)) + \frac{1}{\sqrt{3}} (\operatorname{Im}(Z_{n,m}(f_G)) - \operatorname{Im}(Z_{n,m}(f_B))) \right] \mathbf{i} \\ & + \left[ \operatorname{Re}(Z_{n,m}(f_G)) + \frac{1}{\sqrt{3}} (\operatorname{Im}(Z_{n,m}(f_B)) - \operatorname{Im}(Z_{n,m}(f_R))) \right] \mathbf{j} \\ & + \left[ \operatorname{Re}(Z_{n,m}(f_B)) + \frac{1}{\sqrt{3}} (\operatorname{Im}(Z_{n,m}(f_R)) - \operatorname{Im}(Z_{n,m}(f_G))) \right] \mathbf{k}, \end{aligned} \quad (4.1)$$

where  $f$  is an RGB image,  $f_R, f_G, f_B$  are the red, green and blue color channels respectively and  $Z_{n,m}^R$  denote the right-side quaternion Zernike moments, while  $Z_{n,n}$  denote the conventional Zernike moments.

Again, only the QZMs  $Z_{n,m}^R$  ( $m \geq 0$ ) are calculated, because for  $m < 0$  the  $Z_{n,m}^R(f) = Z_{n,-m}^R(f)^*$  equality can be used.

## Invariants

The class `QZMI` is responsible for computing the combined rotation, scaling and translation (RST) invariant moments, while the class `QZMRI` computes the moments which are invariant only to rotation. The invariants for scaling and rotation are calculated directly using the QZMs, based on the formulas described in Section 2.2.

To achieve translation invariance, the common centroid (introduced in (2.15)) of the RGB image is calculated based on the formulas described by Suk and Flusser [23].

$$\{x_c, y_c\} = \left\{ \frac{M_{10}(f_R) + M_{10}(f_G) + M_{10}(f_B)}{M_{00}(f_R) + M_{00}(f_G) + M_{00}(f_B)}, \frac{M_{01}(f_R) + M_{01}(f_G) + M_{01}(f_B)}{M_{00}(f_R) + M_{00}(f_G) + M_{00}(f_B)} \right\},$$

where  $f_R, f_G, f_B$  are the grayscale images corresponding to the red, green and blue color channels respectively, and  $M_{10}, M_{01}, M_{00}$  are the regular image moments introduced in (2.1). The original image  $f$  is then translated in image coordinates such that the origin falls on the common centroid  $\{x_c, y_c\}$ , and later on scaling and rotation invariants are calculated based on this translated image.

### Quaternion radial harmonic Fourier moments

QRHFs can be calculated using the conventional radial harmonic Fourier moments. This is achieved using a similar formula to the one in (4.1).

As described in Section 2.2, rotation invariants can be constructed by taking modulus of each moment (as used by Liu et al. [8]). However, in order to ensure fair comparison between the capabilities of QZMs and QRHFs, we construct rotation invariants in a similar way to the construction in (2.14) for QZMs.

By calculating the product of left- and right-side QRHFs we get quaternion-valued invariants instead of real-valued invariants. This construction provides rotation invariance, while ensuring that any difference in the capabilities of the methods does not originate from the difference in the construction of the invariants.

## 4.3 New image transformation

The new image transformation, as described in Chapter 3, requires the calculation of the roots of the  $n^{th}$  degree Legendre polynomials  $P_n$ , as well as calculating the integrals of the Lagrange basis polynomials over the roots of  $P_n$ . Furthermore, since when applying the linear transformation from image coordinates to polar coordinates, the pixels of the image do not fall exactly on any point of the new discrete point system, some interpolating methods have to be used to approximate the image values at these points.

Because of the modularity of the previously described classes, it is possible to swap the old transformation with the new one. With some minor modifications during the calculation of the conventional Zernike moments – because of the new discretization formula containing a different measure – the previous classes can be used to obtain the QZMIs using the new method of discretization.

### Roots of Legendre polynomials

The roots of the Legendre polynomial  $P_n$  are essential for the calculation of the new point system. An explicit formula for the roots does not exist, thus an efficient and fast iterative algorithm is utilized to calculate these roots.

The  $n^{th}$  degree Legendre polynomial  $P_n$  satisfies the following differential equation:

$$(1 - x^2)P_n''(x) - 2xP_n'(x) + n(n + 1)P_n(x) = 0. \quad (4.2)$$

A fast algorithm for calculating the roots of  $P_n$ , based on this differential equation was presented by Glaser et al. [51]. The algorithm uses a second-order Runge-Kutta method (namely the midpoint method) to solve (4.2) for some given initial condition. A first approximation for a root of  $P_n$  can be obtained from the solution of the initial value problem. This approximation is then further refined by Newton's method. Subsequent roots can be calculated using the same method but starting from different initial conditions defined by the previous root.

In practice, this algorithm calculates the roots of  $P_n$  with accuracy up to machine precision in only just a few, predefined number of iterations for both the Runge-Kutta and the Newton's method.

### Computing the discrete measure

To calculate the Zernike moments over the discrete orthogonal point system, apart from the  $\lambda_{n,m} = n+1$  scaling factor, a weight function is used as well (as described in (2.17)). To calculate these  $w(r_{k,N}, \theta_{j,N})$  weight values (defined in (3.1)), the following integral has to be calculated:

$$\mathcal{A}_{k,N} = \int_{-1}^1 \ell_{k,N}(x) dx,$$

where  $\ell_{k,N}$  is the  $k^{th}$  Lagrange basis polynomial corresponding to the roots of the Legendre polynomial  $P_N$ , meaning that

$$\ell_{k,N}(x) = \frac{(x - x_1)(x - x_2) \dots (x - x_{k-1})(x - x_{k+1}) \dots (x - x_{N-1})(x - x_N)}{(x_k - x_1)(x_k - x_2) \dots (x_k - x_{k-1})(x_k - x_{k+1}) \dots (x_k - x_{N-1})(x_k - x_N)},$$

where  $x_i$ ,  $i = 1, \dots, N$  are the roots of  $P_N$ .

The Gauss-Legendre quadrature is a numerical method used for approximating the definite integral of a function over the interval  $[-1, 1]$ . This method is based on the roots of the Legendre polynomial  $P_n$  of degree  $n$ . Let  $x_1, x_2, \dots, x_n$  denote the roots of  $P_n$ . The quadrature provides the following approximation of a definite integral:

$$\int_{-1}^1 f(x) dx \approx \sum_{k=1}^n w_k f(x_k),$$

$$w_k = \frac{2}{(1 - x_k^2)(P'_n(x_k))^2}.$$

The quadrature is exact for all polynomials whose degree is at most  $2n - 1$ . Now  $\ell_{k,N}(x)$  is an  $N - 1$  degree polynomial, so the Gauss-Legendre quadrature with  $n = N$  is exact for  $\ell_{k,N}$ . Furthermore,  $\ell_{k,N}$  is defined such that  $\ell_{k,N}(x_i) = 0$  for  $i \neq k$ , and  $\ell_{k,N}(x_k) = 1$

Thus:

$$\int_{-1}^1 \ell_{k,N}(x) dx = w_k = \frac{2}{(1 - x_k^2)(P'_n(x_k))^2}.$$

Since the roots of the Legendre polynomial  $P_N$  must be computed to obtain the point system, this formula gives an easy and fast way to calculate the exact values of the weight function  $w$ .

Additionally, the values of the derivative  $P'_n$  are needed at the point  $x_k$ . For this the recurrence relations presented by Glaser et al. [51] are used:

$$\begin{aligned} P_{n+1}(x) &= \frac{2n+1}{n+1}xP_n(x) - \frac{n}{n+1}P_{n-1}(x) \\ P'_{n+1}(x) &= \frac{2n+1}{n+1}(xP'_n(x) + P_n(x)) - \frac{n}{n+1}P'_{n-1}, \end{aligned}$$

with the initial values of  $P_{-1}(x) = P'_{-1}(x) = P'_0(x) = 0$  and  $P_0(x) = 1$ .

## Image interpolation

After obtaining the discrete orthogonal point system, the values of the image function have to be approximated at each point, because the original pixel values do not fall exactly on the new points. Now we describe some reasonable ways of resampling the image data to fit these new points.

