

# TDK dolgozat

Nagy Gergely



Eötvös Loránd University  
Faculty of Informatics  
Department of Numerical Analysis

## Color image analysis and recognition using quaternion Zernike moments

TDK thesis

*Supervisor:*  
Zsolt Németh  
Assistant Professor

*Author:*  
Gergely Nagy  
Computer Science MSc  
1<sup>st</sup> year

Budapest

Date of closing: 14 May 2020

## Abstract

Image moments and moment invariant features are widely used for image analysis and pattern recognition. The system of orthogonal Zernike polynomials (defined over the complex unit disk) proved to be a useful basis for series expansions because of certain invariance properties.

Conventionally, for multichannel color images, RGB decomposition or grayscale conversion was used. However, in the recent past, quaternion algebra has been employed to various conventional moments to analyze a color image holistically. ? [?] introduced quaternion Fourier-Mellin moments (QFMMs) which are an extension of the conventional Fourier-Mellin moments for the grayscale image. They also proposed their invariants on rotation, scale, and translation for color object recognition. ? [?] proposed the quaternion Zernike moments (QZMs), generally overperforming other similar approaches in these aspects, due to the natural invariances of Zernike functions. The same quaternion techniques were applied successfully to other function systems (e.g. [? ?]), yielding similar results.

In this thesis we introduce a method of transforming a digital RGB image inside the unit circle onto a points system providing discrete orthogonality. Using this points system for the discretization of the QZMIs, we have achieved significant improvements in the image recognition and reconstruction capabilities of the method, especially under noisy conditions.

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Contributions . . . . .	3
1.2	Structure of the thesis . . . . .	3
<b>2</b>	<b>Background</b>	<b>5</b>
2.1	Image moments for monochrome images . . . . .	5
2.2	Image moments for multi-channel, color images . . . . .	5
2.3	Discretization . . . . .	6
2.4	Applications . . . . .	7
<b>3</b>	<b>Proposed method for discretization</b>	<b>8</b>
<b>4</b>	<b>Implementation</b>	<b>11</b>
4.1	Programming language and libraries used . . . . .	11
4.2	Calculating moments and moment invariants . . . . .	11
4.3	New image transformation . . . . .	15
<b>5</b>	<b>Tests and comparison with the original method</b>	<b>21</b>

5.1	Test images . . . . .	21
5.2	Invariance test . . . . .	25
5.3	Image reconstruction . . . . .	28
5.4	Image recognition . . . . .	32
5.5	Template matching . . . . .	39
<b>6</b>	<b>Conclusion</b>	<b>41</b>
6.1	Future possibilities . . . . .	42
	<b>Bibliography</b>	<b>43</b>

# Chapter 1

## Introduction

Moment invariants are widely used in applications for pattern matching [? ], image recognition [? ], or to extract useful features from images [? ].

Most of the moments are defined for single-channel, grayscale images, however extending these techniques to multichannel color images is an important and generally unresolved problem with many possible applications. Conventionally, for color images either RGB decomposition or grayscale conversion was used in order to utilize the methods defined for grayscale images. This may lead to loss of information, for example in the case of grayscale conversion (where the average of the color channels is taken) some 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 [? ], as well as the quaternion Zernike moments as an extension of the conventional Zernike moments [? ].

The Zernike functions are a system of orthogonal functions defined over the unit disk. 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 property of the discretization of these methods defined by orthogonal functions is to preserve the orthogonality over the discrete system, so as to avoid redundancy and achieve high robustness with respect to noise.

The conventional method for discretizing quaternion Zernike moments (used by ? [?]) consists of uniformly distributed points over the unit disk. This method does not achieve discrete orthogonality thus decreasing 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 points 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 method used by ? [? ]. For the tests, image sets from the Columbia Object Image Library [? ] and the Amsterdam Library of Object Images [? ] were used.

Besides the theoretical invariance properties of QZMIs, these are also verified empirically. The image reconstruction capabilities of both methods 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 points 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

Short summary of chapter...

