NON-LINEAR ANALYSIS OF CAMERA CALIBRATION

A Project

Presented to the faculty of the Department of Electrical and Electronic Engineering

California State University, Sacramento

Submitted in partial satisfaction of
the requirements for the degree of

MASTER OF SCIENCE

in

Electrical and Electronic Engineering

by

Greeshma Shailendra

FALL
2018

NON-LINEAR ANALYSIS OF CAMERA CALIBRATION

A Project

by

Greeshma Shailendra

Approved by:

_____, Committee Chair
Dr. Fethi Belkhouche

_____, Second Reader
Dr. Preetham B Kumar

_____
Date

Student: Greeshma Shailendra

I certify that this student has met the requirements for format contained in the University format manual, and this project is suitable for shelving in the Library and credit is to be awarded to the project.

_____, Graduate Coordinator _____
Dr. Preetham B Kumar Date

Department of Electrical and Electronic Engineering

Abstract

of

NON-LINEAR ANALYSIS OF CAMERA CALIBRATION MODEL

by

Greeshma Shailendra

Camera calibration which is also described as camera re-sectioning, involves approximation of various parameters between the lens and the image sensor of a camera. These variables serve different purposes, which include removal of lens distorsion, estimating the distance between the image and the lens which in turn aids to locate the camera position, or determine the size of an object.

The pinhole camera model focuses on the linear mapping of world co-ordinates with image co-ordinates. This project aims to shed light on the significance of non-linear relation between the two-dimensional (2D) image plane and the three-dimensional (3D) object.

_____, Committee Chair
Dr. Fethi Belkhouche

_____
Date

ACKNOWLEDGEMENTS

I would like to sincerely thank Dr. Fethi Belkhouche for guiding through the project with great patience and care. I also thank my graduate coordinator, Dr. Preetham B Kumar for his continuous encouragement and time. I would like to extend my heartfelt appreciation to the entire faculty of the Department of Electrical and Electronics Engineering for the help provided during the program.

I would like to express my profound gratitude to my parents and family for their never-ending support in all my decisions and for motivating me through all the tough choices.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTON

Cameras have been around for a few centuries now. They have undergone immense transformation over the time in all aspects. It is a well-known fact that the initial versions were not only bulky and expensive, but also complicated and difficult to use at times.

However, the 20th century marked a turning point in the field with the invention of new types of cameras which became common in many applications. Today, these devices have outstretched their purposes through contributions in the field of 3D reconstruction, navigation systems and robotics. Unfortunately, every advancement in technology comes with a trade-off. These cameras have significant distorsion, but methods have been developed to cope with it.

It is a simple fact that we live in a 3-dimensional world. When we measure the world, we tend to do it using three co-ordinate locations represented in general as x, y, and z. But when we capture an image, the 3D world becomes a 2D image plane. That plane can be a camera lens or even our eyes. In short, we only have a 2D description of the world and lose the third dimension in the process.

The main purpose of the project is to show how distortions can be further reduced if the mapping between the image and the world co-ordinates is non-linear. This implies that instead of the first order approximation, multiple orders will be considered in the estimation.

Our orientation in the world reference frame is important when we take a picture, as the image at that point projects to different points in the image plane. The geometric models allow us with a lot of intuition to understand how a camera is posed with respect to the environment and the position of the objects in the world. Not only can we infer the geometry of the scene but also how the camera man is looking into the scene.

Consider a simple projection of 3D to 2D (Figure 1). Clipping planes (CL) come in to picture when you want to ignore the obstructions like walls and furniture between the object and the image plane. The far CL does not include obstacles on the projection plane whereas the near CL includes them. But in either case, a 3D to 2D image is obtained on the projection plane of a camera.