First, the image is linearly transformed onto the unit disk using the transformation shown on Figure 2.2, so that the transformed image covers the entire unit disk. This transformation is used as opposed to the one on Figure 2.1, because the point system covers the entire unit disk, so the image also has to cover the whole disk.

There are two ways to approximate the values at each point, depending on the number of points in the discrete orthogonal point system.

**Approximately the same number of points as pixels.** If the number of points is approximately the same as the number of pixels in the image, then bilinear interpolation can be used for each point.

First, the four pixels of the image that are closest to the given point  $(x, y)$  are determined. These four points form a square. Then, the weighted average of these points is calculated along the  $x$  axis, giving approximate function values at two points, which only differ in their  $y$  coordinate. Finally, the weighted average of the function values at these points is

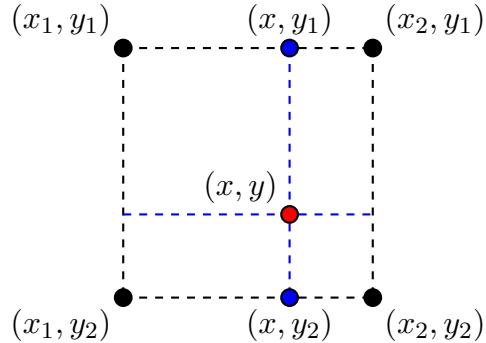


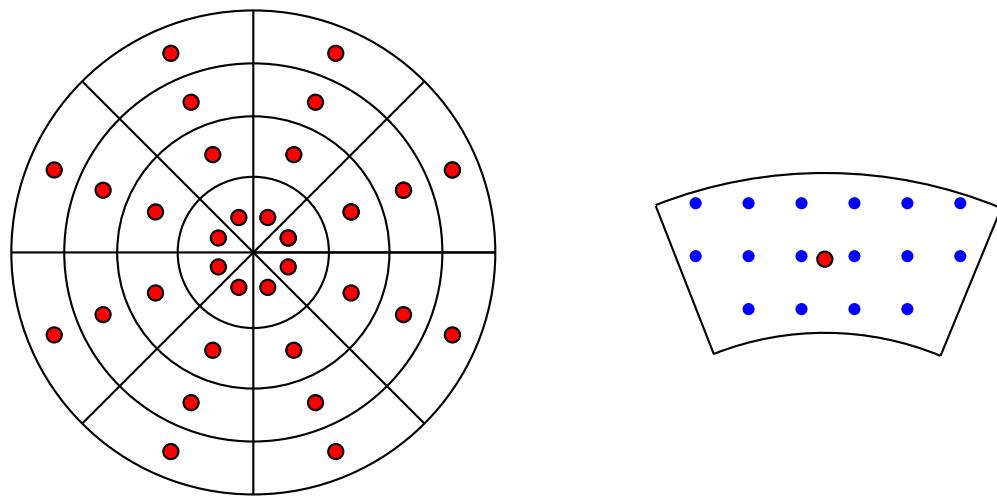
Figure 4.3: The point  $(x, y)$  and its neighboring points, which are used to approximate  $f(x, y)$  by bilinear interpolation

calculated to get the approximation of  $f(x, y)$ . The points involved in this interpolation are shown on Figure 4.3.

**Much fewer points than the number of pixels.** If the number of points is far below the number of pixels, then many of the original pixel values would not be represented in the final approximate function values when using bilinear interpolation. Therefore, the following algorithm is used to approximate function values using discrete integration.

First, assign each pixel of the original image to the point which is the closest to it in the discrete orthogonal system. This results in a set of pixels assigned to each point. Then, for each point in the discrete orthogonal point system, calculate the average of this set of pixels. This algorithm divides the unit disk into annular sections based on proximity to the new points, meaning that the set of pixels belonging to a point form annular sections. This is shown on Figure 4.4.

The comparison between one of the original transformations and the two methods for approximating function values is shown on Figure 4.5 on the peppers image from the USC-SIPI Image Database [52].



(a) The point system on the unit disk and the annular sectors over which the pixel values are averaged.

(b) A single annular sector with the original image pixels (blue) and the point in which the approximate function value has to be calculated (red).

Figure 4.4

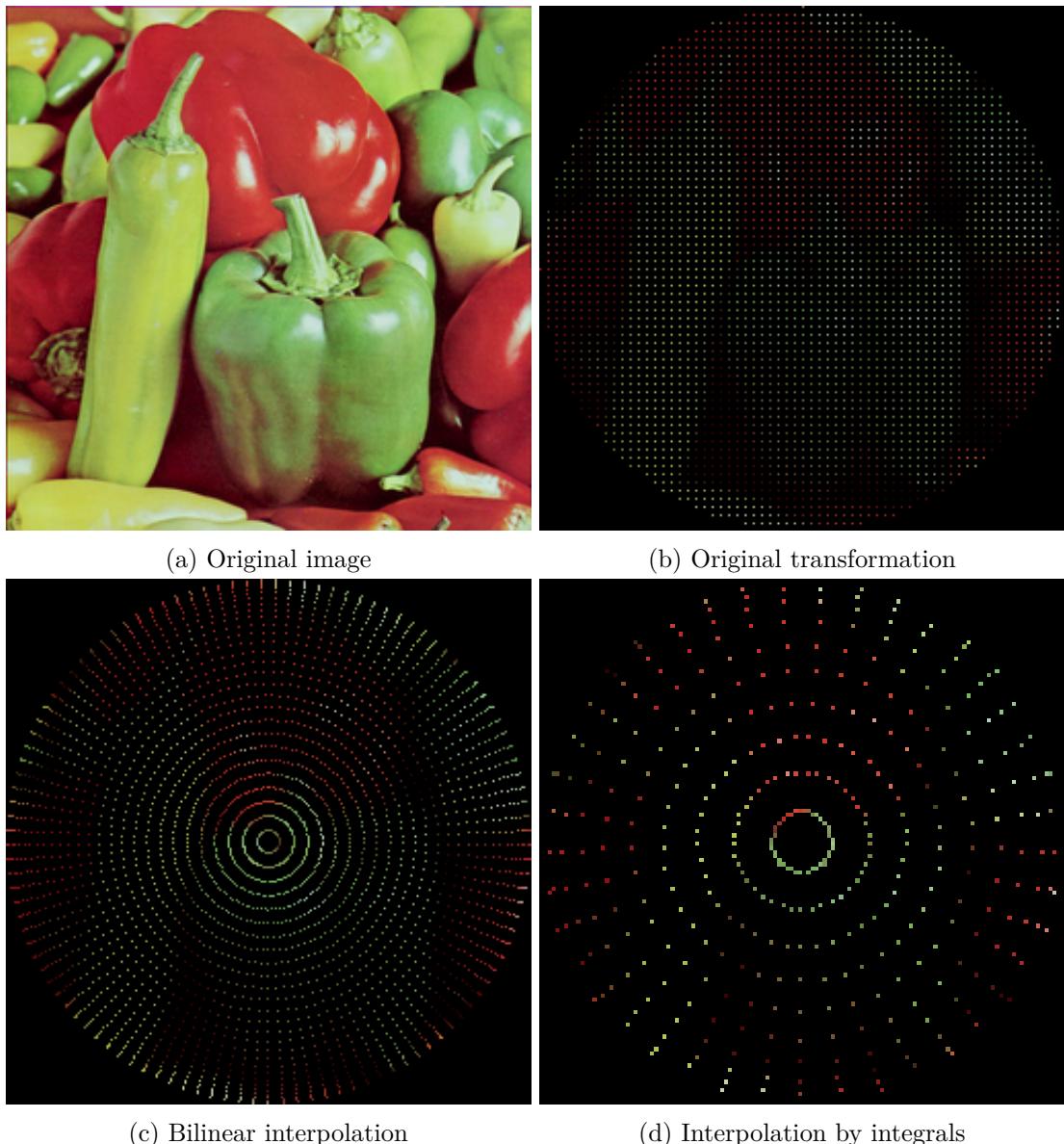


Figure 4.5: Different transformations with different interpolation methods of the peppers image.