### 2.1 Image moments for monochrome images

Altalanos leiras a szurkearnyalatos momentumokrol, nehany peldaval

#### Invariance properties

RST invariancia, definicio, miert akarjuk hogy erre invariants legyen

#### Zernike moments

Bovebben kifejtve, definiciok (Zernike fuggveny, momentum), az invariantsok konstru-alasa, miert jo ez,

### 2.2 Image moments for multi-channel, color images

Szines kepekre milyen modszerek voltak korábban (RGB decomp, greyscale conv)

## **Algebra of quaternions**

kvaterniokrol leiras

## **Quaternion moments**

Altalaban a kvaternios momentumkrol

## **Representing color images with quaternions**

hogyan tekintjuk mostantol a kepet, miert jobb ez

## **Quaternion Zernike moments**

a kvaternios valtozatot leirni

## **Invariants**

hogyan mukodnek az invariantsok, definiciokat leirva

## **Quaternion radial harmonic Fourier moments**

Leirni hogy ez hogyan van definialva, es majd lesz vele osszehasonlitas

## **2.3 Discretization**

Korábbi diszkretizacios modszerek

**Discretization of QZMs**

**Discrete orthogonality**

## **2.4 Applications**

Alkalmazások

## 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 (??) maintains advantageous theoretical properties. The idea behind our approach is motivated by the work [? ].

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 [? ]. 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 (??) 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 (??) 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 (??) expressed with the Jacobi polynomials of (??), 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. [?]), 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.

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

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 (??) in the form

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

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 methods of discretization.

### 4.1 Programming language and libraries used

The implementation was created using the Python programming language [?] and relies on the Numpy [?] and Numba [?] libraries to achieve efficient and fast computation of the moment invariants, as well as the Python Imaging Library (PIL, Pillow) [?] 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 as native code. The disadvantage of using JIT is that it limits the available 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 ??, the quaternion Zernike moments (QZMs) must be calculated first. ? [?] showed that instead of calculating the QZMs directly by using the algebra of

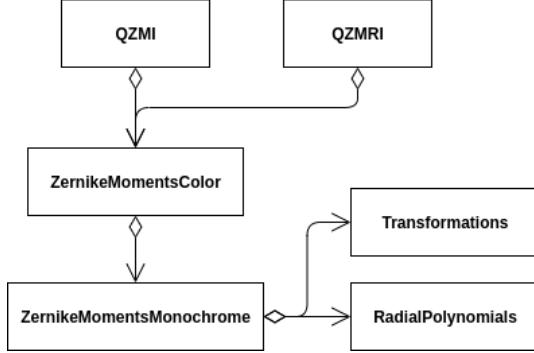


Figure 4.1: The relationships between the classes used to calculate QZMIs and QZMRIs

quaternions, it is possible to calculate the real and the imaginary parts of the quaternion-valued QZM 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, introduced in (??).