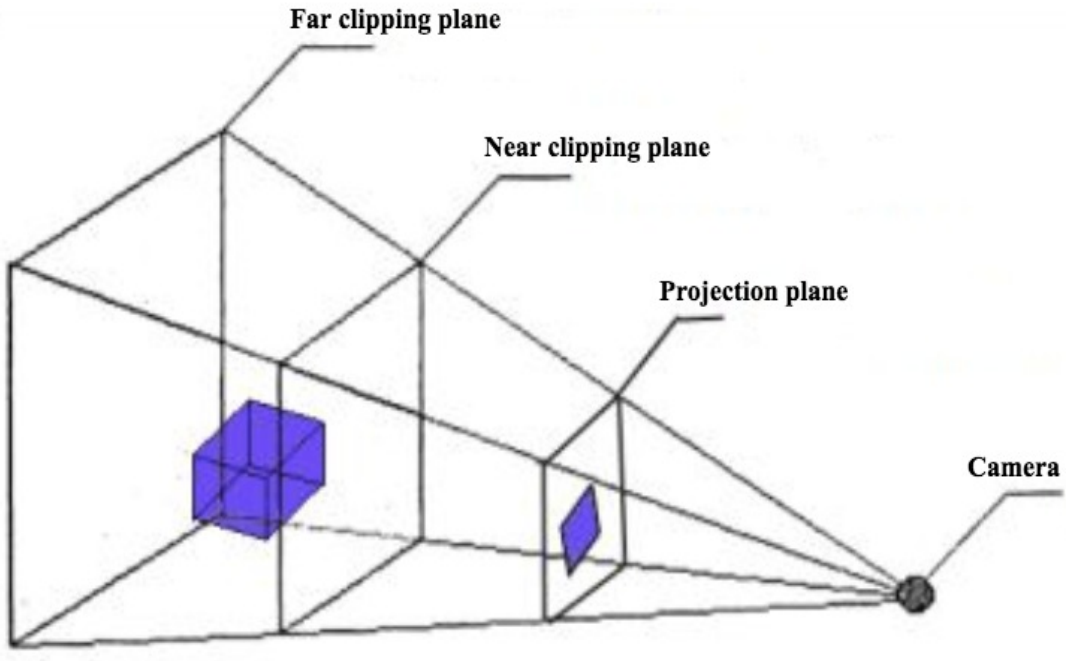

Figure 1: A simple 3D to 2D projection

2. BACKGROUND

A simple pinhole camera comes without a lens and functions on a small aperture, also referred to as a pinhole. It is an effective light-proof box with a tiny opening on one side. The light from a scene goes through the opening and projects an image on the other side of the box. The pinhole camera model had been utilized broadly to depicts the scientific connection between the co-ordinates of a 3D point and its projection onto to the picture plane of a perfect pinhole camera, where the camera opening is portrayed as a point and no focal points are utilized to concentrate light.

The model does exclude, for instance, geometric bends or obscuring of unfocused items caused by focal points and limited measured openings. Additionally, it does not consider that most cameras have discrete image points. This implies the pinhole camera model must be utilized as a first order estimate of the mapping from a 3D scene to a 2D picture. Its credibility depends on the type of the camera. The accuracy diminishes from the focal point of the picture to the edges with an increase in distortion.

Figure 2: Distortion caused by pinhole camera

This distortion can be observed in the picture (Figure 2). A portion of the outcomes that the pinhole camera display does not consider can be redressed, for instance by applying reasonable co-ordinate transforms on the image points; other effects are adequately minor to be ignored if a high specification camera is utilized. This implies the pinhole camera model can be utilized often as a sensible portrayal of how a camera delineates a 3D scene, for instance in PC vision and PC graphics. This results in alleviated distortion (Figure 3).

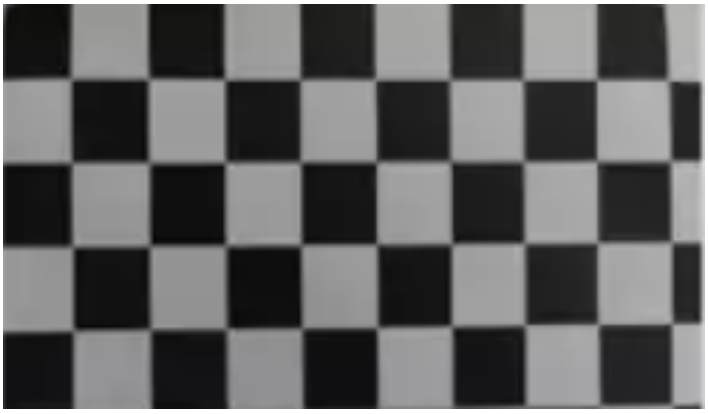

Figure 3: After correcting for distortion

3. CAMERA CALIBRATION

This is a process of characterising the camera through various parameters like focal length, image center, image distortion and even the position of the camera with respect to a fixed reference frame. The x, y, z variables are often used to represent a 3D scene. Also, u and v are used to represent 2D pixel points.