# Chapter 5

## Tests and comparison with the original method

In this chapter we present the different tests that were performed to compare the capabilities of the old and new methods. For each test, the results of both methods of discretization are compared.

In total, four kinds of tests were conducted:

- Invariance
- Image reconstruction
- Image recognition
- Template matching

Finally, the methods were compared with quaternion radial harmonic Fourier moments in terms of image reconstruction and image recognition capabilities.

### 5.1 Test images

The images for testing were acquired from multiple online image libraries. The Lenna and Pepper images [52] (shown on Figure 5.1) were used to test image reconstruction as well as to demonstrate the different point systems.



(a)



(b)

Figure 5.1: The Lenna and Pepper images

For testing image recognition, two sets of images were used. The first set consists of 14 images chosen from the Columbia Object Image Library (COIL-100) [9], shown on Figure 5.2. These images are originally  $128 \times 128$  pixels, but they were placed on a  $204 \times 204$  black background so that the rotated, scaled and translated versions of the images stay completely within these dimensions.

A set of 1008 rotated images was created by rotating each of the 14 images by a degree  $\alpha \in \{0, 5, 10, \dots, 350, 355\}$ . Some examples of the extended and rotated images are shown on Figure 5.3.

Another set of 1176 rotated, scaled and translated images was created by translating each image by -11 pixels in the  $x$  direction and 9 pixels in the  $y$  direction. Then the translated images were rotated by  $\alpha \in \{0, 30, 60, \dots, 300, 330\}$ . Finally, each rotated and translated image was scaled by  $\lambda \in \{0.5, 0.75, \dots, 1.75, 2\}$ . When either scaling or rotation required it, bilinear interpolation was used. Some examples of the RST transformed images are shown on Figure 5.4.

Another set of 13 images was acquired from the Amsterdam Library of Object Images (ALOI) [10]. These are shown on Figure 5.5. Originally, the size of these images was  $768 \times 576$  pixels, but they were downsampled to  $96 \times 72$  and subsequently extended to  $152 \times 128$  by placing the images on a black background. Similarly to the test sets created using the COIL-100 images, the ALOI images were also translated, rotated and scaled, yielding a set of 1092 RST transformed images. The parameters of the transformation were the same as for the COIL-100 images, except for the translation, where  $\Delta x = 8$  and  $\Delta y = 5$  was used.

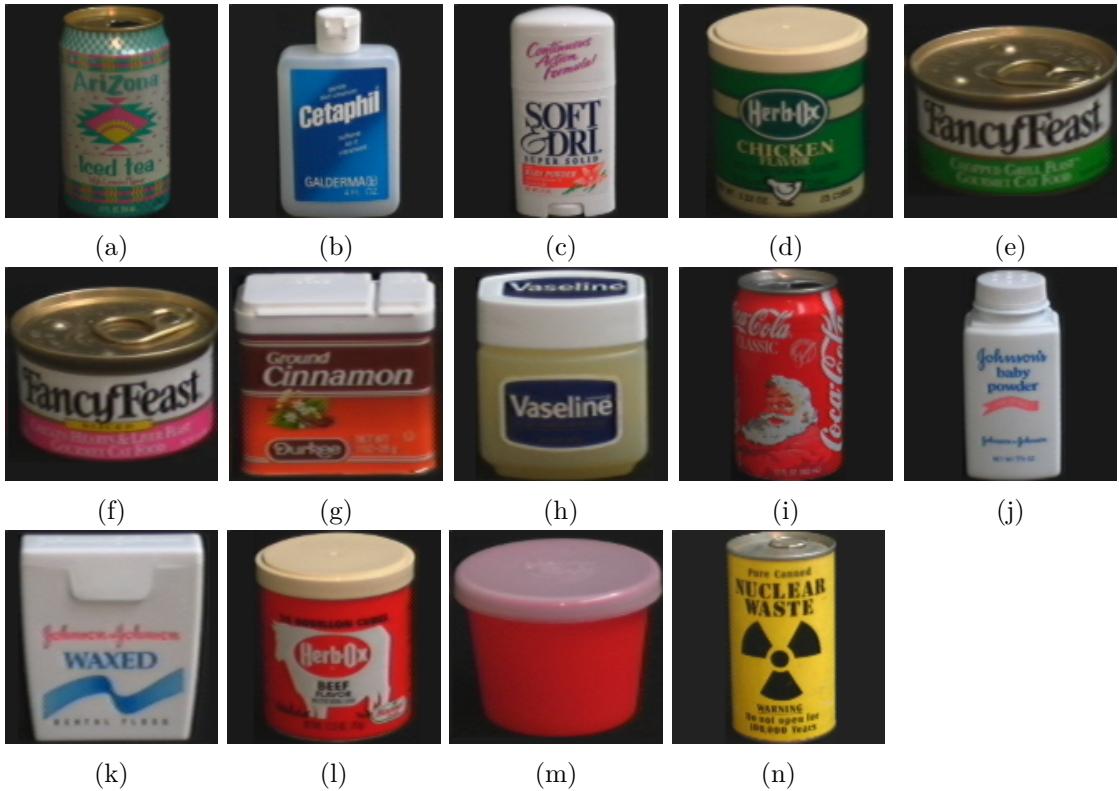


Figure 5.2: The 14 selected images from the Columbia Object Image Library (COIL-100)

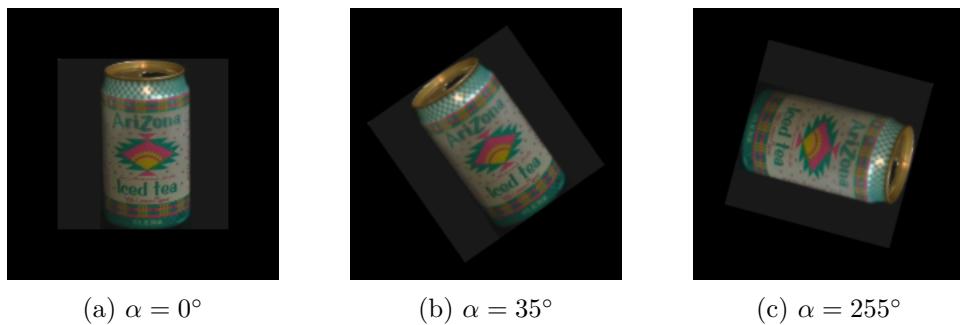


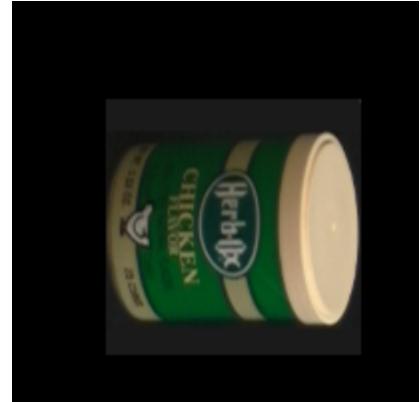
Figure 5.3: Some extended and rotated images from COIL



(a)  $\alpha = 0^\circ, \lambda = 1$



(b)  $\alpha = 150^\circ, \lambda = 0.5$



(c)  $\alpha = 270^\circ, \lambda = 1.5$

Figure 5.4: Some RST transformed images from COIL. All images are translated by  $\Delta x = -11, \Delta y = 9$