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 the rotation invariant QZMs 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.

```

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

## 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 [? ]. This algorithm computes the value of  $R_{n,m}(r)$  for all  $0 \leq |m| \leq n \leq P$ , ( $n - |m|$  is even) with complexity  $\mathcal{O}(N^2)$ . This method is ideal for the precomputation of all radial polynomial values up to a maximum degree.

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

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  $(x, y)$  coordinates transformed to the unit disk,  $\lambda$  is the scaling parameter also given by the transformation (as described in Section ??) and  $f$  is the real-valued, grayscale image.

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

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

## 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 [? ].

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

where  $f$  is an RGB image and  $f_R, f_G, f_B$  are the red, green and blue color channels respectively.

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 ??.

To achieve translation invariance, the common centroid of the RGB image is calculated based on the formulas described by ? [? ].

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

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 (??). The original image  $f$  is then translated in image coordinates such that the origin falls on the common centroid  $\{x_c, y_c\}$ , and the other invariants are then calculated based on this translated image.

## 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 in the new discrete points 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 points 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:

$$(1 - x^2)P_n''(x) - 2xP_n'(x) + n(n + 1)P_n(x) = 0. \quad (4.1)$$

A fast algorithm for calculating the roots of  $P_n$ , based on this differential equation was presented by [? ]. The algorithm uses a second-order Runge-Kutta method (namely the midpoint method) to solve the Prüfer-transformed version of (4.1) 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 points system, besides the  $\lambda_{n,m} = n + 1$  scaling factor, a weight function is used as well (as described in (??)). 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  $P_N$ .