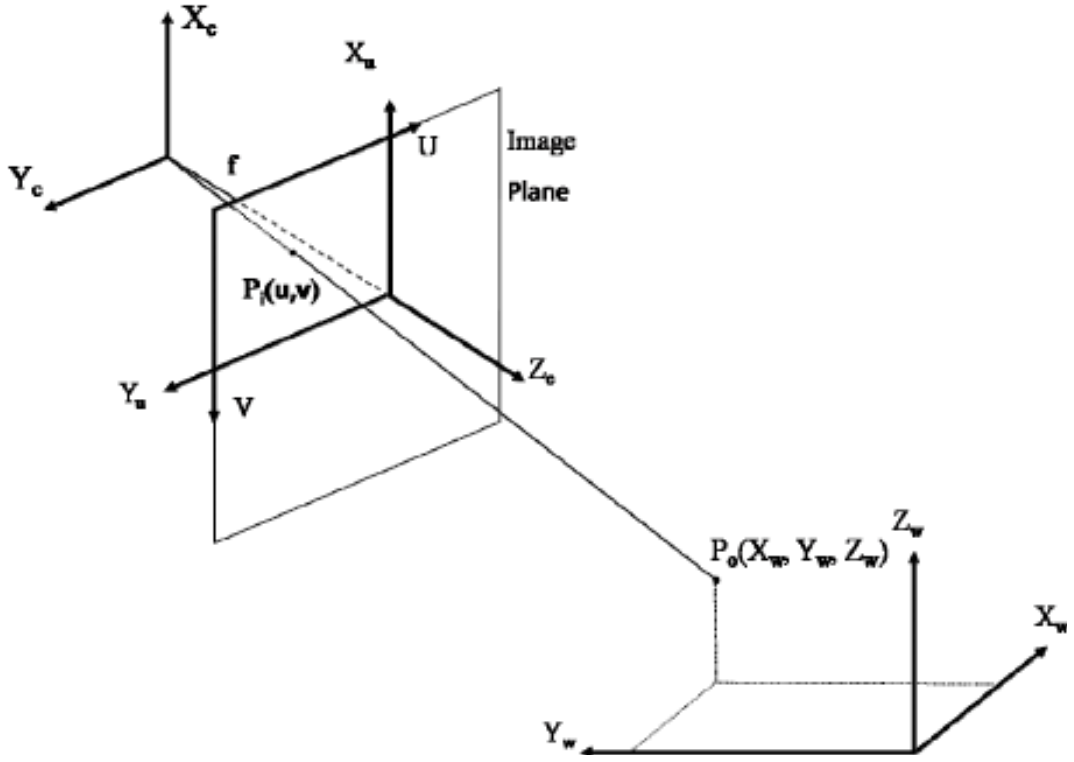

Figure 4: Mapping of 3D to 2D coordinates

Consider the projection point 'P' (Figure 4). In a pinhole camera model, the projection is usually a 3x4 matrix. If we multiply the projection matrix with the world coordinates, we get a new matrix as seen in equation (1).

If we have the world coordinates of a point and the matrix 'P', we can immediately find the unknowns 'u' and 'v'. If we have the values of u and v along with projection matrix, it is still not possible to find the world coordinates x, y, z because of the unknown depth ' d '. Instead, we obtain a range space through the center projection and the pixel. But in both cases, we need the projection matrix 'P'. This matrix depends on the camera parameters, focal length 'f' measured in pixels, image center (u0, v0) measured in pixels and parameters with respect to the outside world what we call as extrinsic parameters 'R' and 'T'. They will be discussed further in later chapters.

$$d\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix} [\, R \quad T \,] \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \mathrm{P} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{1}$$

