Radial Moments for Invariant Image Analysis: Computational and Statistical Aspects

by

Urmila Samanta

A Thesis submitted to the Faculty of Graduate Studies of

The University of Manitoba

in partial fulfilment of the requirements for the degree of

MASTER OF SCIENCE

Department of Electrical and Computer Engineering

University of Manitoba

Winnipeg, Manitoba R3T 5V6 Canada

Contents

List of Tables

List of Figures

Abstract

Zernike moments are sets of mathematical quantities that uniquely characterize an image. It is known that they are invariant under rotation and reflection and robust to noise. In this thesis several other algorithms have been used to calculate these moments. The intent of this thesis is:

1. to use the classical method and the algorithms to reconstruct an image using Zernike moments and study their accuracy and

2. to examine if the invariance and noise insensitivity property of the calculated Zernike moments are upheld by these procedures.

It is found that the constructed images using these algorithms do not resemble the original image. This prevents us from carrying out further study of these algorithms. The classical method has been successfully used to reconstruct an image when the height and width are equal. The classical method is also shown to be invariant under rotation and reflection and robust to Poisson noise.

Acknowledgements

Many thanks to the following persons for their role in the completion of this thesis.

- My advisor Dr. Pawlak for giving me the opportunity to write my Master's degree under his able guidance and for helping me throughout the entire preparation of this thesis.

- My examination committee Dr. Liao and Dr Peters for helping me in my entire program.

- My parents, and my many friends at the university for giving me support in many aspects related to my Master's program.

1 Chapter 1

1.1 Motivation

Zernike moments are sets of mathematical quantities that uniquely characterize an image function defined on the circular domain. They are derived from Zernike functions that constitute a complete and orthogonal basis on the unit circle. As a result the Zernike moments contain complete information about the image function as each Zernike moment carries a unique piece of information about that function. Hence there is no redundancy of information. The fundamental property of the Zernike moments is that they are invariant under rotation and reflection of an image. They are also known to be robust to noise. Due to these salient properties, they have been found useful in a wide variety of applications such as ophthalmology, optical system, watermarking, character identification, identifying gait sequences and image reconstruction.

Since Zernike moments are used in wide range of applications, it is important to know which algorithms lead to the most accurate calculation of Zernike moments. The present study may help other researchers who use Zernike moments in their work. The purpose of this project is to find out:

- which algorithm(s) can be used to produce accurate Zernike moments inorder to reconstruct an image and,
- which algorithm(s) will give accurate Zernike moments that can fare well under rotation and reflection and in the presence of different noise types. The intent of this thesis is to give valuable information about the different algorithms for computing Zernike moments.

1.2 Thesis Outline

A brief outline of our thesis is given below. We introduce moments, orthogonal moments and Zernike moments and discuss their properties. We then consider the applications of

Zernike moments and emphasize their role in image reconstruction. We consider eight different methods to compute Zernike moments. These methods are Direct (Classical), Belkasim, Hybrid, Prata, Kintner, Modified Prata, Q-theta recursive and Factorial Free method. We discuss various types of errors such as in the calculation of geometric errors and discretization (numeric integration) errors of Zernike moments when these moments are computed from digital data. We explain how the geometric error can reduce the accuracy of the Zernike moment in a following section. We have used the above methods to compute Zernike moments inorder to reconstruct images of different sizes. We then consider the use of the above methods to compute the Zernike moments when various types of noise are superimposed on the image. We also consider the use of the above methods to compute the Zernike moments when the image is rotated or reflected. To justify the use of Zernike moments in image reconstruction we test their accuracy using mean square as a measure of error. We performed three experiments. In experiment 1, on the basis of the resemblance between the reconstructed images and the input images of different sizes we study the optimality properties of the computational procedures using the plot of mean squares against moment orders. In experiment 2, we study the effectiveness of the computational procedures when various types of noise such as Gaussian noise (with different variances), Poisson noise and Salt and Pepper noise (with different densities) are separately added to different image sizes. In experiment 3, we study the invariance properties of the classical method under rotation (with different angles) and reflection. We carefully discuss the results obtained in experiment 1, 2 and 3. The final conclusions from these discussions are then carefully stated establishing the optimality properties of the classical method in many respects.

2 Chapter 2

2.1 Moments

Moments are numeric quantities that can describe a function and significant features of that function. They have been used in statistics for a long time to derive the functional form of a probability density function. Moments can be viewed as projections of a function onto a polynomial basis. The general definition of moment is $m_{pq} = \iint\limits_{\Omega} G_{pq}(x,y)f(x,y)dxdy$ where $G_{00}(x,y)$, $G_{10}(x,y)$, $\ldots$ $G_{kj}(x,y)$, $\ldots$ are polynomial basis functions defined on Ω. The existence theorem will now be restated below.

Existence theorem: Assuming that the intensity function $f(x,y)$ is piece-wise continuous and bounded in the region Ω, the moment sequence $\{m_{pq}\}$ is uniquely determined by the intensity function $f(x,y)$ and conversely.

From this theorem it follows that each image function has an unique set of moments $\{m_{pq}\}$ [1].

2.2 Geometric moments

Let $f : \Omega \subset R^2 \rightarrow R, \Omega$ being some compact set, be an image function describing a real scene, such that $f(x,y)$ is non-negative and represents an intensity of the image at a spatial position $(x,y)\epsilon\Omega$ in the image plane. Geometric moments are defined with the basis $x^p y^q$. The two-dimensional geometric moments of order $(p+q)$ about the origin are denoted by m_{pq} and can be defined as $m_{pq} = \iint_{\Omega} x^p y^q f(x,y)dxdy \;\; p,q = 0,1,2,3,\ldots$. These moments are called geometric moments because simple geometric properties of an image such as area, orientation and position can be easily computed using them. The center of mass of the image can be calculated from the moments m_{00}, m_{10}, m_{01}

$$\bar{x} = \frac{m_{10}}{m_{00}}, \tag{2.1}$$

$$\bar{y} = \frac{m_{01}}{m_{00}}. \tag{2.2}$$

The orientation of an image is defined as the axis of least inertia that is the line for which the integral $\iint_\Omega r^2(x,y)x^p y^q f(x,y)dxdy$ is minimum, where $r(x,y)$ is the perpendicular distance from the point (x,y) to the line sought after. It has been shown in [5] that the axis of least inertia can be obtained from second order moments (m_{20}, m_{11}, m_{02}). Higher order moments can provide additional detailed information about the image such as an orientation of the axis of least inertia and symmetry properties [2].

2.3 Central Moments

The central moment corresponding to m_{pq} is defined as

$$\mu_{pq} = \iint_\Omega (x-\bar{x})^p (y-\bar{y})^q f(x,y)dxdy. \tag{2.3}$$

It can be seen that central moments are invariant to translations of the image. If an image is shifted such that its centroid ($\bar{x}$,$\bar{y}$) is at the origin, the geometric moments m_{pq} then become the central moments. Any central moment μ_{pq} can be expressed in terms of moments m_{pq} about the origin [2].

2.4 Moment Invariants

The basic idea is to describe the objects by a set of measurable quantities called invariants that are insensitive to particular deformations and that provide enough discrimination power to distinguish objects belonging to different classes. It is a fundamental problem in image analysis to find image descriptors which are invariant under the following general affine transformations of the image plane

$$\bar{x} = \alpha x + ay + x_0 \tag{2.4}$$

$$\bar{y} = bx + \beta y + y_0 \tag{2.5}$$

Scaling invariance is obtained by proper normalization of each moment. In principle, any moment can be used as a normalizing factor provided that it is nonzero for all images in the experiment. Since low-order moments are more stable to noise and easier to calculate, we normalize most often by a proper power of μ_{00} as shown below

$$\nu_{pq} = \frac{\mu_{pq}}{\mu_{00}^{w}}, \tag{2.6}$$

where $w = \frac{p+q}{2} + 1$

The moments ν_{pq} are called normalized central moments. These moments are scale invariant, that is invariant under the above affine transformations with $a = b = 0$ [1].

2.5 Hu's moment

Invariant moment descriptors are first introduced by Hu who employed the results of the theory of algebraic invariants and derived his seven famous invariants to an in-plane rotation around the origin [3]. His first two invariants are:

$$\phi_1 = m_{20} + m_{02} \tag{2.7}$$

$$\phi_2 = (m_{20} - m_{02})^2 + 4m_{11}^2 \tag{2.8}$$

These descriptors are supposed to be invariant under the above affine transformation but they are actually invariant under shifting, scaling and rotation transformations. The correct version of Hu's invariant moments was established in [6]. These moments are not good at representing complex objects and they form a nonlinear set which makes them unstable. For these reasons they are not used in practice [2].

2.6 Orthogonal Moments

The geometric moments are formed using a monomial basis set $\{x^p y^q\}$. The lack of orthogonality of $\{x^p y^q\}$ results in high correlation between corresponding moments $\{m_{pq}\}$ yielding a highly redundant representation of the image. Therefore the geometric moments $\{m_{pq}\}$ have a great difficulty in distinguishing different image patterns. Teague [7] proposed replacing $\{x^p y^q\}$ by a complete orthogonal polynomial basis $\{V_{pq}(x, y)\}$. The orthogonality of $\{V_{pq}(x, y)\}$ means that we have

$$\iint_{\Omega} V_{pq} V_{p'q'} w(x, y) dx dy = 0, \tag{2.9}$$

for all $(p, q) \neq (p', q')$ where $w(x, y)$ is the weight function. The orthogonal moment of order p,q are now defined by

$$\nu_{pq} = \iint_{\Omega} V_{pq}(x, y) f(x, y) dx dy, \ p, q = 0, 1, 2, \ldots \tag{2.10}$$

The main reasons for introducing an orthogonal or weighted-orthogonal polynomial basis in defining orthogonal moments are:

1. stable and fast numerical implementation
2. avoidance of high dynamic range of moment values and
3. higher robustness to random noise

The following is a simple way to construct two variable counterparts of orthogonal polynomials. Let $\{P_p(x)\}$ be the orthogonal polynomial basis with respect to the weight function $w_1(x)$ which is defined on the set Ω_1. Similarly let $\{Q_q(x)\}$ be the orthogonal polynomial basis with respect to the weight function $w_2(x)$ which is defined on the set Ω_2 and let $\Omega = \Omega_1 \times \Omega_2$. Then it can be shown that $V_{pq}(x, y) = P_p(x) Q_q(y)$ is the orthogonal polynomial of degree $p+q$ with respect to the weight function $w_1(x) w_2(y)$ which is defined on Ω.

Two important cases of orthogonal basis are:

1. Classical orthogonal polynomial in two variables

   Here $\Omega = [-1, 1]^2$ and $\{P_p(x)\}$ and $\{Q_q(x)\}$ are classical orthogonal polynomials. Some examples are Legendre, Chebyshev and Jacobi polynomials.

2. Radial orthogonal functions

   Here $\Omega = \{(x, y) : x^2 + y^2 \leq 1\}$ is the unit circle and the general form of radial orthogonal functions is $V_{pq}(x, y) = R_{pq}(r)e^{jq\theta}$ where $r = \sqrt{(x^2 + y^2)}$, $\phi = \arctan(y/x)$ and $R_{pq}(r)$ is a polynomial in r. Zernike polynomials is one such example [1].

2.7 Zernike Moments

The Zernike function $V_{pq}(x, y)$ is defined as $V_{pq}(x, y) = R_{pq}(r)e^{jq\phi}$ where $r = \sqrt{x^2 + y^2}$ and $\phi = \arctan(x/y)$ and the radial Zernike polynomial $R_{pq}(r)$ is defined by the following formula:

$$R_{pq}(r) = \sum_{s=0}^{\frac{(p-|q|)}{2}} \frac{(-1)^s (p-s)! r^{p-2s}}{s!(\frac{(p+|q|)}{2} - s)!(\frac{(p-|q|)}{2} - s)!} \tag{2.11}$$

$$= \sum_{k=|l|,|l+2|,\ldots}^{n} B_{pqk} r^k \tag{2.12}$$

with $p - |q|$ always even and

$$B_{pqk} r^k = \frac{(-1)^{\frac{n-k}{2}} (\frac{n+k}{2})!}{(\frac{n-k}{2})!(\frac{k+q}{2})!(\frac{k-q}{2})!}. \tag{2.13}$$

The Zernike polynomials (functions) satisfy the relation of orthogonality

$$\int_0^{2\pi} \int_0^1 V_{pq}^*(r, \phi) V_{mk}(r, \phi) r dr d\phi = \frac{\pi}{p+1} \delta_{pm} \delta_{qk}, \tag{2.14}$$

where

$$\delta_{pm} = \begin{cases} 1 & \text{if } m = p \\ 0 & \text{otherwise} \end{cases}$$

and $V^*_{pq}(r, \phi)$ is the conjugate of $V_{pq}(r, \phi)$. In particular

$$\int_0^1 R_{nl}(r) R_{ml}(r) r dr = \frac{1}{2(n+1)} \delta_{mn}. \tag{2.15}$$

The Zernike moment of p^{th} order with repetition q are defined as

$$A_{pq} = \frac{p+1}{\pi} \int_0^{2\pi} \int_0^1 V^*_{pq}(r, \phi) f(r, \phi) r dr d\phi, \ n = 0, 1, 2, \ldots, q = -n, -n+2, \ldots, n. \tag{2.16}$$

The image must be moved to the domain $[-1, 1]^2$ via a mapping transform to obtain accurate Zernike moment values. Since Zernike polynomials form a complete and orthogonal basis defined within the unit circle, the Zernike moments possess all properties of orthogonal moments described before. Teague has shown that Zernike moments are rotation and reflection invariant [7]. That is if the image function is rotated or reflected the corresponding Zernike moments of the transformed image will be the same as the Zernike moments of the original image function. In view of the above desirable properties, Zernike moments are preferred to calculate features of an image and recognize objects in images even if they are transformed in similar images [1],[2].

2.8 Current applications using Zernike moments

Due to the fundamental properties of Zernike moments and because of their robustness to noise, they have become useful in a wide variety of applications such as image reconstruction, shape analysis, corneal surface modelling, ophthalmology [8], [9], [10], and in watermarking [11]. Many of the applications that use Zernike moments to reconstruct an image are character identification [12], [13], gait sequence identification [14] and face recognition [15]. The procedure of image reconstruction from a set of Zernike moments

is commonly used as part of a neural network system [14]. In many of the applications discussed above, for example in the work on gait classification, the images would be divided into training and testing sets. Reconstruction of each image in the training set via Zernike moments would be performed. Each image in the testing set would be reconstructed via Zernike moments and would be compared to all images in the training set. If there is a resemblance then the system has found a match, otherwise the test image is inaccurate and discarded. Iskander et al. [10], have discussed the application of Zernike polynomial in optimal corneal modelling which is given in the following section.

2.8.1 Application of Optical Corneal Modelling

The cornea is an important part of the eye that contributes a majority of the eye's optical power. Some changes to the shape of the eye can produce astigmatism or severe conditions that cannot be corrected with glasses. Therefore, accurate modelling of the corneal surfaces is important for corneal refractive surgery to ensure good optical and visual measurements. They model the corneal surfaces with Zernike polynomials. This involves finding the order of the polynomial expansion based on the measured data to get the most accurate measurements. These measurements can be taken from the videokeratoscope and optics can be applied to get the error. The data received from the videokeratoscope can be modelled as a Zernike polynomial expansion. They assume that the noise is an independent and identically distributed random variable with zero mean and finite variance. Various methods are used to find the optimal moment order and the optimal number of Zernike polynomial terms needed for accurate modelling of the corneal surface. These algorithms are the Akaike information criterion, minimum description length, Hannan and Quinn criterion, corrected Akaike information criterion and a proposed bootstrap method.

Bootstrapping is a technique for assessing the accuracy of a parameter-estimator in situations where conventional techniques are not valid. This technique randomly assigns the observations and recomputes the estimates. The advantage in using bootstrapping is that the knowledge of the distribution of the measurement and modelling error is not necessary

[16].

These researchers have used different versions of Zernike polynomials of order p and repetition q given below:

$$V_{pq}(r,\phi) = \begin{cases} \sqrt{2(p+1)}R_{pq}(\rho)cos(q\phi), & even\ r, q \neq 0 \\ \sqrt{2(p+1)}R_{pq}(\rho)sin(q\phi), & odd\ r, q \neq 0 \\ \sqrt{(p+1)}R_{p0}(\rho)cos(q\phi), & even\ r, q = 0, \end{cases}$$

where the radial polynomial has already been given in equation (2.11). They found that the bootstrapping method outperformed all the other methods considered.

2.9 Image Reconstruction

Image reconstruction using moments is performed only to test the capability of the moments in recognizing objects. Moments have a horrible compression rate compared to JPEG and image reconstruction using moments is never used commercially.

The following are the limitations of geometric moments in determining the shape of an image function.

1. It is well known that there exists an infinite variety of functions whose first N geometric moments coincide and a unique reconstruction of the image function $f(x, y)$ is impossible.

2. The high correlation between the geometric moments yields a highly redundant representation of the image.

3. As the order increases, the geometric moments increase without limit. The Fourier transform of $f(x, y)$ is

$$F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{(-2\pi i)(ux+vy)} f(x,y)dxdy = \sum_{p}\sum_{q} \frac{(-2\pi i)^{p+q}}{p!q!} m_{pq}u^p v^q, \tag{2.17}$$

where m_{pq} are the geometric moments

$f(x, y)$ can be computed using inverse Fourier transform given by

$$f(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{2\pi i(ux+vy)} F(u,v) du dv. \tag{2.18}$$

It follows that image reconstruction from geometric moments is carried out in the Fourier domain and hence cannot be performed directly in the spatial domain.

2.10 Image Reconstruction by Orthogonal Moments

Orthogonal moments are calculated using an orthogonal polynomial basis $\{V_{kj}(x, y)\}$. We know that all polynomial bases of the same degree are equivalent because they generate the same space of functions. In view of this, orthogonal moments of any type can be expressed in terms of geometric moments. However, unlike geometric moments, orthogonal moments are coordinates of the function $f(x, y)$ in the polynomial basis in the common sense used in linear algebra. Due to this, image reconstruction from orthogonal moments can be performed using the following formula

$$\hat{f}(x,y) = \sum_{k,j} M_{kj} V_{kj}(x,y), \tag{2.19}$$

where

$$M_{kj} = \iint_{\Omega} V_{kj} f(x,y) dx dy. \tag{2.20}$$

This reconstruction is optimal because it minimizes the mean-square error when using only a finite set of moments [1].

2.11 Image Reconstruction by Zernike Moments

It has been stated before that for a continuous image function $f(x, y)$, the Zernike moment of order p with repetition q is defined as $A_{pq} = \iint_D f(x, y) V^*_{pq}(x, y) dx dy$ where the double

integral is performed over the unit disk $D = \{\{(x, y) : x^2 + y^2 \leq 1\}$ and $V_{pq}(x, y)$ is the Zernike function defined before. The following formula for approximating A_{pq} has been proposed in [1]

$$\hat{A}_{pq} = \frac{p+1}{\pi} \sum_{(x_i,y_j)\in D} \sum w_{pq}(x_i, y_j) f(x_i, y_j) \tag{2.21}$$

where i & j are taken such that $(x_i, y_j)\epsilon D$ and

$$w_{pq}(x_i, y_j) = \int_{x_i-\frac{\Delta}{2}}^{x_i+\frac{\Delta}{2}} \int_{y_j-\frac{\Delta}{2}}^{y_j+\frac{\Delta}{2}} V^*_{pq}(x, y)dxdy \tag{2.22}$$

where $\Delta = \frac{2}{N}$ is the pixel width. The weight $w_{pq}(x_i, y_j)$ is approximated by

$$w_{pq}(x_i, y_j) \approx \Delta^2 V^*_{pq}(x_i, y_j). \tag{2.23}$$

The formula for reconstructing an image from a finite number of moments L is

$$\hat{f}(x, y) = \sum_{i=0}^{L} \hat{A}_{p_i q_i} V_{p_i q_i}(x, y) + I(q_i \neq 0)\hat{A}_{p_i,-q_i} V_{p_i,-q_i}(x, y) \tag{2.24}$$

where I($*$) is the indicator function.

2.12 Reconstruction Error

The reconstruction error is defined as

$$MSE = \sum_{(x_i,y_j)\in D} \sum (\hat{f}(x_i, y_j) - f(x_i, y_j))^2 \tag{2.25}$$

The reconstruction error is dependent on a number of factors such as the the number of moments used. For Zernike moments, the reconstruction error consists of two parts. One part of the error is due to using a finite number of moments in the calculation and other comes from the inaccuracy of the computed Zenike moments $\hat{A}_{pq}$. The former can be

reduced by increasing L, while the latter is due to the inherent geometric and numerical integration errors and increases with L. Therefore there is a particular moment value that will give the lowest mean square error [2].

2.13 Images

Mathematically speaking an image is a piece-wise continuous function in two variables defined on a compact support $D \subset R \times R$ and having a finite nonzero integral. Since Zernike moments are orthogonal in two dimensional (2D) space image reconstruction should be performed on a 2D image. Examples of 2D images are grayscale or binary images. A grayscale image is such that each pixel carries only intensity information [17]. This type of image consists of pixels of varying shades of gray pixels which may also include black or white pixels. In an 8-bit grayscale image a black pixel has the value 0 while a white pixel has a value 255. A binary image consists of black and white pixels only. In a grayscale image there are black pixels, white pixels and pixels of many shades of gray. The value of the pixels can vary from 0 to 255. In a binary image it has only black and white pixels. In this case, the black pixel has a value of 0 and a white pixel has a value of 1.

Several methods for image compressions methods are available. Some of the more common methods are JPEG (joint photographic experts group), PNG (portable network graphics) and TIFF (tagged image file format). Image compression can be lossless or lossy. Lossless compression is preferred over lossy compression because a lossless compression will contain the exact information of the original image in reconstruction whereas a lossy compression will contain an approximation to the information of the original image in reconstruction. TIFF is an example of an image compression that is lossless [18].

A colour image can be reconstructed from Zernike moments in one of two ways. The first way would be to reconstruct each of the three colour channels, red, blue and green separately and overlap the three resulting reconstructed images. The original problem is now turned into three separate problems. The second way is to find the corresponding three

dimensional (3-D) Zernike moments given by Flusser [1] and do reconstruction directly. The formula for the 3-D Zernike moment is given as

$$\int_0^{2\pi} \int_0^1 \int_{-1}^1 G_r(z) R_{pq}(\varrho) e^{(-qi\phi)} f(\varrho, \phi, z) \varrho dz d\varrho d\phi. \tag{2.26}$$

Where $G_r(z)$ are 1D orthogonal polynomial and $R_{pq}(\varrho)e^{-qj\phi}$ is the kernel function of some moments orthogonal on the disk.

2.14 Errors in computing Zernike moments

The two types of errors that are committed in computing the Zernike moments are:

- Geometric error

and

- Numeric integration error

The geometric error occurs by replacing the domain of the image function by the circular domain of the Zernike functions. The error produced by the mismatch between the two areas (domains) is called the geometric error. The numerical integration error occurs when the involved 2D integration is approximated by a cubature (double summation) formula [19]. Noise can give inaccurate Zernike moment values though it can be reduced by performing filtering on the image [2].

2.15 Explanation of Geometrical Error

To explain geometric error, suppose a circle of the same size as the image is placed on top of the image. The image overlaps onto a circle as is shown in Figure 1.

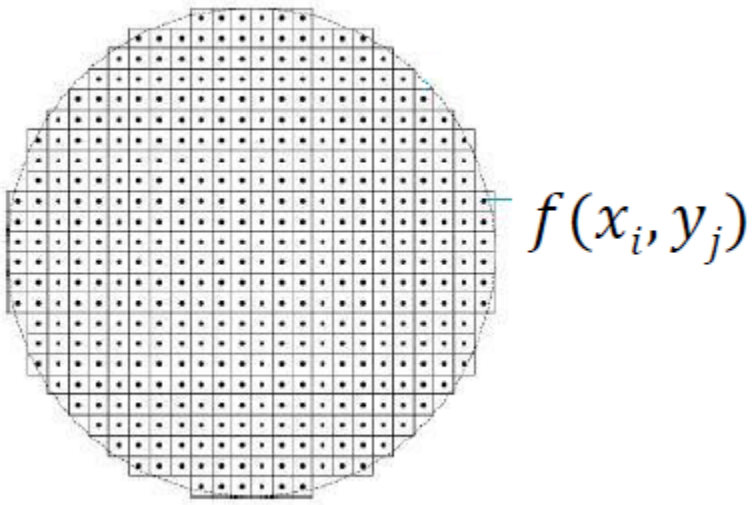

Figure 1: Image to circle overlay

Reconstruction of an image using any algorithm is done in the discrete domain and at discrete point (x_i, y_j) which is the center of the (i, j)th pixel for each pixel. It is assumed that pixels are square in size with width Δ, where $\Delta = x_i - x_{i-1} = y_j - y_{j-1}$. The (i, j)th pixel is defined as $c_{ij} = [x_i - \Delta/2, x_i + \Delta/2] \times [y_j - \Delta/2, y_j + \Delta/2]$.

If a pixel is completely inside the unit circle there is no error. Geometric error is associated with the border pixels and can occur in two ways. We note that the point (x_i, y_j) may or may not lie within the unit circle. First, the center of the pixel (x_i, y_j) lies inside the circle, and the whole pixel is included in the calculation. This is shown in Figure 2.

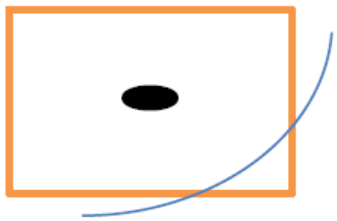

Figure 2: Type of pixel overlay that causes error

The error in this case is due to including that part of the square that lies outside of the circle. In the second case, the center of the pixel (x_i, y_j) lies outside the circle, but there is a part of the pixel inside the circle. This is shown in Figure 3.

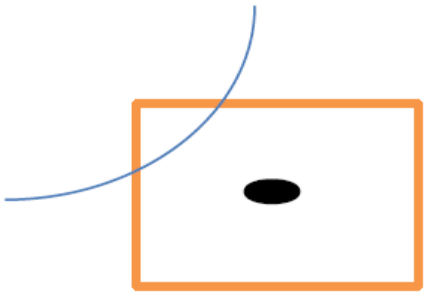

Figure 3: Type of pixel overlay that causes error

Due to the fact that the center of the pixel is outside the circle, this pixel is not included in the calculation. In view of this, part of the pixel inside the circle is left out. Geometric error can be reduced if the image is moved to a circular transform or if the image is encapsulated by a circle [2].

2.16 Determining Accuracy

2.16.1 Accuracy in image reconstruction

Due to the errors discussed in the previous sections, correct Zernike moment values may not be found and accurate image reconstruction may not be achieved. Since image reconstruction is done using the estimates $\hat{A}_{pq}$ of the current Zernike moments A_{pq}, it is important to determine the closeness between the reconstructed image and the original image. The mean squared error (MSE) can be used to assess this amount. The formula for the MSE is given by

$$MSE = \frac{1}{M \times N} \sum_{x_i} \sum_{y_j} \frac{((\hat{f}(x_i, y_j) - f(x_i, y_j))^2}{f^2(x_i, y_j)} \tag{2.27}$$

Where $\hat{f}(x, y)$ is the reconstructed image of the original image $f(x, y)$ and where MxN is the size of the images. A small value for the MSE is an indication that the reconstructed image is close to the original image [2].

2.16.2 Accuracy in invariance properties

Suppose $A_{p_iq_i}, i = 1 \dots L$ are the Zernike moments and $A^{(r)}_{p_iq_i}, \ i = 1 \dots L$ are the corresponding Zernike moments under the operation of rotation or reflection. An agreement between the two sets is given by the following formula

$$\frac{1}{L}\sum_{i=1}^{L}(|A_{p_iq_i}| - |A^{(r)}_{p_iq_i}|)^2. \tag{2.28}$$

Since $A_{p_iq_i}$ and$A^{(r)}_{p_iq_i}, \ i = 1 \dots L$ are unknown, the above formula is estimated using the estimates of $\hat{A}_{p_iq_i}$ and $\hat{A}^{(r)}_{p_iq_i}, \ i = 1 \dots L$. We therefore define the mean square error (MSE) formula as

$$MSE(|\hat{A}|, |\hat{A}^{(r)}|) = \frac{1}{L}\sum_{i=1}^{L}(|\hat{A}_{p_iq_i}| - |\hat{A}^{(r)}_{p_iq_i}|)^2 \tag{2.29}$$

where L is the number of Zernike moments.

$\hat{A}_{p_iq_i}$ and $\hat{A}^{(r)}_{p_iq_i}$ are computed using formula (2.21) in section 2.11. A small value of the mean square error is an indication that Zernike moments are invariant under the operation of rotation or reflection [2].

2.17 Algorithms to calculate Zernike moments

In our thesis, we have considered eight algorithms, namely the traditional method (the Direct method), Belkasim [20], Kintner [20], Factorial Free method [24], Prata [20], Hybrid [22], Q-theta recursive [23] and Modified Prata [21]. Each of these algorithms use formula (2.16) for the Zernike moment given in section 2.7.

In the Direct method, the formula for the radial polynomial is the same as given in section 2.7. The other algorithms use different versions of the polynomials. All algorithms except the Direct and the Factorial Free methods use recursion to find the radial polynomials of higher orders.

2.17.1 Direct

The following flowchart shows how the Direct method obtains radial polynomials for each and every possible combination of moment order and repetition value.

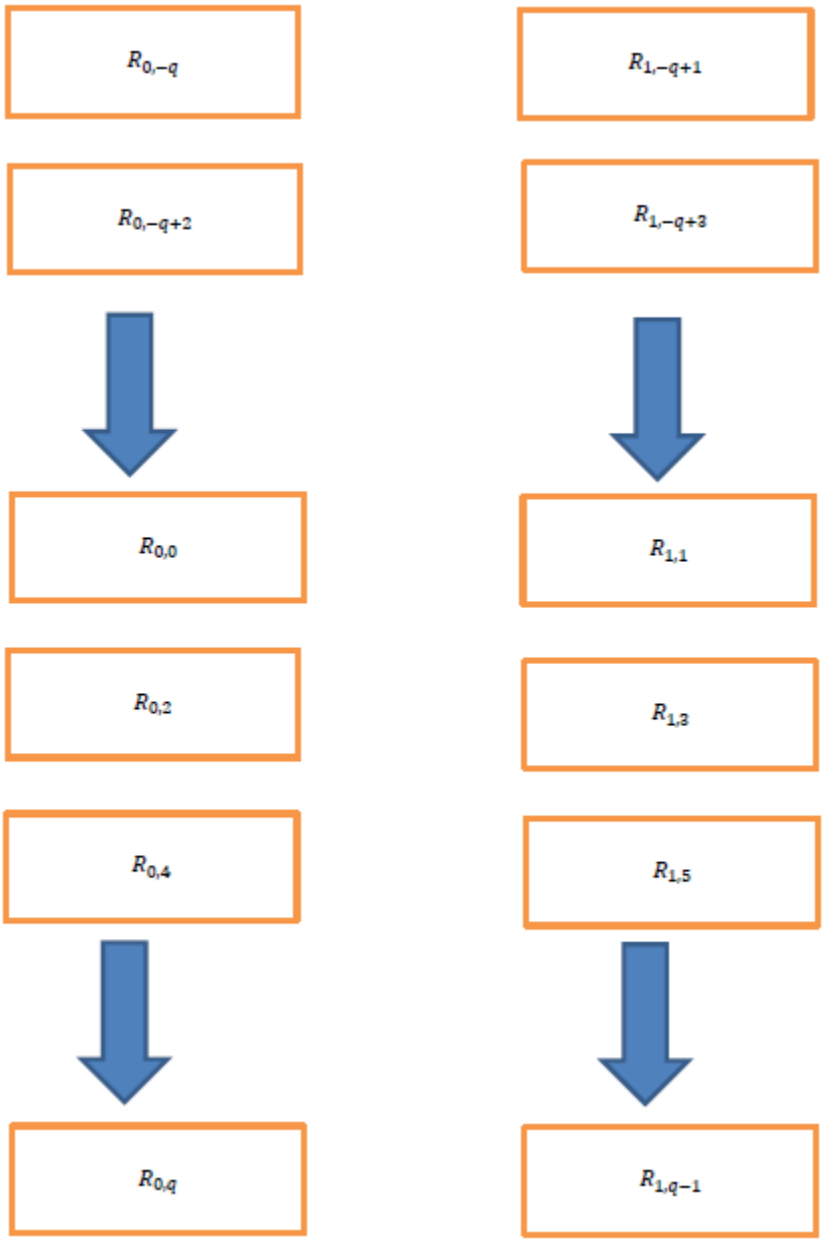

Figure 4: Flowchart for the Direct method

This diagram states that the order begins at 0, goes up continuously until the maximum order is achieved.

2.17.2 Belkasim

In the Belkasim algorithm we express the radial polynomial recursively using the following formula:

$$R_{pq}(r) = B_{pqp}R_{pp}(r) + B_{pq(p-2)}R_{(p-2)(p-2)}(r) + \cdots + B_{pqk}R_{kk}(r) + \ldots B_{pqq}R_{qq}(r) \quad (2.30)$$

This is a recursive way of obtaining each radial polynomial term. This method only uses previous radial terms where the moment order (p) and the repetition value (q) is the same. This is shown in the following diagram.

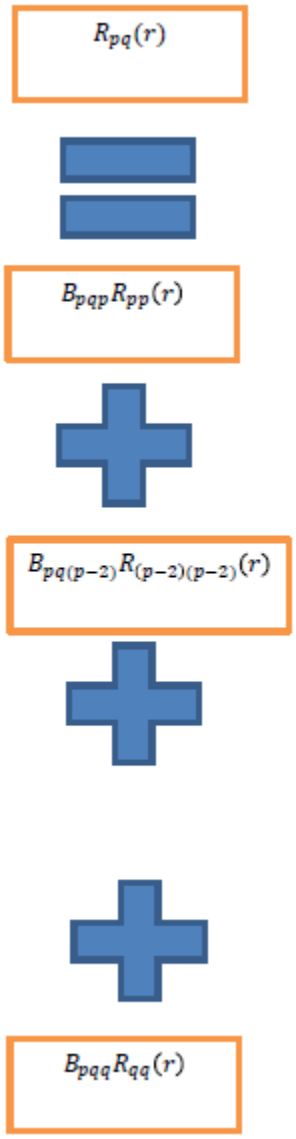

Figure 5: Flowchart for computation of Belkasim method

2.17.3 Kintner

In the Kintner method the radial polynomial is expressed by the following formula:

1. When $p = q$,

$$R_{pq} = r^p \tag{2.31}$$

.