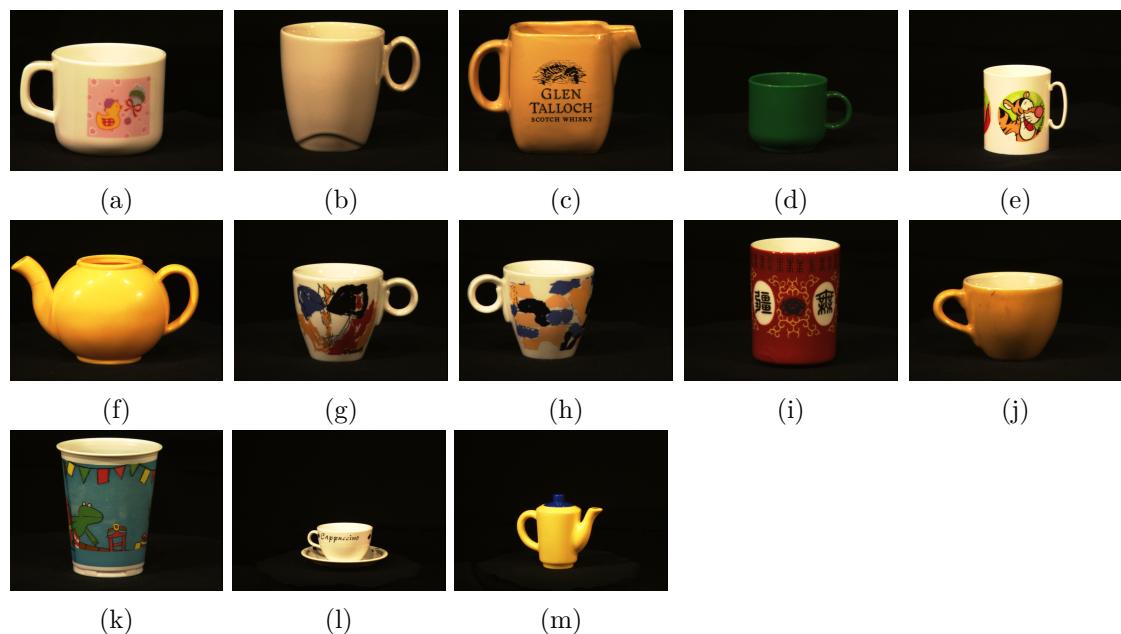


Figure 5.5: The 13 selected images from the Amsterdam Library of Object Images (ALOI)

## 5.2 Invariance test

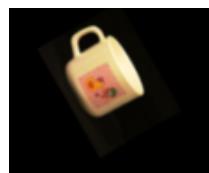
In order to test quaternion Zernike moment invariants with respect to rotation, scaling and translation, the QZMIs of order 1 to 4 were calculated for a given image and all of its (previously generated) RST transformations. Then, the modulus of these QZMIs was calculated, as well as the mean ( $\mu$ ), standard deviation ( $\sigma$ ) and  $\frac{\sigma}{\mu}$  for the same moments of all transformed images.

### Results

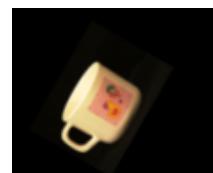
The moduli of the QZMIs for the transformed images on Figure 5.6 is shown in Table 5.1 for the old method of discretization and in Table 5.2 for the proposed method of discretization. The coefficient of variation ( $\frac{\sigma}{\mu}$ ) shows that using both methods, the moments are invariant to RST transformation. The only rows where  $\frac{\sigma}{\mu}$  is higher are the ones where the modulus of the moment is very close to zero, and thus small numerical errors impact this number significantly. Comparing the two methods, the proposed one yields slightly lower values for the coefficient of variation for all moments.



(a)



(b)



(c)



(d)



(e)



(f)

Figure 5.6: The images with different rotation and scale, which are used to test the invariance of the methods.

	Fig.5.6a	Fig.5.6b	Fig.5.6c	Fig.5.6d	Fig.5.6e	Fig.5.6f	$\frac{\sigma}{\mu}$
$ \bar{\Psi}_{1,1}^1 $	1.50e-2	1.58e-2	1.54e-2	1.61e-2	1.58e-2	1.49e-2	3.73%
$ \bar{\Psi}_{2,0}^0 $	2.965	2.964	2.964	2.963	2.964	2.965	0.028%
$ \bar{\Psi}_{2,2}^0 $	8.793	8.785	8.789	8.783	8.786	8.794	0.057%
$ \bar{\Psi}_{2,2}^2 $	1.50e-4	1.65e-4	1.57e-4	1.69e-4	1.65e-4	1.47e-4	6.87%
$ \bar{\Psi}_{3,1}^1 $	5.91e-2	6.25e-2	6.09e-2	6.34e-2	6.22e-2	5.89e-2	3.71%
$ \bar{\Psi}_{3,3}^1 $	0.233	0.247	0.241	0.250	0.246	0.233	3.69%
$ \bar{\Psi}_{3,3}^3 $	1.43e-6	1.61e-6	1.50e-6	1.67e-6	1.62e-6	1.38e-6	9.40%
$ \bar{\Psi}_{4,0}^0 $	4.827	4.821	4.824	4.820	4.822	4.828	0.086%
$ \bar{\Psi}_{4,2}^0 $	14.316	14.291	14.302	14.284	14.292	14.318	0.114%
$ \bar{\Psi}_{4,2}^2 $	7.39e-4	8.13e-4	7.74e-4	8.35e-4	8.12e-4	7.26e-4	6.85%
$ \bar{\Psi}_{4,4}^0 $	23.309	23.249	23.276	23.232	23.251	23.314	0.171%
$ \bar{\Psi}_{4,4}^2 $	3.64e-3	4.00e-3	3.81e-3	4.11e-3	4.00e-3	3.58e-3	6.83%
$ \bar{\Psi}_{4,4}^4 $	1.28e-8	1.52e-8	1.39e-8	1.59e-8	1.51e-8	1.25e-8	12.00%

Table 5.1: The modulus of QZMIs using the old method for discretization. Note that  $\frac{\sigma}{\mu}$  was calculated using the QZMIs for all transformation of the image, not just the values shown in the table.

	Fig.5.6a	Fig.5.6b	Fig.5.6c	Fig.5.6d	Fig.5.6e	Fig.5.6f	$\frac{\sigma}{\mu}$
$ \bar{\Psi}_{1,1}^1 $	1.49e-2	1.58e-2	1.54e-2	1.60e-2	1.57e-2	1.49e-2	3.72%
$ \bar{\Psi}_{2,0}^0 $	2.965	2.964	2.964	2.963	2.964	2.965	0.028%
$ \bar{\Psi}_{2,2}^0 $	8.793	8.785	8.789	8.783	8.786	8.793	0.056%
$ \bar{\Psi}_{2,2}^2 $	1.49e-4	1.64e-4	1.55e-4	1.68e-4	1.63e-4	1.47e-4	6.82%
$ \bar{\Psi}_{3,1}^1 $	5.90e-2	6.25e-2	6.08e-2	6.33e-2	6.21e-2	5.89e-2	3.70%
$ \bar{\Psi}_{3,3}^1 $	0.233	0.247	0.240	0.250	0.245	0.233	3.68%
$ \bar{\Psi}_{3,3}^3 $	1.41e-6	1.61e-6	1.48e-6	1.65e-6	1.60e-6	1.37e-6	9.32%
$ \bar{\Psi}_{4,0}^0 $	4.828	4.821	4.824	4.820	4.822	4.828	0.085%
$ \bar{\Psi}_{4,2}^0 $	14.317	14.291	14.304	14.286	14.293	14.318	0.113%
$ \bar{\Psi}_{4,2}^2 $	7.35e-4	8.13e-4	7.68e-4	8.29e-4	8.07e-4	7.25e-4	6.80%
$ \bar{\Psi}_{4,4}^0 $	23.312	23.249	23.279	23.236	23.254	23.314	0.170%
$ \bar{\Psi}_{4,4}^2 $	3.62e-3	4.00e-3	3.79e-3	4.09e-3	3.98e-3	3.58e-3	6.78%
$ \bar{\Psi}_{4,4}^4 $	1.28e-8	1.52e-8	1.37e-8	1.56e-8	1.50e-8	1.24e-8	11.94%

Table 5.2: The modulus of QZMIs using the new method for discretization. Note that  $\frac{\sigma}{\mu}$  was calculated using the QZMIs for all transformation of the image, not just the values shown in the table.

### 5.3 Image reconstruction

As described in Section 2.2, the quaternion Zernike moments of an image can be used to approximately reconstruct the original image using the formulas in (2.13). This reconstruction requires the computation of QZMs of up to a finite degree  $M$ .