In addition to these, we might encounter some parameters which cannot be modelled linearly in algebraic equations. This happens when we have lenses similar to fish eyes which gather a very big filter view. This leads to distorsion. The image of the outside world which appears to be nice and straight to our naked eye will appear curved in the picture. All these curved lines will be distorted with respect to the center. This is called radial distorsion which means that the pixel point is distorted proportionally to the radius from the center. This can be modelled with a polynomial as seen in equation (2). Radius 'r' can be easily found as the square root of u^2 and v^2. The unknowns k1, k2,.. are the radial distorsion parameters which we have to find through a procedure called calibration.

$$u^{dist} = u(1 + k_1 w + k_2 w^2 + ..) \tag{3}$$

$$v^{dist} = v(1 + k_1 w + k_2 w^2 + ..) \quad (4)$$

Where

$$w^2 = u^2 + v^2 \quad (5)$$

The calibration estimates the intrinsic parameters of a camera (f, u_0, v_0, k_1, k_2,..). In the past when there were no computers, they were calculated based on the readings of the specifications of a camera. These intrinsic parameters as opposed to the extrinsic parameters will really depend on where is the camera with respect to a fixed frame. What is really required is to remove radial distorsion from the image through calibration as it distracts us from the linearity of the equation [1].

To perform this calibration, we need to compute the intrinsic and extrinsic parameters with some world coordinates of some points in the world and the corresponding image coordinates. After we find the rotation matrix, the translation vector and the intrinsic 'K' and having removed the radial distortion, then the projection of rays in the world can be found, which can help to compute many other factors such as the motion of the camera or regulation of a point.

3.1 Geometric camera parameters

In a geometric model, all the distances are measured with respect to the camera's reference frame and the image coordinates have their origin at the principle point. It can be inferred that the world and the pixel coordinate system are related by a set of

physical parameters such as the focal length, pixel size, principle point and the camera position.

Let us consider a pinhole camera model with x, y, z world coordinates and u, v image plane coordinates. The perceptive projection is expressed relative to a positive number of ‘k’ and focal length ‘λ’ as:

$$K\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} u \\ v \\ \lambda \end{bmatrix} \tag{6}$$

Which can be written as

$$Kz = \lambda$$

$$K = \frac{\lambda}{z} \tag{7}$$

therefore,

$$u = \frac{\lambda x}{z} \text{ and } v = \frac{\lambda y}{z} \tag{8}$$

Equation (6) is called the perceptive projection. The pixel coordinates of images can be used to reconstruct a 3D environment with the help of two types of camera parameters as follows [1].

3.2 Intrinsic camera parameters

These are the internal factors which link the pixel coordinates of an image with the camera coordinates. This specifically represents the image plane coordinates (2D) and pixel coordinates (2D). To map image plane points (u, v) to a 3D scene, we use homogeneous coordinates notation as

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A_0 \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{9}$$

$$A_0 = \begin{bmatrix} \frac{\lambda}{z} & 0 & 0 \\ 0 & \frac{\lambda}{z} & 0 \\ 0 & 0 & \frac{\lambda}{z} \end{bmatrix} \tag{10}$$

In addition, the first-degree approximation of the pixel coordinate system (r, c) can also be defined similarly as,

$$r = u_0 + u \tag{11}$$

$$c = v_0 + v \tag{12}$$

$$r = u_0 + \frac{\lambda x}{z} \tag{13}$$

$$c = v_0 + \frac{\lambda y}{z} \tag{14}$$

where (u_0, v_0) are the image plane origin points. They can be represented in matrix format as,

$$\begin{bmatrix} r \\ c \\ 1 \end{bmatrix} = \begin{bmatrix} \lambda/z & 0 & u_0/z \\ 0 & \lambda/z & v_0/z \\ 0 & 0 & \lambda/z \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{15}$$

To express all the point locations in world units to pixel units, the image plane coordinates are divided by the pixel size. Let us consider s_x and s_y as the horizontal and vertical pixel dimension. We obtain,

$$r = u_0 + \frac{u}{s_x} \tag{16}$$

$$c = v_0 + \frac{v}{s_y} \tag{17}$$

$$u = s_x (r - u_0) \tag{18}$$

$$v = s_x (r - u_0) \tag{19}$$

3.3 Extrinsic camera parameters

These are the external factors of a camera (such as position and orientation) which undergo changes keeping the outside world as a reference frame. To find these entities, we will further need to determine the translation vector (T) with respect to the two reference frames and rotation matrix (R) which aligns the axes of the two frames onto each other.