2. When $p - q = 2$,

$$R_{pq} = (q+2)R_{(q+2)(q+2)}(r) - (q+1)R_{qq}(r) \tag{2.32}$$

3. Otherwise,

$$R_{pq}(r) = \frac{(K_2 r^2 + K_3)R_{(p-2)q}(r) + K_4 R_{(p-4)q}}{K_1} \tag{2.33}$$

where

$$K_1 = \frac{(p+q)(p-q)(p-2)}{2} \tag{2.34}$$

$$K_2 = 2(p-1)(p-2) \tag{2.35}$$

$$K_3 = -q^2(p-1) - p(p-1)(p-2) \tag{2.36}$$

$$K_4 = \frac{-p(p+q-2)(p-q-2)}{2} \tag{2.37}$$

The flowchart for obtaining the radial polynomials ($R_{pq}(r)$) using the Kintner method is shown below.

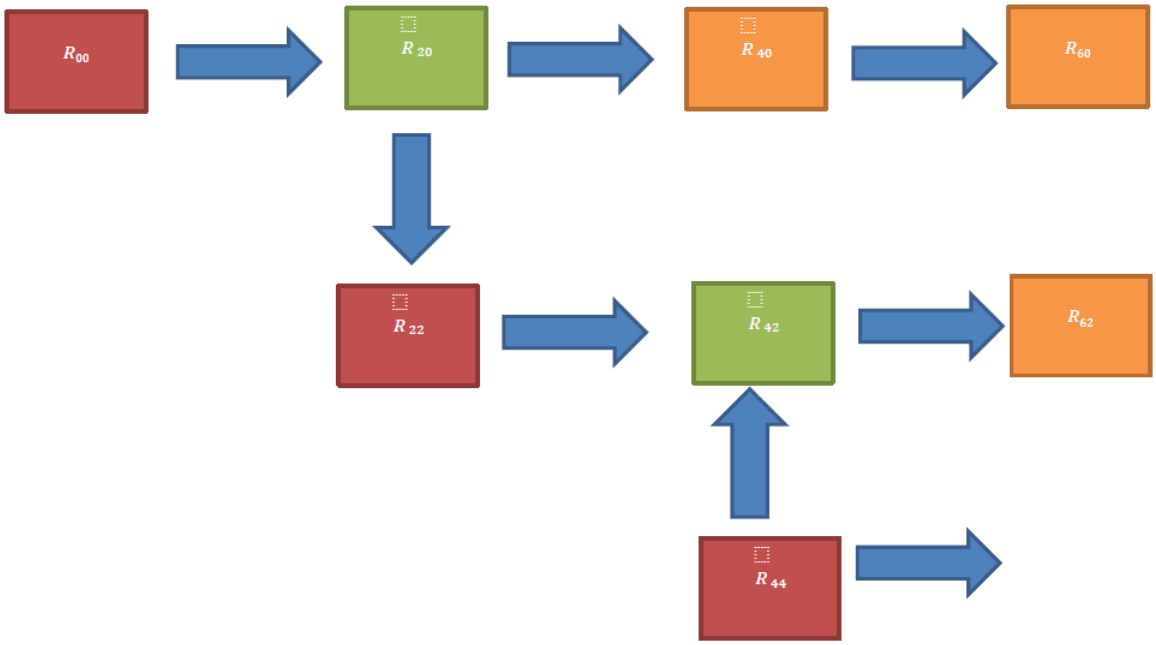

Figure 6: Flowchart for computation of Kintner method

The red boxes represent the terms where the moment order and repetition values are the same and the green boxes represent where the order and repetition values differ by 2. The orange boxes are the terms found from relation (2.33).

2.17.4 Factorial Free

The Factorial Free method finds the radial polynomial using the following formula where

$$R_{pq}(r) = \sum_{s=0}^{(\frac{p-|q|}{2})} (-1^s) T_{pq}(s) r^{p-2s}, \tag{2.38}$$

where

$$T_{qs}(s) = \frac{e^2}{2\pi} \frac{\prod_{i=0}^{2} K_i}{K_1} \frac{(\prod_{i=0}^{3} k_i)^{\frac{1}{2}}}{k_3} \tag{2.39}$$

$$K_i = \frac{k_i^{(1-k_i)}}{k_i + \frac{1}{12}} \tag{2.40}$$

$$k_0 = s + 1 \tag{2.41}$$

$$k_1 = \frac{p + |q|}{2} - s + 1 \tag{2.42}$$

$$k_2 = \frac{p - |q|}{2} - s + 1 \tag{2.43}$$

$$k_3 = p - s + 1. \tag{2.44}$$

We note that like the Direct method, the Factorial Free method does not use any recursive relation. Therefore, the flowchart for this method will be similar to that of the Direct method.

2.17.5 Prata

The steps to obtaining the radial polynomial function through this method is given below:

1. If the repetition value is 0, use the Direct method

2. If the repetition value and moment order are the same, (ie. $p = q$), relation (2.31) is used.

3. Otherwise,

$$R_{pq}(r) = L_1 R_{(p-1)(q-1)}(r) + L_2 R_{(p-2)q}(r) \tag{2.45}$$

$$L_1 = \frac{2rp}{p+q} \tag{2.46}$$

and

$$L_2 = -\frac{p-q}{p+q} \tag{2.47}$$

Terms such as R_{20} and R_{31} are found using relation (2.45).

$$R_{20}(r) = L_1 R_{11} + L_2 R_{00},$$
$$R_{31}(r) = L_1 R_{20} + L_2 R_{11}.$$

The following diagram describes the Prata algorithm.

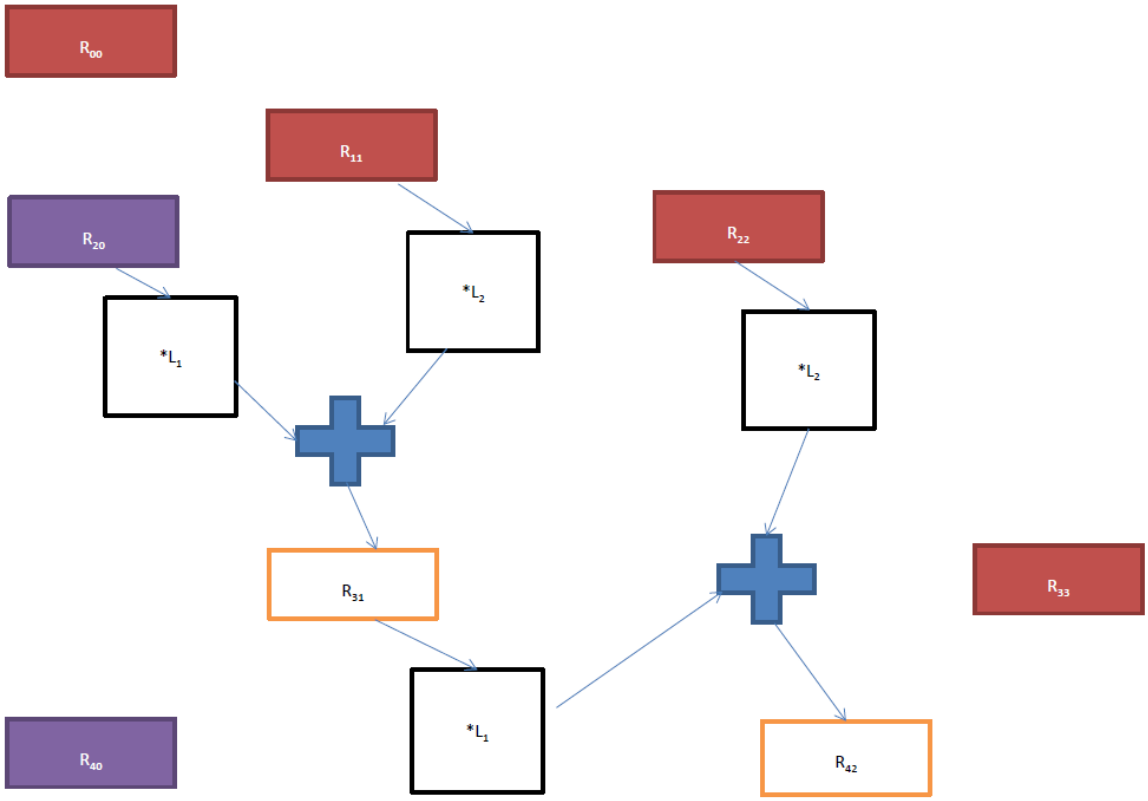

Figure 7: Flowchart for computation of Prata method

The red boxes represent the terms obtained from relation (2.31) while the purple boxes represent the first situation and are therefore calculated from the Direct method. Terms corresponding to these two situations are calculated independently. The white boxes with the orange borders represent terms obtained from relation (2.45).

2.17.6 Hybrid

Note: The hybrid method uses several algorithms such as Kintner, Prata, etc. The hybrid method is expressed by the following steps:

- When the moment order and the repetition value is the same, the radial polynomial is computed using equation (2.31).
- For the specific case when the moment order is 2 and the repetition value is 0, (ie.

R_{20}), the radial polynomial value is found from

$$R_{p(p-2)}(r) = pR_{pp}(r) - (p-1)R_{(p-2)(p-2)}(r) \tag{2.48}$$

- For the remaining cases when the repetition value is equal to 0, the radial polynomial terms are found from Kintner's method given by relation (2.33).
- Prata's method given by relation (2.45) is used to obtain the radial polynomial for other the cases.

It then follows that:

$$\begin{aligned}
R_{00}(r) &= 1 \\
R_{20}(r) &= r^2 - 1 \\
R_{40}(r) &= \frac{(K_2 r^2 + K_3)(r^2 - 1) + K_4}{K_1} \\
R_{31}(r) &= L_1 R_{20} + L_2 R_{11} \\
R_{42}(r) &= L_1 R_{31}(r) + L_2 R_{22}.
\end{aligned}$$

The following diagram shows the Hybrid method.

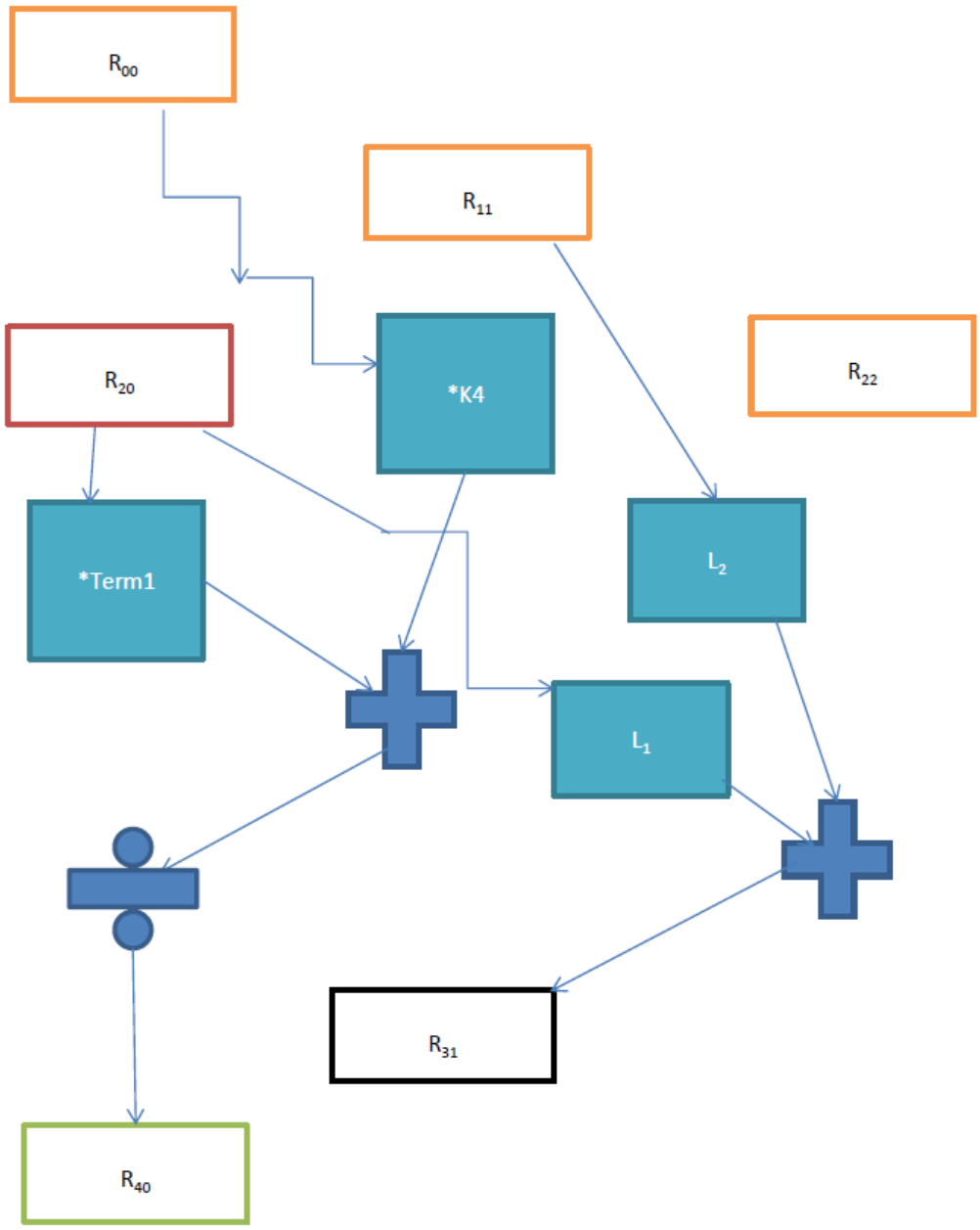

Figure 8: Flowchart for computation of Hybrid method

The white box with the green border is computed from Kintner's method while the white box with the black border is computed from Prata's method.

2.17.7 Q-theta

The Q-theta (q-θ) recursive algorithm employs recurrence relationships of the trigonometric functions. The algorithm for q-theta recursive is given below:

- If $p = q$, use equation (2.31).
- If $p - q = 2$,

$$R_{p,(p-2)}(r) = pr^p - (p-1)r^{p-2} \tag{2.49}$$

- Otherwise

$$R_{pq}(r) = H_1 R_{p,q+4} + (H_2 + \frac{H_3}{r^2}) R_{p,q+2}, \tag{2.50}$$

where the terms H_1, H_2 and H_3 are given as

$$H_1 = \frac{(q+4)(q+3)}{2} - (q+4)H_2 + H_3 \frac{(p+q+6)(p-q-4)}{8} \tag{2.51}$$

$$H_2 = \frac{H_3(p+q+4)(p-q-2)}{4(q+3)} + q + 2 \tag{2.52}$$

$$H_3 = \frac{4(q+2)(q+1)}{(p+q+2)(p-q)}. \tag{2.53}$$

Using this method R_{40} will be calculated in the following manner:

$$R_{40}(r) = H_1 R_{44} + \frac{(H_2 + H_3))R_{42}(r)}{r^2}.$$

The following diagram shows the Q-theta method.

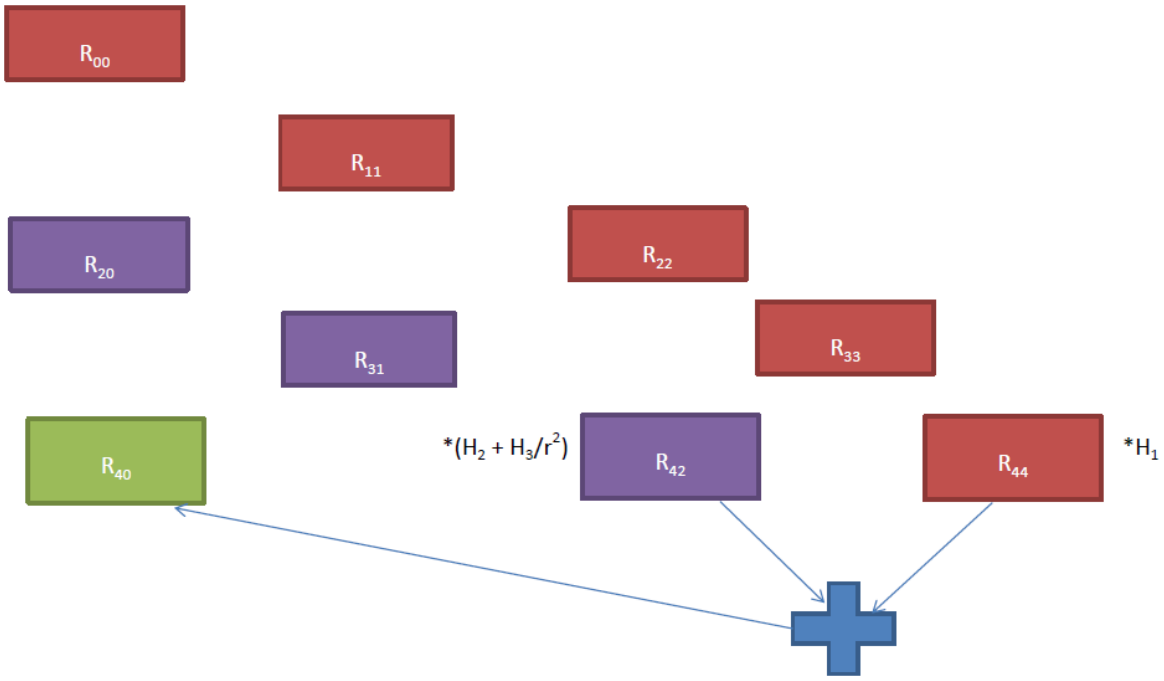

Figure 9: Flowchart for computation of Q-θ method

2.17.8 Modified Prata

The equations used in the Modified Prata method are as follows:

- If $p = q$, use equation (2.31).
- If $p - q = 2$,

$$R_{pq}(r) = pR_{pp}(r) - (p-1)R_{(p-2)(p-2)}(r) \tag{2.54}$$

- In the general case,

$$R_{(p+q)q}(r) = L_1 R_{(p+q-1)|q-1|}(r) + L_2 R_{(p+q-2)q}(r) \tag{2.55}$$

From the above formula it follows that

$$R_{00}(r) = 1$$

$$R_{11}(r) = r$$

$$R_{22}(r) = r^2$$

$$R_{31}(r) = 2R_{33}(r) - 2R_{11}(r) = 3r^3 - 2r$$

$$R_{33}(r) = r^3$$

$$R_{20}(r) = 2R_{22} - R_{00} = 2r^2 - 1$$

$$R_{42}(r) = 4R_{44}(r) - 3R_{22}(r) = 4r^4 - 3r^2.$$

The following diagram shows the method.

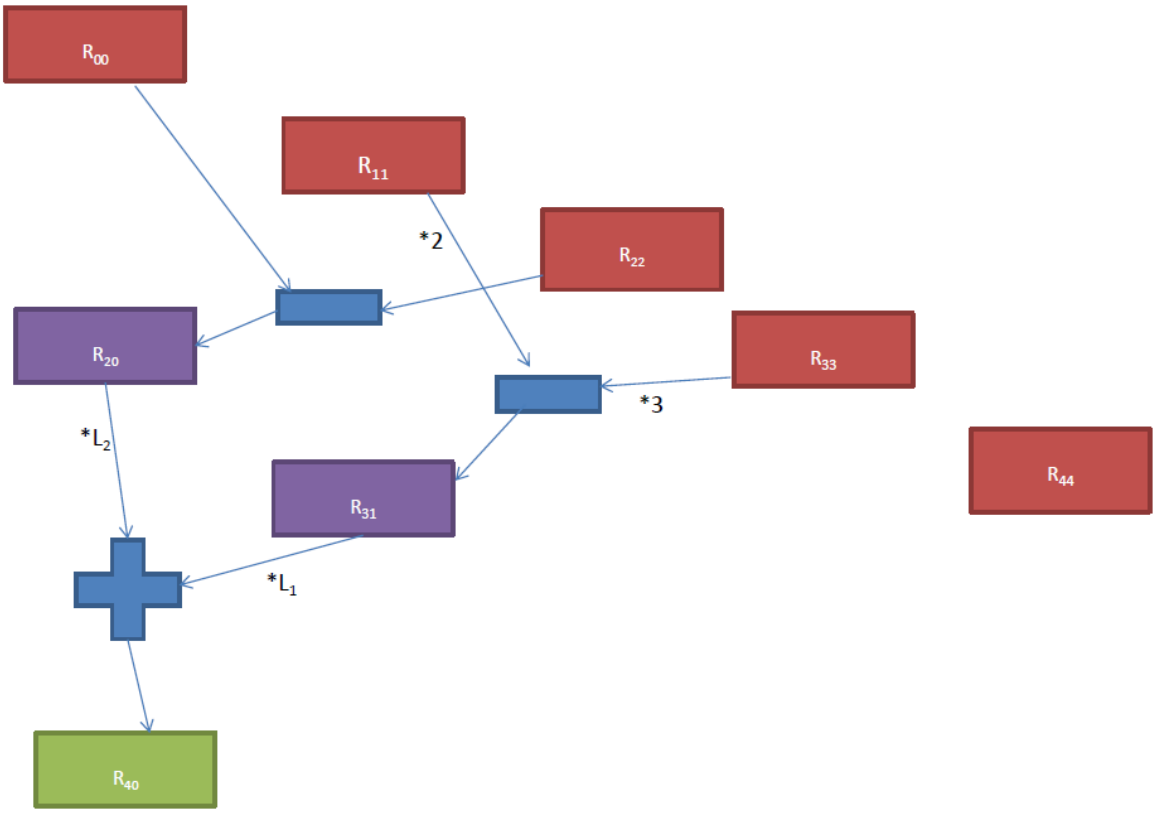

Figure 10: Flowchart for computation of Modified Prata method

The purple boxes are obtained from relation (2.54) and the green box is obtained from relation (2.55).

3 Chapter 3: Noise types that occur in images

3.1 Noise

Noise is an unwanted component of an image function. Due to the random nature of noise it can be described by a random variable. Let $f(x, y)$ be the original (noise free version) image function and $g(x, y)$ be the noisy version of $f(x, y)$. The noisy version can be decomposed into two parts: namely the original function $f(x, y)$ and a noise component $z(x, y)$. Image decomposition models can be classified as additive or multiplicative. Additive image decomposition models are more common than multiplicative decomposition models. Gaussian noise is often used in an additive noise model. An additive noise model is expressed mathematically as

$$g(x, y) = f(x, y) + z(x, y). \tag{3.1}$$

A multiplicative noise model is expressed mathematically as

$$g(x, y) = f(x, y)z(x, y). \tag{3.2}$$

A type of noise often modelled as multiplicative is speckle noise. An additive noise model can be changed into a multiplicative noise model by taking the exponent of both sides of relation (3.1). Similarly, a multiplicative noise model can be transformed into an additive noise model if the logarithm of both sides of relation (3.2) is taken. There are some types of noise that cannot be used in either the additive or multiplicative noise models. Poisson noise and salt & pepper noise are examples of this kind of noise.

3.2 Consistent Estimation of an Image Function in the Presence of White Noise in an additive image independent noise model

Ideally it would be nice to eliminate noise; however in reality the best that can be done is to find ways to reduce noise. This problem turns into a statistical estimation problem. We use the notations in Section 3.1. According to the additive noise model we have $g(x, y) = f(x, y) + z(x, y)$ where $z(x, y)$ is the zero-mean stationary noise process with auto-correlation function

$$Cov(z(x, y), z(x', y')) = \sigma^2 k(x - x', y - y') \tag{3.3}$$

where $k(x, y)$ is a positive function and is symmetric with respect to each argument. If $k(x, y) = \delta(x, y)$ (with $\delta(x, y)$ being a delta function) then $z(x, y)$ is a white noise. Zernike moment of order p and repetition q of the image function $f(x, y)$ is defined as follows:

$$A_{pq}(f) = \int_D V^*_{pq}(x, y) f(x, y) dxdy \tag{3.4}$$

where $V^*_{pq}(x, y)$ is the conjugate of $V_{pq}(x, y)$, the Zernike function of order p and repetition q defined in section 2.7 and $\int_D$ denotes double integration over the unit disk. Since f is unknown the Zernike moment is unknown. The corresponding Zernike moment using $g(x, y)$ in place of $f(x, y)$ is:

$$A_{pq}(g) = \int_D V^*_{pq}(x, y) g(x, y) dxdy \tag{3.5}$$

We want to quantify the basic statistical properties of $A_{pq}(g)$ by evaluating the mean and variance $A_{pq}(g)$. We assume the continuous model for the original image and the observed noisy image. Hence the discretization error is ignored. For the above additive model we have $E\{A_{pq}(g)\} = A_{pq}(f)$, that is $A_{pq}(g)$ is an unbiased estimate of $A_{pq}(f)$. The following calculation gives the variance of $A_{pq}(g)$:

Using relation (3.3) we get,

$$var\{A_{pq}(g)\} = Cov(\int_D V_{pq}^*(x,y)g(x,y)dxdy, \int_D V_{pq}^*(x,y)g(x,y)dxdy) \tag{3.6}$$

$$= \int_D \int_D V_{pq}^*(x,y)V_{pq}(x',y')cov\{g(x,y), g(x',y')\}dxdydx'dy' \tag{3.7}$$

$$= \sigma^2 \int_D \int_D V_{pq}^*(x,y)V_{pq}(x',y')k(x-x',y-y')dxdydx'dy' \tag{3.8}$$

For the white noise the expression in (3.8) takes the following form

$$Var\{A_{pq}(g)\} = \sigma^2 \int_D \int_D V_{pq}^*(x,y)V_{pq}(x',y')\delta(x-x',y-y')dxdydx'dy', \tag{3.9}$$

$$= \sigma^2 \int_D |V_{pq}(x,y)|^2 dxdy = \sigma^2 \frac{\pi}{p+1}. \tag{3.10}$$

We note that the variance of $A_{pq}(g)$ is the same as the mean square error $E(A_{pq}(g) - A_{pq}(f))^2$ becomes smaller and smaller as p becomes larger and larger. This is true for all q values, where, $|q| \leq p, p-|q|$ is an even number. We therefore conclude that in the presence of white noise in an additive noise model, $A_{pq}(g)$ is a consistent estimate of $A_{pq}(f)$ as p (the order) tends to infinity.

Further Notes: For an additive image dependent noise model given by

$$g(x,y) = f(x,y) + f^{\alpha}(x,y)z(x,y) \tag{3.11}$$

where $0 \leq \alpha \leq 1$ and $z(x,y)$ is a zero-mean white noise with variance σ^2, a straightforward calculation shows that $E\{A_{pq}(g)\} = A_{pq}(f)$ and $Var\{A_{pq}(g)\} = \sigma^2 \int_D |V_{pq}(x,y)|^2 f^{2\alpha}(x,y)dxdy$. Similarly, for a multiplicative noise model given by $g(x,y) = f(x,y)z(x,y)$ where $z(x,y)$ is a stationary noise process with $Ez(x,y) = 1$, a straightforward calculation shows $E\{A_{pq}(g)\} = A_{pq}(f)$ and $Var\{A_{pq}(g)\} = \sigma^2 \int_D |V_{pq}(x,y)|^2 f^2(x,y)dxdy$. The following plots show the above three noise models

for the same periodic sinusoidal signal.

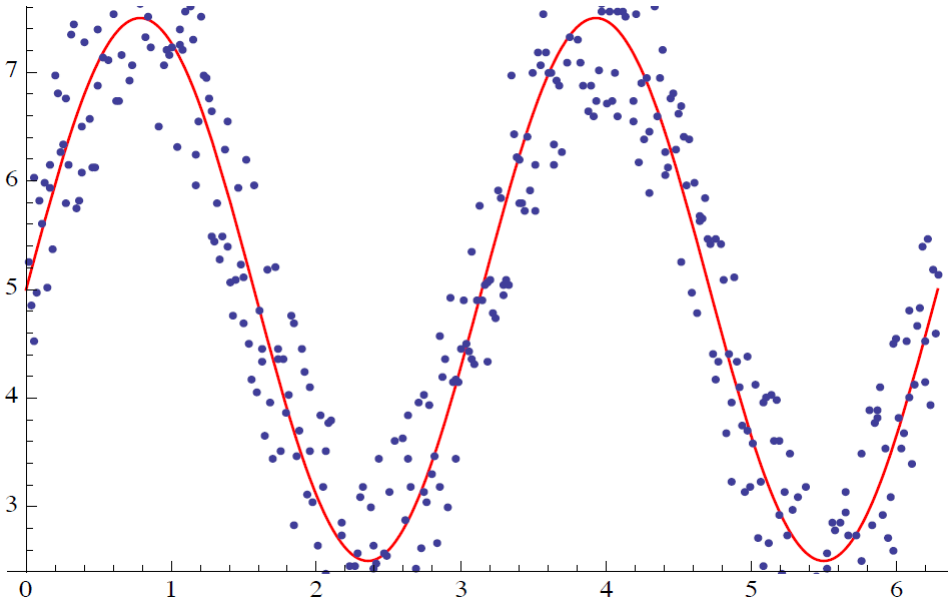

Figure 11: Additive Image Independent Noise Model

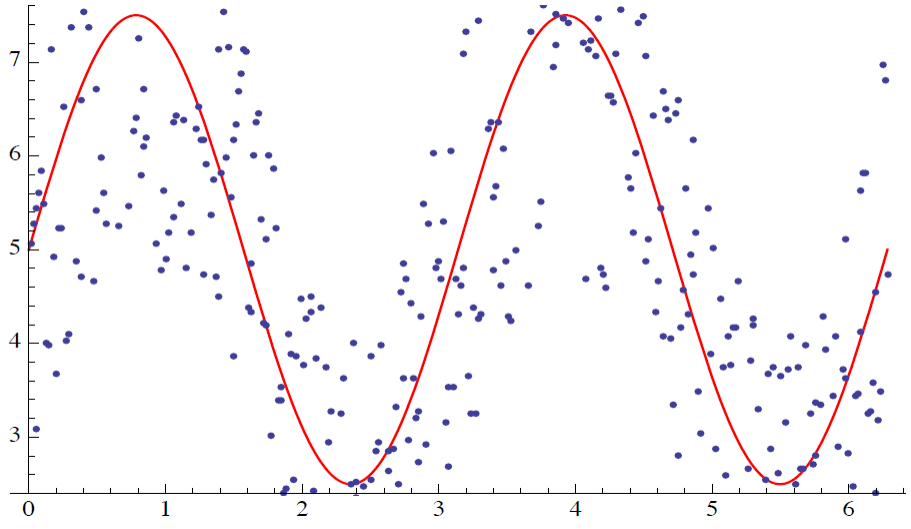

Figure 12: Additive Image Dependent Noise Model (Film Grain Noise with $\alpha = 0.5$

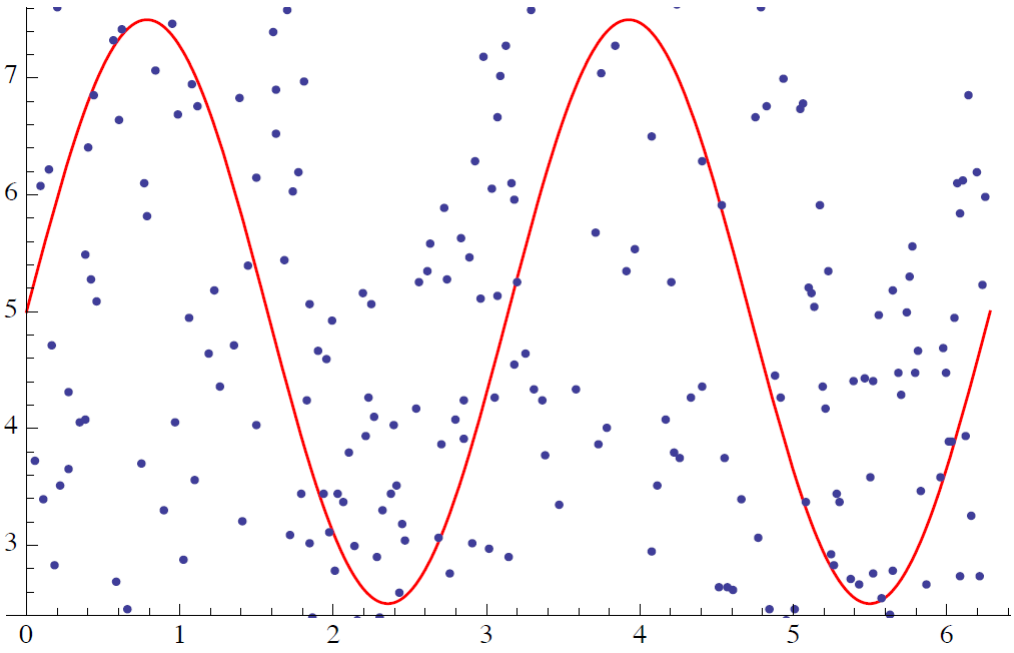

Figure 13: Multiplicative Noise Model

3.3 Types of Noise Distributions

In the following we shall briefly describe three types of noise.

1. Gaussian Noise

   The Gaussian noise is the most frequently occurring noise in additive noise models. It is used to model thermal noise, film grain noise and photon counting noise. The density function of uninvariate Gaussian noise q, with mean μ and variance σ^2 is

$$p_q(x) = (2\pi\sigma^2)^{(1/2)} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \tag{3.12}$$

$$-\infty < x < \infty, -\infty < \mu < \infty, 0 < \sigma^2 < \infty$$

2. Heavy-Tailed Noise

   Tails of some noise distributions approaches 0 more slowly than the tails of the Gaussian noise distribution. This type of noise is known as a heavy-tailed noise.

An example of a heavy-tailed noise is static on a weak broadcast AM radio station during a lightning storm. The density functions of some of the heavy-tailed noises are given below:

- Double Exponential

$$p_a(x) = \frac{1}{2\sigma} e^{\frac{-|x-\mu|}{\sigma}} \tag{3.13}$$

$$-\infty < x < \infty, -\infty < \mu < \infty, 0 < \sigma < \infty$$

- Negative Exponential

  The pdf is

$$p_a(x) = \frac{1}{\mu} e^{-\frac{x}{\mu}} \tag{3.14}$$

$$0 < x < \infty, 0 < \mu < \infty$$

- Alpha Stable

  The pdf of a simple alpha stable noise model is the Caucy pdf given by

$$p_a(x) = \frac{\sigma^2}{\pi} \frac{1}{\sigma^2 + (x-\mu)^2} \tag{3.15}$$

$$-\infty < x < \infty, -\infty < \mu < \infty, 0 < \sigma^2 < \infty$$

- Gaussian mixture model

  The pdf is

$$p_a(x) = (1-\alpha)p_0(x) + \alpha p_1(x) \tag{3.16}$$

  where $p_0(x)$ and $p_1(x)$ are Gaussian densities with different means of μ_0 and μ_1 and different variances σ_0^2 and σ_1^2 respectively and $0 \leq \alpha \leq 1$.