In order to reconstruct the discrete image, the formula

$$f(r_{x,y}, \theta_{x,y}) \approx \sum_{n=0}^M \sum_{m=-n}^n Z_{n,m}^R(f) R_{n,m}(r_{x,y}) e^{\mu m \theta_{x,y}}$$

was used for each pixel with image coordinates  $(x, y)$ , where  $(r_{x,y}, \theta_{x,y})$  are the polar coordinates obtained by performing the linear transformation of the image onto the unit disk, using the transformation shown on Figure 2.1 ( $tf_1$ ) for the old method and the transformation shown on Figure 2.2 ( $tf_2$ ) for the new one. The reason for this difference in transformation is that for the proposed discrete orthogonal point system, the interpolated pixel values are calculated using  $tf_2$ , while conventionally for (quaternion) Zernike moments,  $tf_1$  is used [2].

To measure the error of the reconstruction the normalized mean squared error ( $\varepsilon^2$ ) was used. If  $f(x, y)$  is the original and  $\hat{f}(x, y)$  is the reconstructed image, both with size  $N \times N$ , then the normalized mean squared error is defined as:

$$\varepsilon^2 = \frac{\sum_{x=1}^N \sum_{y=1}^N |f(x, y) - \hat{f}(x, y)|^2}{\sum_{x=1}^N \sum_{y=1}^N |f(x, y)|^2}.$$

In the cases, where  $tf_2$  is used only the part of the image falling inside the unit circle is reconstructed and thus the mean squared error is calculated over this part of the image only.

For the new method of discretization the number of points on the unit disk was chosen to be approximately the same as the number of pixels falling inside the inscribed circle of the image, or the minimum required to satisfy the discrete orthogonality relation (from Theorem 1) if the number of pixels is too few.

## Results

Image reconstruction was performed for both the Lenna and the Pepper images [52], using image sizes ranging from  $64 \times 64$  up to  $256 \times 256$ . QZMs of up to degree 350

M	50	100	150	250	350
old method $\varepsilon^2$	0.02659	0.01341	0.00868	0.00428	0.00279
new method $\varepsilon^2$	0.01611	0.00789	0.00460	0.00164	0.00066
change	-39.4%	-41.1%	-47.0%	-61.7%	-76.3%

Table 5.3: Comparison of the normalized mean squared errors between the two method, for the  $256 \times 256$  Lenna image.

were used to reconstruct the image. In the case of smaller images, for example a  $64 \times 64$  image, the reconstruction error of the conventional method starts to increase after QZMs with degree more than 100 are used, as using higher degrees results in trying to extract more information from the original image than there is information contained in the pixel values, thus increasing the error of reconstruction. Because of the interpolation used by the proposed method, this issue does not affect it. As long as the number of points  $N$  in the discrete orthogonal system satisfies the constraints of Theorem 1, the reconstruction error decreases as moments of higher degree are used.

Figure 5.7 shows some of the reconstructed images and their normalized mean squared error when using the old method of discretization. The previously mentioned phenomenon is visible on this figure, for example in the  $M > 100$  cases for the  $64 \times 64$  Lenna image. Figure 5.8 shows the same data using the new method. In this case the reconstruction error is always smaller when more moments are used.

Comparing the error of reconstruction between the two methods, using the discrete orthogonal point system provides much lower normalized mean squared errors for all levels of reconstruction. Table 5.3 shows the comparison between the  $\varepsilon^2$  values and the change in the error when using the proposed method instead of the old one. The decrease in the error of reconstruction is significant for both low and high maximal degrees of QZMs.

M	Lenna 64 × 64	Lenna 128 × 128	Lenna 256 × 256	Pepper 256 × 256
Original				
M = 50				
	0.01257	0.01948	0.02659	0.03777
M = 100				
	0.00468	0.00719	0.01341	0.01596
M = 150				
	0.02076	0.00388	0.00868	0.00885
M = 250				
	0.57565	0.00823	0.00428	0.00378
M = 350				
	0.97677	0.02695	0.00279	0.00253

Figure 5.7: Reconstructed images using the old method, with the normalized mean squared error shown below each image.

M	Lenna 64 × 64	Lenna 128 × 128	Lenna 256 × 256	Pepper 256 × 256
Original				
M = 50				
	0.00366	0.00901	0.01611	0.01778
M = 100				
	0.00066	0.00238	0.00791	0.00682
M = 150				
	0.00027	0.00083	0.00461	0.00347
M = 250				
	0.00011	0.00030	0.00164	0.00111
M = 350				
	0.00005	0.00013	0.00066	0.00048

Figure 5.8: Reconstructed images using the proposed new method, with the normalized mean squared error shown below each image.

## 5.4 Image recognition

In order to test image recognition capabilities, the COIL-100 [9] and ALOI [10] images and the respective RST transformed sets of images were used. The exact parameters for the transformations performed to obtain these sets of images is described in Section 5.1.

During the image recognition test, the goal is to correctly identify the original (non-transformed) image based on the invariant moments of a transformed image. For each method we calculate the percentage of correctly recognized images.

To recognize an image, first the QZMIs have to be calculated. In this thesis, we used QZMIs of up to degree 4, but not all possible QZMIs were used. The selected QZMIs were:  $\bar{\Psi}_{1,1}^1, \bar{\Psi}_{2,0}^0, \bar{\Psi}_{2,2}^2, \bar{\Psi}_{3,1}^1, \bar{\Psi}_{3,3}^3, \bar{\Psi}_{4,0}^0, \bar{\Psi}_{4,2}^2, \bar{\Psi}_{4,4}^4$ . These 8 quaternion valued invariants contain a total of 20 real valued invariants, since while each quaternion could provide 4 real invariants, the QZMIs  $\bar{\Psi}_{n,k}^m$  with  $n = k$  necessarily have  $\text{Im } \bar{\Psi}_{n,n}^m = 0$ . This follows directly from the definition (see Section 2.2):

$$\bar{\Psi}_{n,n}^m = \bar{L}_{n,m}^R(f)(\bar{L}_{n,m}^R(f))^* = \left| \bar{L}_{n,m}^R(f) \right|^2 \in \mathbb{R}.$$

Thus from the selected 8 QZMIs four provide a single real-valued invariant and another four provide 4 real-valued invariants. These 20 real invariants are then used to construct a vector  $I$  of length 20. This vector is normalized using the method presented by Suk and Flusser [23]:

$$I_k = \text{sgn}(I_k) |I_k|^{\frac{1}{2}} \quad (k = 1, 2, \dots, 20).$$

This invariant vector is calculated for an RST transformed image and all of the original images. Then the minimum Euclidean distance is used to choose from the original images. The recognition algorithm classifies the transformed image as this chosen image.

### Noise generation

To test robustness against noise the tests were also performed with different levels of real-valued Gaussian noise added to all images. The parameters of the Gaussian noise were as follows: the mean of the distribution was always at 0, while standard deviation ( $\sigma$ ) values from 1 up to 60 were used for the QZMIs and standard deviation ranging from 40 to 120 was used to test the (rotation invariant) QZMRIs. QZMRIs are more robust against noise than QZMIs, so higher noise values could be used. Apart from Gaussian noise, robustness against salt-and-pepper noise was also tested. Salt-and-pepper noise was added with densities ( $p$ ) ranging from 0.2% up to 15% for the QZMIs and ranging from 5% to 30% for the QZMRIs. Some examples of both types of noise are shown on Figure 5.9.