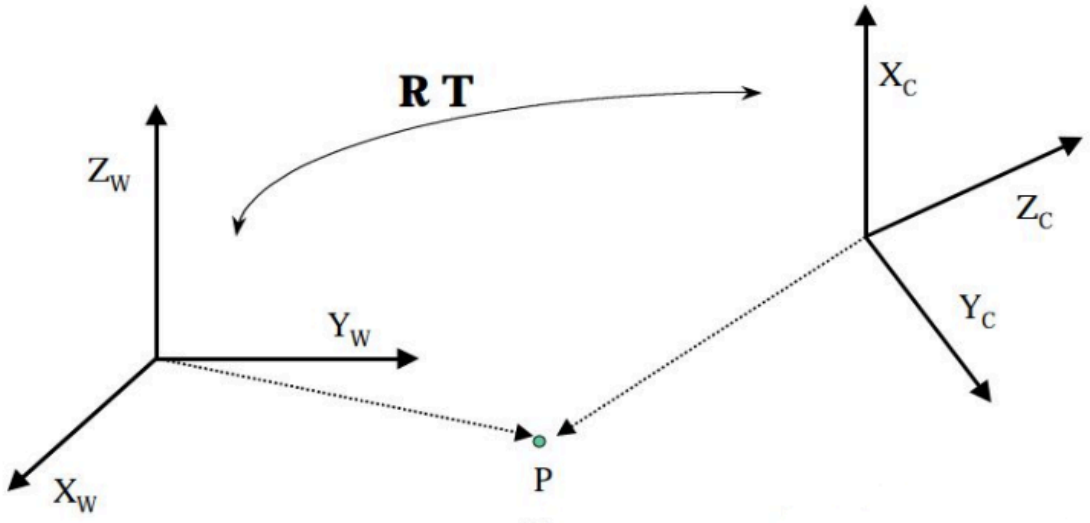

Figure 5: Translation and rotation matrix relative to two reference frames

Consider a point 'P' with respect to the world coordinates 'p_w' and camera coordinates 'p_c' . The rotation matrix is always a 3X3 orthogonal matrix and the translation matrix is 3X1 matrix. The relationship of point 'P' with respect to the world and camera reference frame is given by,

$$p_c = R\, p_w + T \tag{20}$$

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = R \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + T \tag{21}$$

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + \begin{bmatrix} T_{11} \\ T_{21} \\ T_{31} \end{bmatrix} \quad (22)$$

therefore,

$$x_c = R_1^T (p_w + T) \quad (23)$$

$$y_c = R_2^T (p_w + T) \quad (24)$$

$$z_c = R_3^T (p_w + T) \quad (25)$$

For example, to find the translation matrix 'T', we consider that the world coordinate system 'p_w' is zero and apply this assumption to equation (20). This makes the translation matrix equal to the camera coordinates. This implies that the 'p_c' vector starts from camera point and goes to the origin of the world point.

$$p_c = R(0) + T \quad (26)$$

$$p_c = T \quad (27)$$

Similarly, if we assume T=0 and replace p_w = [1 0 0] for the x-axis of the world, then the camera coordinate is equal to the first row of rotation matrix 'R'. This implies that the row 1 of rotation matrix is the x-axis of the world with respect to camera. The same method can be repeated for y-axis and z-axis.

4. LEAST SQUARES ESTIMATION

Statistical estimation is a crucial part of perceptive projection theory. Several methods are available to perform this function. The least square method is one of the many published methodologies [2] [5]. Many other procedures require entire description of the problem factors like the joint probability function, but the least squares technique just requires variances, covariances and mean values.