3. Salt & Pepper Noise

   Salt & Pepper noise refers to many processes which results in a few pixels being very affected by noise. Salt & Pepper noise is an example of (very) heavy-tailed noise. A simple model is the following. Let $f(x, y)$ be the original image and $g(x, y)$ be the image after it have been altered by salt & pepper. The probability function of this noise is:

$$P(g = f) = 1 - \alpha \tag{3.17}$$

$$P(g = max) = \frac{\alpha}{2} \tag{3.18}$$

$$P(g = min) = \frac{\alpha}{2} \tag{3.19}$$

$$0 < \alpha < 1$$

   where max and min refer to the maximum and minimum pixel values in the image. This means that with probability 1-α, the pixels are unaltered and with probability α, the pixels are changed to either the largest or smallest value. The altered pixels look like black and white dots sprinkled over the image. To learn more about the different noise types, please consult [25].

4. Zernike Moments from Noisy Images: Discrete Models

   We now assume that discrete data are observed on the lattice points (pixels) defined within D. Hence the NxN noisy image in the additive image independent noise model is given by

$$g(x_i, y_i) = f(x_i, y_i) + z(x_i, y_i);\ 1 \leq i, j \leq N \tag{3.20}$$

   We assume that the image resolution is controlled by the pixel size Δ^2 where $\Delta = \frac{2}{N}$. The Poisson noise model is one discrete noise model that we now describe. Under

the Poisson noise model,

$$g(x_i.y_j) = Poisson\{f(x_i, y_j)\} \tag{3.21}$$

Using the basic properties of Poisson random variable we have $E\{g(x_i, y_j\} = f(x_i, y_j)$ and $Var\{g(x_i, y_j)\} = f(x_i, y_j)$. Hence $g(x_i, y_j)$ is virtually equivalent to the model $g(x_i, y_j) = f(x_i, y_j) + \sqrt{f(x_i, y_j)}z(x_i, y_j)$ where $z(x_i, y_j)$ is a white noise with mean 0 and variance 1 for each $i, j, 1 \leq i, j \leq N$.

We next discuss the salt & pepper noise model. The Cartesian coordinate estimate of the Zernike moment is

$$\tilde{A}_{pq}(g) = \sum_{(x_i,y_j)\in D} \int_{x_i-\frac{\Delta}{2}}^{x_i+\frac{\Delta}{2}} \int_{y_j-\frac{\Delta}{2}}^{y_j+\frac{\Delta}{2}} V_{pq}^*(x, y)dxdy. \tag{3.22}$$

Similarly, the Zernike moment estimate in polar coordinates is given by the following formula

$$\tilde{A}_{pq}(g) = \sum_{l=1}^{U}\sum_{k=1}^{V} \tilde{g}(\rho_l, \theta_k) \int_{\rho_l-\Delta\rho}^{\rho_l+\Delta\rho} \int_{\theta_l-\Delta\theta}^{\theta_l+\Delta\theta} R_{pq}(p)e^{-jq\theta}\rho d\rho d\theta. \tag{3.23}$$

4 Chapter 4

4.1 Design of Experiment

The algorithms considered in our work were implemented using Matlab software and were run on a PC computer with Windows 7 as the operating system. These programs were similar in format. The input images used were all of TIFF file format. The reason for choosing TIFF images is because the TIFF file format is a lossless compression format and this will ensure that the most accurate results are acheived. The input image 'Lenna' that is shown in Appendix 1, 2 and 3 is allowed for usage by the original owner [26]. The image sizes considered are listed in Table 1.

Image size
10x10
32x32
64x64
128x128
256x256
291x240
367x490
480x640

Table 1: List of Image sizes used.

Three experiments are performed. All image sizes except 128x128 are used in the first experiment and the image sizes listed in the first four rows of the above table are used in the first part of the second experiment. The image size 64x64 is used in the second and third parts of Experiment 2. In the first experiment, using each of the seven algorithms and the classical method, image reconstruction is performed. In the second experiment, using the classical method, image reconstruction with a noisy image as input is performed. In the third experiment, using the classical method, invariance properties of the Zernike moments

is examined. In this experiment, only one image size (64x64) is used.

4.2 Experiment 1

In the first experiment, an input image of a particular size is decomposed into a set of Zernike moments using formula (2.16). These Zernike moments are estimated using formula (2.21). This set of estimated moments are used to reconstruct an image using (2.24). The reconstructed image is then compared with the input image using the normalized mean square as a measure of error. The formula for this mean square error is given in relation (2.27). This procedure is performed for moment order 0, 10, 20 and so on. The mean square errors are plotted against the moment orders. This procedure was repeated for images of other sizes.

4.3 Observations

The reconstructed images for different image sizes and for different computational procedures are shown in Appendix 1. It is seen that the output images do not resemble the input image for any of the seven algorithms. For the classical method there is a resemblance between the constructed image and the input image only when the image has equal height and width. For images with unequal height and width there is some resemblance between the constructed image and the input image although the reconstructed image has some aberrations. The plots of the mean square errors against the moment orders for the classical method corresponding to these different image sizes are shown in the following diagrams.

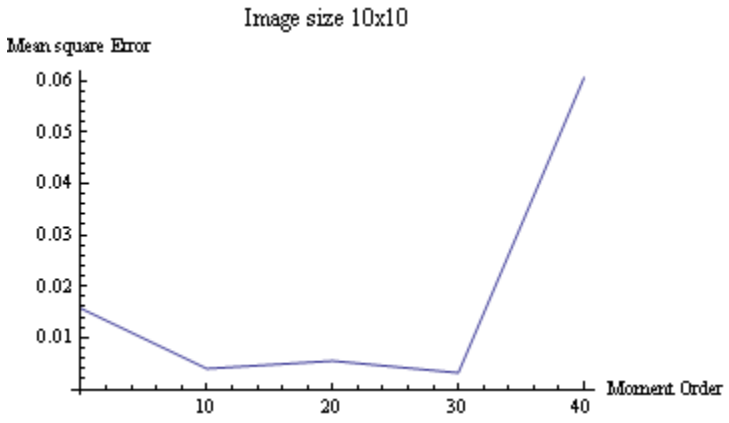

Figure 14: Plot of mean square error against moment order for 10x10 image

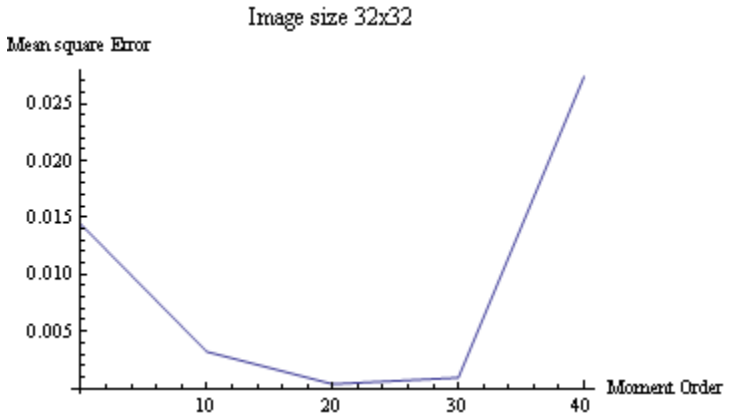

Figure 15: Plot of mean square error against moment order for 32x32 image

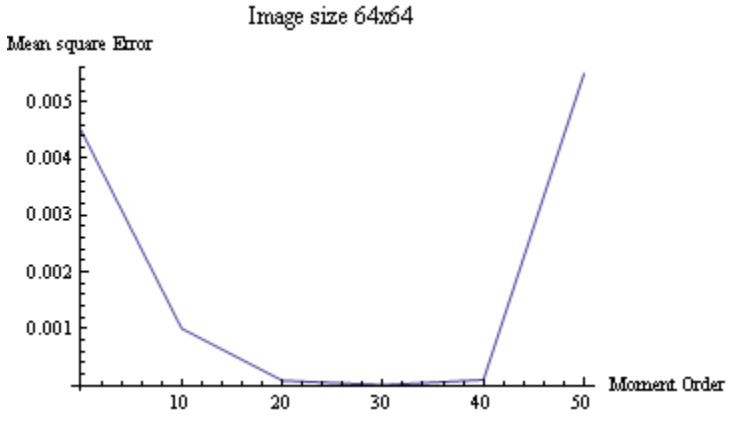

Figure 16: Plot of mean square error against moment order for 64x64 image

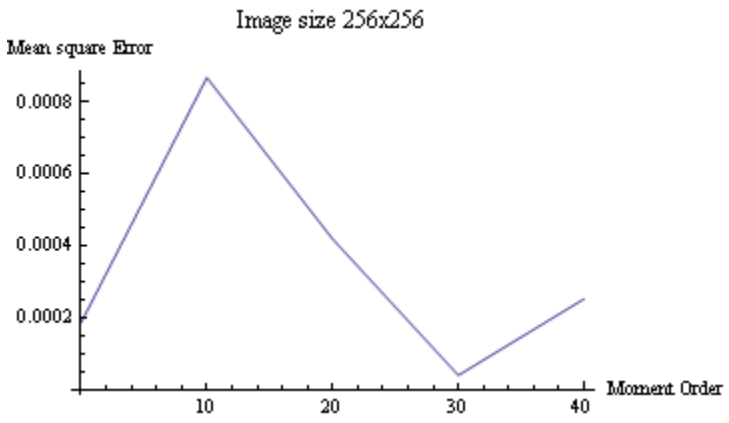

Figure 17: Plot of mean square error against moment order for 256x256 image

For smaller image sizes, the above plots possess different U-shapes, but this property is not shared by the plot for image size 256x256. We now define the optimal moment order as that particular value of p for which the reconstructed image most resembles the input image.

The following table shows the optimal moment order for the classical method when there is a complete resemblance between the reconstructed image and the input square image. The optimal moment order can also mean the moment order for which the mean square attends the smallest value. This is true for all image sizes except for 256x256.

Image size	10x10	32x32	64x64	256x256
Optimal Moment Order	30	30	40	40

Table 2: The optimal moment order for the classical method for different image sizes.

From the table it is seen that for the classical method the optimal order does change as the image size varies. In fact the optimal order does not decrease as the image size

increases. For the sake of completeness, the plots of the mean square errors against the moment orders for the other image sizes for the classical method are shown below.

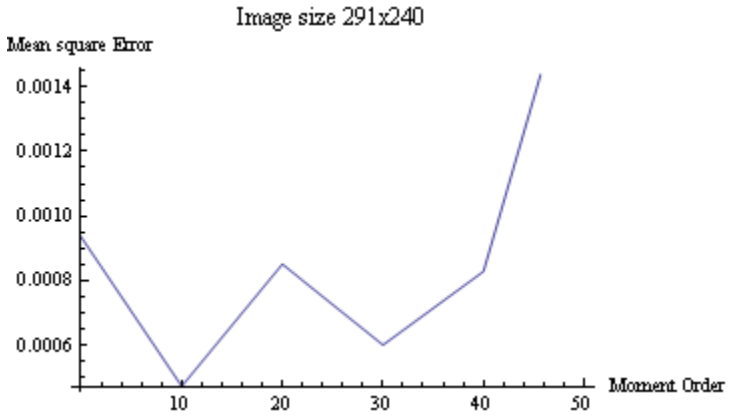

Figure 18: Plot of mean square error against moment order for 291x240 image

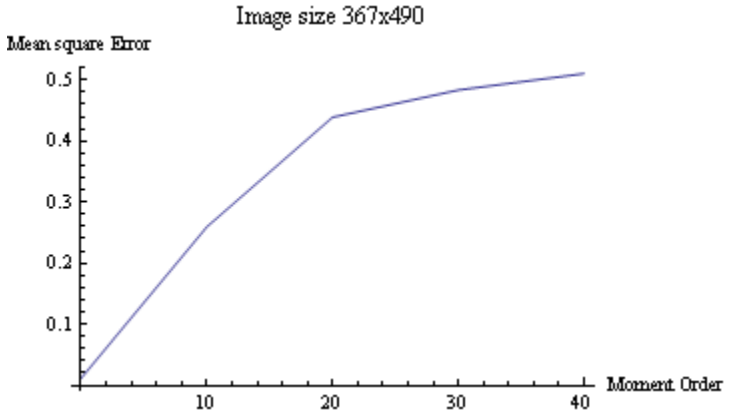

Figure 19: Plot of mean square error against moment order for 367x490 image

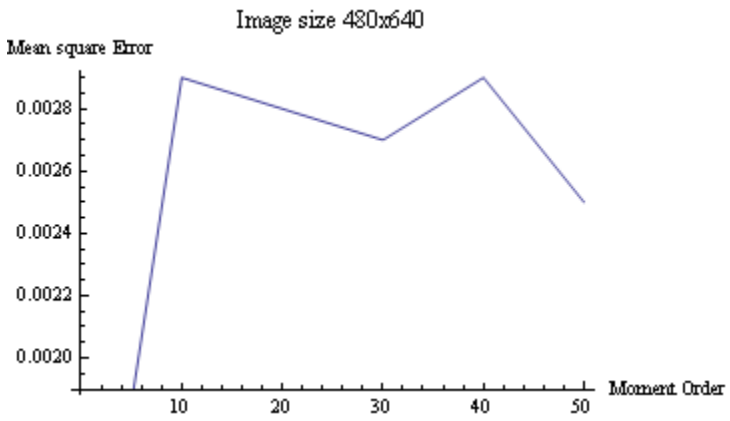

Figure 20: Plot of mean square error against moment order for 480x640 image

For the sake of completeness, the mean square errors against the moment orders for all algorithms corresponding to one image size (64x64) are shown below.

The Belkasim Method

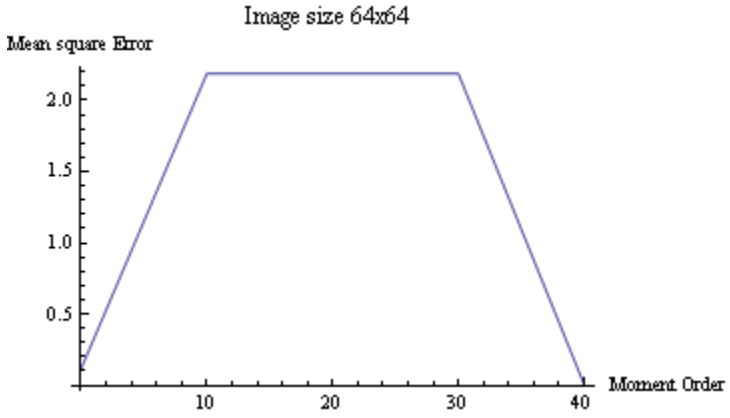

Figure 21: Plot of mean square error against moment order for Belkasim method

The Factorial Free Method

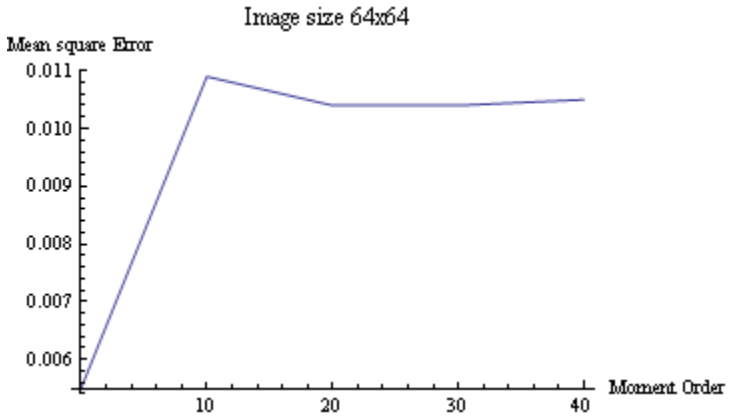

Figure 22: Plot of mean square error against moment order for Factorial Free method

The Hybrid Method

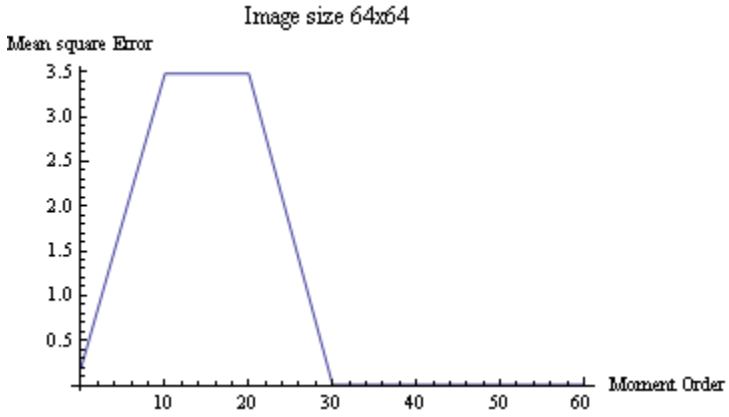

Figure 23: Plot of mean square error against moment order for Hybrid method

The Kintner Method

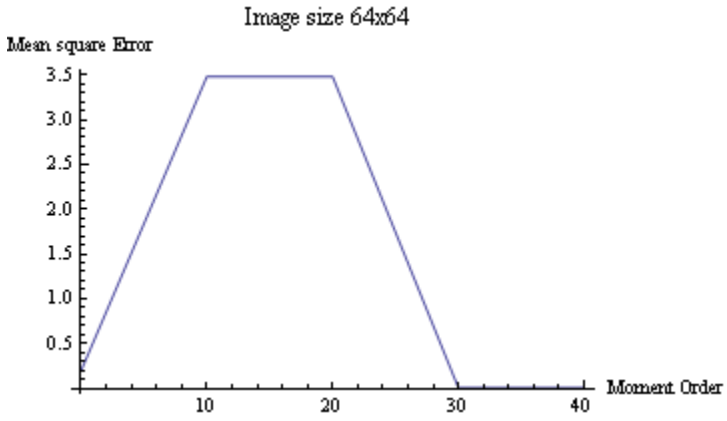

Figure 24: Plot of mean square error against moment order for Kintner method

The Modified Prata Method

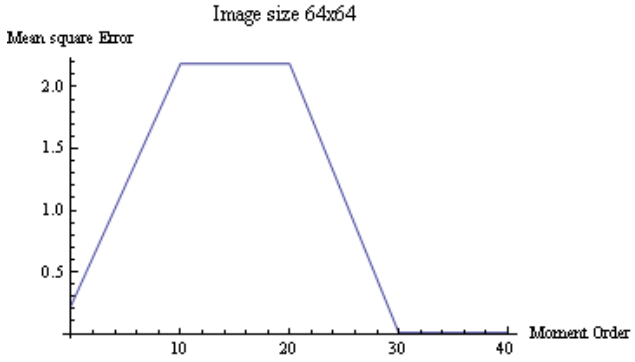

Figure 25: Plot of mean square error against moment order for Modified Prata method

The Prata Method

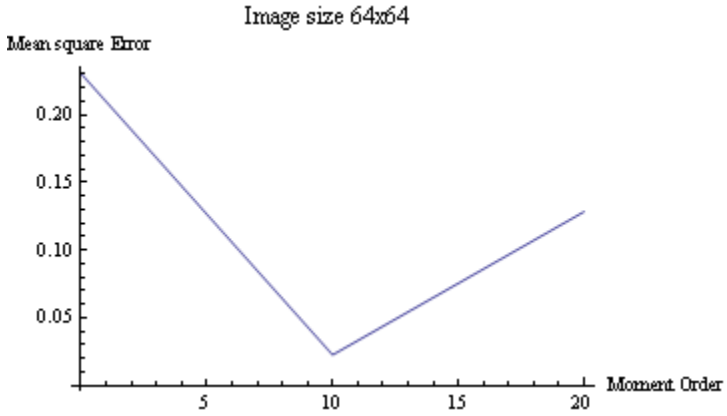

Figure 26: Plot of mean square error against moment order for Prata method

The Q-theta Method

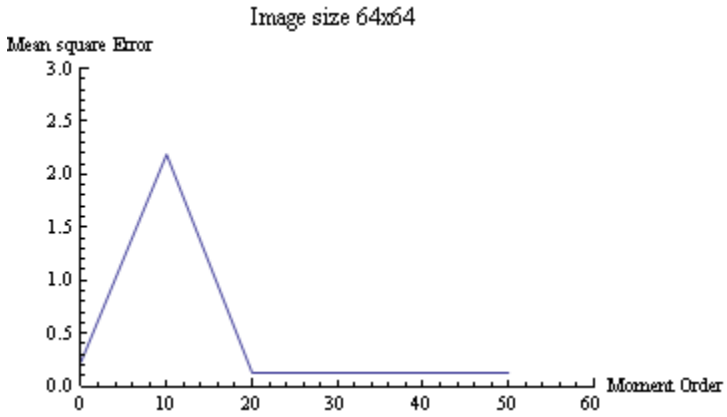

Figure 27: Plot of mean square error against moment order for Q-theta method

We have seen in experiment 1 that the output images produced by each of the seven algorithms under study do not resemble the input image. Hence the classical procedure with square images has been used alone in experiment 2 and experiment 3.

4.4 Experiment 2

The experiment has three parts.

4.5 (a) Part 1

This part of the experiment consists of adding Gaussian white noise (with mean 0 and variance 1) to images of four sizes (10x10, 32x32, 64x64, 128x128). Using the procedures outlined in Experiment 1, the input image is decomposed into a set of Zernike moments and the reconstructed image is obtained using the estimates of the Zernike moments. A plot of the mean square errors against the moment orders is made and the optimal moment order is determined when the output image resembles the input image.

4.6 Observations

The reconstructed images produced by the classical procedure are listed in Appendix 2. These reconstructed images do not resemble the input images. The classical methods is not robust in the presence of Gaussian noise. The plots of the mean square errors against the moment orders for different image sizes are shown below.

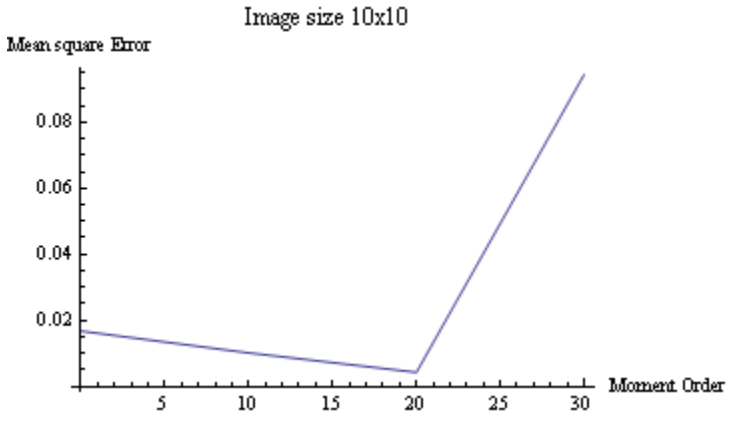

Figure 28: Plot of mean square errors against moment orders with Gaussian noise

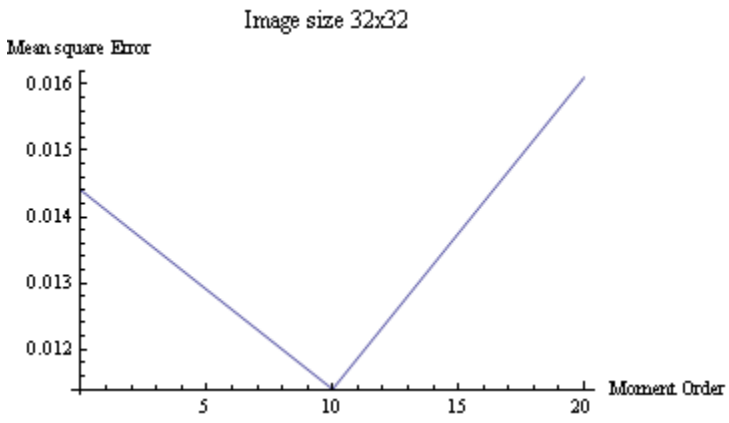

Figure 29: Plot of mean square errors against moment orders with Gaussian noise

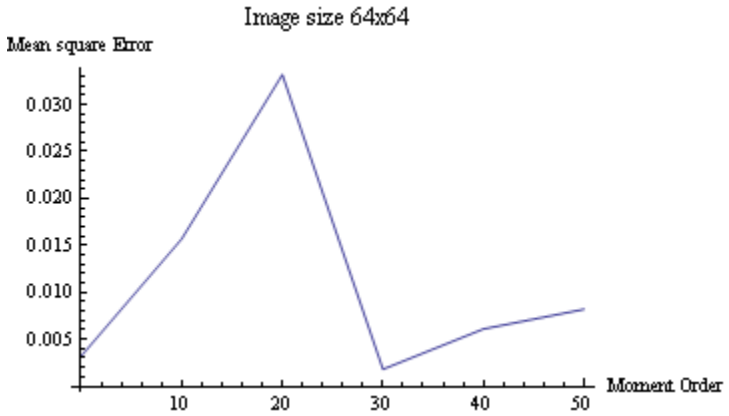

Figure 30: Plot of mean square errors against moment orders with Gaussian noise

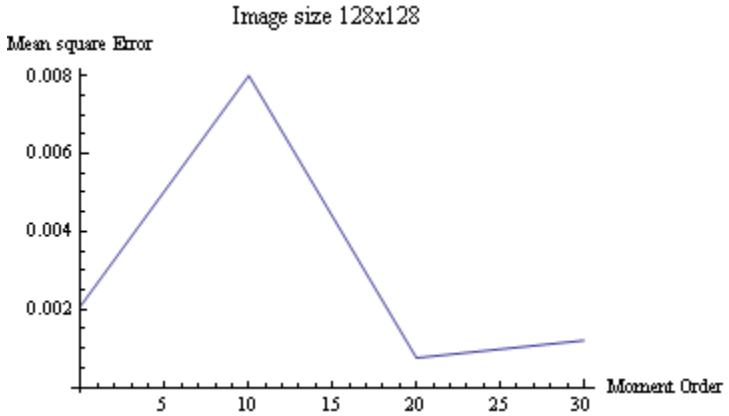

Figure 31: Plot of mean square errors against moment orders with Gaussian noise

4.7 (b) Part 2

In the second part of the experiment different noise types are added to the input image of one size (64x64). The noise types considered were Gaussian, Poisson and salt & pepper. We then proceed as in Part 1.

4.8 Observations

The reconstructed images for different noise types is listed in Appendix 3. It is seen that with Poisson noise, there is a resemblance between the input image and the output image. The plots of the mean square errors against the moment orders for different noise types are shown in the following diagrams.

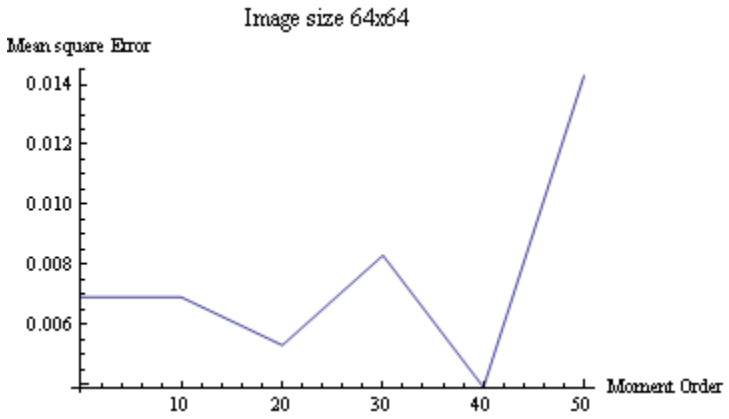

Figure 32: Plot of mean square errors against moment orders with Gaussian noise

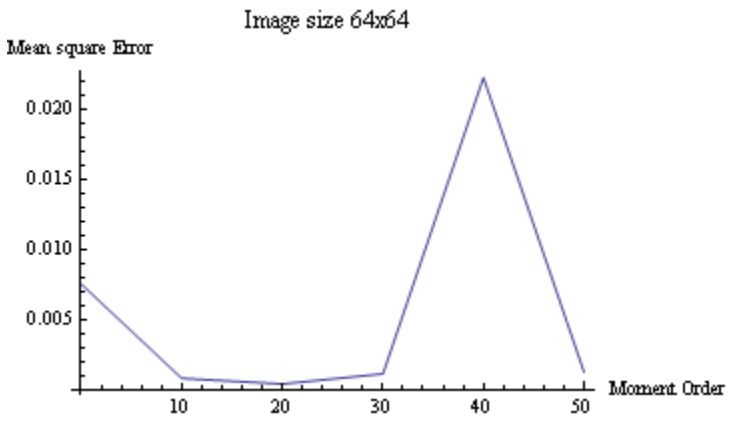

Figure 33: Plot of mean square errors against moment orders with Poisson noise

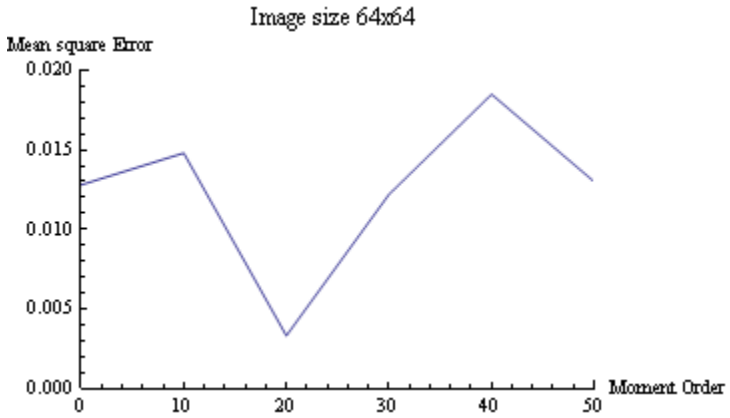

Figure 34: Plot of mean square errors against moment orders with Salt & Pepper noise

A comparison between Figure 16 and Figure 33 reveals that the optimum moment order for image reconstruction with Poisson noise is smaller than the optimum order for image reconstruction without Poisson noise (20 vs. 40).

4.9 (c) Part 3

Part 3 of the experiment is similar to Part 2 with the difference that the Gaussian noise has a fixed mean 0 and different variances from 0.1 to 1 (in increments of 0.1) and from 1 to 5 (in increments of 1). The density for the salt & pepper noise changes from 0.1 to 0.9 with an increment of 0.1.

4.10 Observations

The plots of the mean square errors against Gaussian variances and the plots of the mean square errors against the different density values of the salt & pepper noise are shown below. In both cases, the reconstructed images do not resemble the original image although the mean square errors are small (less than 1).

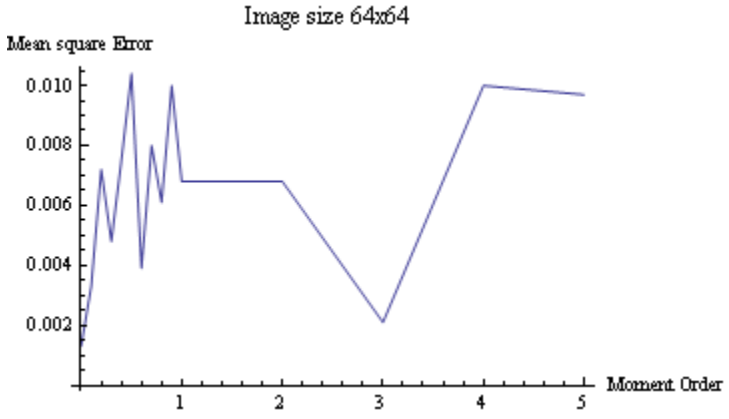

Figure 35: Plot of mean square errors against the variances of the Gaussian noise

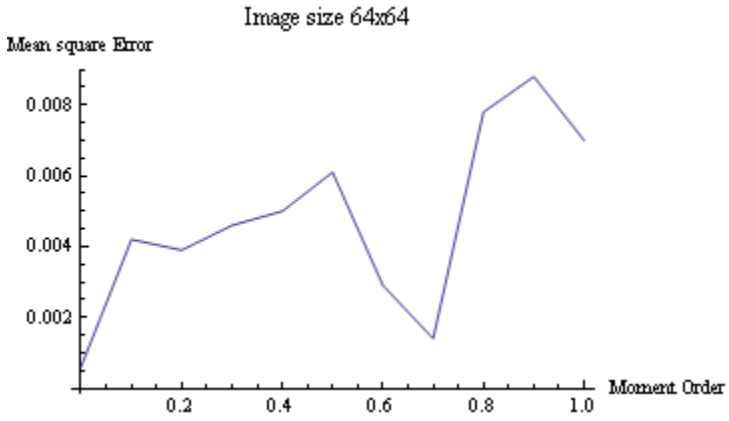

Figure 36: Plot of mean square errors against parameter values of the salt & pepper noise

4.11 Experiment 3

4.12 (a) Part 1

Part 1 of Experiment 3 deals with the verification of invariance of the derived sets of Zernike moments under reflection and rotation. Using the classical method, the input image (64x64) is once again decomposed into a set of Zernike moments. Using Matlab built-in operations the input image is also transformed and stored as a separate image. The Matlab operations are:

1. imrotate was used to rotate the image function at an angle of 45^o
2. fliplr was used to reflect the image function vertically,
3. flipud was used to reflect the image function horizontally, and
4. imrotate in conjunction with fliplr was used to reflect the image function about a line that makes 45^o angle to the horizontal line.

The classical method is then used to calculate the Zernike moments of the transformed images. Zernike moments of each of these four transformed images are then separately compared with the Zernike moments of the original image. In each case, the mean square error given in formula (2.29) is used as a measure of error. We therefore calculate the mean square for each Matlab operation and moment order p (p=0, 10, 20, 30, 40).

4.13 Observations for the first part of experiment 3

The mean square error for all combinations of Matlab operations and moment orders are listed in Table 3.

The mean square errors of the classical method for each Matlab operation is less than 1 when the moment order varies between 10 and 40. The small values of the mean square errors suggest that the classical method is invariant under rotation and reflection.

Matlab Operation	Moment Order					
	P=0	P=10	P=20	P=30	P=40	P=50
Operation (1)	1.3851	0.0172	8.5321e-004	0.1988	0.1885	5.6350
Operation (2)	1.3848	0.0252	0.0250	0.1697	0.1981	5.4578
Operation (3)	1.3844	0.0151	0.0052	0.2065	0.1846	5.5716
Operation (4)	1.3848	0.0252	0.0250	0.1697	0.1981	5.5716

Table 3: The mean square error values for the classical method

4.14 (b) Part 2

In Part 2 of Experiment 3 we examine the effect of the degree of rotation on the invariance of Zernike moments. Each of the Matlab operations (1) and (4) are used inconjunction with a change of rotation angle from 0 degrees to 90 degrees (in increments of 5 degrees). The mean square error for order p=40 is calculated for each change of rotation angle (we note that p=40 is the optimal order for the classical method when the image size 64x64 as shown in Figure 16). The plots of the mean square errors against the line angles for both operations are shown in the following two graphs.