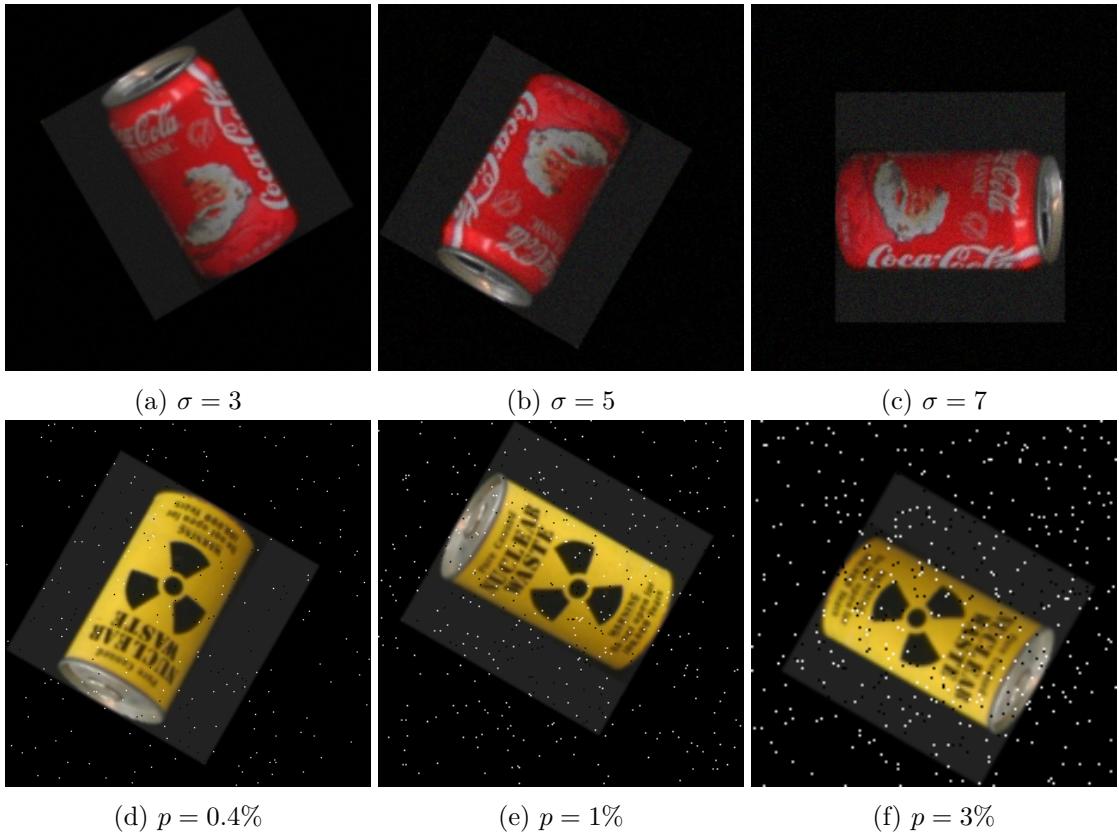


Figure 5.9: Some test images with Gaussian noise (top row) or salt-and-pepper noise (bottom row) with different parameters added to them.

The tests were performed directly using the raw, noisy images, without any kind of filtering applied to the image. The goal was to see the robustness of these methods against noise.

### Test cases

The recognition tests were performed using both the old and the proposed method for discretization. Furthermore, the new method was tested with two different  $N$  values for creating the discrete orthogonal system of  $8N^2$  points.

In the first case  $N$  was chosen such that the total number of points in the system should

be approximately the same as the pixels of the image falling inside its inscribed circle.

$$\begin{aligned} N &\in \mathbb{N} \\ 8N^2 &\approx n^2 \frac{\pi}{4} \\ N &= \left\lceil 2n\sqrt{\pi} \right\rceil. \end{aligned}$$

where the size of the image is  $n \times n$ . Based on the interpolating methods described in Section 4.3 this means that bilinear interpolation should be used.

In the other case  $N$  was chosen such that the number of points is close to the minimum required to achieve discrete orthogonality of moments up to degree 4. In this case  $N = 10$  was chosen. Since the number of points is far less than the number of pixels, based on Section 4.3 the interpolation should be done using the integral interpolation formula.

Note that for the old method, the number of points used is the same as the number of pixels in the image.

Three sets of transformed images were tested, which were generated as described in Section 5.1:

- COIL-100 rotated (1008 images)
- COIL-100 rotated, scaled, translated (1176 images)
- ALOI rotated, scaled, translated (1092 images)

For the set where only rotation was applied to the images, instead of using QZMIs the rotation invariant QZMRIs were used.

## Results

When the images have Gaussian noise, the new method with the higher number of points performs the best for all levels of noise, far outperforming the original method in recognition capabilities. The rate of recognition for Gaussian noise in case of the RST transformed image sets is shown in Table 5.4. The recognition rate of the new method remains relatively high (around 80%) even for high noise values, while the recognition rate of the original method drops significantly (below 20% in some cases).

When using only the minimal possible number of points in the new method, for all noise levels the recognition rate remains roughly around the same level as with the original

<b>Image set</b>	<b>Noise stdev.</b>	<b>Old method (%)</b>	<b>New method many points (%)</b>	<b>New method few points (%)</b>
COIL - RST	No noise	99.06	99.15	98.21
	1	98.98	99.49	98.81
	2	98.98	99.74	98.81
	3	98.55	99.83	98.04
	5	95.15	99.49	94.64
	7	95.15	98.72	91.67
	9	76.87	98.47	89.20
	40	52.89	88.52	51.87
	50	48.21	84.10	45.07
	60	41.58	85.80	39.12
ALOI - RST	No noise	99.91	100.00	94.60
	1	94.51	99.08	86.63
	2	84.89	93.13	86.35
	3	78.85	88.55	74.81
	5	72.07	93.31	58.24
	7	63.28	94.23	61.81
	9	55.04	94.32	48.81
	40	18.41	90.84	15.29
	50	19.32	82.51	15.93
	60	13.19	84.89	13.83

Table 5.4: Percentage of images recognized correctly by QZMIs in the case of using the RST transformed image sets with added Gaussian noise.

method, but the computation required to achieve such levels of image recognition is much lower because of the lower number of points.

The same observation can be made about the QZMRIs used to recognize the rotated images. Even for extremely high noise values, the recognition rate of the new method remains well above 90%. The results for the QZMRIs are shown in Table 5.5.

The reason for such differences between the old and the new methods is that the old method uses a non-orthogonal discretization, which means some redundancy occurs among the QZMs. This leads to the value of the added noise (numerical errors) increasing when these QZMs are multiplied to construct the QZMIs. On the other hand, the new method uses a discrete orthogonal system, which means no redundancy between the QZMs and thus it more robust against Gaussian noise.

When salt-and-pepper noise is added to the images, the recognition capabilities of both methods are high, but no clear difference is visible between them. Even using the minimal possible number of points in the new method, the recognition rates are almost the same as with the other two methods. These same observation can be made about the recognition capabilities of the QZMRIs. The results for salt-and-pepper noise are shown for QZMIs and QZMRIs in Table 5.6 and Table 5.5 respectively.

The reason for no significant difference between the methods in the case of salt-and-pepper noise is that this kind of noise adds a high frequency component to the image. Choosing only moments with low order for image recognition behaves as a low-pass filter, since these moments are most sensitive to low frequency components. Thus both methods are able to handle salt-and-pepper noise well.

Noise type	Noise param.	Old method (%)	New method many points (%)	New method few points (%)
No noise		100.00	100.00	100.00
Gauss	40	91.37	99.90	92.56
	50	86.51	100.00	94.44
	60	84.42	99.60	93.35
	70	80.85	99.90	83.93
	80	76.98	98.12	80.45
	90	75.79	99.01	83.83
	100	64.98	98.91	79.46
	110	64.78	97.62	69.94
	120	68.55	96.23	72.12
Salt	5	100.00	100.00	100.00
	10	100.00	100.00	100.00
	15	100.00	100.00	100.00
	20	100.00	100.00	100.00
	25	100.00	100.00	100.00
	30	100.00	100.00	100.00

Table 5.5: Percentage of images recognized correctly by QZMRIs in the case of using the rotated COIL image set with either Gaussian or salt-and-pepper noise. The noise parameter means the standard deviation in case of the Gaussian noise, and the density in case of the salt-and-pepper noise.

<b>Image set</b>	<b>Noise density</b>	<b>Old method (%)</b>	<b>New method many points (%)</b>	<b>New method few points (%)</b>
COIL - RST	No noise	99.06	99.15	98.21
	0.2	99.66	99.32	94.98
	0.4	99.91	99.74	99.15
	0.6	99.91	99.91	99.40
	1	98.98	99.91	99.66
	2	99.66	93.96	99.74
	3	99.40	99.40	96.34
	5	97.87	94.90	97.87
	10	99.91	93.03	98.72
	15	99.91	93.20	97.87
ALOI - RST	No noise	99.91	100.00	94.60
	0.2	88.64	90.75	78.11
	0.4	86.08	91.30	80.95
	0.6	83.97	90.11	89.19
	1	95.97	94.60	94.51
	2	98.44	94.96	95.97
	3	97.61	96.06	97.25
	5	97.99	97.25	89.56
	10	98.44	87.27	93.22
	15	93.50	91.30	97.71

Table 5.6: Percentage of images recognized correctly by QZMIs in the case of using the RST transformed image sets with added salt-and-pepper noise.