The least squares method falls into the category of regression analysis. Regression analysis is an arrangement of factual procedures for evaluating the connections between variables. It incorporates numerous methods for demonstrating and investigating several factors, when the attention is on the relationship between a variable and at least one autonomous variable.

Consider a problem which has many more sets of equations than the unknown variables. These are called overdetermined systems. By using least squares to solve the problem, it reduces the overall mean of the residuals from each of the equation. There are two classifications of this method: linear and non-linear least squares. As the name suggests, the linear method handles a regression model which consists of only first order parameters [6]. Non-linear systems are commonly solved through a repetitive process and at the end of each iteration, estimation of the system is done linearly.

For linear cases,

$$f(x, \alpha) = \sum_{i=1}^{n} \alpha_i \; A_i(x) \tag{28}$$

where, A_i is a function of x and

$$\alpha = (X^T X)^{-1} X^T y \tag{29}$$

For non-linear cases,

$$f(x_i, \alpha) = f^k(x_i, \alpha) + \sum_j A_{ij} \; \Delta \alpha_j \tag{30}$$

where $\Delta \alpha_j$ is the augmentation for each cycle. The residuals are given by,

$$r_i = \Delta y_j - \sum_{j=1}^{m} A_{ij} \; \Delta \alpha_j \tag{31}$$

The sum of squares is minimized by setting the gradient equation to zero and solving for $\Delta \alpha_j$. This results in simultaneous linear equations. These equations are written in matrix notation as,

$$(A^T A) \, \Delta\alpha = A^T \Delta y \tag{32}$$

In this project, we are dealing with non-linear least squares and the iteration is performed up to the fifth order of the perceptive projection equations.

5. CHI-SQUARED TEST

The Chi-squared method is a test of statistical hypothesis. Consider a group of data points. This method is applied to determine the significant variation between observed and expected frequencies of the data locations. The observations include mutually exclusive cases.

This approach is usually referred to as the total of squared errors for various samples. There are many published methods for Chi-Squared [3] [4]. But the most common and widely used method is Pearson's Chi squared test also called X^2 test. It was applied on a group of random samples that belonged to 'k' number of mutually exclusive cases. For example, this project includes five mutually exclusive cases in the form of 1^{st}, 2^{nd}, 3^{rd}, 4^{th} and 5^{th} order polynomials expressions. Each of these cases have 50 odd samples. Once the error function of these samples is determined, the Chi-squared method is applied on the error functions. These squared functions are finally plotted to test the sampling distribution.

6. RESULTS AND ANALYSIS

This chapter mainly focuses on the procedure followed to obtain the final results and its analysis.

6.1 Step by step analysis of the procedure

This project uses 50 data inputs from the various camera parameters. They will be used in every iteration from first to fifth order of equations. Let us first solve for the first order. Recall equations (16) and (17). We can further derive them as,

$$r = u_0 + \frac{\lambda x}{z s_x} \tag{33}$$

$$c = v_0 + \frac{\lambda y}{z s_y} \tag{34}$$

Let us denote the image plane coordinates (u, v) with (α, β) for further derivations. So, equations (33) and (34) can be rewritten as,

$$r = \alpha_0 + \alpha_1 \frac{x}{z} \tag{35}$$

$$c = \beta_0 + \beta_1 \frac{y}{z} \tag{36}$$

For 50 data points, we obtain

$$\begin{aligned} r_1 &= \alpha_0 + \alpha_1 \frac{x_1}{z_1} \\ r_2 &= \alpha_0 + \alpha_1 \frac{x_2}{z_2} \\ &\vdots \\ r_{50} &= \alpha_0 + \alpha_1 \frac{x_{50}}{z_{50}} \end{aligned} \tag{38}$$

and

$$c_1 = \beta_0 + \beta_1 \frac{y_1}{z_1}$$
$$c_2 = \beta_0 + \beta_1 \frac{y_2}{z_2}$$
$$\vdots$$
$$c_{50} = \beta_0 + \beta_1 \frac{y_{50}}{z_{50}} \quad (39)$$