4.15 Observations for second part of the experiment

The small values of the mean square errors in Figure 37 suggest that the classical method is invariant under the operation of rotation when the angle of rotation changes.

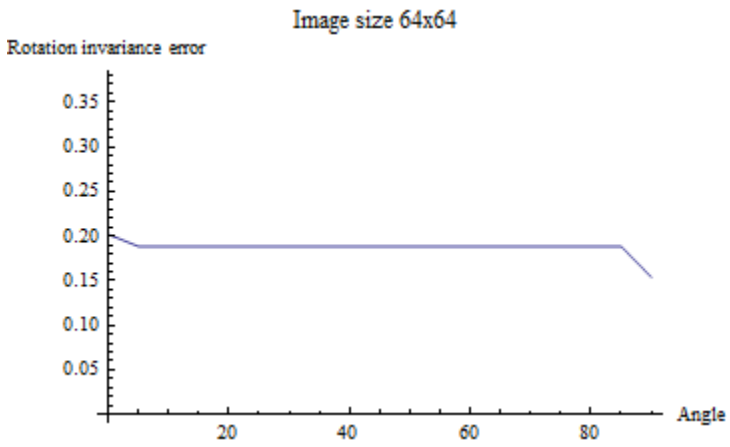

Figure 37: Plot of mean square errors against rotation angles Matlab operation (1)

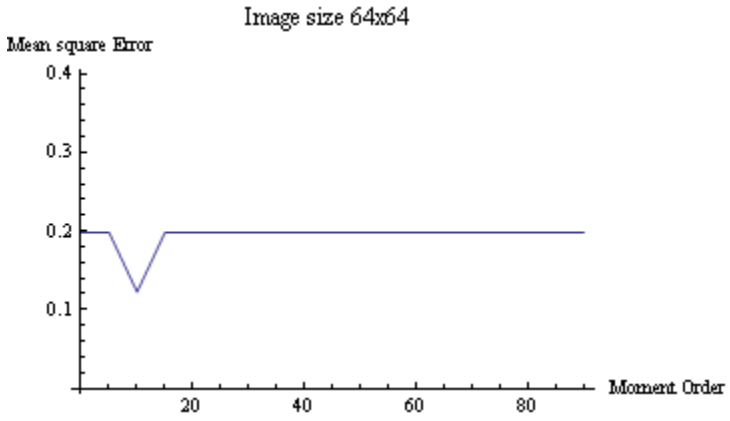

Figure 38: Plot of mean square error against line angle Matlab operation (4)

The classical method is invariant to reflection about any line when the angle of inclination of the line is greater than 20.

4.16 Possible explanation of the performance of the other procedures

If the images produced by another algorithm resemble the input image, the Zernike moments associated with the algorithm will be close to the Zernike moments produced by the classical method. We are therefore inclined to think that the difference between the radial polynomial for the classical method and the radial polynomial for the other methods will be close to zero for all moment orders. We therefore have computed the difference

between the radial polynomials of the classical method and the other methods in Appendix 4.

The variables are represented in tabular form where each row represents the moment order and each column represents the repetition value. The maximum moment order was set to n=30. From each table, we see that many values differ from 0 and in some cases by a huge amount. The radial polynomial function is included in the calculation of the Zernike moment. From this we can hypothesize that this may be a reason as to why the Zernike moments produced by these other methods are not the same as the Zernike moments produced by the classical method.

5 Chapter 5

5.1 Conclusions

It is found that among the eight computational procedures to compute Zernike moments for the purpose of image reconstruction, only the classical method is effective when the images have equal width and height. The classical method is effective to a certain extent for images with unequal height and width. The remaining seven algorithms are useless for the purpose of image reconstruction. For the classical method the optimum moment order (the moment order for which the reconstructed image resembles the input image) varies with the image size. The classical method is robust to Poisson noise in the sense that it can reconstruct an image when Poisson noise is added to the input image. However the classical method cannot reconstruct an image when either the Gaussian noise (with changing variance) or the salt & pepper noise (with changing densities) is added to the input image. The classical method is studied for its invariance under rotation and reflection about any line when the angle of that line to the horizontal axis varies from 0 to 90 degrees with an increment of 5 degrees. The mean square error for the classical method for any combination of the angle of the line of rotation and reflection about any line is always less than one, establishing the invariance of the classical method.

5.2 Contributions to thesis

The main contribution in this thesis is the implementation of eight computational procedures to derive radial Zernike polynomials and corresponding sets of Zernike moments of different orders for a continuous image function and then use estimates of the Zernike moments to reconstruct an image. From an exhaustive study of reconstruction of images of different sizes, it is found that the classical method can reconstruct a square input image properly. For rectangular images the classical method is partially effective in the sense that there is some resemblance between the constructed image and the input image although the reconstructed image has some aberrations. The remaining seven algorithms

are not useful for effective image reconstruction as these procedures leads to reconstructed images having no resemblance to the original image. Based on this finding the classical method has been further studied for image reconstruction under various noise types. It is found that the classical method is robust to Poisson noise in the sense that it can properly reconstruct an image in the presence of Poisson noise. However the classical method is not robust to Gaussian noise (with changing variances) and salt and pepper noise (with changing densities). The final contribution of the thesis deals with a demonstration of the invariance of the classical method under rotation about any line and under reflection about any line when the angle of that line varies.

5.3 Future Work

Some future possible applications of Zernike moments are listed below:

1. It is possible to use image reconstruction for coin identification and classification. This application could be in the form of a neural network system as described in section 2.8. This has not been done with Zernike moments [29][30]. This might be a useful project done in conjunction with the Royal Canadian Mint.

2. Similarly one can use Zernike moments in detection of cancer cells. A similar analysis as in 1) can be done. Zernike moments have been used to reconstruct holographic images representing leukemia cells but Zernike moments have not been used to reconstruct actual images showing cancer cells [27] [28]. CancerCare Manitoba may be interested in such an application.

3. Also Zernike moments can be used in identification of circular objects and objects that are nearly circular. The application of image reconstruction via Zernike moments should be tested on circular objects such as planets or objects that have circular parts like arcs to determine the accuracy and limitations to the application.

4. One can pursue further studies on the invariance of Zernike moments under the

presence of other noise types such as shot noise and speckle noise and under different image sizes. Other noise types such as additive image dependent noise and multiplicative noise and heavy-tailed noise maybe included in this study.

5. One can undertake a detailed study of the reconstructed error (as measured by the mean square error) of the classical method. Suppose one deals with the reconstruction of N similar objects of the same size using Zernike moments. For each of these N objects the mean square errors can be plotted against the moment orders. One can calculate the average of N mean square errors for a particular moment order for these N images. These mean square errors can be averaged to produce a grand mean square error for this moment order for the classical method. This mean square error is a better measure of the accuracy of the classical method for this particular moment order. This procedure may now be repeated for all moment orders. The grand mean square errors may now be plotted against the moment orders.

6. We have seen that the Zernike moments cannot properly reconstruct an image in the presence of a white Gaussian noise and salt & pepper noise. It would be interesting to use an appropriate filter to eliminate noise from the original noisy image and then use the Zernike moments of this filtered image to reconstruct an image using the appropriate procedure. However a comparison of the final reconstructed image with the original noisy image may not be possible in this situation.

6 References

[1] J. Flusser, T. Suk and B. Zitova. Moments and Moment Invariants in Pattern Recognition. Southern Gate, Chichester, John Wiley & Sons Ltd, 2009.

[2] M. Pawlak. Image Analysis by Moments: Reconstruction and Computational Aspects. Wroclaw, Wybrze'ze Wyspianskiego, Wroclaw Belmont, CA. Oficyna Wydawnicza Politechniki Wroclawskiej, Available: http://www.dbc.wroc.pl/Content/1432/pawlak.pdf, 2006.

[3] M. Hu. (1962, February). "Visual pattern recognition by moment invariants." IRE Transactions on Information Theory [Online], vol. 8 (2), pp. 179-187. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1057692&tag=1, [April 25, 2013].

[4] T.H. Reiss, (1991, August). "The revised fundamental theorem of moment invariants." IEEE Transactions on Pattern Analysis and Machine Intelligence, vol (13) pp.830-834, Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=85675, [April 25, 2013].

[5] Prokop R.J. and Reeves A.P., (1992, September). "A survey of moment-based techniques for unoccluded object representation and recognition." Graphical Models and Image Processing, vol 54 (5) pp. 438?460, Available: http://www.via.cornell.edu/ece547/text/survey.pdf, [April 25, 2013].

[6] Flusser J. and Suk T., (1993, January). "Pattern recognition by affine moment invariants." Pattern Recognition, vol 26(1), pp. 167-17426:167174, Available: http://www.sciencedirect.com/science/article/pii/003132039390098H, [April 25, 2013].

[7] M.R. Teague, (1980, August). "Image analysis via the general theory of moments". Journal of the Optical Society of America, Vol. 70 (8), pp. 920-930, Available: http://www.opticsinfobase.org/josa/abstract.cfm?uri=josa-70-8-920, [April 25, 2013].

[8] A. Ruggeri and S. Pajaro, (2002). "Automatic recognition of cell layers in corneal confocal microscopy images." Computer Methods and Programs in Biomedicine, pp. 25-35, Available: http://www.sciencedirect.com/science/article/pii/S0169260701001535, [April 25, 2013]

[9] J. Yu, S.S.R. Abidi, P. Artes, A. McIntyre and M. Heywood, (2005). "Diagnostic Support for Glaucoma Using Retinal Images: A Hybrid Image Analysis and Data Mining Approach", Connecting Medical Informatics and Bio-Informatics, pp. 187-192, Available: http://www.ncbi.nlm.nih.gov/pubmed/16160257, [April 25, 2013].

[10] D.R. Iskander, M.J. Collins and B. Davis. (2001, January). "Optimal Modeling of Corneal Surfaces with Zernike Polynomials." IEEE Transactions on Biomedical Engineering, [Online], vol. 48, pp.87-95, Available: http://www.ncbi.nlm.nih.gov/pubmed/11235595, [April 25, 2013].

[11] H. S. Kim and H. Lee, (2003, August). "Invariant Image Watermark Using Zernike Moments". IEEE Transactions On circuits and systems for video technology, vol. 13, pp. 766-775, Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1227606, [April 25, 2013].

[12] A. Khotanzad and Y.H. Hong, (1990, May). "Invariant Image Recognition by Zernike Moments". IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, pp. 489-497, Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=55109, [April 25, 2013].

[13] R.R. Bailey and M. Srinath. (1996, April). "Orthogonal Moment Features for Use With Parametric and Non-Parametric Classifiers", IEEE Transactions on pattern analysis and machine intelligence, vol. 18, pp. 389-399, Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=00491620, [April 25, 2013].

[14] J.D. S and M.S. Nixon, "Zernike Velocity Moments for Description and Recognition of Moving Shapes," British Machine Vision Conference 2008, pp. 705-714. Available: http://www.comp.leeds.ac.uk/bmvc2008/proceedings/2001/papers/28/accepted_28.pdf, [April 25, 2013].

[15] C. Singh, E. Walia and N. Mittal. (2012), "Fusion of Zernike moments and SIFT features for improved face recogntion", International Conference on Recent Advances and Future Trends in Information Technology, West Sussex, pp. 26-31. Available: http://www.ijcaonline.org/proceedings/irafit/number6/5890-1046, [April 25, 2013].

[16]"Bootstrapping" [Online]. Available: http://en.wikipedia.org/wiki/Bootstrapping, [April 25, 2013].

[17] "Grayscale" [Online]. Available: http://en.wikipedia.org/wiki/Grayscale, [April 25,2013].

[18] "Image compression" [Online]. Available: http://en.wikipedia.org/wiki/Image_compression, [April 25, 2013].

[19] S. X. Liao and M. Pawlak, (1998, December), "On the accuracy of Zernike moments for image analysis", pp. 1358-1364, Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=735809, [April 25, 2013].

[20] C. Chong, P. Raveendran and R. Mukundan. (2003, March). "A comparative analysis of algorithms for fast computation of Zernike moments," Pattern Recognition [Online], vol. 32 (3) pp. 731-742. Available: http://www.sciencedirect.com/science/article/pii/S0031320302000912, [April 25, 2013].

[21] C. Singh and E. Walia (2010, July). "Fast and numerically stable methods for the computation of Zernike moments." Pattern Recognition [Online], vol. 43 (7), pp. 2497-2506. Available: http://www.sciencedirect.com/science/article/pii/S003132031000083X, [April 25, 2013]

[22] C. Yee, R. Paramesran and F. Takeda (2004, February). "New computational methods for full and subset Zernike moments." Information Sciences [Online], vol. 159, (3-4), pp. 203-220. Available: http://dl.acm.org/citation.cfm?id=987197, [April 25, 2013].

[23]. C. Singh and E. Walia. (2011). "Algorithms for fast computation of Zernike moments and their numerical stability," Image and Vision Computing 29, pp.

251-259. Available: http://www.sciencedirect.com/science/article/pii/S0262885610001423, [April 25, 2013].

[24] G.A. Papakostas, Y.S. Boutalis, D.A. Karras and B.G. Mertzios (2007, July), "A new class of Zernike moments for computer vision applications." Information Sciences [Online], vol. 177 (13), pp. 2802-2819. Available: http://www.sciencedirect.com/science/article/pii/S0020025507000394, [April 25, 2013].

[25] C. Boncelet. Image Noise Models in Handbook of Image and Video Processing (Second Edition) Edited by A. Bovik. Burlington, MA, Elsevier Academic Press Inc. , 2005.

[26]"Lenna", http://en.wikipedia.org/wiki/Lenna, [April 25, 2013].

[27] R. Mukundan, (2004). "A new class of rotational invariants using discrete orthogonal moments," in 6th IASTED Conference on Signal and Image Processing [Online], Honolulu, USA, vol. (6), pp. 80-84. Available: http://ir.canterbury.ac.nz/bitstream/10092/843/1/12594002_IASTED_SIP04.pdf, [April 25, 2013].

[28] Z. Chen and S. Sun (2010, January). "A Zernike moment phase-based descriptor for local image representation and matching," IEEE Transactions on image processing, vol. 19(1), pp. 205-219. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=05256258, [April 25, 2013].

[29] W. Liyun, L. Hefei, Z. Fuhao, L. Zhengding and W. Zhendi (2009, July). "Spermatogonium image recognition using Zernike moments," Computer Methods and Programs in Biomedicine [Online], vol 96 (1), Available: http://www.sciencedirect.com.proxy2.lib.umanitoba.ca/science/article/pii/S0169260709000352 [April 25, 2013]

[30] M.R. Asadi, A. Vahedi and H. Amindavar, "Leukemia cell recognition with Zernike moments of holographic images," in Signal Processing Symposium, 2006. NORSIG 2006. Proceedings of the 7th Nordic [Online], 2006, vol. 7, pp. 214-217. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?

arnumber=4052221, [April 25, 2013].

Appendix 1

For experiment 1, the input and output images are shown below for each combination of image size, computational procedure and moment order.

Image size: 10x10

Computational procedure: Classical

N=0

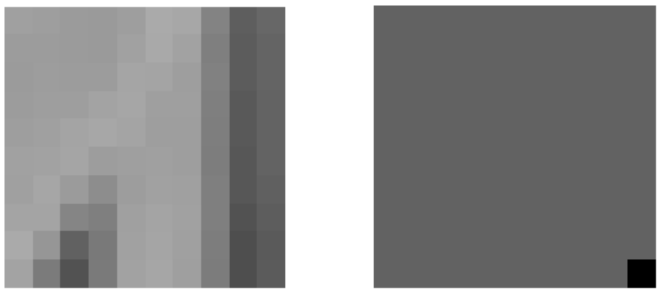

Figure 39: Image reconstruction results for Direct method: Image size: 10x10, Moment order: N=0

N=10

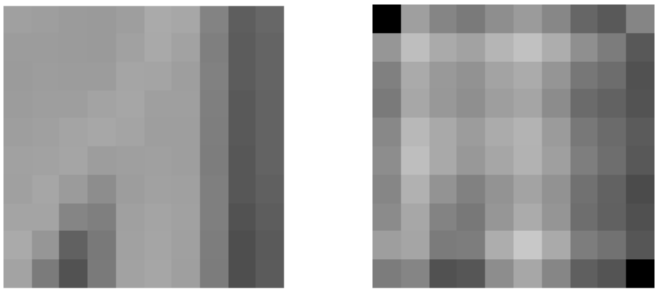

Figure 40: Image reconstruction results for Direct method: Image size: 10x10, Moment order: N=10

N=20

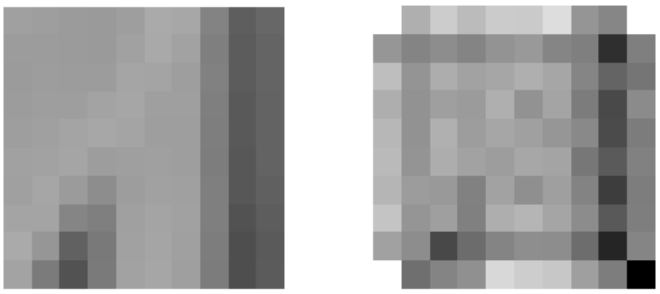

Figure 41: Image reconstruction results for Direct method: Image size: 10x10, Moment order: N=20

N=30

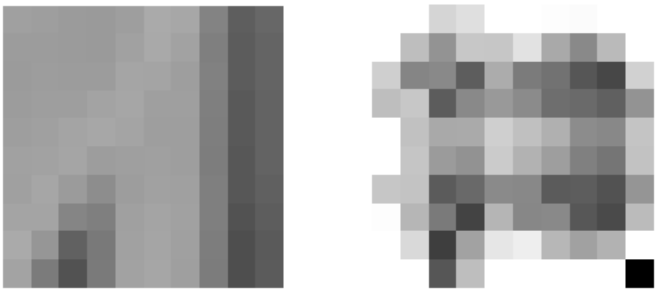

Figure 42: Image reconstruction results for Direct method: Image size: 10x10, Moment order: N=30

N=40

Figure 43: Image reconstruction results for Direct method: Image size: 10x10, Moment order: N=40

N=50

Figure 44: Image reconstruction results for Direct method: Image size: 10x10, Moment order: N=50

Computational procedure: Belkasim

N=0

Figure 45: Image reconstruction results for Belkasim method: Image size: 10x10, Moment order: N=0

N=10

Figure 46: Image reconstruction results for Belkasim method: Image size: 10x10, Moment order: N=10

N=20

Figure 47: Image reconstruction results for Belkasim method: Image size: 10x10, Moment order: N=20

N=30

Figure 48: Image reconstruction results for Belkasim method : Image size: 10x10, Moment order: N=30

N=40

Figure 49: Image reconstruction results for Belkasim method : Image size: 10x10, Moment order: N=40

N=50

Figure 50: Image reconstruction results for Belkasim method : Image size: 10x10, Moment order: N=50

Computational procedure: Factorial Free

N=0

Figure 51: Image reconstruction results for Factorial Free method : Image size: 10x10, Moment order: N=0

N=10

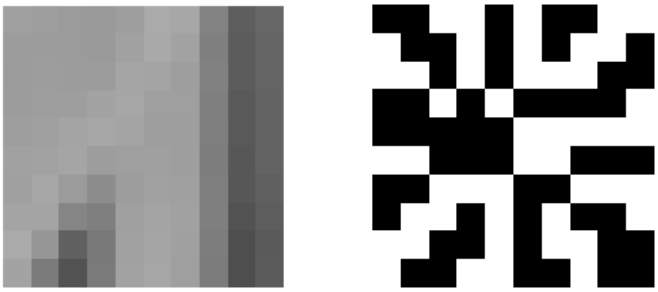

Figure 52: Image reconstruction results for Factorial Free method : Image size: 10x10, Moment order: N=10

N=20

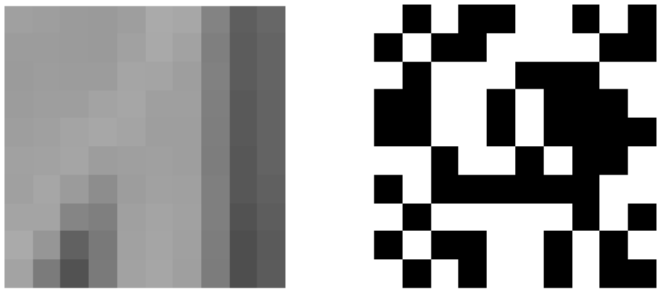

Figure 53: Image reconstruction results for Factorial Free method : Image size: 10x10, Moment order: N=20

N=30

Figure 54: Image reconstruction results for Factorial Free method : Image size: 10x10, Moment order: N=30

N=40

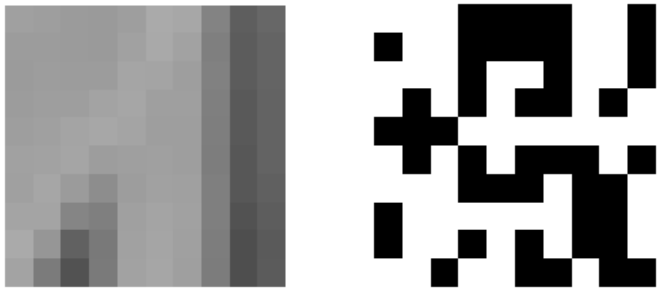

Figure 55: Image reconstruction results for Factorial Free method : Image size: 10x10, Moment order: N=40

N=50

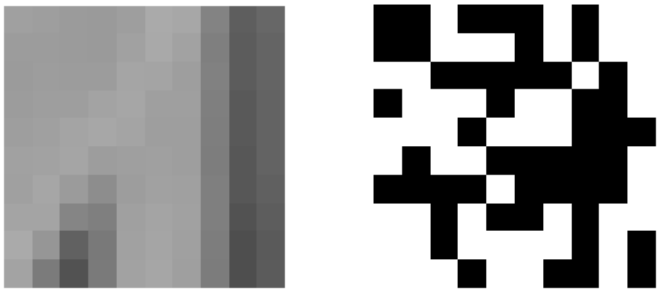

Figure 56: Image reconstruction results for Factorial Free method : Image size: 10x10, Moment order: N=50

Computational procedure: Hybrid

N=0

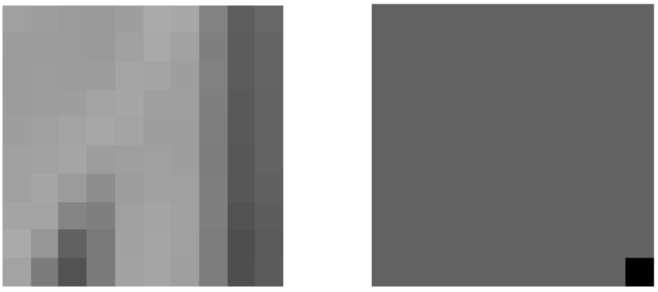

Figure 57: Image reconstruction results for Hybrid method : Image size: 10x10, Moment order: N=0

N=10

Figure 58: Image reconstruction results for Hybrid method : Image size: 10x10, Moment order: N=10

N=20

Figure 59: Image reconstruction results for Hybrid method : Image size: 10x10, Moment order: N=20

N=30

Figure 60: Image reconstruction results for Hybrid method : Image size: 10x10, Moment order: N=30

N=40

Figure 61: Image reconstruction results for Hybrid method : Image size: 10x10, Moment order: N=40

N=50

Figure 62: Image reconstruction results for Hybrid method : Image size: 10x10, Moment order: N=50

Computational procedure: Kintner

N=0

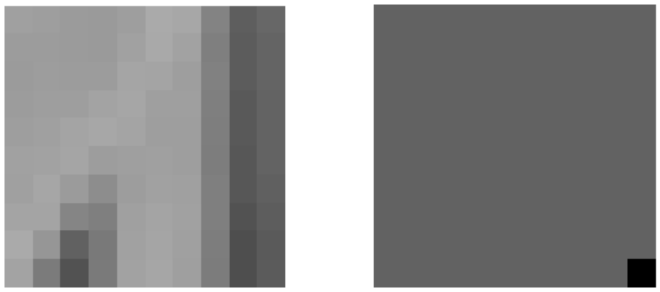

Figure 63: Image reconstruction results for Kintner method : Image size: 10x10, Moment order: N=0

N=10

Figure 64: Image reconstruction results for Kintner method : Image size: 10x10, Moment order: N=10

N=20

Figure 65: Image reconstruction results for Kintner method : Image size: 10x10, Moment order: N=20

N=30

Figure 66: Image reconstruction results for Kintner method : Image size: 10x10, Moment order: N=30

N=40

Figure 67: Image reconstruction results for Kintner method : Image size: 10x10, Moment order: N=40

N=50

Figure 68: Image reconstruction results for Kintner method : Image size: 10x10, Moment order: N=50

Computational procedure: Modified Prata

N=0

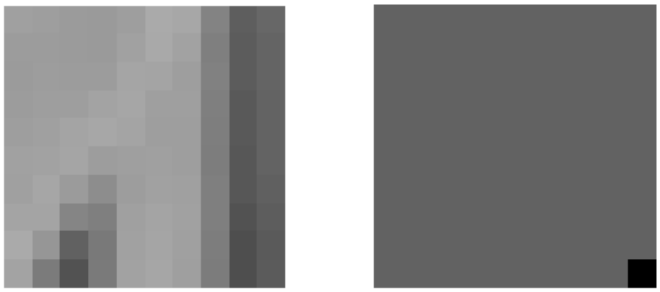

Figure 69: Image reconstruction results for Modified Prata method : Image size: 10x10, Moment order: N=0

N=10

Figure 70: Image reconstruction results for Modified Prata method : Image size: 10x10, Moment order: N=10

N=20

Figure 71: Image reconstruction results for Modified Prata method : Image size: 10x10, Moment order: N=20

N=30

Figure 72: Image reconstruction results for Modified Prata method : Image size: 10x10, Moment order: N=30

N=40

Figure 73: Image reconstruction results for Modified Prata method : Image size: 10x10, Moment order: N=40

N=50

Figure 74: Image reconstruction results for Modified Prata method : Image size: 10x10, Moment order: N=50

Computational procedure: Prata

N=0

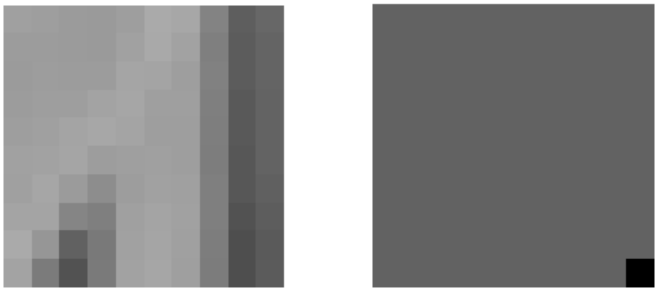

Figure 75: Image reconstruction results for Prata method : Image size: 10x10, Moment order: N=0

N=10

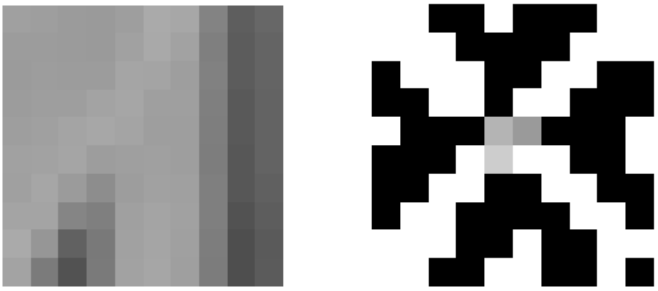

Figure 76: Image reconstruction results for Prata method : Image size: 10x10, Moment order: N=10

N=20

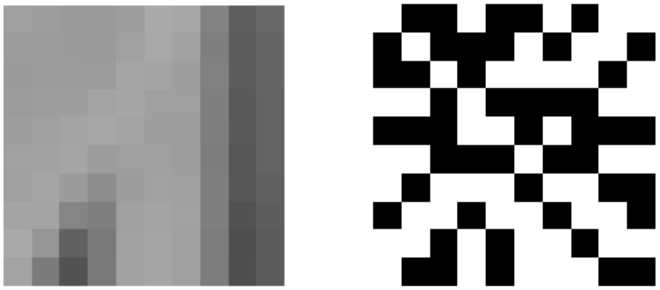

Figure 77: Image reconstruction results for Prata method : Image size: 10x10, Moment order: N=20

N=30

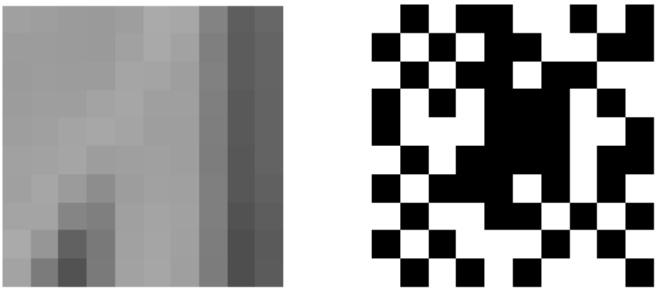

Figure 78: Image reconstruction results for Prata method : Image size: 10x10, Moment order: N=30

N=40

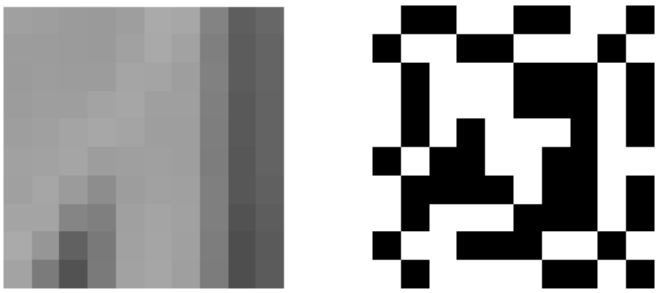

Figure 79: Image reconstruction results for Prata method : Image size: 10x10, Moment order: N=40

N=50

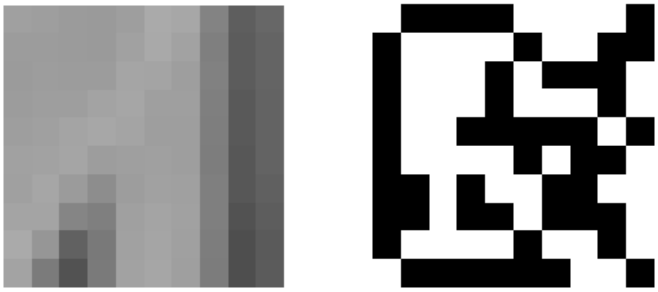

Figure 80: Image reconstruction results for Prata method : Image size: 10x10, Moment order: N=50

Computational procedure: Q-θ

N=0

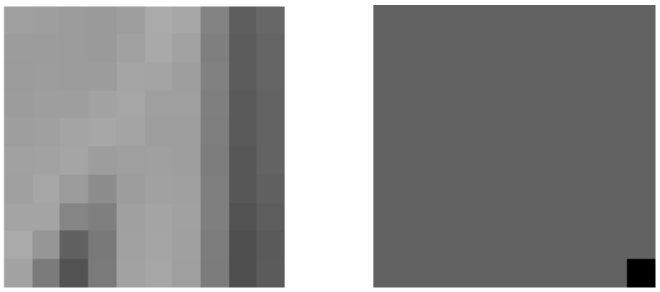

Figure 81: Image reconstruction results for Q-theta method : Image size: 10x10, Moment order: N=0

N=10

Figure 82: Image reconstruction results for Q-theta method : Image size: 10x10, Moment order: N=10

N=20

Figure 83: Image reconstruction results for Q-theta method : Image size: 10x10, Moment order: N=20

N=30

Figure 84: Image reconstruction results for Q-theta method : Image size: 10x10, Moment order: N=30

N=40

Figure 85: Image reconstruction results for Q-theta method : Image size: 10x10, Moment order: N=40

N=50

Figure 86: Image reconstruction results for Q-theta method : Image size: 10x10, Moment order: N=50

Image size: 32x32

Computational procedure: Classical

N=0

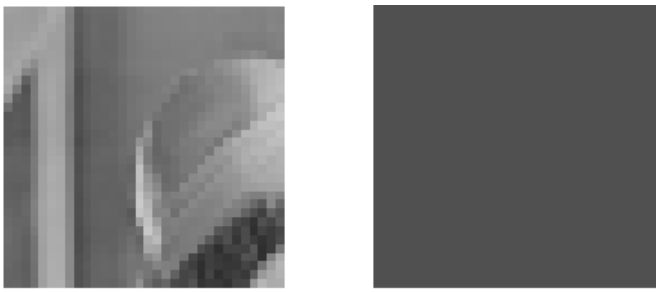

Figure 87: Image reconstruction results for Direct method: Image size: 32x32, Moment order: N=0

N=10

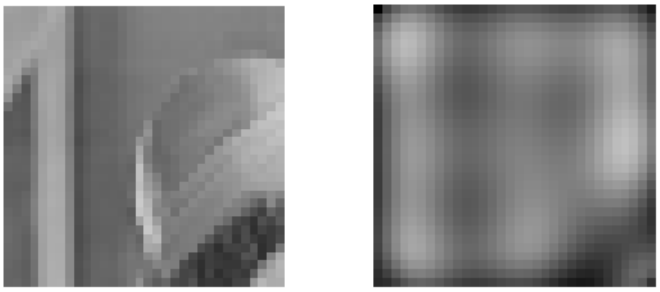

Figure 88: Image reconstruction results for Direct method: Image size: 32x32, Moment order: N=10

N=20

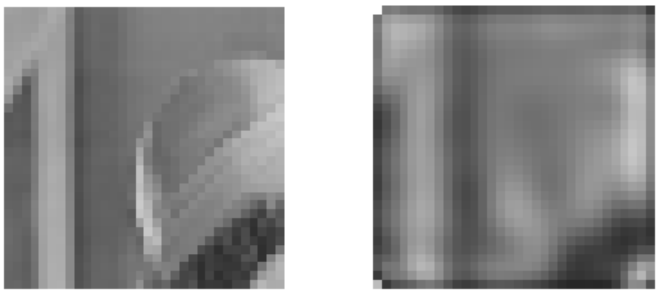

Figure 89: Image reconstruction results for Direct method: Image size: 32x32, Moment order: N=20