## 5.5 Template matching

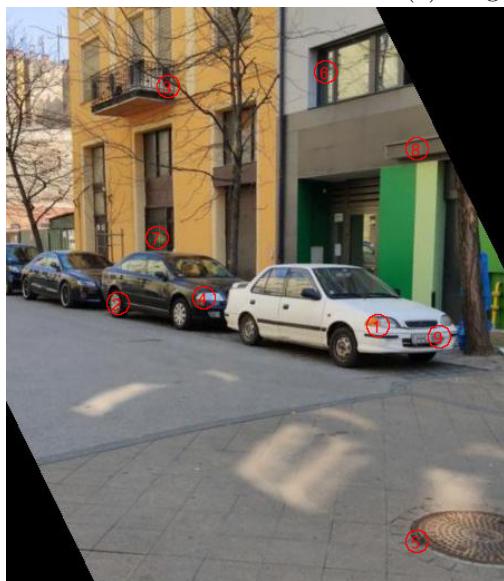
This final test aims to present some possible application of the pattern matching capabilities of the QZMIs. Some pictures were taken using a the camera on a Xiaomi Mi 9 smartphone with autofocus enabled. The images were taken of the same scene but with different focus and different rotation of the camera.

A total of 9 circles with radii of 10 pixels were chosen so that they represent some unique area on the pictures. Then, on another picture taken with different rotation and focus, after determining the scaling factor between the two images, a sliding circular window was moved across the second image. The invariant vector described in the image recognition test was constructed for each window, and then the minimal Euclidean distance was used to find each area of the original image on the second image.

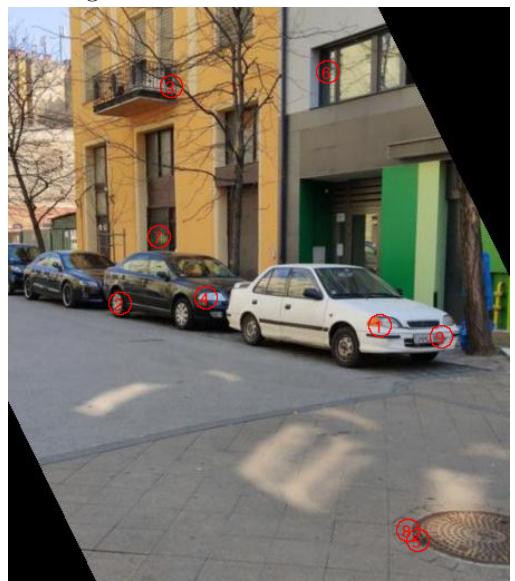
The original image and the results of both methods are shown on Figure 5.10. Both methods managed to correctly find 8 out of the 9 templates on the other image. Neither managed to correctly locate template number 8.



(a) Original image



(b) Old method



(c) New method

Figure 5.10: The original image with the templates and the result of both methods.

## 5.6 Comparisons with QRHFMs

Image reconstruction

Image recognition

# Chapter 6

## Conclusion

In this chapter we summarize the work and results presented in this thesis. We also present a short overview of further possibilities based on this work.

We have constructed a system of points on the unit disk, over which the quaternion extension of the Zernike functions is discrete orthogonal. Using this method of discretization, the quaternion Zernike moments are more robust to noise than with the previously used method.

Many tests have been performed on large sets of images to verify and quantify the improvements of the method. First, the invariance properties of the QZMIs have been verified empirically using the proposed method.

By comparing the image reconstruction capabilities of the original and the proposed method, we found that the mean square error of reconstruction can decrease significantly, by more than 50% in some cases, when using the proposed method.

Significant improvements in image recognition have also been achieved by the new method, especially under highly noisy environments. With respect to Gaussian noise the new method is much more robust, achieving a recognition rate of more than 80% even for extreme noise values, where the original method only achieved a rate of recognition of around 15%. The reason for this improvement is that due to the discrete orthogonality property of the new method, there is no redundancy between different moments.

With respect to salt-and-pepper noise, no significant difference could be determined between the methods, since this kind of noise adds a very high frequency component to the image, which is filtered by using only moments with low orders in both methods.

We have also found that in order to achieve almost the same recognition rate with the new method as the original one, the number of points in the system can be reduced to almost the minimum number required to still achieve discrete orthogonality. This reduces the computational costs significantly.

## 6.1 Future possibilities

One possibility for future work is to employ the techniques described in this thesis in some applications, which currently use other Zernike moment based methods.

In the proposed discretization, a more efficient, FFT-based method for the computation of moments up to some given degree could be constructed. This would provide a huge boost to the speed of image decomposition and reconstruction.

It is also possible to utilize the techniques described in this thesis to try and improve other quaternion moment invariant based methods, e.g. the ones presented in works [12], [27, 31, 32] and [8, 28, 38, 39].

Finally, the methods described could be further generalized to three dimensional space, where possible applications include pattern recognition in point clouds produced by LiDAR sensors [53].

# Acknowledgment

The Project is supported by the Hungarian Government and co-financed by the European Social Fund ( EFOP-3.6.3-VEKOP-16-2017-00001: Talent Management in Autonomous Vehicle Control Technologies )

62

**SZÉCHENYI 2020**



HUNGARIAN  
GOVERNMENT

**European Union**  
European Social  
Fund



**INVESTING IN YOUR FUTURE**

# Bibliography

- [1] B. Chen, H. Shu, H. Zhang, G. Chen, and L. Luo. Color Image Analysis by Quaternion Zernike Moments. *In Proceedings of the 20th International Conference on Pattern Recognition*, pages 625–628, 2010.
- [2] B. Chen, H. Shu, H. Zhang, G. Chen, C. Toumoulin, J.-L. Dillenseger, and L. Luo. Quaternion zernike moments and their invariants for color image analysis and object recognition. *Signal Processing*, 92:308–318, 2012.
- [3] Y. H. Lin and C. Chen. Template matching using the parametric template vector with translation, rotation and scale invariance. *Pattern Recognition*, 41(7):2413–2421, 2008.
- [4] S. O. Belkasim, M. Shridhar, and M. Ahmadi. Pattern recognition with moment invariants: A comparative study and new results. *Pattern Recognition*, 24(12):1117 – 1138, 1991.
- [5] M. Saaidia, L. Sylvie, and V. Vigneron. Face detection by neural network trained with Zernike moments. *In Proceedings of the 6th WSEAS International Conference on Signal Processing, Robotics and Automation*, pages 36–41, 2007.
- [6] L.-Q. Guo and M. Zhu. Quaternion Fourier–Mellin moments for color images. *Pattern Recognition*, 44(2):187 – 195, 2011.
- [7] F. Zernike. Beugungstheorie des Schneidenverfahrens und seiner verbesserten Form, der Phasenkontrastmethode. *Physica*, 1(7):689 – 704, 1934.
- [8] Y. Liu, S. Zhang, G. Li, H. Wang, and J. Yang. Accurate quaternion radial harmonic Fourier moments for color image reconstruction and object recognition. *Pattern Analysis and Applications*, 2020. doi: 10.1007/s10044-020-00877-6.
- [9] S. A. Nene, S. K. Nayar, and H. Murase. Columbia Object Image Library (COIL-100). *Technical Report CU-CS-006-96*, 1996.
- [10] J. M. Geusebroek, G. J. Burghouts, and A. W. M. Smeulders. ALOI: Amsterdam Library of Object Images. *International Journal of Computer Vision*, 61(1):103–122, 2005.