The Gauss-Legendre quadrature is based on the roots of the Legendre polynomial  $P_n$  of degree  $n$ . Let  $x_1, x_2, \dots, x_n$  be the roots of  $P_n$ .

$$\begin{aligned} \int_{-1}^1 f(x) dx &\approx \sum_{k=1}^n w_k f(x_k), \\ w_k &= \frac{2}{(1 - x_k^2)(P'_n(x_k))^2}. \end{aligned}$$

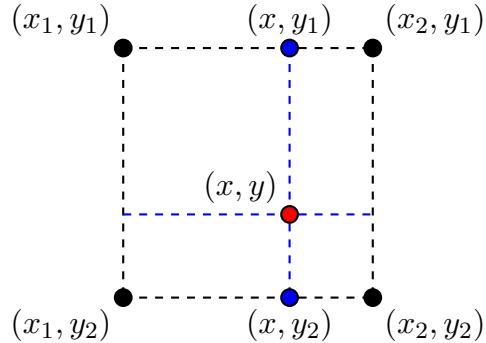


Figure 4.3: The point  $(x, y)$  and its neighboring points, which are used to approximate  $f(x, y)$  by bilinear interpolation

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 points system, this formula gives an easy and fast way to calculate the exact values of the weight function  $w$ .

### Image interpolation

After obtaining the discrete orthogonal points 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 ??, so that the the transformed image covers the entire unit disk. This transformation is used as opposed to the one on Figure ??, because the points 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 points system.

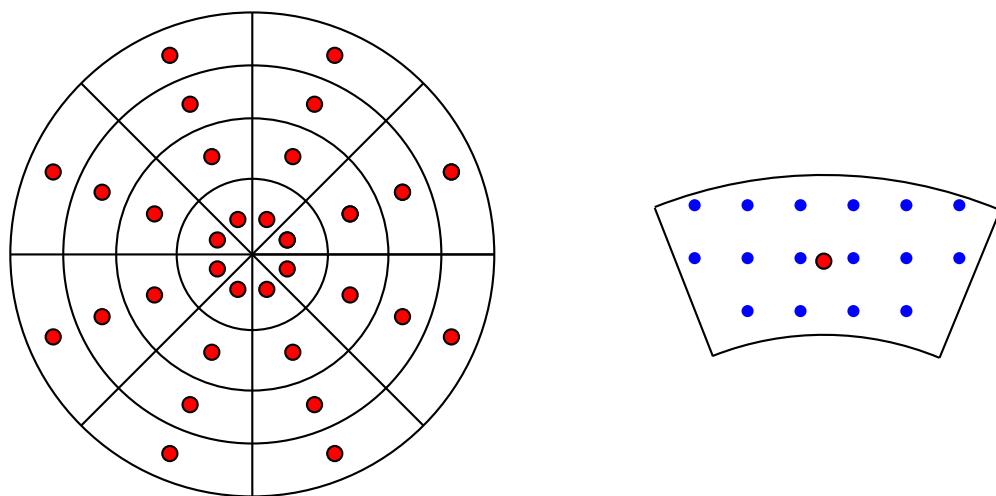
**Approximately the same number of points as pixels.** If the number of points is approximately the same as the number of pixels in the image, then bilinear interpolation can be used for each point.

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

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

First, for each pixel in the original image, determine which point it falls closest to after the linear transformation. Then, for each point in the discrete orthogonal points system, calculate the average of the pixels that are closest to that point. This algorithm divides the unit disk into annular sections based on the proximity to the new points. 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 [? ].



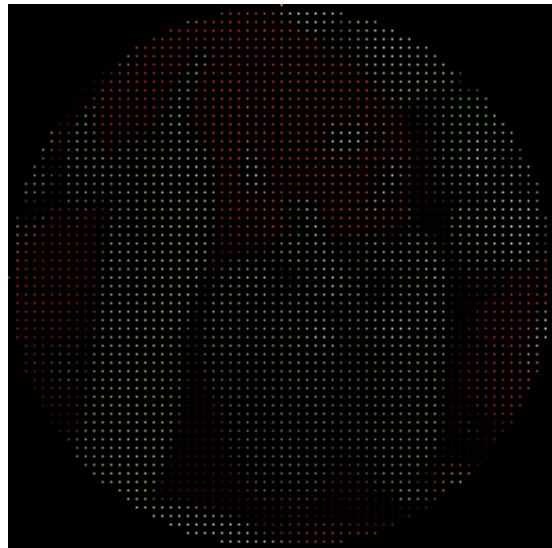
(a) The points system on the unit disk and the annular sectors over which the pixel values are averaged.

(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).