N=30

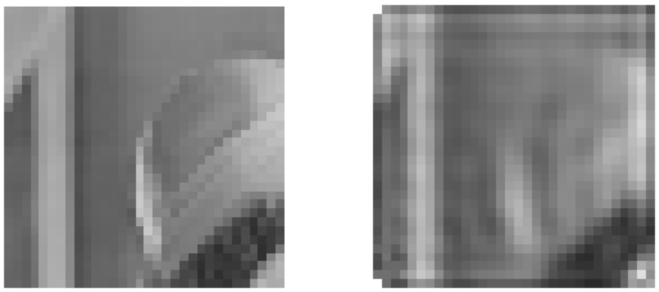

Figure 90: Image reconstruction results for Direct method: Image size: 32x32, Moment order: N=30

N=40

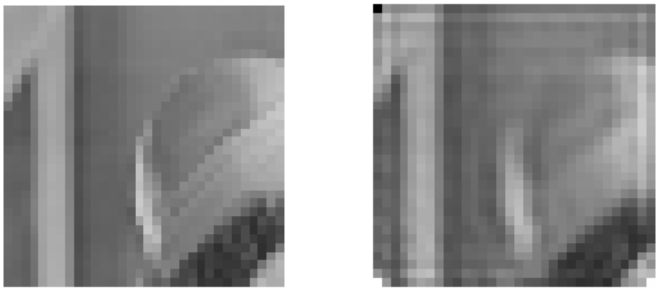

Figure 91: Image reconstruction results for Direct method: Image size: 32x32, Moment order: N=40

N=50

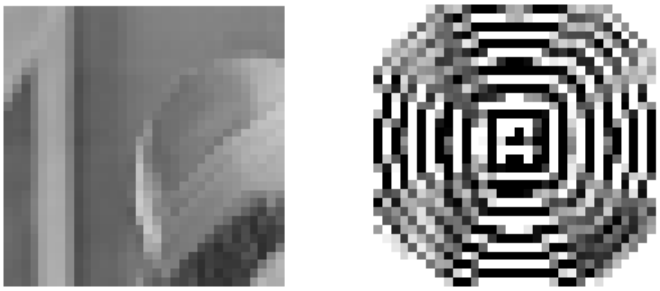

Figure 92: Image reconstruction results for Direct method: Image size: 32x32, Moment order: N=50

Computational procedure: Belkasim

N=0

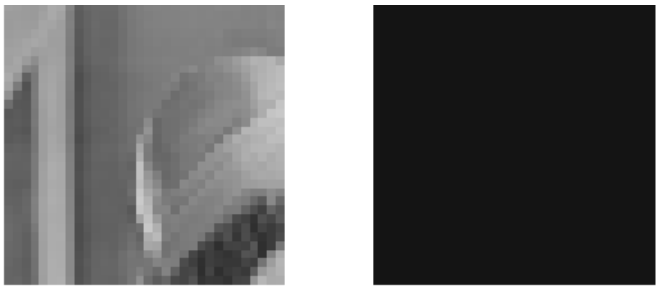

Figure 93: Image reconstruction results for Belkasim method: Image size: 32x32, Moment order: N=0

N=10

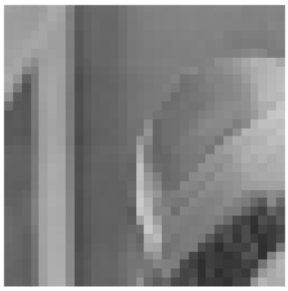

Figure 94: Image reconstruction results for Belkasim method: Image size: 32x32, Moment order: N=10

N=20

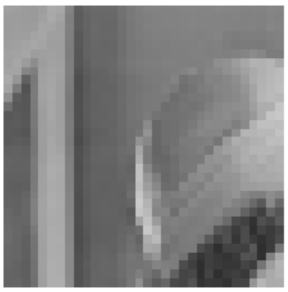

Figure 95: Image reconstruction results for Belkasim method: Image size: 32x32, Moment order: N=20

N=30

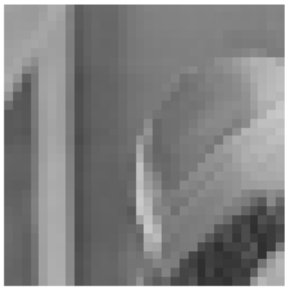

Figure 96: Image reconstruction results for Belkasim method : Image size: 32x32, Moment order: N=30

N=40

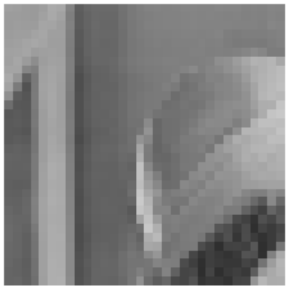

Figure 97: Image reconstruction results for Belkasim method : Image size: 32x32, Moment order: N=40

N=50

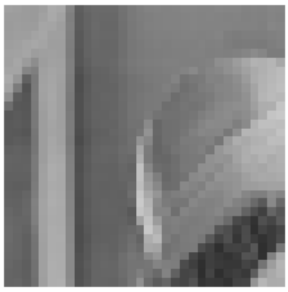

Figure 98: Image reconstruction results for Belkasim method : Image size: 32x32, Moment order: N=50

Computational procedure: Factorial Free

N=0

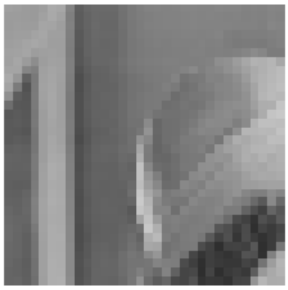

Figure 99: Image reconstruction results for Factorial Free method : Image size: 32x32, Moment order: N=0

N=10

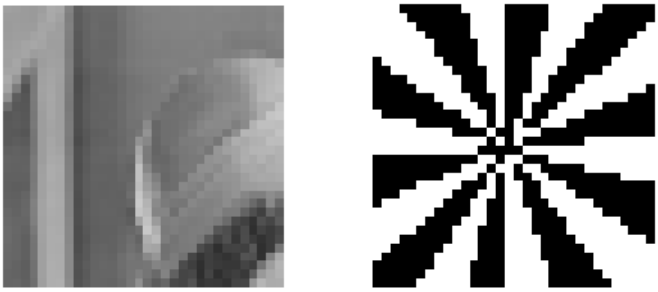

Figure 100: Image reconstruction results for Factorial Free method : Image size: 32x32, Moment order: N=10

N=20

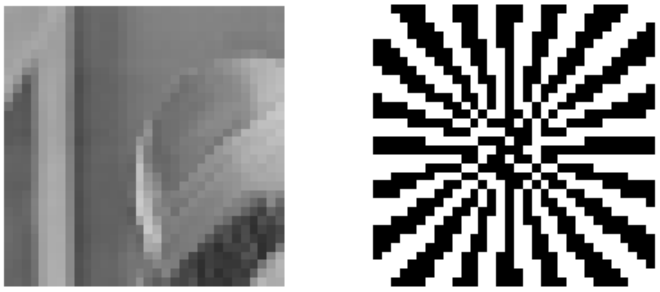

Figure 101: Image reconstruction results for Factorial Free method : Image size: 32x32, Moment order: N=20

N=30

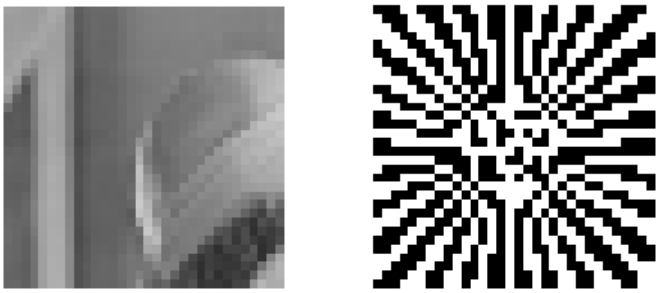

Figure 102: Image reconstruction results for Factorial Free method : Image size: 32x32, Moment order: N=30

N=40

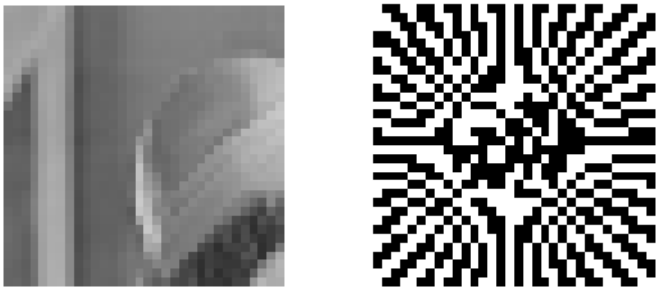

Figure 103: Image reconstruction results for Factorial Free method : Image size: 32x32, Moment order: N=40

N=50

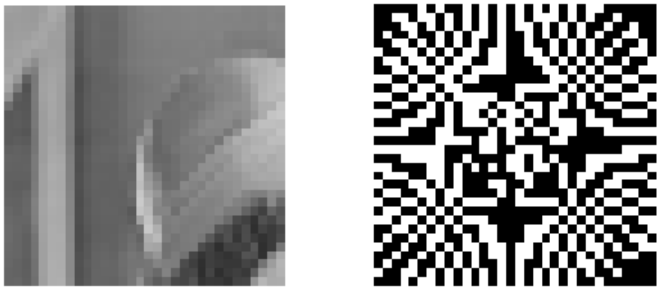

Figure 104: Image reconstruction results for Factorial Free method : Image size: 32x32, Moment order: N=50

Computational procedure: Hybrid

N=0

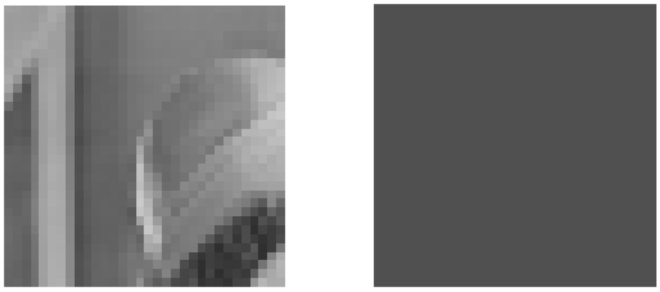

Figure 105: Image reconstruction results for Hybrid method : Image size: 32x32, Moment order: N=0

N=10

Figure 106: Image reconstruction results for Hybrid method : Image size: 32x32, Moment order: N=10

N=20

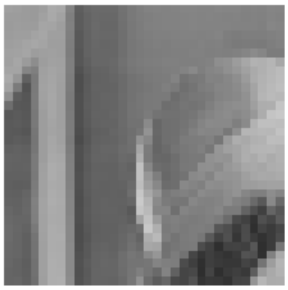

Figure 107: Image reconstruction results for Hybrid method : Image size: 32x32, Moment order: N=20

N=30

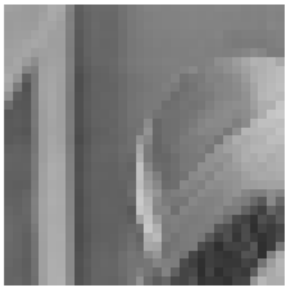

Figure 108: Image reconstruction results for Hybrid method : Image size: 32x32, Moment order: N=30

N=40

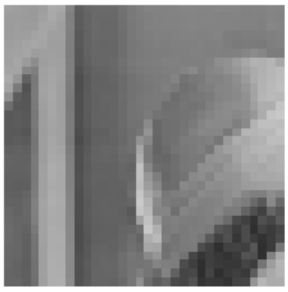

Figure 109: Image reconstruction results for Hybrid method : Image size: 32x32, Moment order: N=40

N=50

Figure 110: Image reconstruction results for Hybrid method : Image size: 32x32, Moment order: N=50

Computational procedure: Kintner

N=0

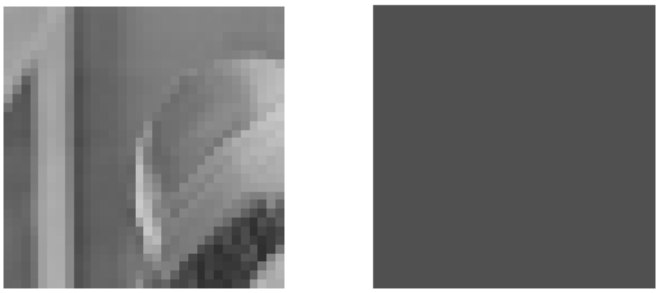

Figure 111: Image reconstruction results for Kintner method : Image size: 32x32, Moment order: N=0

N=10

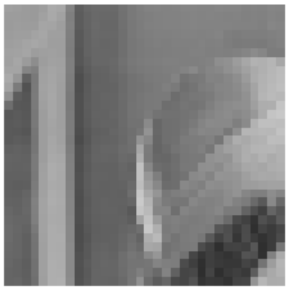

Figure 112: Image reconstruction results for Kintner method : Image size: 32x32, Moment order: N=10

N=20

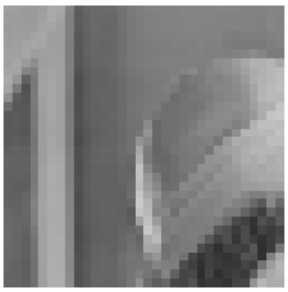

Figure 113: Image reconstruction results for Kintner method : Image size: 32x32, Moment order: N=20

N=30

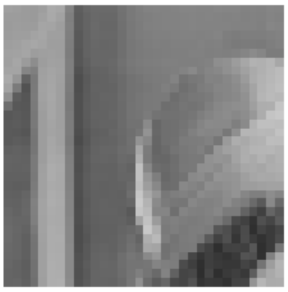

Figure 114: Image reconstruction results for Kintner method : Image size: 32x32, Moment order: N=30

N=40

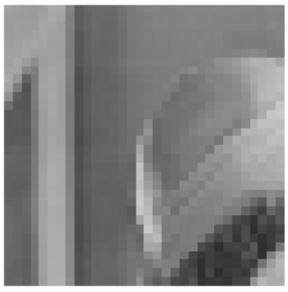

Figure 115: Image reconstruction results for Kintner method : Image size: 32x32, Moment order: N=40

N=50

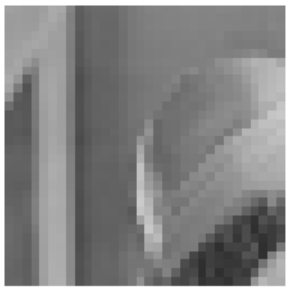

Figure 116: Image reconstruction results for Kintner method : Image size: 32x32, Moment order: N=50

Computational procedure: Modified Prata

N=0

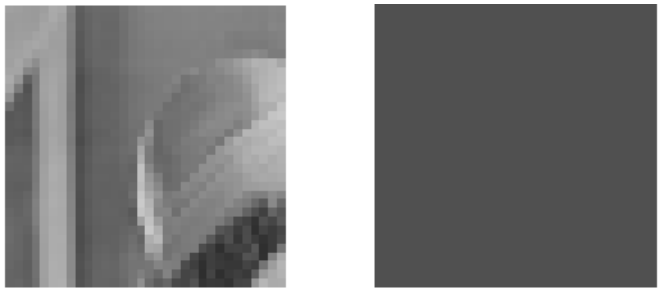

Figure 117: Image reconstruction results for Modified Prata method : Image size: 32x32, Moment order: N=0

N=10

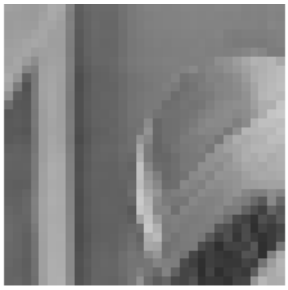

Figure 118: Image reconstruction results for Modified Prata method : Image size: 32x32, Moment order: N=10

N=20

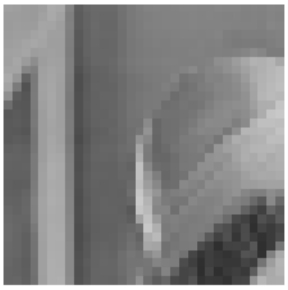

Figure 119: Image reconstruction results for Modified Prata method : Image size: 32x32, Moment order: N=20

N=30

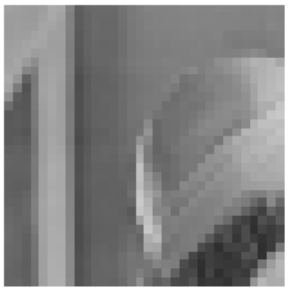

Figure 120: Image reconstruction results for Modified Prata method : Image size: 32x32, Moment order: N=30

N=40

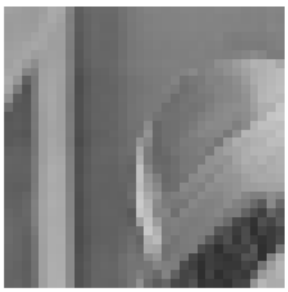

Figure 121: Image reconstruction results for Modified Prata method : Image size: 32x32, Moment order: N=40

N=50

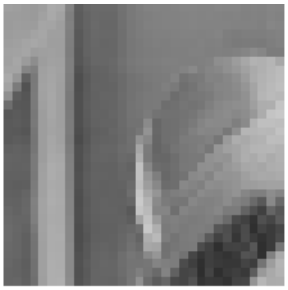

Figure 122: Image reconstruction results for Modified Prata method : Image size: 32x32, Moment order: N=50

Computational procedure: Prata

N=0

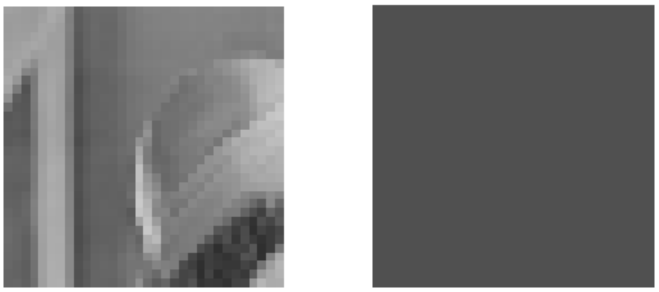

Figure 123: Image reconstruction results for Prata method : Image size: 32x32, Moment order: N=0

N=10

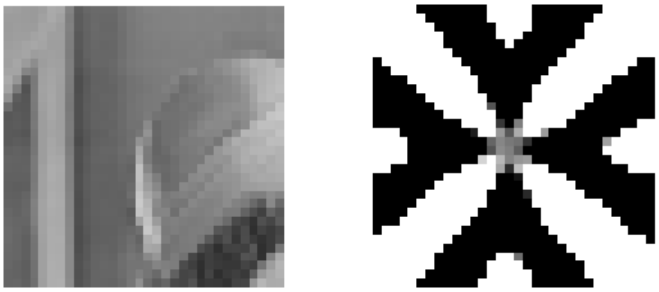

Figure 124: Image reconstruction results for Prata method : Image size: 32x32, Moment order: N=10

N=20

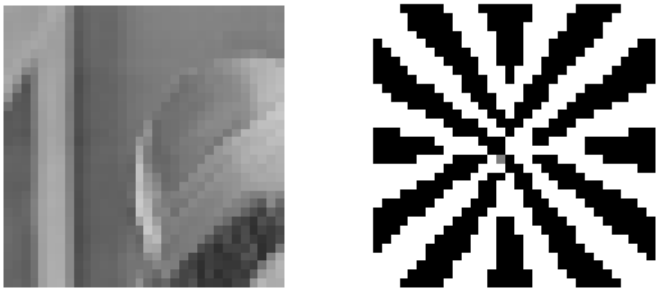

Figure 125: Image reconstruction results for Prata method : Image size: 32x32, Moment order: N=20

N=30

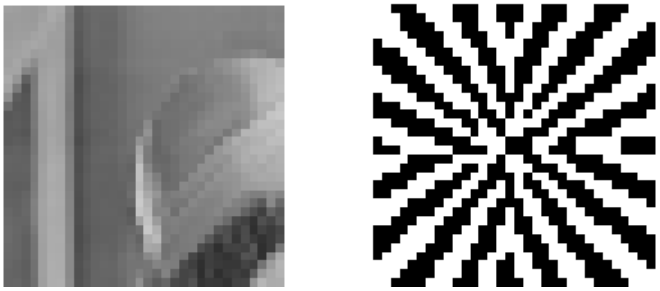

Figure 126: Image reconstruction results for Prata method : Image size: 32x32, Moment order: N=30

N=40

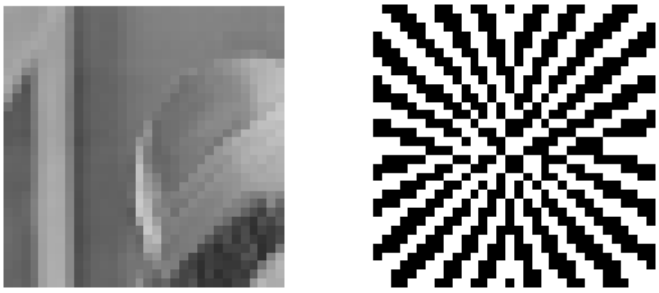

Figure 127: Image reconstruction results for Prata method : Image size: 32x32, Moment order: N=40

N=50

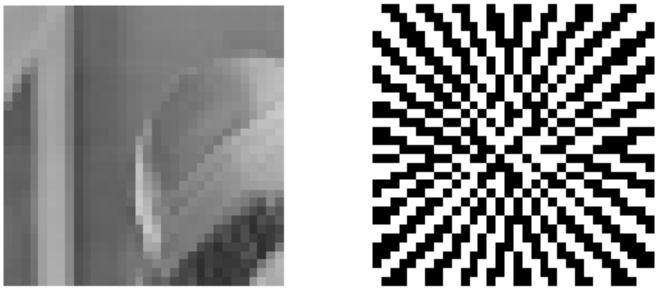

Figure 128: Image reconstruction results for Prata method : Image size: 32x32, Moment order: N=50

Computational procedure: Q-θ

N=0

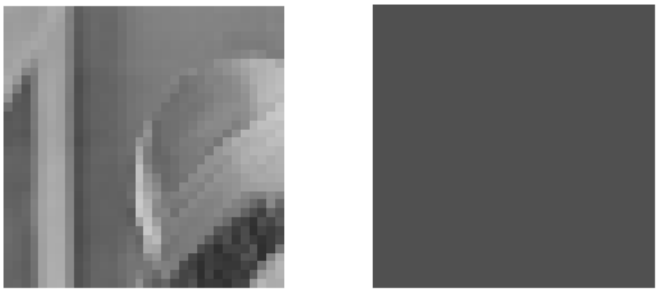

Figure 129: Image reconstruction results for Q-theta method : Image size: 32x32, Moment order: N=0

N=10

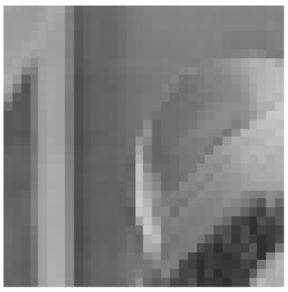

Figure 130: Image reconstruction results for Q-theta method : Image size: 32x32, Moment order: N=10

N=20

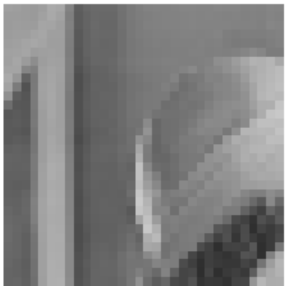

Figure 131: Image reconstruction results for Q-theta method : Image size: 32x32, Moment order: N=20

N=30

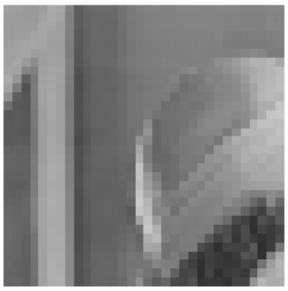

Figure 132: Image reconstruction results for Q-theta method : Image size: 32x32, Moment order: N=30

N=40

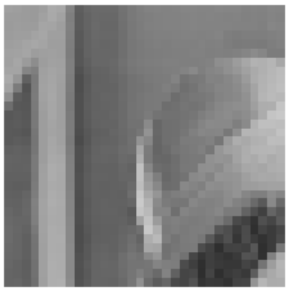

Figure 133: Image reconstruction results for Q-theta method : Image size: 32x32, Moment order: N=40

N=50

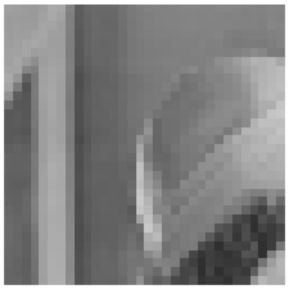

Figure 134: Image reconstruction results for Q-theta method : Image size: 32x32, Moment order: N=50

Image size: 64x64

Computational procedure: Classical

N=0

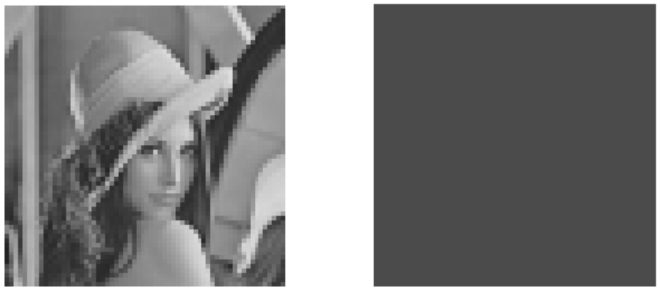

Figure 135: Image reconstruction results for Direct method: Image size: 64x64, Moment order: N=0

N=10

Figure 136: Image reconstruction results for Direct method: Image size: 64x64, Moment order: N=10

N=20

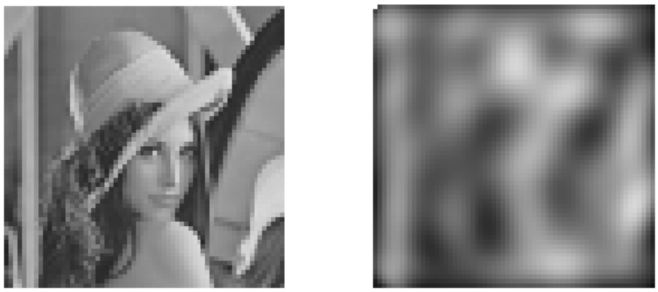

Figure 137: Image reconstruction results for Direct method: Image size: 64x64, Moment order: N=20

N=30

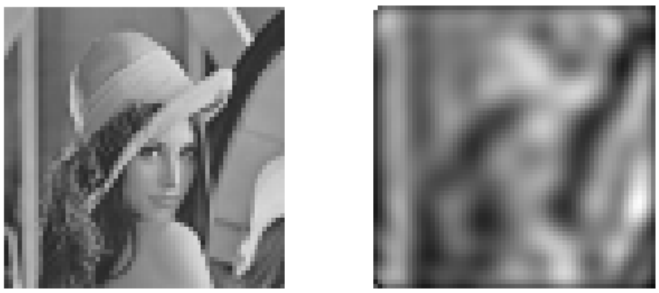

Figure 138: Image reconstruction results for Direct method: Image size: 64x64, Moment order: N=30

N=40

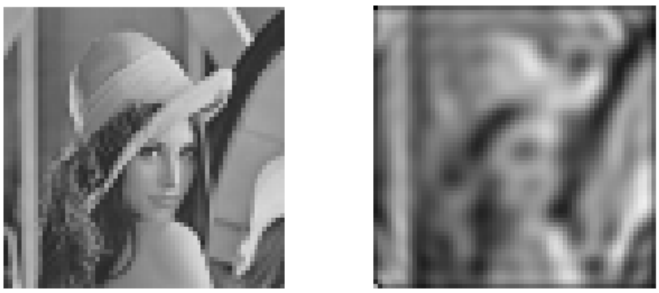

Figure 139: Image reconstruction results for Direct method: Image size: 64x64, Moment order: N=40

N=50

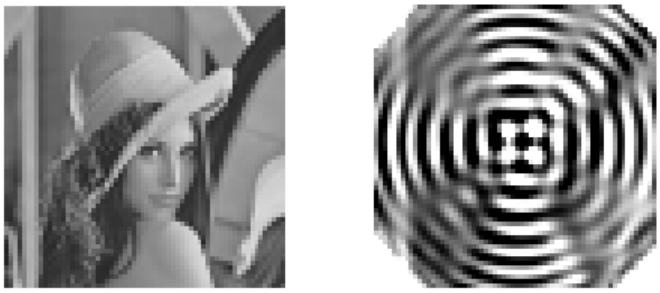

Figure 140: Image reconstruction results for Direct method: Image size: 64x64, Moment order: N=50

Computational procedure: Belkasim

N=0

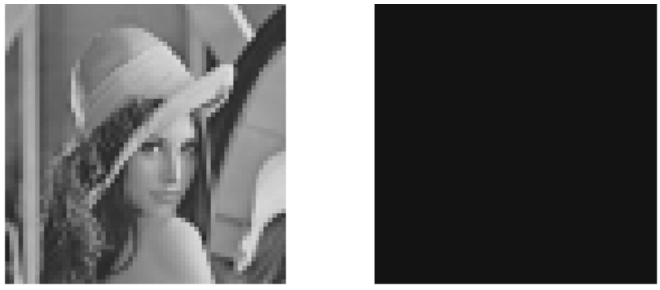

Figure 141: Image reconstruction results for Belkasim method: Image size: 64x64, Moment order: N=0

N=10

Figure 142: Image reconstruction results for Belkasim method: Image size: 64x64, Moment order: N=10

N=20

Figure 143: Image reconstruction results for Belkasim method: Image size: 64x64, Moment order: N=20

N=30

Figure 144: Image reconstruction results for Belkasim method : Image size: 64x64, Moment order: N=30

N=40

Figure 145: Image reconstruction results for Belkasim method : Image size: 64x64, Moment order: N=40

N=50

Figure 146: Image reconstruction results for Belkasim method : Image size: 64x64, Moment order: N=50

Computational procedure: Factorial Free

N=0

Figure 147: Image reconstruction results for Factorial Free method : Image size: 64x64, Moment order: N=0

N=10

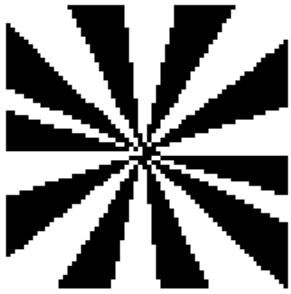

Figure 148: Image reconstruction results for Factorial Free method : Image size: 64x64, Moment order: N=10

N=20

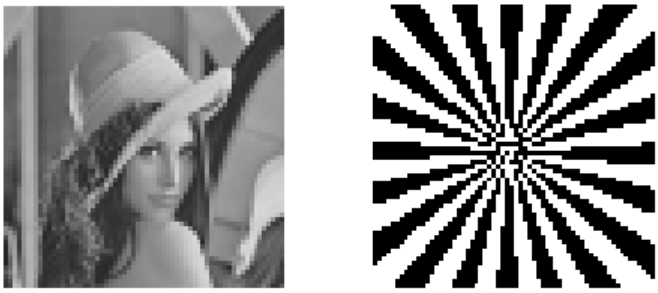

Figure 149: Image reconstruction results for Factorial Free method : Image size: 64x64, Moment order: N=20

N=30

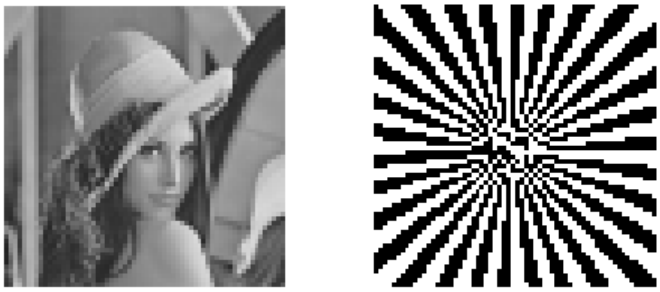

Figure 150: Image reconstruction results for Factorial Free method : Image size: 64x64, Moment order: N=30

N=40

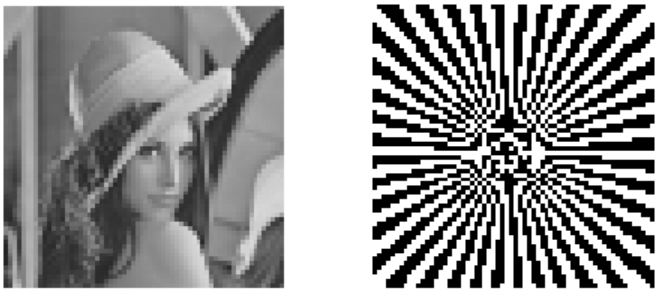

Figure 151: Image reconstruction results for Factorial Free method : Image size: 64x64, Moment order: N=40

N=50

Figure 152: Image reconstruction results for Factorial Free method : Image size: 64x64, Moment order: N=50

Computational procedure: Hybrid

N=0

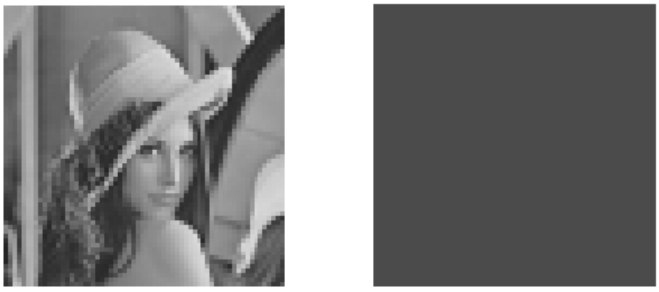

Figure 153: Image reconstruction results for Hybrid method : Image size: 64x64, Moment order: N=0

N=10

Figure 154: Image reconstruction results for Hybrid method : Image size: 64x64, Moment order: N=10

N=20

Figure 155: Image reconstruction results for Hybrid method : Image size: 64x64, Moment order: N=20

N=30

Figure 156: Image reconstruction results for Hybrid method : Image size: 64x64, Moment order: N=30

N=40

Figure 157: Image reconstruction results for Hybrid method : Image size: 64x64, Moment order: N=40

N=50

Figure 158: Image reconstruction results for Hybrid method : Image size: 64x64, Moment order: N=50

Computational procedure: Kintner

N=0


Figure 159: Image reconstruction results for Kintner method : Image size: 64x64, Moment order: N=0

N=10

Figure 160: Image reconstruction results for Kintner method : Image size: 64x64, Moment order: N=10

N=20

Figure 161: Image reconstruction results for Kintner method : Image size: 64x64, Moment order: N=20

N=30

Figure 162: Image reconstruction results for Kintner method : Image size: 64x64, Moment order: N=30

N=40

Figure 163: Image reconstruction results for Kintner method : Image size: 64x64, Moment order: N=40

N=50

Figure 164: Image reconstruction results for Kintner method : Image size: 64x64, Moment order: N=50