- [11] R. Mukundan and K. R. Ramakrishnan. *Moment Functions in Image Analysis — Theory and Applications*. World Scientific, 1998.
- [12] H. Zhu, Y. Yang, Z. Gui, Y. Zhu, and Z. Chen. Image analysis by generalized Chebychev–Fourier and generalized pseudo-Jacobi–Fourier moments. *Pattern Recognition*, 51:1 – 11, 2016.
- [13] Y. Lui, S. Zhang, G. Li, H. Wang, and J. Yang. Accurate quaternion radial harmonic fourier moments for color image reconstruction and object recognition. *Pattern Analysis and Applications*, 23(4):1551–1567, 2020.
- [14] A. Khotanzad and Y. H. Hong. Invariant image recognition by Zernike moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(5):489–497, 1990.
- [15] F. Zhang, S.-Q. Liu, D.-B. Wang, and W. Guan. Aircraft recognition in infrared image using wavelet moment invariants. *Image and Vision Computing*, 27:313–318, 2009.
- [16] F. Mindru, T. Tuytelaars, L. V. Gool, and T. Moons. Moment invariants for recognition under changing viewpoint and illumination. *Computer Vision and Image Understanding*, 94(1):3 – 27, 2004.
- [17] Y. Zhang, S. Wang, X. Yang, Z. Dong, G. Liu, P. Phillips, and T.-F. Yuan. Pathological brain detection in MRI scanning by wavelet packet Tsallis entropy and fuzzy support vector machine. *SpringerPlus*, 4(1):716, 2015.
- [18] J. C. Wyant and K. Creath. Basic Wavefront Aberration Theory for Optical Metrology. *Applied Optics and Optical Engineering*, 11:2–55, 1992.
- [19] L. A. V. de Carvalho and J. C. de Castro. Preliminary Results of an Instrument for Measuring the Optical Aberrations of the Human Eye. *Brasilian Journal of Physics*, 33(1):140–147, 2003.
- [20] V. L. Genberg. Optical Surface Evaluation. In *Proceedings of SPIE 0450, Structural Mechanics of Optical Systems I*, pages 81 – 87, 1984.
- [21] M. Pap and F. Schipp. Discrete orthogonality of Zernike functions. *Mathematica Pannonica*, 16(1):137–144, 2005.
- [22] G. Szegő. *Orthogonal Polynomials*. Amer. Math. Soc. Colloq. Publ. 23, AMS, Cambridge, R.I., 1967.
- [23] T. Suk and J. Flusser. Affine Moment Invariants of Color Images. In *Proceedings of the 13th International Conference on Computer Analysis of Images and Patterns*, pages 334–341, 2009.
- [24] W. R. Hamilton. *Elements of Quaternions*. Longmans, Green & Co., London, U.K., 1866.

- [25] T. A. Ell and S. J. Sangwine. Hypercomplex Fourier Transforms of Color Images. *IEEE Transactions on Image Processing*, 16(1):22–35, 2007.
- [26] Z. Shao, H. Shu, J. Wu, B. Chen, and J. L. Coatrieux. Quaternion Bessel–Fourier moments and their invariant descriptors for object reconstruction and recognition. *Pattern Recognition*, 47(2):603 – 611, 2014.
- [27] X. Wang, W. Li, H. Yang, P. Niu, and Y. Li. Invariant quaternion radial harmonic Fourier moments for color image retrieval. *Optics & Laser Technology*, 66:78 – 88, 2015.
- [28] C. Wang, X. Wang, and Z. Xia. Geometrically invariant image watermarking based on fast Radial Harmonic Fourier Moments. *Signal Processing: Image Communication*, 45:10–23, 2016.
- [29] Y. N. Li. Quaternion Polar Harmonic Transforms for Color Images. *IEEE Signal Processing Letters*, 20(8):803–806, 2013.
- [30] L. Guo, M. Dai, and M. Zhu. Quaternion moment and its invariants for color object classification. *Information Sciences*, 273:132 – 143, 2014.
- [31] C. Singh and J. Singh. Quaternion generalized Chebyshev-Fourier and pseudo-Jacobi-Fourier moments for color object recognition. *Optics & Laser Technology*, 106:234 – 250, 2018.
- [32] T. Yang, J. Ma, Y. Miao, X. Wang, B. Xiao, B. He, and Q. Meng. Quaternion weighted spherical Bessel-Fourier moment and its invariant for color image reconstruction and object recognition. *Information Sciences*, 505:388 – 405, 2019.
- [33] C. Chong, P. Raveendran, and R. Mukundan. A comparative analysis of algorithms for fast computation of Zernike moments. *Pattern Recognition*, 36(3):731 – 742, 2003.
- [34] C.-W. Chong, P. Raveendran, and R. Mukundan. Translation invariants of Zernike moments. *Pattern Recognition*, 36(8):1765 – 1773, 2003.
- [35] S. Liao and M. Pawlak. On the accuracy of Zernike moments for image analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:1358 – 1364, 1999.
- [36] M. Pawlak and S. Liao. On the recovery of a function on a circular domain. *IEEE Transactions on Information Theory*, 48:2736 – 2753, 2002.
- [37] Y. Xin, M. Pawlak, and S. Liao. Accurate Computation of Zernike Moments in Polar Coordinates. *IEEE Transactions on Image Processing*, 16(2):581–587, 2007.
- [38] K. Hosny and M. Darwish. Invariant color images representation using accurate quaternion Legendre–Fourier moments. *Pattern Analysis and Applications*, 22(3):1105 – 1122, 2019.

- [39] K. Hosny and M. Darwish. New set of quaternion moments for color images representation and recognition. *Journal of Mathematical Imaging and Vision*, 60:717 – 736, 2018.
- [40] R. Castro-Ortega, C. Toxqui-Quitl, A. Padilla-Vivanco, J. F. Solís-Villarreal, and E. E. Orozco-Guillén. Zernike moment invariants for hand vein pattern description from raw biometric data. *Journal of Electronic Imaging*, 28(5):1 – 11, 2019.
- [41] H. Shojanazeri, W. A. W. Adnan, S. Ahmad, and S. Rahimipour. Authentication of images using Zernike moment watermarking. *Multimedia Tools and Applications*, 76:577–606, 2015.
- [42] N. Singhal, Y. Lee, C. Kim, and S. Lee. Robust image watermarking using local Zernike moments. *Journal of Visual Communication and Image Representation*, 20 (6):408 – 419, 2009.
- [43] H. S. Kim and H.-K. Lee. Invariant Image Watermark Using Zernike Moments. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(8):766–775, 2003.
- [44] S. Zheng, Y. Song, T. Leung, and I. Goodfellow. Improving the Robustness of Deep Neural Networks via Stability Training. *In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 4480 – 4488, 2016.
- [45] M. Liu, S. Liu, H. Su, K. Cao, and J. Zhu. Analyzing the Noise Robustness of Deep Neural Networks. *IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 60–71, 2018.
- [46] L. Carvalho. Preliminary Results of Neural Networks and Zernike Polynomials for Classification of Videokeratography Maps. *Optometry and vision science : official publication of the American Academy of Optometry*, 82:151–8, 2005.
- [47] Python Software Foundation. The Python Language Reference. <https://docs.python.org/3/reference/>, 2020. [Accessed: 2020.05.06.]
- [48] The SciPy community. NumPy v1.18 Manual. <https://numpy.org/doc/stable/>, 2020. [Accessed: 2020.05.06.]
- [49] Anaconda Inc. et al. Numba Reference Manual. <http://numba.pydata.org/numba-doc/latest/reference/index.html>, 2020. [Accessed: 2020.05.06.]
- [50] A. Clark et al. Pillow Reference Manual. <https://pillow.readthedocs.io/en/stable/>, 2020. [Accessed: 2020.05.06.]
- [51] A. Glaser, X. Liu, and V. Rokhlin. A Fast Algorithm for the Calculation of the Roots of Special Functions. *SIAM Journal on Scientific Computing*, 29:1420–1438, 2007.

- [52] The USC-SIPI Image Database. <http://sipi.usc.edu/database>. [Accessed: 2020.05.09].
- [53] B. Ma, Y. Zhang, and S. Tian. Building Reconstrucion Using Three-Dimensional Zernike Moments in Digital Surface Model. *2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 1324–1327, 2018.