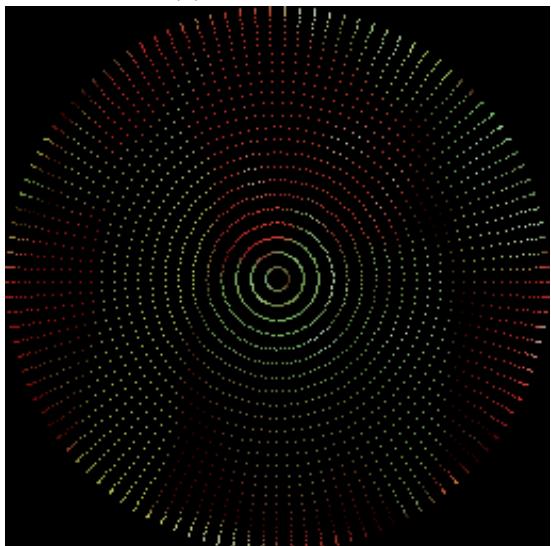
Figure 4.4



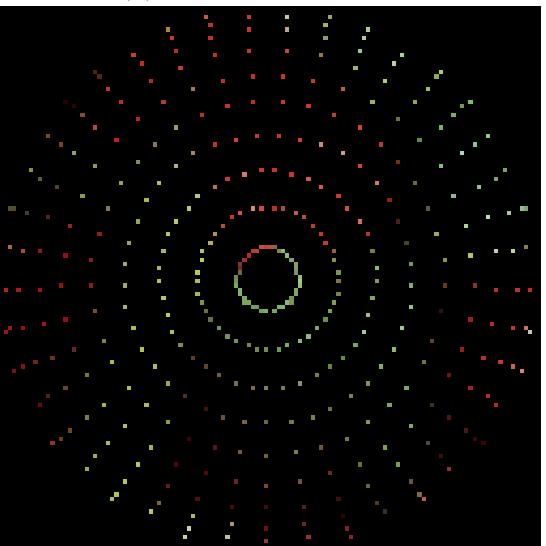
(a) Original image



(b) Original transformation



(c) Bilinear interpolation



(d) Interpolation by integrals

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

### 5.1 Test images

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

For the image recognition tests, two sets of images were used. The first set consists of 14 images chosen from the Columbia Object Image Library (COIL-100) [? ], shown on Figure 5.2. These images are originally  $128 \times 128$  pixels, but they were placed on a



(a)



(b)

Figure 5.1: The Lenna and Pepper images

$204 \times 204$  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, 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) [? ]. These are shown on Figure 5.5. Originally, these 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)

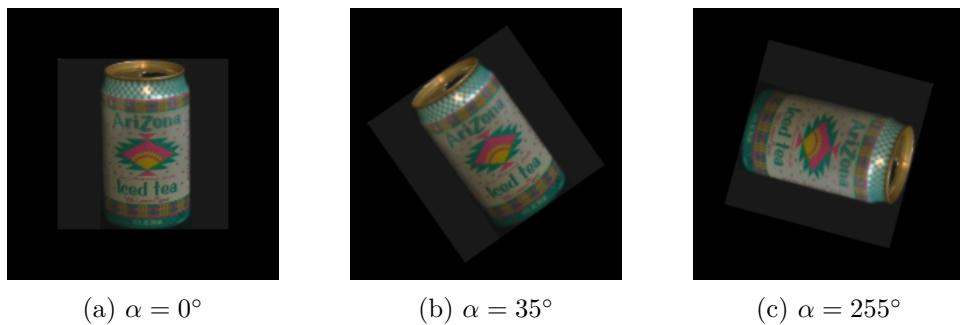


Figure 5.3: Some extended and rotated images from COIL



(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  $\Delta x = -11, \Delta y = 9$

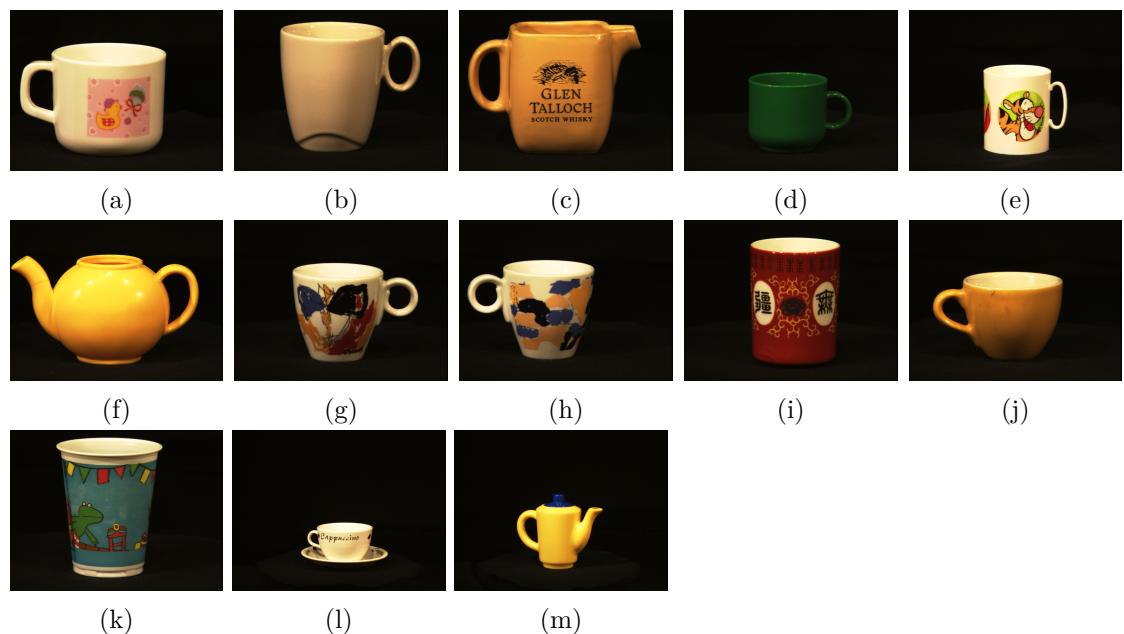


Figure 5.5: The 13 selected images from the Amsterdam Library of Object Images (ALOI)

## 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 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 moment of all transformed images.

### Results

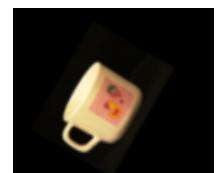
The moduli of the QZMIs for the transformed images shown on Figure 5.6 is shown in Table 5.1 for the old method of discretization and in Table 5.2 for the new, proposed method of discretization. The coefficient of variation ( $\frac{\sigma}{\mu}$ ) shows that using both methods, the moments are invariant 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)



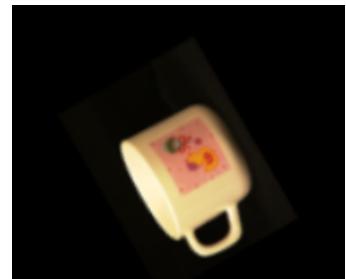
(c)



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

In order to reconstruct the discrete image, 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 ?? ( $tf_1$ ) for the old method and the transformation shown on Figure ?? ( $tf_2$ ) for the new one. The reason for this difference in transformation is that for the proposed discrete orthogonal points system, the interpolated pixel values are calculated using  $tf_2$ , while conventionally for (quaternion) Zernike moments,  $tf_1$  is used [? ].