Computational procedure: Modified Prata

N=0

Figure 165: Image reconstruction results for Modified Prata method : Image size: 64x64, Moment order: N=0

N=10

Figure 166: Image reconstruction results for Modified Prata method : Image size: 64x64, Moment order: N=10

N=20

Figure 167: Image reconstruction results for Modified Prata method : Image size: 64x64, Moment order: N=20

N=30

Figure 168: Image reconstruction results for Modified Prata method : Image size: 64x64, Moment order: N=30

N=40

Figure 169: Image reconstruction results for Modified Prata method : Image size: 64x64, Moment order: N=40

N=50

Figure 170: Image reconstruction results for Modified Prata method : Image size: 64x64, Moment order: N=50

Computational procedure: Prata

N=0

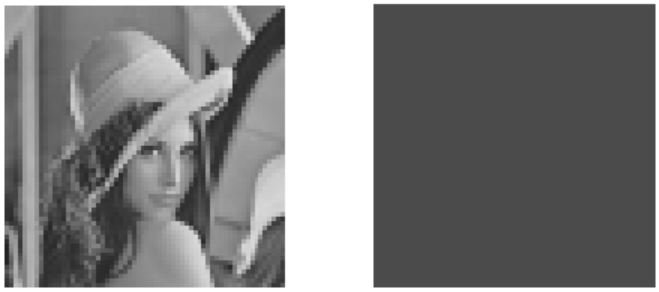

Figure 171: Image reconstruction results for Prata method : Image size: 64x64, Moment order: N=0

N=10

Figure 172: Image reconstruction results for Prata method : Image size: 64x64, Moment order: N=10

N=20

Figure 173: Image reconstruction results for Prata method : Image size: 64x64, Moment order: N=20

N=30

Figure 174: Image reconstruction results for Prata method : Image size: 64x64, Moment order: N=30

N=40

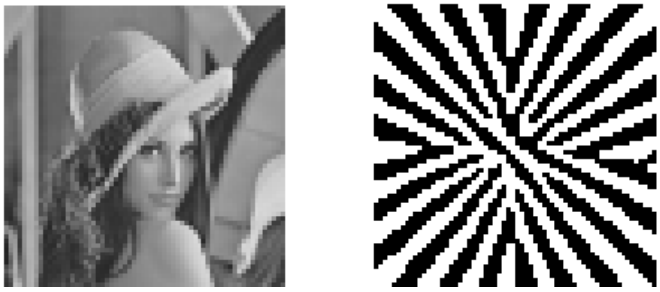

Figure 175: Image reconstruction results for Prata method : Image size: 64x64, Moment order: N=40

N=50

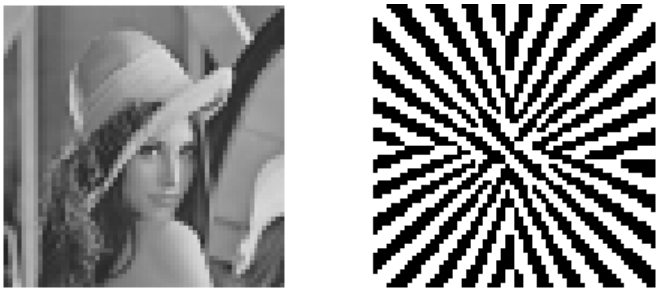

Figure 176: Image reconstruction results for Prata method : Image size: 64x64, Moment order: N=50

Computational procedure: Q-θ

N=0

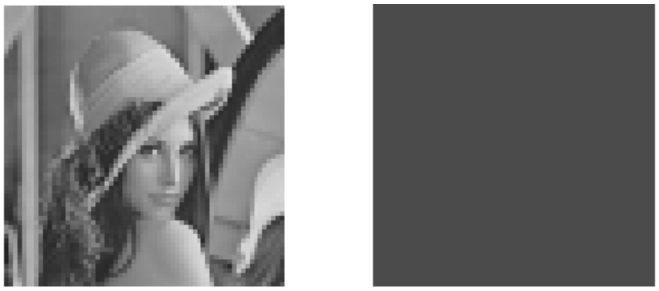

Figure 177: Image reconstruction results for Q-theta method : Image size: 64x64, Moment order: N=0

N=10

Figure 178: Image reconstruction results for Q-theta method : Image size: 64x64, Moment order: N=10

N=20

Figure 179: Image reconstruction results for Q-theta method : Image size: 64x64, Moment order: N=20

N=30

Figure 180: Image reconstruction results for Q-theta method : Image size: 64x64, Moment order: N=30

N=40

Figure 181: Image reconstruction results for Q-theta method : Image size: 64x64, Moment order: N=40

N=50

Figure 182: Image reconstruction results for Q-theta method : Image size: 64x64, Moment order: N=50

Image size: 256x256

Computational procedure: Classical

N=0


Figure 183: Image reconstruction results for Direct method: Image size: 256x256, Moment order: N=0

N=10


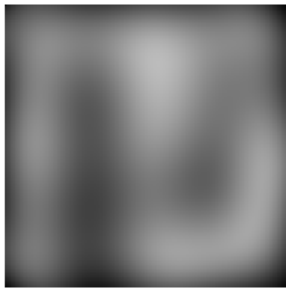

Figure 184: Image reconstruction results for Direct method: Image size: 256x256, Moment order: N=10

N=20

Figure 185: Image reconstruction results for Direct method: Image size: 256x256, Moment order: N=20

N=30

Figure 186: Image reconstruction results for Direct method: Image size: 256x256, Moment order: N=30

N=40

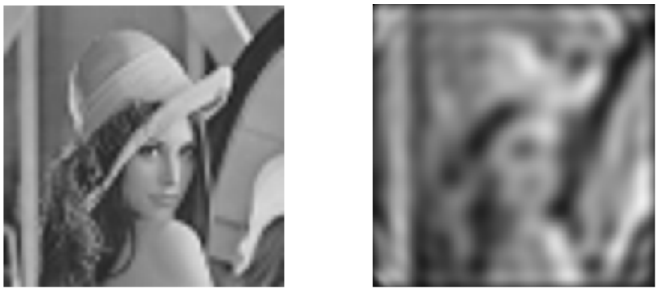

Figure 187: Image reconstruction results for Direct method: Image size: 256x256, Moment order: N=40

N=50

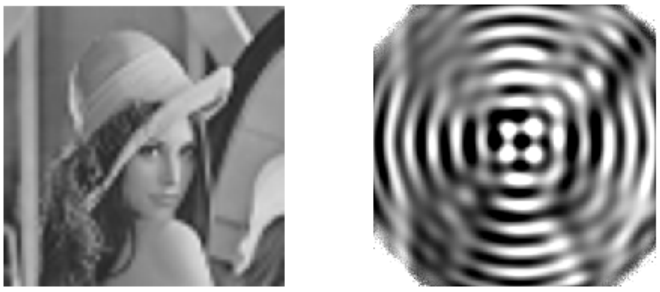

Figure 188: Image reconstruction results for Direct method: Image size: 256x256, Moment order: N=50

Computational procedure: Belkasim

N=0

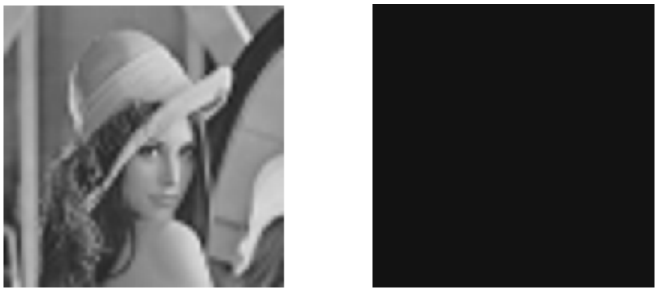

Figure 189: Image reconstruction results for Belkasim method: Image size: 256x256, Moment order: N=0

N=10

Figure 190: Image reconstruction results for Belkasim method: Image size: 256x256, Moment order: N=10

N=20

Figure 191: Image reconstruction results for Belkasim method: Image size: 256x256, Moment order: N=20

N=30

Figure 192: Image reconstruction results for Belkasim method : Image size: 256x256, Moment order: N=30

N=40

Figure 193: Image reconstruction results for Belkasim method : Image size: 256x256, Moment order: N=40

N=50

Figure 194: Image reconstruction results for Belkasim method : Image size: 256x256, Moment order: N=50

Computational procedure: Factorial Free

N=0

Figure 195: Image reconstruction results for Factorial Free method : Image size: 256x256, Moment order: N=0

N=10

Figure 196: Image reconstruction results for Factorial Free method : Image size: 256x256, Moment order: N=10

N=20

Figure 197: Image reconstruction results for Factorial Free method : Image size: 256x256, Moment order: N=20

N=30

Figure 198: Image reconstruction results for Factorial Free method : Image size: 256x256, Moment order: N=30

N=40

Figure 199: Image reconstruction results for Factorial Free method : Image size: 256x256, Moment order: N=40

N=50

Figure 200: Image reconstruction results for Factorial Free method : Image size: 256x256, Moment order: N=50

Computational procedure: Hybrid

N=0

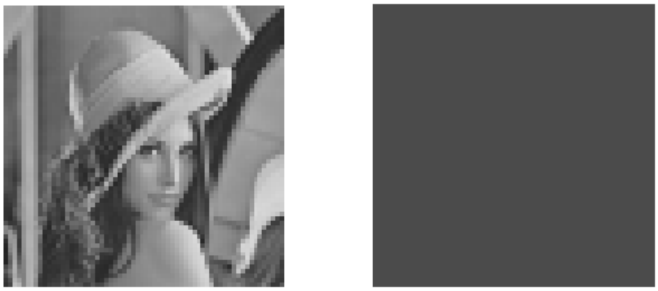

Figure 201: Image reconstruction results for Hybrid method : Image size: 256x256, Moment order: N=0

N=10

Figure 202: Image reconstruction results for Hybrid method : Image size: 256x256, Moment order: N=10

N=20

Figure 203: Image reconstruction results for Hybrid method : Image size: 256x256, Moment order: N=20

N=30

Figure 204: Image reconstruction results for Hybrid method : Image size: 256x256, Moment order: N=30

N=40

Figure 205: Image reconstruction results for Hybrid method : Image size: 256x256, Moment order: N=40

N=50

Figure 206: Image reconstruction results for Hybrid method : Image size: 256x256, Moment order: N=50

Computational procedure: Kintner

N=0

Figure 207: Image reconstruction results for Kintner method : Image size: 256x256, Moment order: N=0

N=10

Figure 208: Image reconstruction results for Kintner method : Image size: 256x256, Moment order: N=10

N=20

Figure 209: Image reconstruction results for Kintner method : Image size: 256x256, Moment order: N=20

N=30

Figure 210: Image reconstruction results for Kintner method : Image size: 256x256, Moment order: N=30

N=40

Figure 211: Image reconstruction results for Kintner method : Image size: 256x256, Moment order: N=40

N=50

Figure 212: Image reconstruction results for Kintner method : Image size: 256x256, Moment order: N=50

Computational procedure: Modified Prata

N=0

Figure 213: Image reconstruction results for Modified Prata method : Image size: 256x256, Moment order: N=0

N=10

Figure 214: Image reconstruction results for Modified Prata method : Image size: 256x256, Moment order: N=10

N=20

Figure 215: Image reconstruction results for Modified Prata method : Image size: 256x256, Moment order: N=20

N=30

Figure 216: Image reconstruction results for Modified Prata method : Image size: 256x256, Moment order: N=30

N=40

Figure 217: Image reconstruction results for Modified Prata method : Image size: 256x256, Moment order: N=40

N=50

Figure 218: Image reconstruction results for Modified Prata method : Image size: 256x256, Moment order: N=50

Computational procedure: Prata

N=0


Figure 219: Image reconstruction results for Prata method : Image size: 256x256, Moment order: N=0

N=10


Figure 220: Image reconstruction results for Prata method : Image size: 256x256, Moment order: N=10

N=20

Figure 221: Image reconstruction results for Prata method : Image size: 256x256, Moment order: N=20

N=30

Figure 222: Image reconstruction results for Prata method : Image size: 256x256, Moment order: N=30

N=40

Figure 223: Image reconstruction results for Prata method : Image size: 256x256, Moment order: N=40

N=50

Figure 224: Image reconstruction results for Prata method : Image size: 256x256, Moment order: N=50

Computational procedure: Q-θ

N=0

Figure 225: Image reconstruction results for Q-theta method : Image size: 256x256, Moment order: N=0

N=10

Figure 226: Image reconstruction results for Q-theta method : Image size: 256x256, Moment order: N=10

N=20

Figure 227: Image reconstruction results for Q-theta method : Image size: 256x256, Moment order: N=20

N=30

Figure 228: Image reconstruction results for Q-theta method : Image size: 256x256, Moment order: N=30

N=40

Figure 229: Image reconstruction results for Q-theta method : Image size: 256x256, Moment order: N=40

N=50

Figure 230: Image reconstruction results for Q-theta method : Image size: 256x256, Moment order: N=50

Image size: 291x240

Computational procedure: Classical

N=0

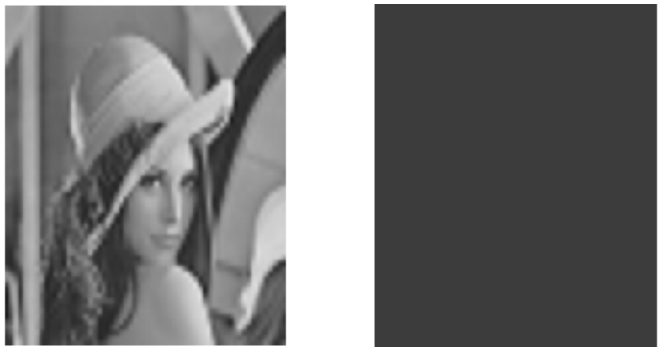

Figure 231: Image reconstruction results for Direct method: Image size: 291x240, Moment order: N=0

N=10

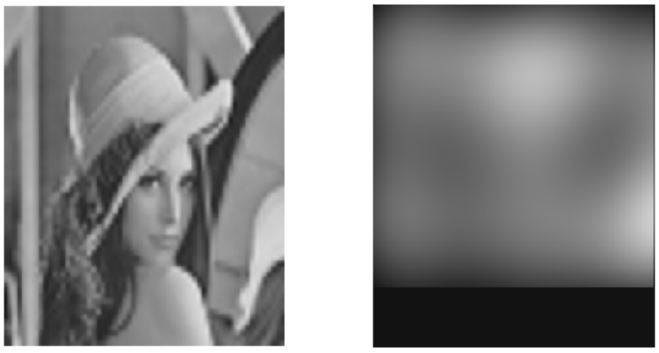

Figure 232: Image reconstruction results for Direct method: Image size: 291x240, Moment order: N=10

N=20

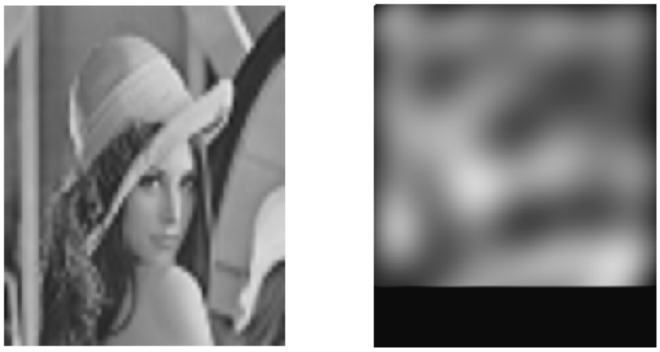

Figure 233: Image reconstruction results for Direct method: Image size: 291x240, Moment order: N=20

N=30

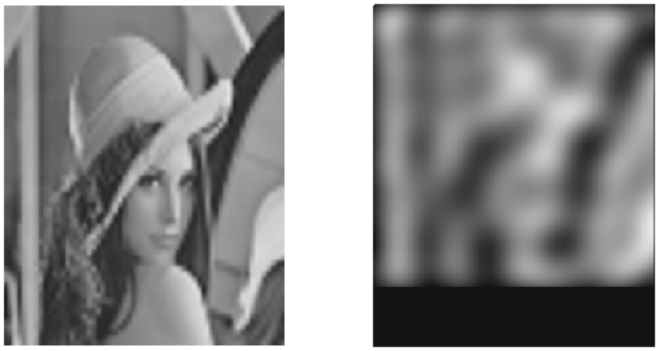

Figure 234: Image reconstruction results for Direct method: Image size: 291x240, Moment order: N=30

N=40

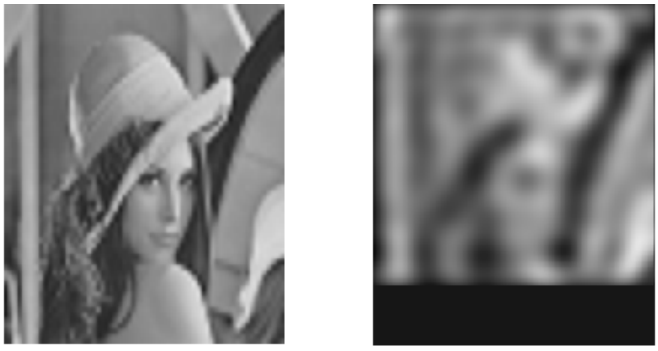

Figure 235: Image reconstruction results for Direct method: Image size: 291x240, Moment order: N=40

N=50

Figure 236: Image reconstruction results for Direct method: Image size: 291x240, Moment order: N=50

Computational procedure: Belkasim

N=0

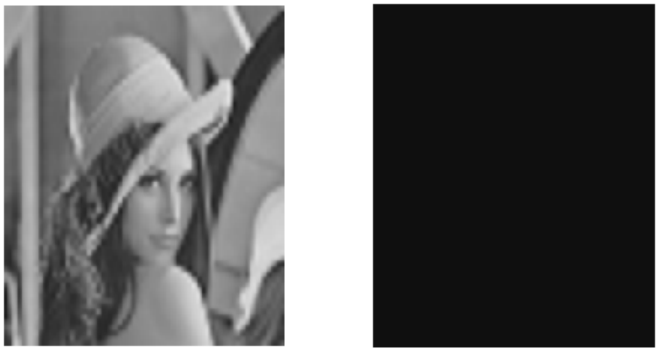

Figure 237: Image reconstruction results for Belkasim method: Image size: 291x240, Moment order: N=0

N=10

Figure 238: Image reconstruction results for Belkasim method: Image size: 291x240, Moment order: N=10

N=20

Figure 239: Image reconstruction results for Belkasim method: Image size: 291x240, Moment order: N=20

N=30

Figure 240: Image reconstruction results for Belkasim method : Image size: 291x240, Moment order: N=30

N=40

Figure 241: Image reconstruction results for Belkasim method : Image size: 291x240, Moment order: N=40

N=50

Figure 242: Image reconstruction results for Belkasim method : Image size: 291x240, Moment order: N=50

Computational procedure: Factorial Free

N=0

Figure 243: Image reconstruction results for Factorial Free method : Image size: 291x240, Moment order: N=0

N=10

Figure 244: Image reconstruction results for Factorial Free method : Image size: 291x240, Moment order: N=10

N=20

Figure 245: Image reconstruction results for Factorial Free method : Image size: 291x240, Moment order: N=20

N=30

Figure 246: Image reconstruction results for Factorial Free method : Image size: 291x240, Moment order: N=30

N=40

Figure 247: Image reconstruction results for Factorial Free method : Image size: 291x240, Moment order: N=40

N=50

Figure 248: Image reconstruction results for Factorial Free method : Image size: 291x240, Moment order: N=50

Computational procedure: Hybrid

N=0

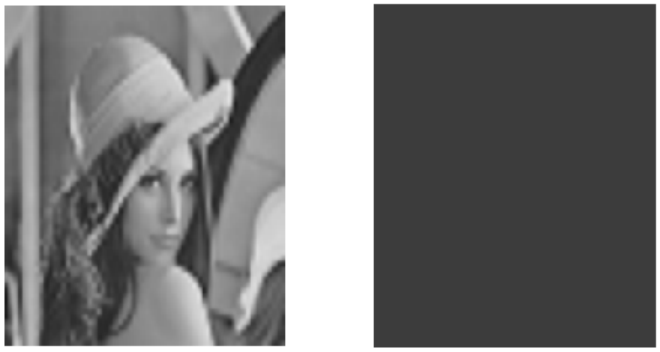

Figure 249: Image reconstruction results for Hybrid method : Image size: 291x240, Moment order: N=0

N=10

Figure 250: Image reconstruction results for Hybrid method : Image size: 291x240, Moment order: N=10

N=20

Figure 251: Image reconstruction results for Hybrid method : Image size: 291x240, Moment order: N=20

N=30

Figure 252: Image reconstruction results for Hybrid method : Image size: 291x240, Moment order: N=30

N=40

Figure 253: Image reconstruction results for Hybrid method : Image size: 291x240, Moment order: N=40

N=50

Figure 254: Image reconstruction results for Hybrid method : Image size: 291x240, Moment order: N=50

Computational procedure: Kintner

N=0

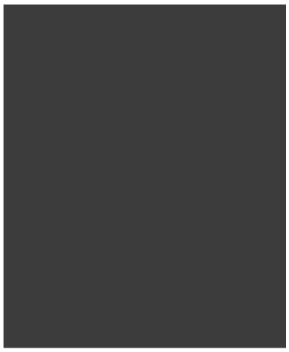

Figure 255: Image reconstruction results for Kintner method : Image size: 291x240, Moment order: N=0

N=10

Figure 256: Image reconstruction results for Kintner method : Image size: 291x240, Moment order: N=10

N=20

Figure 257: Image reconstruction results for Kintner method : Image size: 291x240, Moment order: N=20

N=30

Figure 258: Image reconstruction results for Kintner method : Image size: 291x240, Moment order: N=30

N=40

Figure 259: Image reconstruction results for Kintner method : Image size: 291x240, Moment order: N=40

N=50

Figure 260: Image reconstruction results for Kintner method : Image size: 291x240, Moment order: N=50

Computational procedure: Modified Prata

N=0

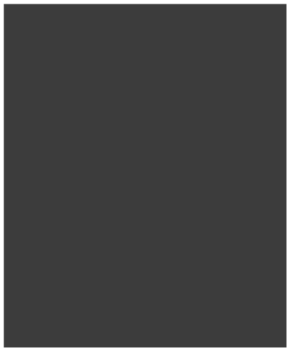

Figure 261: Image reconstruction results for Modified Prata method : Image size: 291x240, Moment order: N=0

N=10

Figure 262: Image reconstruction results for Modified Prata method : Image size: 291x240, Moment order: N=10

N=20

Figure 263: Image reconstruction results for Modified Prata method : Image size: 291x240, Moment order: N=20

N=30

Figure 264: Image reconstruction results for Modified Prata method : Image size: 291x240, Moment order: N=30

N=40

Figure 265: Image reconstruction results for Modified Prata method : Image size: 291x240, Moment order: N=40

N=50

Figure 266: Image reconstruction results for Modified Prata method : Image size: 291x240, Moment order: N=50

Computational procedure: Prata

N=0

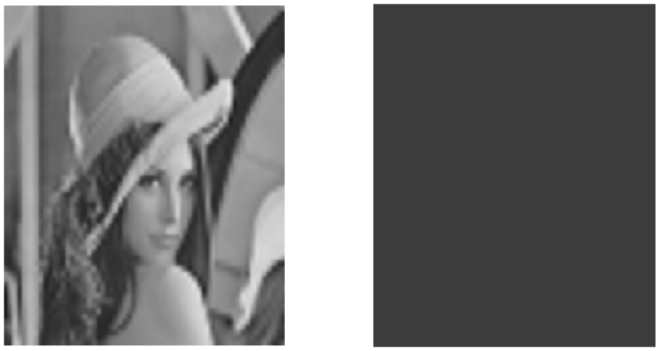

Figure 267: Image reconstruction results for Prata method : Image size: 291x240, Moment order: N=0

N=10

Figure 268: Image reconstruction results for Prata method : Image size: 291x240, Moment order: N=10

N=20

Figure 269: Image reconstruction results for Prata method : Image size: 291x240, Moment order: N=20

N=30

Figure 270: Image reconstruction results for Prata method : Image size: 291x240, Moment order: N=30

N=40

Figure 271: Image reconstruction results for Prata method : Image size: 291x240, Moment order: N=40

N=50

Figure 272: Image reconstruction results for Prata method : Image size: 291x240, Moment order: N=50

Computational procedure: Q-θ

N=0

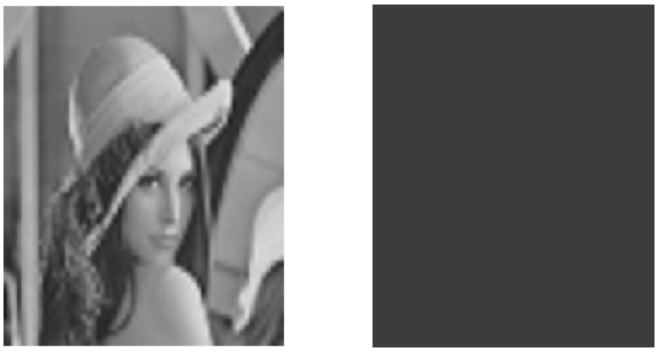

Figure 273: Image reconstruction results for Q-theta method : Image size: 291x240, Moment order: N=0

N=10

Figure 274: Image reconstruction results for Q-theta method : Image size: 291x240, Moment order: N=10

N=20

Figure 275: Image reconstruction results for Q-theta method : Image size: 291x240, Moment order: N=20

N=30

Figure 276: Image reconstruction results for Q-theta method : Image size: 291x240, Moment order: N=30

N=40

Figure 277: Image reconstruction results for Q-theta method : Image size: 291x240, Moment order: N=40

N=50

Figure 278: Image reconstruction results for Q-theta method : Image size: 291x240, Moment order: N=50

Image size: 367x490

Computational procedure: Classical

N=0

Figure 279: Image reconstruction results for Direct method: Image size: 367x490, Moment order: N=0

N=10

Figure 280: Image reconstruction results for Direct method: Image size: 367x490, Moment order: N=10

N=20

Figure 281: Image reconstruction results for Direct method: Image size: 367x490, Moment order: N=20

N=30

Figure 282: Image reconstruction results for Direct method: Image size: 367x490, Moment order: N=30

N=40

Figure 283: Image reconstruction results for Direct method: Image size: 367x490, Moment order: N=40

N=50

Figure 284: Image reconstruction results for Direct method: Image size: 367x490, Moment order: N=50

Computational procedure: Belkasim

N=0

Figure 285: Image reconstruction results for Belkasim method: Image size: 367x490, Moment order: N=0

N=10

Figure 286: Image reconstruction results for Belkasim method: Image size: 367x490, Moment order: N=10

N=20

Figure 287: Image reconstruction results for Belkasim method: Image size: 367x490, Moment order: N=20

N=30

Figure 288: Image reconstruction results for Belkasim method : Image size: 367x490, Moment order: N=30

N=40

Figure 289: Image reconstruction results for Belkasim method : Image size: 367x490, Moment order: N=40

N=50

Figure 290: Image reconstruction results for Belkasim method : Image size: 367x490, Moment order: N=50

Computational procedure: Factorial Free

N=0

Figure 291: Image reconstruction results for Factorial Free method : Image size: 367x490, Moment order: N=0

N=10

Figure 292: Image reconstruction results for Factorial Free method : Image size: 367x490, Moment order: N=10

N=20

Figure 293: Image reconstruction results for Factorial Free method : Image size: 367x490, Moment order: N=20

N=30

Figure 294: Image reconstruction results for Factorial Free method : Image size: 367x490, Moment order: N=30

N=40

Figure 295: Image reconstruction results for Factorial Free method : Image size: 367x490, Moment order: N=40

N=50

Figure 296: Image reconstruction results for Factorial Free method : Image size: 367x490, Moment order: N=50

Computational procedure: Hybrid

N=0

Figure 297: Image reconstruction results for Hybrid method : Image size: 367x490, Moment order: N=0

N=10

Figure 298: Image reconstruction results for Hybrid method : Image size: 367x490, Moment order: N=10

N=20

Figure 299: Image reconstruction results for Hybrid method : Image size: 367x490, Moment order: N=20

N=30

Figure 300: Image reconstruction results for Hybrid method : Image size: 367x490, Moment order: N=30

N=40

Figure 301: Image reconstruction results for Hybrid method : Image size: 367x490, Moment order: N=40

N=50

Figure 302: Image reconstruction results for Hybrid method : Image size: 367x490, Moment order: N=50

Computational procedure: Kintner

N=0

Figure 303: Image reconstruction results for Kintner method : Image size: 367x490, Moment order: N=0

N=10

Figure 304: Image reconstruction results for Kintner method : Image size: 367x490, Moment order: N=10

N=20

Figure 305: Image reconstruction results for Kintner method : Image size: 367x490, Moment order: N=20

N=30

Figure 306: Image reconstruction results for Kintner method : Image size: 367x490, Moment order: N=30

N=40

Figure 307: Image reconstruction results for Kintner method : Image size: 367x490, Moment order: N=40

N=50

Figure 308: Image reconstruction results for Kintner method : Image size: 367x490, Moment order: N=50

Computational procedure: Modified Prata

N=0

Figure 309: Image reconstruction results for Modified Prata method : Image size: 367x490, Moment order: N=0

N=10

Figure 310: Image reconstruction results for Modified Prata method : Image size: 367x490, Moment order: N=10

N=20

Figure 311: Image reconstruction results for Modified Prata method : Image size: 367x490, Moment order: N=20

N=30

Figure 312: Image reconstruction results for Modified Prata method : Image size: 367x490, Moment order: N=30

N=40

Figure 313: Image reconstruction results for Modified Prata method : Image size: 367x490, Moment order: N=40

N=50

Figure 314: Image reconstruction results for Modified Prata method : Image size: 367x490, Moment order: N=50

Computational procedure: Prata

N=0


Figure 315: Image reconstruction results for Prata method : Image size: 367x490, Moment order: N=0

N=10


Figure 316: Image reconstruction results for Prata method : Image size: 367x490, Moment order: N=10

N=20


Figure 317: Image reconstruction results for Prata method : Image size: 367x490, Moment order: N=20

N=30

Figure 318: Image reconstruction results for Prata method : Image size: 367x490, Moment order: N=30

N=40

Figure 319: Image reconstruction results for Prata method : Image size: 367x490, Moment order: N=40

N=50

Figure 320: Image reconstruction results for Prata method : Image size: 367x490, Moment order: N=50

Computational procedure: Q-θ

N=0

Figure 321: Image reconstruction results for Q-theta method : Image size: 367x490, Moment order: N=0

N=10

Figure 322: Image reconstruction results for Q-theta method : Image size: 367x490, Moment order: N=10

N=20

Figure 323: Image reconstruction results for Q-theta method : Image size: 367x490, Moment order: N=20

N=30

Figure 324: Image reconstruction results for Q-theta method : Image size: 367x490, Moment order: N=30

N=40

Figure 325: Image reconstruction results for Q-theta method : Image size: 367x490, Moment order: N=40

N=50

Figure 326: Image reconstruction results for Q-theta method : Image size: 367x490, Moment order: N=50

Image size: 480x640

Computational procedure: Classical

N=0

Figure 327: Image reconstruction results for Direct method: Image size: 480x640, Moment order: N=0

N=10

Figure 328: Image reconstruction results for Direct method: Image size: 480x640, Moment order: N=10

N=20

Figure 329: Image reconstruction results for Direct method: Image size: 480x640, Moment order: N=20

N=30

Figure 330: Image reconstruction results for Direct method: Image size: 480x640, Moment order: N=30

N=40

Figure 331: Image reconstruction results for Direct method: Image size: 480x640, Moment order: N=40

N=50

Figure 332: Image reconstruction results for Direct method: Image size: 480x640, Moment order: N=50

Computational procedure: Belkasim

N=0

Figure 333: Image reconstruction results for Belkasim method: Image size: 480x640, Moment order: N=0

N=10

Figure 334: Image reconstruction results for Belkasim method: Image size: 480x640, Moment order: N=10

N=20

Figure 335: Image reconstruction results for Belkasim method: Image size: 480x640, Moment order: N=20

N=30

Figure 336: Image reconstruction results for Belkasim method : Image size: 480x640, Moment order: N=30

N=40

Figure 337: Image reconstruction results for Belkasim method : Image size: 480x640, Moment order: N=40

N=50

Figure 338: Image reconstruction results for Belkasim method : Image size: 480x640, Moment order: N=50

Computational procedure: Factorial Free

N=0

Figure 339: Image reconstruction results for Factorial Free method : Image size: 480x640, Moment order: N=0

N=10

Figure 340: Image reconstruction results for Factorial Free method : Image size: 480x640, Moment order: N=10

N=20

Figure 341: Image reconstruction results for Factorial Free method : Image size: 480x640, Moment order: N=20

N=30

Figure 342: Image reconstruction results for Factorial Free method : Image size: 480x640, Moment order: N=30

N=40

Figure 343: Image reconstruction results for Factorial Free method : Image size: 480x640, Moment order: N=40

N=50

Figure 344: Image reconstruction results for Factorial Free method : Image size: 480x640, Moment order: N=50

Computational procedure: Hybrid

N=0

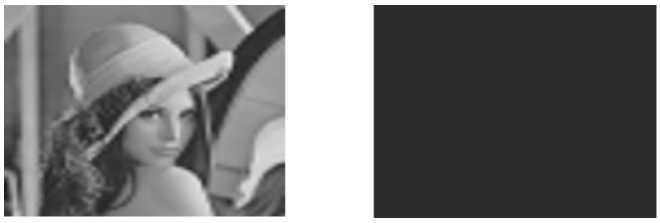

Figure 345: Image reconstruction results for Hybrid method : Image size: 480x640, Moment order: N=0

N=10

Figure 346: Image reconstruction results for Hybrid method : Image size: 480x640, Moment order: N=10

N=20

Figure 347: Image reconstruction results for Hybrid method : Image size: 480x640, Moment order: N=20

N=30

Figure 348: Image reconstruction results for Hybrid method : Image size: 480x640, Moment order: N=30

N=40

Figure 349: Image reconstruction results for Hybrid method : Image size: 480x640, Moment order: N=40

N=50

Figure 350: Image reconstruction results for Hybrid method : Image size: 480x640, Moment order: N=50

Computational procedure: Kintner

N=0

Figure 351: Image reconstruction results for Kintner method : Image size: 480x640, Moment order: N=0

N=10

Figure 352: Image reconstruction results for Kintner method : Image size: 480x640, Moment order: N=10

N=20

Figure 353: Image reconstruction results for Kintner method : Image size: 480x640, Moment order: N=20

N=30

Figure 354: Image reconstruction results for Kintner method : Image size: 480x640, Moment order: N=30

N=40

Figure 355: Image reconstruction results for Kintner method : Image size: 480x640, Moment order: N=40

N=50

Figure 356: Image reconstruction results for Kintner method : Image size: 480x640, Moment order: N=50

Computational procedure: Modified Prata

N=0

Figure 357: Image reconstruction results for Modified Prata method : Image size: 480x640, Moment order: N=0

N=10

Figure 358: Image reconstruction results for Modified Prata method : Image size: 480x640, Moment order: N=10

N=20

Figure 359: Image reconstruction results for Modified Prata method : Image size: 480x640, Moment order: N=20

N=30

Figure 360: Image reconstruction results for Modified Prata method : Image size: 480x640, Moment order: N=30

N=40

Figure 361: Image reconstruction results for Modified Prata method : Image size: 480x640, Moment order: N=40

N=50

Figure 362: Image reconstruction results for Modified Prata method : Image size: 480x640, Moment order: N=50

Computational procedure: Prata

N=0

Figure 363: Image reconstruction results for Prata method : Image size: 480x640, Moment order: N=0

N=10

Figure 364: Image reconstruction results for Prata method : Image size: 480x640, Moment order: N=10

N=20

Figure 365: Image reconstruction results for Prata method : Image size: 480x640, Moment order: N=20

N=30

Figure 366: Image reconstruction results for Prata method : Image size: 480x640, Moment order: N=30

N=40

Figure 367: Image reconstruction results for Prata method : Image size: 480x640, Moment order: N=40

N=50

Figure 368: Image reconstruction results for Prata method : Image size: 480x640, Moment order: N=50

Computational procedure: Q-θ

N=0

Figure 369: Image reconstruction results for Q-theta method : Image size: 480x640, Moment order: N=0

N=10

Figure 370: Image reconstruction results for Q-theta method : Image size: 480x640, Moment order: N=10

N=20

Figure 371: Image reconstruction results for Q-theta method : Image size: 480x640, Moment order: N=20

N=30

Figure 372: Image reconstruction results for Q-theta method : Image size: 480x640, Moment order: N=30

N=40

Figure 373: Image reconstruction results for Q-theta method : Image size: 480x640, Moment order: N=40

N=50

Figure 374: Image reconstruction results for Q-theta method : Image size: 480x640, Moment order: N=50

Appendix 2

For part 1 of experiment 2, the input and output images for the classical method are shown below for each combination of image size and moment order.

Image size (10x10)

N=10

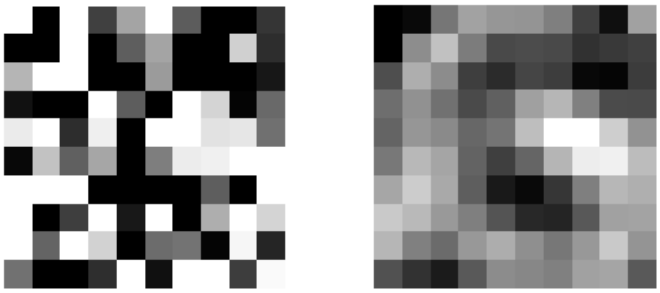

Figure 375: Image reconstruction results for Direct method with Gaussian noise : Image size: 10x10, Moment order: N=10

N=20

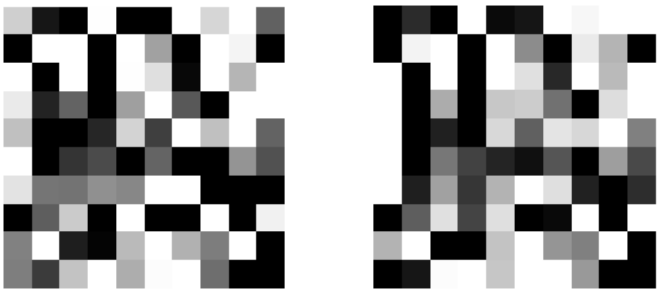

Figure 376: Image reconstruction results for Direct method with Gaussian noise : Image size: 10x10, Moment order: N=20

N=30

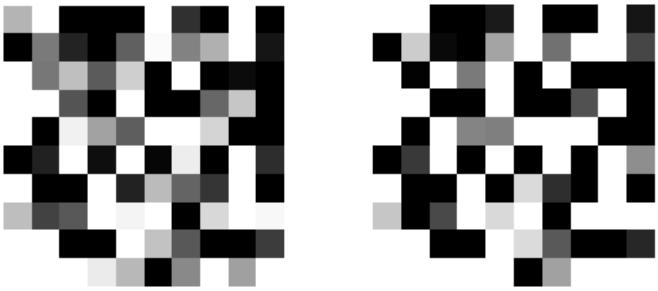

Figure 377: Image reconstruction results for Direct method with Gaussian noise : Image size: 10x10, Moment order: N=30

N=40

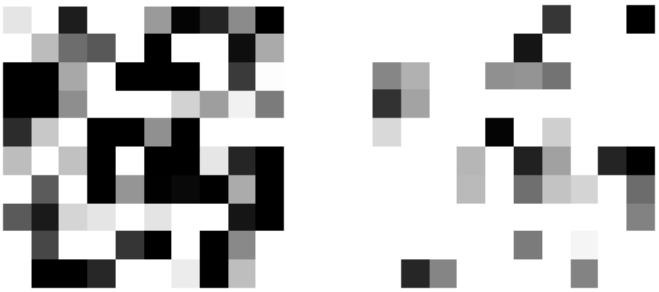

Figure 378: Image reconstruction results for Direct method with Gaussian noise : Image size: 10x10, Moment order: N=40

N=50

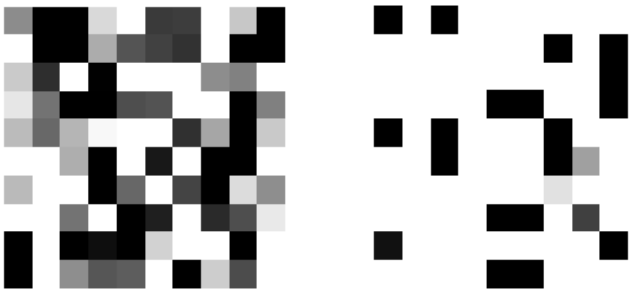

Figure 379: Image reconstruction results for Direct method with Gaussian noise : Image size: 10x10, Moment order: N=50

Image size (32x32)

N=0

Figure 380: Image reconstruction results for Direct method with Gaussian noise : Image size: 32x32, Moment order: N=0

N=10

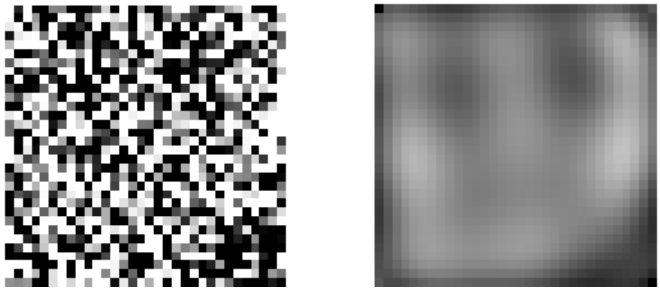

Figure 381: Image reconstruction results for Direct method with Gaussian noise : Image size: 32x32, Moment order: N=10

N=20

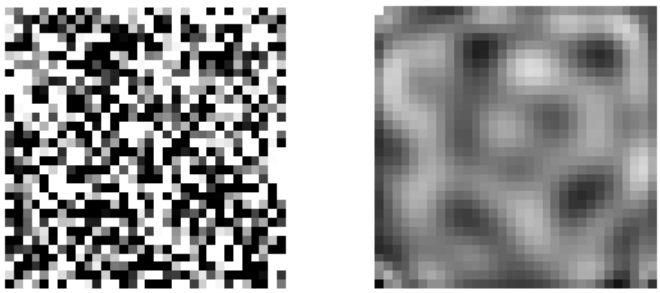

Figure 382: Image reconstruction results for Direct method with Gaussian noise : Image size: 32x32, Moment order: N=20

N=30

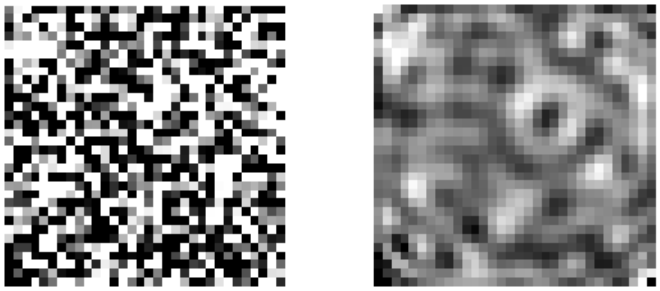

Figure 383: Image reconstruction results for Direct method with Gaussian noise : Image size: 32x32, Moment order: N=30

N=40

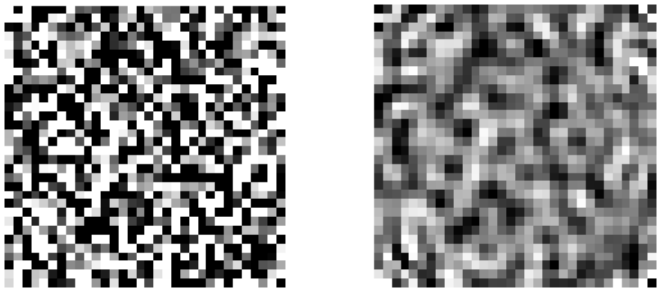

Figure 384: Image reconstruction results for Direct method with Gaussian noise : Image size: 32x32, Moment order: N=40

N=50

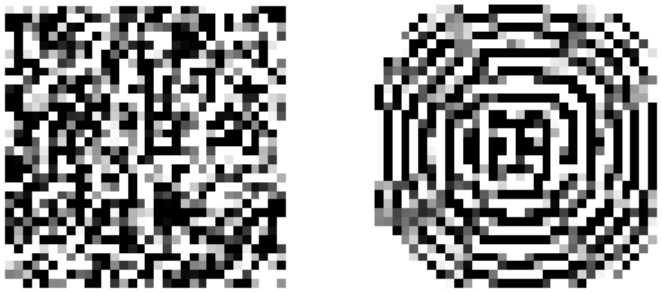

Figure 385: Image reconstruction results for Direct method with Gaussian noise : Image size: 32x32, Moment order: N=50

Image size (64x64)

N=0

Figure 386: Image reconstruction results for Direct method with Gaussian noise : Image size: 64x64, Moment order: N=0

N=10

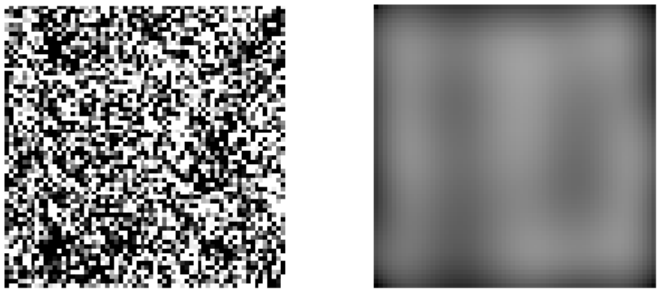

Figure 387: Image reconstruction results for Direct method with Gaussian noise : Image size: 64x64, Moment order: N=10

N=20

Figure 388: Image reconstruction results for Direct method with Gaussian noise : Image size: 64x64, Moment order: N=20

N=30

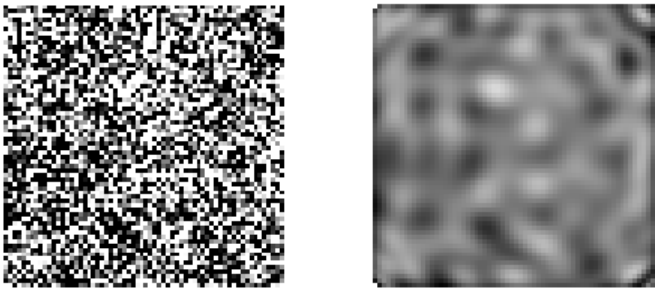

Figure 389: Image reconstruction results for Direct method with Gaussian noise : Image size: 64x64, Moment order: N=30

N=40

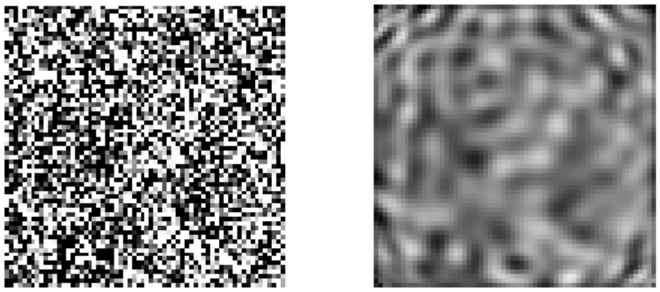

Figure 390: Image reconstruction results for Direct method with Gaussian noise : Image size: 64x64, Moment order: N=40

N=50

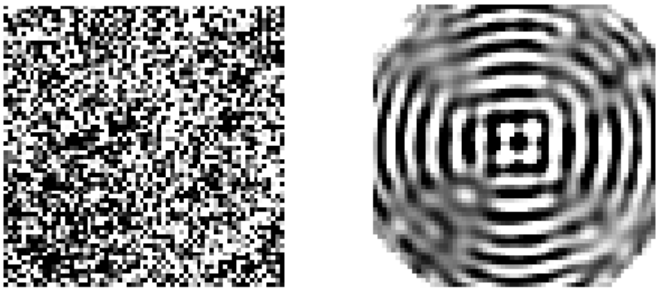

Figure 391: Image reconstruction results for Direct method with Gaussian noise : Image size: 64x64, Moment order: N=50

Image size (128x128)

N=0

Figure 392: Image reconstruction results for Direct method with Gaussian noise : Image size: 128x128, Moment order: N=0

N=10

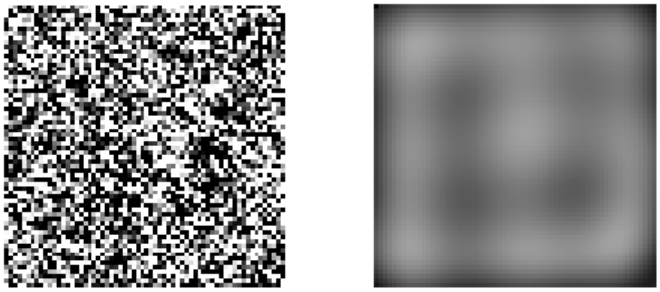

Figure 393: Image reconstruction results for Direct method with Gaussian noise : Image size: 128x128, Moment order: N=10

N=20

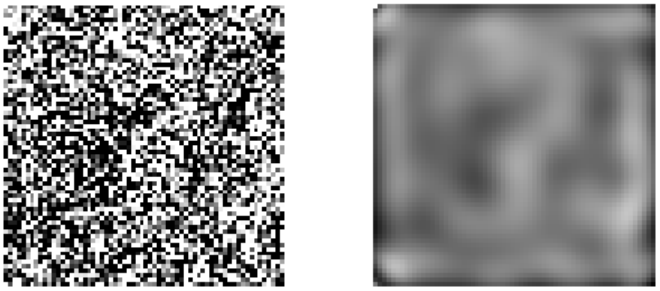

Figure 394: Image reconstruction results for Direct method with Gaussian noise : Image size: 128x128, Moment order: N=20

N=30

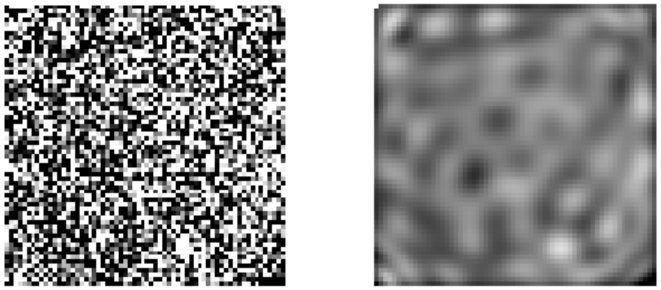

Figure 395: Image reconstruction results for Direct method with Gaussian noise : Image size: 128x128, Moment order: N=30

N=40

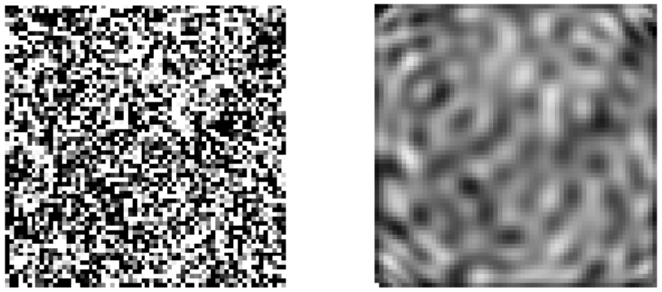

Figure 396: Image reconstruction results for Direct method with Gaussian noise : Image size: 128x128, Moment order: N=40

N=50

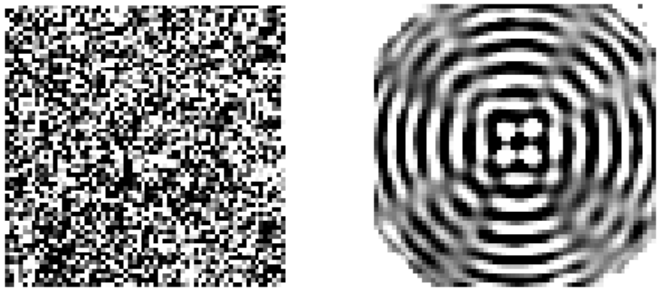

Figure 397: Image reconstruction results for Direct method with Gaussian noise : Image size: 128x128, Moment order: N=50

Appendix 3

For part 2 of experiment 2, the input image and output images are shown below for each combination of noise type and moment order

Noise type: Gaussian white noise

Image size (64x64)

N=0

Figure 398: Image reconstruction results for Direct method with Gaussian noise : Image size: 64x64, Moment order: N=0

N=10

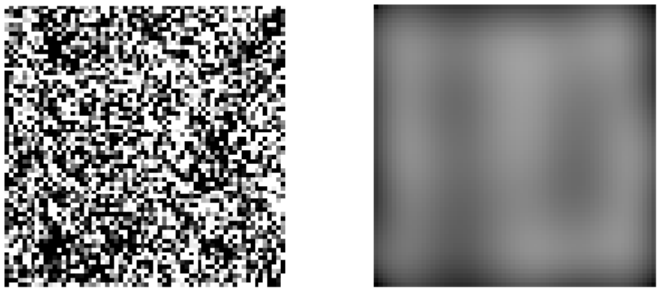

Figure 399: Image reconstruction results for Direct method with Gaussian noise : Image size: 64x64, Moment order: N=10

N=20

Figure 400: Image reconstruction results for Direct method with Gaussian noise : Image size: 64x64, Moment order: N=20

N=30

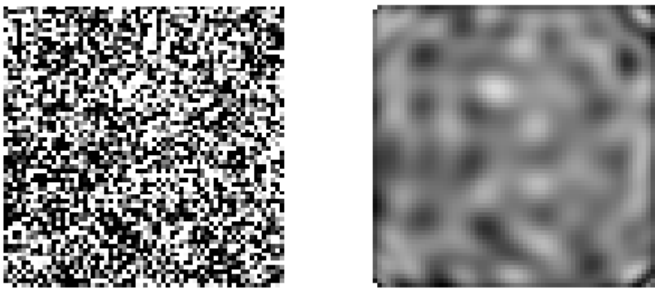

Figure 401: Image reconstruction results for Direct method with Gaussian noise : Image size: 64x64, Moment order: N=30

N=40

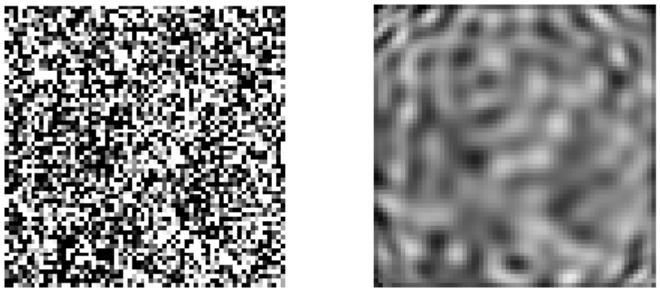

Figure 402: Image reconstruction results for Direct method with Gaussian noise : Image size: 64x64, Moment order: N=40

N=50

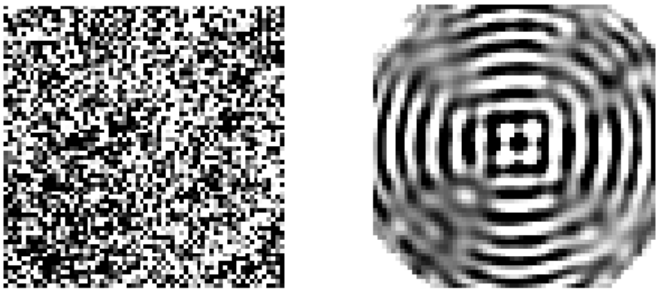

Figure 403: Image reconstruction results for Direct method with Gaussian noise : Image size: 64x64, Moment order: N=50

Noise type: Poisson noise

Image size (64x64)

N=0

Figure 404: Image reconstruction results for Direct method with Poisson noise : Image size: 64x64, Moment order: N=0

N=10

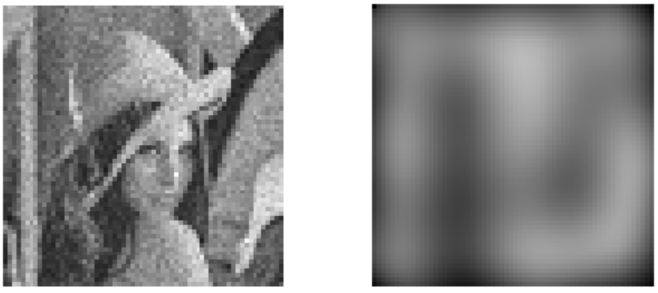

Figure 405: Image reconstruction results for Direct method with Poisson noise : Image size: 64x64, Moment order: N=10

N=20

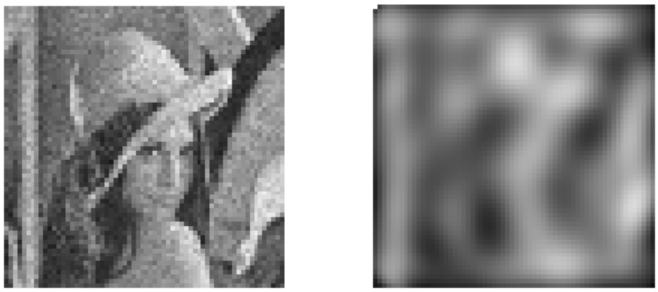

Figure 406: Image reconstruction results for Direct method with Poisson noise : Image size: 64x64, Moment order: N=20

N=30

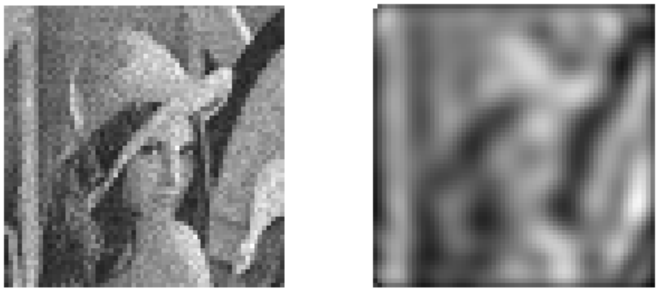

Figure 407: Image reconstruction results for Direct method with Poisson noise : Image size: 64x64, Moment order: N=30

N=40

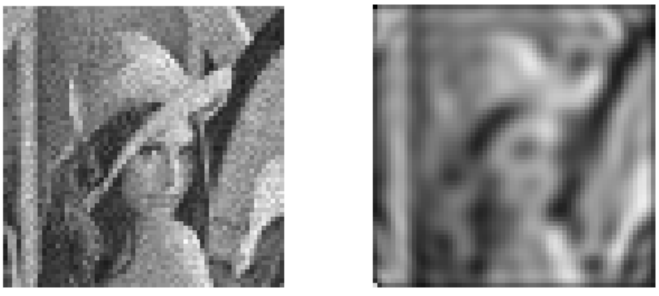

Figure 408: Image reconstruction results for Direct method with Poisson noise : Image size: 64x64, Moment order: N=40

N=50

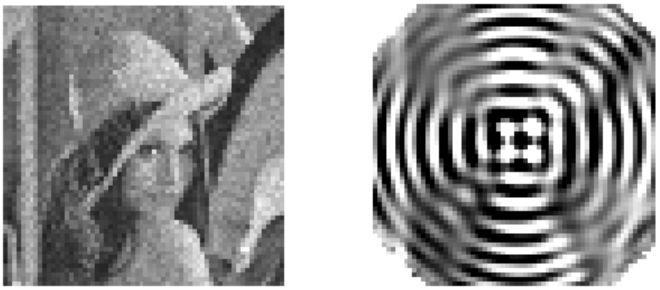

Figure 409: Image reconstruction results for Direct method with Poisson noise : Image size: 64x64, Moment order: N=50

Noise type: Salt & Pepper noise

Image size (64x64)

N=0

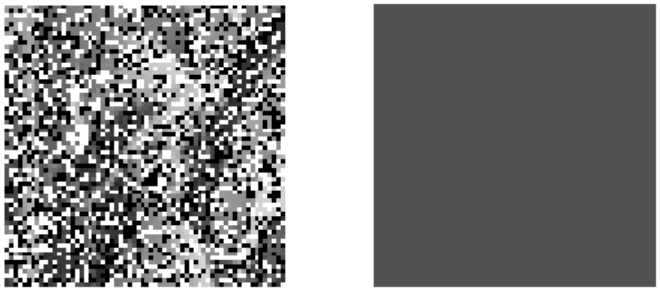

Figure 410: Image reconstruction results for Direct method with Salt & Pepper noise : Image size: 64x64, Moment order: N=0

N=10

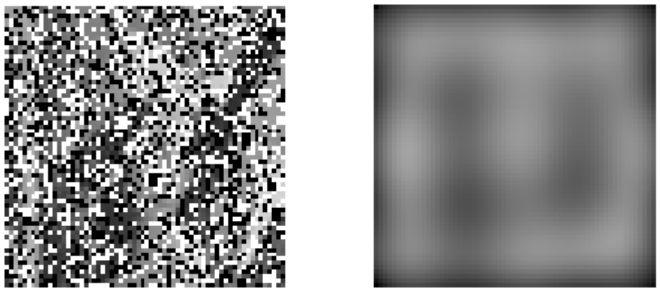

Figure 411: Image reconstruction results for Direct method with Salt & Pepper noise : Image size: 64x64, Moment order: N=10

N=20

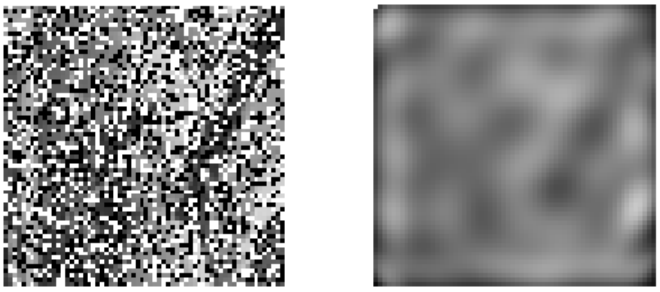

Figure 412: Image reconstruction results for Direct method with Salt & Pepper noise : Image size: 64x64, Moment order: N=20

N=30

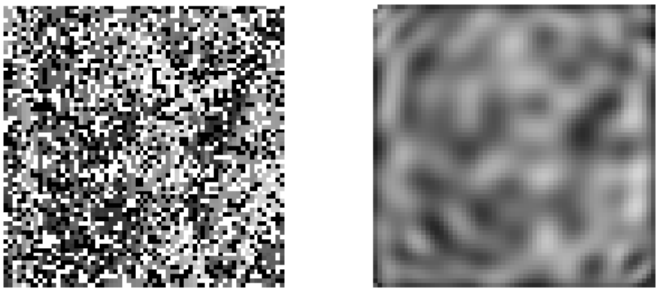

Figure 413: Image reconstruction results for Direct method with Salt & Pepper noise : Image size: 64x64, Moment order: N=30

N=40

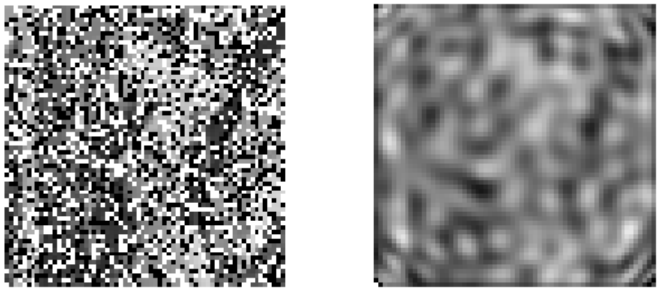

Figure 414: Image reconstruction results for Direct method with Salt & Pepper noise : Image size: 64x64, Moment order: N=40

N=50

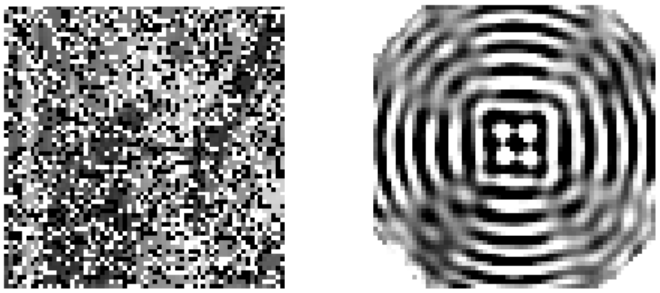

Figure 415: Image reconstruction results for Direct method with Salt & Pepper noise : Image size: 64x64, Moment order: N=50

Appendix 4

The difference between the Direct method and the other methods are shown below.