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 only this part of the image.

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.

## Results

Image reconstruction was performed for both the Lenna and the Pepper images [? ], using image sizes ranging from  $64 \times 64$  up to  $256 \times 256$ . Depending on the size of the

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	0.00790	0.00463	0.00190	0.00238
change	-39.4%	-41.1%	-46.7%	-55.6%	-14.7%

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

image, QZMs of up to degree 350 were used reconstruct the image. However, in the case of smaller images, for example a  $64 \times 64$  image, only QZMs of up to 100 degree could be used, as using higher degrees would result 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. This phenomenon is visible, for example on Figure 5.8 for the  $64 \times 64$  Lenna,  $M = 100$  case.

Figure 5.7 shows some of the reconstructed images and their normalized mean squared error when using the old method of discretization, while Figure 5.8 shows the same data using the new method.

Comparing the error of reconstruction between the two methods, using the discrete orthogonal points system provides much lower normalized mean squared errors for almost 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.

Due to the lower number of points used by the second method (because of the different transformation of the image onto the unit disk) the previously described increase in reconstruction error occurs for lower  $M$ s than in the original method.

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				
	0.00388	0.00868	0.00885	
M = 250				
		0.00428	0.00378	
M = 350				
		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				
	0.00380	0.00901	0.01611	0.01778
M = 100				
	0.01181	0.00246	0.00790	0.00681
M = 150				
		0.00152	0.00463	0.00349
M = 250				
			0.00190	0.00131
M = 350				
			0.00238	0.00229