0.5	0	0	0	0	0	0	0	0	0	0	0	0
0	-21.454	0	0	0	0	0	0	0	0	0	0	0
-20.8667	0	-20.8073	0	0	0	0	0	0	0	0	0	0
0	-20.2953	0	-45100.5	0	0	0	0	0	0	0	0	0
-43763	0	-43741.2	0	-2022690	0	0	0	0	0	0	0	0
0	-1961743	0	-1961722	0	-9.1E+07	0	0	0	0	0	0	0
-8.8E+07	0	-8.8E+07	0	-8.8E+07	0	-4.1E+09	0	0	0	0	0	0
0	-3.9E+09	0	-3.9E+09	0	-3.9E+09	0	-1.8E+11	0	0	0	0	0
-1.8E+11	0	-1.8E+11	0	-1.8E+11	0	-1.8E+11	0	-8.2E+12	0	0	0	0
0	-7.9E+12	0	-7.9E+12	0	-7.9E+12	0	-7.9E+12	0	-3.7E+14	0	0	0
-3.6E+14	0	-3.6E+14	0	-3.6E+14	0	-3.6E+14	0	-3.6E+14	0	-1.65E+16	0	0
0	-1.60E+16	0	-1.60E+16	0	-1.60E+16	0	-1.60E+16	0	-1.60E+16	0	-7.38E+17	0
-7.16E+17	0	-7.16E+17	0	-7.16E+17	0	-7.16E+17	0	-7.16E+17	0	-7.16E+17	0	-3.31E+19
0	-3.21E+19	0	-3.21E+19	0	-3.21E+19	0	-3.21E+19	0	-3.21E+19	0	-3.21E+19	0
-1.44E+21	0	-1.44E+21	0	-1.44E+21	0	-1.44E+21	0	-1.44E+21	0	-1.44E+21	0	-1.44E+21
0	-6.46E+22	0	-6.46E+22	0	-6.46E+22	0	-6.46E+22	0	-6.46E+22	0	-6.46E+22	0
-2.90E+24	0	-2.90E+24	0	-2.90E+24	0	-2.90E+24	0	-2.90E+24	0	-2.90E+24	0	-2.90E+24
0	-1.30E+26	0	-1.30E+26	0	-1.30E+26	0	-1.30E+26	0	-1.30E+26	0	-1.30E+26	0
-5.82E+27	0	-5.82E+27	0	-5.82E+27	0	-5.82E+27	0	-5.82E+27	0	-5.82E+27	0	-5.82E+27
0	-2.61E+29	0	-2.61E+29	0	-2.61E+29	0	-2.61E+29	0	-2.61E+29	0	-2.61E+29	0
-1.17E+31	0	-1.17E+31	0	-1.17E+31	0	-1.17E+31	0	-1.17E+31	0	-1.17E+31	0	-1.17E+31
0	-5.25E+32	0	-5.25E+32	0	-5.25E+32	0	-5.25E+32	0	-5.25E+32	0	-5.25E+32	0
-2.36E+34	0	-2.36E+34	0	-2.36E+34	0	-2.36E+34	0	-2.36E+34	0	-2.36E+34	0	-2.36E+34
0	-1.06E+36	0	-1.06E+36	0	-1.06E+36	0	-1.06E+36	0	-1.06E+36	0	-1.06E+36	0
-4.74E+37	0	-4.74E+37	0	-4.74E+37	0	-4.74E+37	0	-4.74E+37	0	-4.74E+37	0	-4.74E+37
0	-2.13E+39	0	-2.13E+39	0	-2.13E+39	0	-2.13E+39	0	-2.13E+39	0	-2.13E+39	0
-9.53E+40	0	-9.53E+40	0	-9.53E+40	0	-9.53E+40	0	-9.53E+40	0	-9.53E+40	0	-9.53E+40
0	-4.27E+42	0	-4.27E+42	0	-4.27E+42	0	-4.27E+42	0	-4.27E+42	0	-4.27E+42	0
-1.92E+44	0	-1.92E+44	0	-1.92E+44	0	-1.92E+44	0	-1.92E+44	0	-1.92E+44	0	-1.92E+44
0	-8.60E+45	0	-8.60E+45	0	-8.60E+45	0	-8.60E+45	0	-8.60E+45	0	-8.60E+45	0
-3.86E+47	0	-3.86E+47	0	-3.86E+47	0	-3.86E+47	0	-3.86E+47	0	-3.86E+47	0	-3.86E+47

Figure 416: Difference between Direct and Belkasim radial polynomial for moment order up to N=30

0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
-1.48E+21	0	0	0	0	0	0	0	0	0	0	0	0
0	-6.66E+22	0	0	0	0	0	0	0	0	0	0	0
-6.46E+22	0	-2.99E+24	0	0	0	0	0	0	0	0	0	0
0	-2.90E+24	0	-1.34E+26	0	0	0	0	0	0	0	0	0
-1.30E+26	0	-1.30E+26	0	-6.01E+27	0	0	0	0	0	0	0	0
0	-5.82E+27	0	-5.82E+27	0	-2.69E+29	0	0	0	0	0	0	0
-2.61E+29	0	-2.61E+29	0	-2.61E+29	0	-1.21E+31	0	0	0	0	0	0
0	-1.17E+31	0	-1.17E+31	0	-1.17E+31	0	-5.42E+32	0	0	0	0	0
-5.25E+32	0	-5.25E+32	0	-5.25E+32	0	-5.25E+32	0	-2.43E+34	0	0	0	0
0	-2.36E+34	0	-2.36E+34	0	-2.36E+34	0	-2.36E+34	0	-1.09E+36	0	0	0
-1.06E+36	0	-1.06E+36	0	-1.06E+36	0	-1.06E+36	0	-1.06E+36	0	-4.89E+37	0	0
0	-4.74E+37	0	-4.74E+37	0	-4.74E+37	0	-4.74E+37	0	-4.74E+37	0	-2.19E+39	0
-2.13E+39	0	-2.13E+39	0	-2.13E+39	0	-2.13E+39	0	-2.13E+39	0	-2.13E+39	0	-9.83E+40
0	-9.53E+40	0	-9.53E+40	0	-9.53E+40	0	-9.53E+40	0	-9.53E+40	0	-9.53E+40	0
-4.27E+42	0	-4.27E+42	0	-4.27E+42	0	-4.27E+42	0	-4.27E+42	0	-4.27E+42	0	-4.27E+42
0	-1.92E+44	0	-1.92E+44	0	-1.92E+44	0	-1.92E+44	0	-1.92E+44	0	-1.92E+44	0
-8.60E+45	0	-8.60E+45	0	-8.60E+45	0	-8.60E+45	0	-8.60E+45	0	-8.60E+45	0	-8.60E+45
0	-3.86E+47	0	-3.86E+47	0	-3.86E+47	0	-3.86E+47	0	-3.86E+47	0	-3.86E+47	0

Figure 417: Difference between Direct and Belkasim radial polynomial for moment order up to N=30

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
-4.41E+42	0	0	0	0
0	-1.98E+44	0	0	0
-1.92E+44	0	-8.87E+45	0	0
0	-8.60E+45	0	-3.98E+47	0
-3.86E+47	0	-3.86E+47	0	-1.78E+49

Figure 418: Difference between Direct and Belkasim radial polynomial for moment order up to N=30

-9.71388	0	0	0	0	0	0	0	0	0	0	0	0
0	-1847.08	0	0	0	0	0	0	0	0	0	0	0
-143494	0	-551984	0	0	0	0	0	0	0	0	0	0
0	-6.1E+07	0	-2.3E+08	0	0	0	0	0	0	0	0	0
-5E+09	0	-3.3E+10	0	-1.3E+11	0	0	0	0	0	0	0	0
0	-3.3E+12	0	-2.2E+13	0	-8.5E+13	0	0	0	0	0	0	0
-2.8E+14	0	-2.62E+15	0	-1.74E+16	0	-6.71E+16	0	0	0	0	0	0
0	-2.54E+17	0	-2.39E+18	0	-1.59E+19	0	-6.12E+19	0	0	0	0	0
-2.16E+19	0	-2.62E+20	0	-2.47E+21	0	-1.65E+22	0	-6.33E+22	0	0	0	0
0	-2.50E+22	0	-3.03E+23	0	-2.86E+24	0	-1.90E+25	0	-7.32E+25	0	0	0
-2.14E+24	0	-3.19E+25	0	-3.88E+26	0	-3.65E+27	0	-2.43E+28	0	-9.36E+28	0	0
0	-3.00E+27	0	-4.47E+28	0	-5.43E+29	0	-5.12E+30	0	-3.41E+31	0	-1.31E+32	0
-2.59E+29	0	-4.57E+30	0	-6.80E+31	0	-8.27E+32	0	-7.79E+33	0	-5.19E+34	0	-2.00E+35
0	-4.27E+32	0	-7.52E+33	0	-1.12E+35	0	-1.36E+36	0	-1.28E+37	0	-8.53E+37	0
-3.71E+34	0	-7.54E+35	0	-1.33E+37	0	-1.98E+38	0	-2.40E+39	0	-2.26E+40	0	-1.51E+41
0	-7.00E+37	0	-1.42E+39	0	-2.51E+40	0	-3.73E+41	0	-4.54E+42	0	-4.28E+43	0
-6.10E+39	0	-1.41E+41	0	-2.86E+42	0	-5.04E+43	0	-7.51E+44	0	-9.13E+45	0	-8.60E+46
0	-1.30E+43	0	-3.00E+44	0	-6.10E+45	0	-1.08E+47	0	-1.60E+48	0	-1.95E+49	0
-1.14E+45	0	-2.93E+46	0	-6.76E+47	0	-1.38E+49	0	-2.42E+50	0	-3.61E+51	0	-4.39E+52
0	-2.70E+48	0	-6.97E+49	0	-1.61E+51	0	-3.27E+52	0	-5.76E+53	0	-8.58E+54	0
-2.37E+50	0	-6.75E+51	0	-1.74E+53	0	-4.02E+54	0	-8.17E+55	0	-1.44E+57	0	-2.14E+58
0	-6.21E+53	0	-1.77E+55	0	-4.56E+56	0	-1.05E+58	0	-2.14E+59	0	-3.77E+60	0
-5.45E+55	0	-1.70E+57	0	-4.85E+58	0	-1.25E+60	0	-2.89E+61	0	-5.87E+62	0	-1.03E+64
0	-1.56E+59	0	-4.87E+60	0	-1.39E+62	0	-3.58E+63	0	-8.27E+64	0	-1.68E+66	0
-1.37E+61	0	-4.66E+62	0	-1.46E+64	0	-4.15E+65	0	-1.07E+67	0	-2.47E+68	0	-5.02E+69
0	-4.27E+64	0	-1.45E+66	0	-4.52E+67	0	-1.29E+69	0	-3.33E+70	0	-7.67E+71	0
-3.76E+66	0	-1.38E+68	0	-4.68E+69	0	-1.46E+71	0	-4.17E+72	0	-1.07E+74	0	-2.48E+75
0	-1.26E+70	0	-4.62E+71	0	-1.57E+73	0	-4.90E+74	0	-1.40E+76	0	-3.60E+77	0
-1.11E+72	0	-4.38E+73	0	-1.60E+75	0	-5.45E+76	0	-1.70E+78	0	-4.85E+79	0	-1.25E+81
0	-3.99E+75	0	-1.57E+77	0	-5.77E+78	0	-1.96E+80	0	-6.12E+81	0	-1.74E+83	0
-3.53E+77	0	-1.48E+79	0	-5.85E+80	0	-2.15E+82	0	-7.28E+83	0	-2.27E+85	0	-6.48E+86

Figure 419: Difference between Direct and Factorial Free radial polynomial for moment order up to N=30

0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
-3.28E+38	0	0	0	0	0	0	0	0	0	0	0	0
0	-5.80E+41	0	0	0	0	0	0	0	0	0	0	0
-2.85E+44	0	-1.10E+45	0	0	0	0	0	0	0	0	0	0
0	-5.72E+47	0	-2.20E+48	0	0	0	0	0	0	0	0	0
-1.83E+50	0	-1.22E+51	0	-4.70E+51	0	0	0	0	0	0	0	0
0	-4.13E+53	0	-2.75E+54	0	-1.06E+55	0	0	0	0	0	0	0
-1.04E+56	0	-9.82E+56	0	-6.54E+57	0	-2.52E+58	0	0	0	0	0	0
0	-2.61E+59	0	-2.45E+60	0	-1.63E+61	0	-6.29E+61	0	0	0	0	0
-5.62E+61	0	-6.83E+62	0	-6.43E+63	0	-4.28E+64	0	-1.65E+65	0	0	0	0
0	-1.54E+65	0	-1.87E+66	0	-1.76E+67	0	-1.17E+68	0	-4.52E+68	0	0	0
-2.96E+67	0	-4.41E+68	0	-5.36E+69	0	-5.05E+70	0	-3.36E+71	0	-1.29E+72	0	0
0	-8.85E+70	0	-1.32E+72	0	-1.60E+73	0	-1.51E+74	0	-1.00E+75	0	-3.86E+75	0
-1.56E+73	0	-2.75E+74	0	-4.09E+75	0	-4.98E+76	0	-4.69E+77	0	-3.12E+78	0	-1.20E+79
0	-5.04E+76	0	-8.88E+77	0	-1.32E+79	0	-1.61E+80	0	-1.51E+81	0	-1.01E+82	0
-8.31E+78	0	-1.69E+80	0	-2.98E+81	0	-4.43E+82	0	-5.39E+83	0	-5.07E+84	0	-3.38E+85
0	-2.89E+82	0	-5.87E+83	0	-1.03E+85	0	-1.54E+86	0	-1.87E+87	0	-1.76E+88	0
-4.50E+84	0	-1.04E+86	0	-2.11E+87	0	-3.72E+88	0	-5.54E+89	0	-6.73E+90	0	-6.34E+91
0	-1.67E+88	0	-3.86E+89	0	-7.85E+90	0	-1.38E+92	0	-2.06E+93	0	-2.50E+94	0

Figure 420: Difference between Direct and Factorial Free radial polynomial for moment order up to N=30

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
-3.88E+82	0	0	0	0
0	-1.30E+86	0	0	0
-1.17E+89	0	-4.52E+89	0	0
0	-4.22E+92	0	-1.62E+93	0
-2.36E+95	0	-1.57E+96	0	-6.04E+96

Figure 421: Difference between Direct and Factorial Free radial polynomial for moment order up to N=30

0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
6399.426	0	0	0	0	0	0	0	0	0	0	0	0
0	-6.28311	0	0	0	0	0	0	0	0	0	0	0
13948.17	0	-8.27379	0	0	0	0	0	0	0	0	0	0
0	-123.434	0	-42.3838	0	0	0	0	0	0	0	0	0
1549259	0	-9.0011	0	-778.116	0	0	0	0	0	0	0	0
0	-352.769	0	-52.609	0	253.8544	0	0	0	0	0	0	0
667935.6	0	-1812.44	0	-1730.78	0	-28.2201	0	0	0	0	0	0
0	-23391.4	0	-8671.03	0	566.2134	0	-50.381	0	0	0	0	0
44383443	0	1800.955	0	-131383	0	-62.3197	0	389.5581	0	0	0	0
0	-29105.5	0	3727.979	0	41000.57	0	-115.345	0	-202.128	0	0	0
19583168	0	-46824.8	0	-113023	0	-4522.68	0	847.0207	0	20.21872	0	0
0	-613336	0	-212459	0	36882.7	0	-7155.01	0	-443.923	0	39.98748	0
2.87E+08	0	43524.55	0	-3309983	0	-3562.09	0	57831.05	0	43.54798	0	-250.971
0	-746810	0	90115.98	0	1012977	0	-8720.32	0	-29634.4	0	83.05859	0
1.09E+08	0	-250536	0	-2868296	0	-109144	0	54011.05	0	2916.391	0	-542.401
0	-4191172	0	-1065540	0	920350	0	-173665	0	-27194.3	0	5163.669	0
4.75E+08	0	292465.1	0	-2.1E+07	0	-87796.1	0	1360598	0	2696.169	0	-35023.2
0	-4641930	0	836213.7	0	6444712	0	-211390	0	-688146	0	6177.907	0
2.67E+08	0	-188447	0	-1.7E+07	0	-674229	0	1293476	0	66881.23	0	-33323.4
0	-9431289	0	-244468	0	5436838	0	-1154584	0	-644970	0	119128.9	0
2.54E+08	0	471448	0	-4.1E+07	0	-503003	0	8473169	0	63381	0	-789394
0	-1.1E+07	0	2386343	0	12740266	0	-1307805	0	-4245567	0	143919.4	0
2.19E+08	0	392115	0	-3.7E+07	0	-1253919	0	7524550	0	409979.3	0	-769520
0	-1.1E+07	0	3575104	0	12256011	0	-2696484	0	-3729127	0	768432.5	0
1.07E+08	0	400881.1	0	-3.8E+07	0	-1109380	0	16897029	0	364004	0	-4812156
0	-1.2E+07	0	4557880	0	12746165	0	-3160115	0	-8411576	0	869256	0
7.12E+08	0	476616.3	0	-3.6E+07	0	-1112455	0	16799109	0	810402.7	0	-4413064
0	-1.1E+07	0	5614683	0	12685456	0	-3660502	0	-8334246	0	1769734	0
2.61E+08	0	-266677	0	-2.9E+07	0	-1070189	0	17647068	0	805866.1	0	-9654629

Figure 422: Difference between Direct and Hybrid radial polynomial for moment order up to N=30

0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
147.7221	0	0	0	0	0	0	0	0	0	0	0	0
0	-14.3607	0	0	0	0	0	0	0	0	0	0	0
322.9837	0	-32.8377	0	0	0	0	0	0	0	0	0	0
0	-30.5291	0	172.9579	0	0	0	0	0	0	0	0	0
20458.85	0	-64.1852	0	-111.623	0	0	0	0	0	0	0	0
0	-1939.13	0	374.063	0	10.58283	0	0	0	0	0	0	0
19376.61	0	-4044.58	0	-243.933	0	27.81944	0	0	0	0	0	0
0	-1903.69	0	23242.87	0	21.28155	0	-124.018	0	0	0	0	0
457320.3	0	-4653.15	0	-14894.3	0	55.52221	0	86.85208	0	0	0	0
0	-43054.5	0	22406.08	0	1326.758	0	-268.351	0	-9.37669	0	0	0
444456.1	0	-90047.8	0	-14392.6	0	3394.182	0	187.8521	0	-22.0177	0	0
0	-43465.2	0	508954.7	0	1401.561	0	-16227	0	-17.5391	0	91.37181	0
2772437	0	-105638	0	-324404	0	3694.053	0	11253.39	0	-44.7804	0	-69.7961
0	-261747	0	504382	0	28870.98	0	-15814.3	0	-1107.98	0	197.9032	0
2539774	0	-565432	0	-322292	0	73660.07	0	11061.11	0	-2643.77	0	-149.423
0	-249343	0	3054945	0	31336.64	0	-348027	0	-1220.43	0	11708.84	0

Figure 423: Difference between Direct and Hybrid radial polynomial for moment order up to N=30

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
9.504925	0	0	0	0
0	16.5798	0	0	0
16.99389	0	-68.5391	0	0

Figure 424: Difference between Direct and Hybrid radial polynomial for moment order up to N=30

0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
6399.426	0	0	0	0	0	0	0	0	0	0	0	0
0	-6.28311	0	0	0	0	0	0	0	0	0	0	0
13948.17	0	-8.27379	0	0	0	0	0	0	0	0	0	0
0	-123.434	0	-42.3838	0	0	0	0	0	0	0	0	0
1549259	0	-9.0011	0	-778.116	0	0	0	0	0	0	0	0
0	-352.769	0	-52.609	0	253.8544	0	0	0	0	0	0	0
667935.6	0	-1812.44	0	-1730.78	0	-28.2201	0	0	0	0	0	0
0	-23391.4	0	-8671.03	0	566.2134	0	-50.381	0	0	0	0	0
44383443	0	1800.955	0	-131383	0	-62.3197	0	389.5581	0	0	0	0
0	-29105.5	0	3727.979	0	41000.57	0	-115.345	0	-202.128	0	0	0
19583168	0	-46824.8	0	-113023	0	-4522.68	0	847.0207	0	20.21872	0	0
0	-613336	0	-212459	0	36882.7	0	-7155.01	0	-443.923	0	39.98748	0
2.87E+08	0	43524.55	0	-3309983	0	-3562.09	0	57831.05	0	43.54798	0	-250.971
0	-746810	0	90115.98	0	1012977	0	-8720.32	0	-29634.4	0	83.05859	0
1.09E+08	0	-250536	0	-2868296	0	-109144	0	54011.05	0	2916.391	0	-542.401
0	-4191172	0	-1065540	0	920350	0	-173665	0	-27194.3	0	5163.669	0
4.75E+08	0	292465.1	0	-2.1E+07	0	-87796.1	0	1360598	0	2696.169	0	-35023.2
0	-4641930	0	836213.7	0	6444712	0	-211390	0	-688146	0	6177.907	0
2.67E+08	0	-188447	0	-1.7E+07	0	-674229	0	1293476	0	66881.23	0	-33323.4
0	-9431289	0	-244468	0	5436838	0	-1154584	0	-644970	0	119128.9	0
2.54E+08	0	471448	0	-4.1E+07	0	-503003	0	8473169	0	63381	0	-789394
0	-1.1E+07	0	2386343	0	12740266	0	-1307805	0	-4245567	0	143919.4	0
2.19E+08	0	392115	0	-3.7E+07	0	-1253919	0	7524550	0	409979.3	0	-769520
0	-1.1E+07	0	3575104	0	12256011	0	-2696484	0	-3729127	0	768432.5	0
1.07E+08	0	400881.1	0	-3.8E+07	0	-1109380	0	16897029	0	364004	0	-4812156
0	-1.2E+07	0	4557880	0	12746165	0	-3160115	0	-8411576	0	869256	0
7.12E+08	0	476616.3	0	-3.6E+07	0	-1112455	0	16799109	0	810402.7	0	-4413064
0	-1.1E+07	0	5614683	0	12685456	0	-3660502	0	-8334246	0	1769734	0
2.61E+08	0	-266677	0	-2.9E+07	0	-1070189	0	17647068	0	805866.1	0	-9654629

Figure 425: Difference between Direct and Kintner radial polynomial for moment order up to N=30

0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
147.7221	0	0	0	0	0	0	0	0	0	0	0	0
0	-14.3607	0	0	0	0	0	0	0	0	0	0	0
322.9837	0	-32.8377	0	0	0	0	0	0	0	0	0	0
0	-30.5291	0	172.9579	0	0	0	0	0	0	0	0	0
20458.85	0	-64.1852	0	-111.623	0	0	0	0	0	0	0	0
0	-1939.13	0	374.063	0	10.58283	0	0	0	0	0	0	0
19376.61	0	-4044.58	0	-243.933	0	27.81944	0	0	0	0	0	0
0	-1903.69	0	23242.87	0	21.28155	0	-124.018	0	0	0	0	0
457320.3	0	-4653.15	0	-14894.3	0	55.52221	0	86.85208	0	0	0	0
0	-43054.5	0	22406.08	0	1326.758	0	-268.351	0	-9.37669	0	0	0
444456.1	0	-90047.8	0	-14392.6	0	3394.182	0	187.8521	0	-22.0177	0	0
0	-43465.2	0	508954.7	0	1401.561	0	-16227	0	-17.5391	0	91.37181	0
2772437	0	-105638	0	-324404	0	3694.053	0	11253.39	0	-44.7804	0	-69.7961
0	-261747	0	504382	0	28870.98	0	-15814.3	0	-1107.98	0	197.9032	0
2539774	0	-565432	0	-322292	0	73660.07	0	11061.11	0	-2643.77	0	-149.423
0	-249343	0	3054945	0	31336.64	0	-348027	0	-1220.43	0	11708.84	0

Figure 426: Difference between Direct and Kintner radial polynomial for moment order up to N=30

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
9.504925	0	0	0	0
0	16.5798	0	0	0
16.99389	0	-68.5391	0	0

Figure 427: Difference between Direct and Kintner radial polynomial for moment order up to N=30

0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
-10605.1	0	0	0	0	0	0	0	0	0	0	0	0
0	-6.28311	0	0	0	0	0	0	0	0	0	0	0
-810995	0	18.14311	0	0	0	0	0	0	0	0	0	0
0	231.1287	0	92.88624	0	0	0	0	0	0	0	0	0
-5.3E+07	0	1025.813	0	1347.225	0	0	0	0	0	0	0	0
0	13741.48	0	5076.03	0	-440.37	0	0	0	0	0	0	0
-3.1E+09	0	61383.88	0	78883.51	0	51.47881	0	0	0	0	0	0
0	794047.7	0	293866	0	-25071.5	0	78.62718	0	0	0	0	0
-1.8E+11	0	3468629	0	4457334	0	2806.874	0	-685.983	0	0	0	0
0	44061537	0	16306857	0	-1391090	0	4504.136	0	361.7583	0	0	0
-9.6E+12	0	1.9E+08	0	2.44E+08	0	153487.8	0	-36774.3	0	-36.2904	0	0
0	2.38E+09	0	8.8E+08	0	-7.5E+07	0	242849.5	0	19021.61	0	-61.2541	0
-5.1E+14	0	1.01E+10	0	1.3E+10	0	8189359	0	-1961531	0	-1886.64	0	449.5875
0	1.26E+11	0	4.65E+10	0	-4E+09	0	12833620	0	1005360	0	-3370.91	0
-2.70E+16	0	5.3E+11	0	6.81E+11	0	4.29E+08	0	-1E+08	0	-98840.1	0	22986.36
0	6.54E+12	0	2.42E+12	0	-2.1E+11	0	6.68E+08	0	52300573	0	-175064	0
-1.39E+18	0	2.74E+13	0	3.52E+13	0	2.22E+10	0	-5.3E+09	0	-5108387	0	1187761
0	3.36E+14	0	1.24E+14	0	-1.1E+13	0	3.43E+10	0	2.69E+09	0	-8994703	0
-7.12E+19	0	1.40E+15	0	1.80E+15	0	1.13E+12	0	-2.7E+11	0	-2.6E+08	0	60705845
0	1.71E+16	0	6.32E+15	0	-5.4E+14	0	1.74E+12	0	1.37E+11	0	-4.6E+08	0
-3.61E+21	0	7.09E+16	0	9.11E+16	0	5.74E+13	0	-1.4E+13	0	-1.3E+10	0	3.07E+09
0	8.62E+17	0	3.19E+17	0	-2.72E+16	0	8.8E+13	0	6.9E+12	0	-2.3E+10	0
-1.81E+23	0	3.56E+18	0	4.58E+18	0	2.89E+15	0	-6.9E+14	0	-6.6E+11	0	1.55E+11
0	4.32E+19	0	1.60E+19	0	-1.36E+18	0	4.41E+15	0	3.45E+14	0	-1.2E+12	0
-9.05E+24	0	1.78E+20	0	2.29E+20	0	1.44E+17	0	-3.45E+16	0	-3.3E+13	0	7.72E+12
0	2.15E+21	0	7.95E+20	0	-6.79E+19	0	2.20E+17	0	1.72E+16	0	-5.8E+13	0
-4.50E+26	0	8.84E+21	0	1.14E+22	0	7.16E+18	0	-1.71E+18	0	-1.65E+15	0	3.83E+14
0	1.06E+23	0	3.94E+22	0	-3.36E+21	0	1.09E+19	0	8.52E+17	0	-2.85E+15	0
-2.22E+28	0	4.37E+23	0	5.62E+23	0	3.54E+20	0	-8.47E+19	0	-8.15E+16	0	1.89E+16

Figure 428: Difference between Direct and Modified Prata radial polynomial for moment order up to N=30

0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
-267.049	0	0	0	0	0	0	0	0	0	0	0	0
0	25.34116	0	0	0	0	0	0	0	0	0	0	0
-13505	0	50.50772	0	0	0	0	0	0	0	0	0	0
0	1286.131	0	-313.224	0	0	0	0	0	0	0	0	0
-693975	0	2698.443	0	202.748	0	0	0	0	0	0	0	0
0	65698.93	0	-15564.2	0	-17.5467	0	0	0	0	0	0	0
-3.5E+07	0	136938.1	0	10014.41	0	-44.6431	0	0	0	0	0	0
0	3327470	0	-788214	0	-895.435	0	225.81	0	0	0	0	0
-1.8E+09	0	6908172	0	505292.5	0	-2299.07	0	-157.275	0	0	0	0
0	1.67E+08	0	-4E+07	0	-44941.7	0	11017.8	0	14.71032	0	0	0
-8.9E+10	0	3.46E+08	0	25312494	0	-115009	0	-7664.39	0	35.86585	0	0
0	8.35E+09	0	-2E+09	0	-2244501	0	550248.8	0	757.791	0	-166.629	0
-4.4E+12	0	1.72E+10	0	1.26E+09	0	-5727386	0	-381702	0	1811.337	0	126.0101
0	4.15E+11	0	-9.8E+10	0	-1.1E+08	0	27329310	0	37538.84	0	-8031.66	0
-2.2E+14	0	8.54E+11	0	6.25E+10	0	-2.8E+08	0	-1.9E+07	0	89678.56	0	6083.072
0	2.05E+13	0	-4.9E+12	0	-5.5E+09	0	1.35E+09	0	1855605	0	-397052	0

Figure 429: Difference between Direct and Modified Prata radial polynomial for moment order up to N=30

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
-14.8345	0	0	0	0
0	-26.4691	0	0	0
-758.497	0	124.4966	0	0

Figure 430: Difference between Direct and Modified Prata radial polynomial for moment order up to N=30

0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	-6.28311	0	0	0	0	0	0	0	0	0	0	0
0	0	-1.62307	0	0	0	0	0	0	0	0	0	0
0	-15.0505	0	-1.03412	0	0	0	0	0	0	0	0	0
0	0	5.455355	0	-425.642	0	0	0	0	0	0	0	0
0	-107.49	0	44.27042	0	263.081	0	0	0	0	0	0	0
0	0	-11.8157	0	979.7459	0	-97.3304	0	0	0	0	0	0
0	-391.614	0	181.2372	0	167.5798	0	-240.7	0	0	0	0	0
0	0	19.47507	0	-1800.69	0	335.1225	0	6089.389	0	0	0	0
0	-1389.28	0	849.2924	0	2137.989	0	-201.281	0	-4706.24	0	0	0
0	0	-26.5298	0	3001.576	0	-869.204	0	-25660.6	0	1518.119	0	0
0	-4859.28	0	3253.218	0	4587.012	0	-2991.21	0	8226.836	0	3730.845	0
0	0	31.84716	0	-4668.92	0	1905.229	0	80987.67	0	-7927.5	0	-84902.6
0	-17124.5	0	12435.03	0	20427.51	0	-6933.4	0	-60586.5	0	-6006.33	0
0	0	-36.098	0	6851.729	0	-3729.53	0	-213373	0	30264.75	0	526363.8
0	-60399.7	0	46280.8	0	64461.17	0	-36745.1	0	46641.74	0	64804.47	0
0	0	40.45396	0	-9605.5	0	6724.635	0	494822.3	0	-94506.9	0	-2357628
0	-214386	0	171164.3	0	236851	0	-122322	0	-485317	0	-35848.2	0
0	0	-44.7234	0	13007.38	0	-11373	0	-1042249	0	255810	0	8535441
0	-765383	0	629903.6	0	825947.3	0	-495268	0	-277177	0	690734.4	0
0	0	48.06299	0	-17136.5	0	18253.98	0	2035638	0	-621378	0	-2.6E+07
0	-2748497	0	2316335	0	2967310	0	-1817423	0	-3980751	0	585346.3	0
0	0	-50.7819	0	22057.37	0	-28064.4	0	-3740512	0	1385652	0	73127816
0	-9924576	0	8521851	0	10636691	0	-6915844	0	-8656906	0	7080451	0
0	0	53.97849	0	-27829.7	0	41655.01	0	6535076	0	-2882191	0	-1.8E+08
0	-3.6E+07	0	31400820	0	38470936	0	-2.6E+07	0	-4.2E+07	0	16826969	0
0	0	-58.0162	0	34522.93	0	-60036.8	0	-1.1E+07	0	5657537	0	4.28E+08
0	-1.3E+08	0	1.16E+08	0	1.4E+08	0	-9.7E+07	0	-1.3E+08	0	85093018	0
0	0	62.29195	0	-42210.9	0	84367.44	0	17669013	0	-1.1E+07	0	-9.3E+08

Figure 431: Difference between Direct and Prata radial polynomial for moment order up to N=30

0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
66204.34	0	0	0	0	0	0	0	0	0	0	0	0
0	-21180.2	0	0	0	0	0	0	0	0	0	0	0
-252425	0	-57025.4	0	0	0	0	0	0	0	0	0	0
0	151418.9	0	1140833	0	0	0	0	0	0	0	0	0
1586880	0	226758.4	0	-914130	0	0	0	0	0	0	0	0
0	-774662	0	-9331912	0	278807.8	0	0	0	0	0	0	0
-4231575	0	-1682933	0	5343088	0	889661.8	0	0	0	0	0	0
0	3175541	0	54113601	0	-2516353	0	-1.5E+07	0	0	0	0	0
18122467	0	4713658	0	-3.7E+07	0	-5652311	0	12588583	0	0	0	0
0	-1.1E+07	0	-2.5E+08	0	16052394	0	1.54E+08	0	-4317472	0	0	0
-3.2E+07	0	-2.4E+07	0	1.5E+08	0	43277586	0	-9.9E+07	0	-1.2E+07	0	0
0	34124872	0	9.71E+08	0	-8.1E+07	0	-1.1E+09	0	47252510	0	2E+08	0
1.43E+08	0	43814962	0	-6.5E+08	0	-1.9E+08	0	7.84E+08	0	1.06E+08	0	-1.8E+08
0	-9.5E+07	0	-3.3E+09	0	3.45E+08	0	6.12E+09	0	-3.6E+08	0	-2.4E+09	0
-1.1E+08	0	-2.2E+08	0	1.98E+09	0	9.05E+08	0	-4.1E+09	0	-9.1E+08	0	1.73E+09
0	2.44E+08	0	1.02E+10	0	-1.3E+09	0	-2.9E+10	0	2.18E+09	0	2.04E+10	0

Figure 432: Difference between Direct and Prata radial polynomial for moment order up to N=30

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
77678446	0	0	0	0
0	1.61E+08	0	0	0
-1E+09	0	-2.6E+09	0	0

Figure 433: Difference between Direct and Prata radial polynomial for moment order up to N=30

0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
6399.426	0	0	0	0	0	0	0	0	0	0	0	0
0	-6.28311	0	0	0	0	0	0	0	0	0	0	0
13948.17	0	-8.27379	0	0	0	0	0	0	0	0	0	0
0	-123.434	0	-42.3838	0	0	0	0	0	0	0	0	0
1549259	0	-9.0011	0	-778.116	0	0	0	0	0	0	0	0
0	-352.769	0	-52.609	0	253.8544	0	0	0	0	0	0	0
667935.6	0	-1812.44	0	-1730.78	0	-28.2201	0	0	0	0	0	0
0	-23391.4	0	-8671.03	0	566.2134	0	-50.381	0	0	0	0	0
44383443	0	1800.955	0	-131383	0	-62.3197	0	389.5581	0	0	0	0
0	-29105.5	0	3727.979	0	41000.57	0	-115.345	0	-202.128	0	0	0
19583168	0	-46824.8	0	-113023	0	-4522.68	0	847.0207	0	20.21872	0	0
0	-613336	0	-212459	0	36882.7	0	-7155.01	0	-443.923	0	39.98748	0
2.87E+08	0	43524.55	0	-3309983	0	-3562.09	0	57831.05	0	43.54798	0	-250.971
0	-746810	0	90115.98	0	1012977	0	-8720.32	0	-29634.4	0	83.05859	0
1.09E+08	0	-250536	0	-2868296	0	-109144	0	54011.05	0	2916.391	0	-542.401
0	-4191172	0	-1065540	0	920350	0	-173665	0	-27194.3	0	5163.669	0
4.75E+08	0	292465.1	0	-2.1E+07	0	-87796.1	0	1360598	0	2696.169	0	-35023.2
0	-4641930	0	836213.7	0	6444712	0	-211390	0	-688146	0	6177.907	0
2.67E+08	0	-188447	0	-1.7E+07	0	-674229	0	1293476	0	66881.23	0	-33323.4
0	-9431289	0	-244468	0	5436838	0	-1154584	0	-644970	0	119128.9	0
2.54E+08	0	471448	0	-4.1E+07	0	-503003	0	8473169	0	63381	0	-789394
0	-1.1E+07	0	2386343	0	12740266	0	-1307805	0	-4245567	0	143919.4	0
2.19E+08	0	392115	0	-3.7E+07	0	-1253919	0	7524550	0	409979.3	0	-769520
0	-1.1E+07	0	3575104	0	12256011	0	-2696484	0	-3729127	0	768432.5	0
1.07E+08	0	400881.1	0	-3.8E+07	0	-1109380	0	16897029	0	364004	0	-4812156
0	-1.2E+07	0	4557880	0	12746165	0	-3160115	0	-8411576	0	869256	0
7.12E+08	0	476616.3	0	-3.6E+07	0	-1112455	0	16799109	0	810402.7	0	-4413064
0	-1.1E+07	0	5614683	0	12685456	0	-3660502	0	-8334246	0	1769734	0
2.61E+08	0	-266677	0	-2.9E+07	0	-1070189	0	17647068	0	805866.1	0	-9654629

Figure 434: Difference between Direct and Q-theta radial polynomial for moment order up to N=30

0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
147.7221	0	0	0	0	0	0	0	0	0	0	0	0
0	-14.3607	0	0	0	0	0	0	0	0	0	0	0
322.9837	0	-32.8377	0	0	0	0	0	0	0	0	0	0
0	-30.5291	0	172.9579	0	0	0	0	0	0	0	0	0
20458.85	0	-64.1852	0	-111.623	0	0	0	0	0	0	0	0
0	-1939.13	0	374.063	0	10.58283	0	0	0	0	0	0	0
19376.61	0	-4044.58	0	-243.933	0	27.81944	0	0	0	0	0	0
0	-1903.69	0	23242.87	0	21.28155	0	-124.018	0	0	0	0	0
457320.3	0	-4653.15	0	-14894.3	0	55.52221	0	86.85208	0	0	0	0
0	-43054.5	0	22406.08	0	1326.758	0	-268.351	0	-9.37669	0	0	0
444456.1	0	-90047.8	0	-14392.6	0	3394.182	0	187.8521	0	-22.0177	0	0
0	-43465.2	0	508954.7	0	1401.561	0	-16227	0	-17.5391	0	91.37181	0
2772437	0	-105638	0	-324404	0	3694.053	0	11253.39	0	-44.7804	0	-69.7961
0	-261747	0	504382	0	28870.98	0	-15814.3	0	-1107.98	0	197.9032	0
2539774	0	-565432	0	-322292	0	73660.07	0	11061.11	0	-2643.77	0	-149.423
0	-249343	0	3054945	0	31336.64	0	-348027	0	-1220.43	0	11708.84	0

Figure 435: Difference between Direct and Q-theta radial polynomial for moment order up to N=30

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
9.504925	0	0	0	0
0	16.5798	0	0	0
16.99389	0	-68.5391	0	0

Figure 436: Difference between Direct and Q-theta radial polynomial for moment order up to N=30