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 [? ] and ALOI [? ] images and the 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. The goal of the image recognition test is to see what percentage of the transformed images can each method correctly identify by comparing their moment invariants to the ones of the original, non-transformed 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 of 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 ??):

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

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 ? [? ].

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

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 QZMRIs. QZMRIs are more robust against noise than QZMIs, so higher noise values could be used. Instead of using Gaussian noise, for some test cases 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 for both types of noise are shown on Figure 5.9.

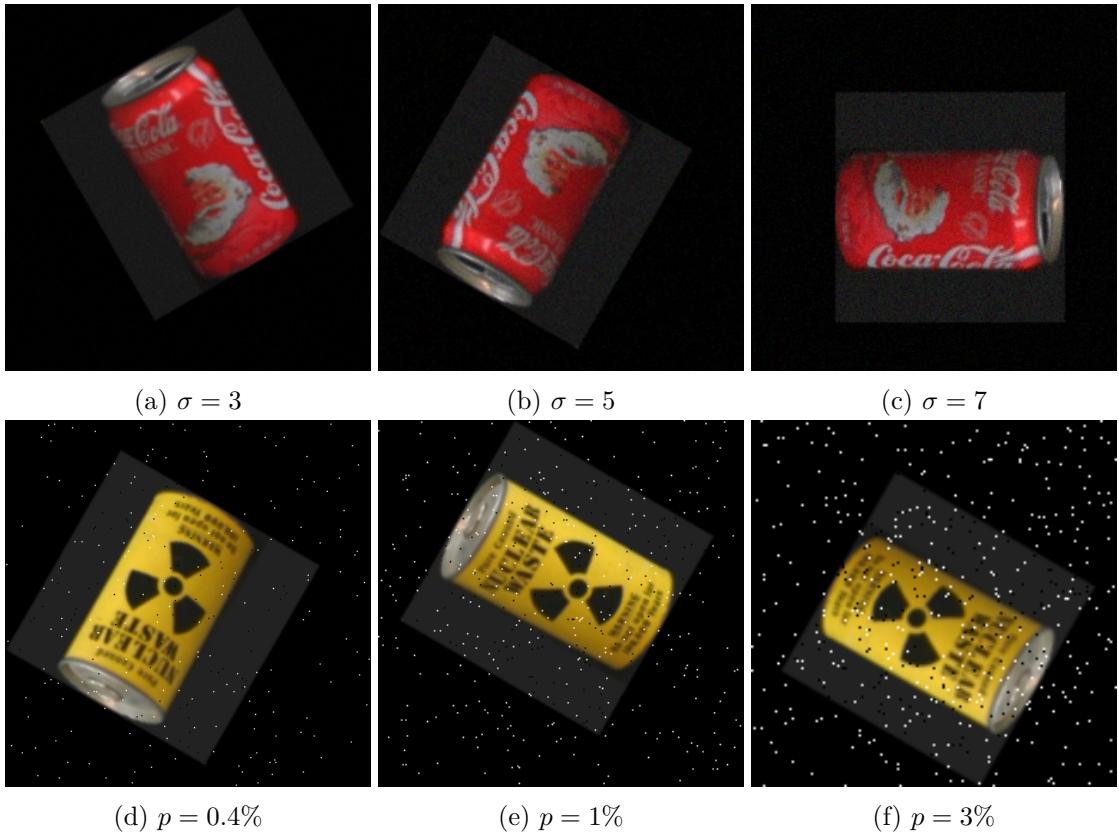


Figure 5.9: Some test images with Gaussian noise (top row) or salt-and-pepper noise (bottom row) with different parameters added to them.

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 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

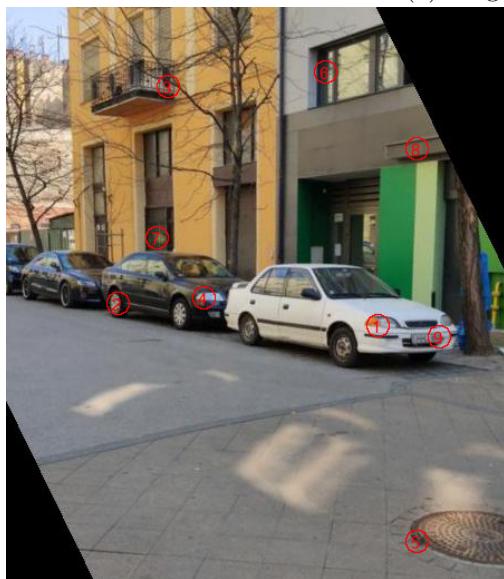
This final test aims to present some possible application of the pattern matching capabilities of the QZMIs. Some pictures were taken using a the camera on a Xiaomi Mi 9 smartphone with autofocus enabled. The images were taken of the same scene but with different focus and different rotation of the camera.

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.

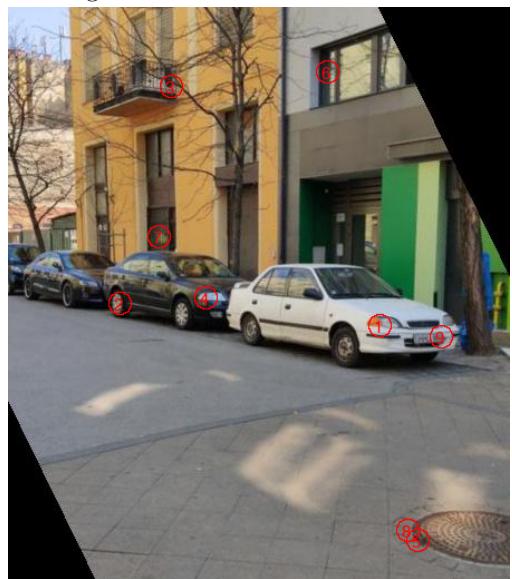
The original image and the results of both methods are shown on Figure 5.10. Both methods managed to correctly find 8 out of the 9 templates on the other image. Neither managed to correctly locate template number 8.



(a) Original image



(b) Old method



(c) New method

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

# 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 points system 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 [?], [? ? ?] and [? ? ? ?].

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 [?].

# 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 )