In matrix notation, they can be written as,

$$\begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_{50} \end{bmatrix} = \begin{bmatrix} 1 & x_1/z_1 \\ 1 & x_2/z_2 \\ \vdots & \vdots \\ 1 & x_{50}/z_{50} \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} \quad (40)$$

$$R = X\,A \quad (41)$$

By multiplying the transpose of matrix X on both sides of equation, we get

$$X^T R = X^T X\,A \quad (42)$$

$$A = (X^T X)^{-1} X^T R \quad (43)$$

and

$$\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{50} \end{bmatrix} = \begin{bmatrix} 1 & y_1/z_1 \\ 1 & y_2/z_2 \\ \vdots & \vdots \\ 1 & y_{50}/z_{50} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} \quad (44)$$

$$C = Y\,B \quad (45)$$

By multiplying the transpose of matrix Y on both sides of equation, we get

$$Y^T C = Y^T Y\,B \quad (46)$$

$$B = (Y^T Y)^{-1} Y^T C \quad (47)$$

Let us apply ‘Least squares’ to find the error in the data points [7]. Substitute the values obtained in equations (43) and (47) to the corresponding unknown parameters. Once, the error is obtained, the Chi-Squared method is applied. As a result, the Chi-squared square error at each data location is obtained. Let ‘err_r’ and ‘err_c’ represent the square

of errors for pixel point (r, c). First, the residual error for each of the 50 data points is found. Then it is squared at each point. From equations (38) and (39),

$$\begin{aligned} err\_r_1 &= (r_1 - \alpha_0 - \alpha_1 \frac{x_1}{z_1})^2 \\ err\_r_2 &= (r_2 - \alpha_0 - \alpha_1 \frac{x_2}{z_2})^2 \\ &\vdots \\ err\_r_{50} &= (r_{50} - \alpha_0 - \alpha_1 \frac{x_{50}}{z_{50}})^2 \end{aligned} \tag{48}$$

and

$$\begin{aligned} err\_c_1 &= (c_1 - \beta_0 - \beta_1 \frac{y_1}{z_1})^2 \\ err\_c_2 &= (c_2 - \beta_0 - \beta_1 \frac{y_2}{z_2})^2 \\ &\vdots \\ err\_c_{50} &= (c_{50} - \beta_0 - \beta_1 \frac{y_{50}}{z_{50}})^2 \end{aligned} \tag{49}$$

For second order polynomials, (r, c) are expressed as follows,

$$r = \alpha_0 + \alpha_1 \frac{x}{z} + \alpha_2 \left(\frac{x}{z}\right)^2 \tag{50}$$

$$c = \beta_0 + \beta_1 \frac{x}{z} + \beta_2 \left(\frac{y}{z}\right)^2 \tag{51}$$

The process is repeated for 50 points following equations (38) and (39).

Similarly, the equations for third, fourth and fifth order polynomials are as follows:

$$r = \alpha_0 + \alpha_1 \frac{x}{z} + \alpha_2 \left(\frac{x}{z}\right)^2 + \alpha_3 \left(\frac{x}{z}\right)^3$$

$$r = \alpha_0 + \alpha_1 \frac{x}{z} + \alpha_2 \left(\frac{x}{z}\right)^2 + \alpha_3 \left(\frac{x}{z}\right)^3 + \alpha_4 \left(\frac{x}{z}\right)^4$$

$$r = \alpha_0 + \alpha_1 \frac{x}{z} + \alpha_2 \left(\frac{x}{z}\right)^2 + \alpha_3 \left(\frac{x}{z}\right)^3 + \alpha_4 \left(\frac{x}{z}\right)^4 + \alpha_5 \left(\frac{x}{z}\right)^5 \quad (52)$$

and

$$c = \beta_0 + \beta_1 \frac{x}{z} + \beta_2 \left(\frac{y}{z}\right)^2 + \beta_3 \left(\frac{y}{z}\right)^3$$

$$c = \beta_0 + \beta_1 \frac{x}{z} + \beta_2 \left(\frac{y}{z}\right)^2 + \beta_3 \left(\frac{y}{z}\right)^3 + \beta_4 \left(\frac{y}{z}\right)^3$$

$$c = \beta_0 + \beta_1 \frac{x}{z} + \beta_2 \left(\frac{y}{z}\right)^2 + \beta_3 \left(\frac{y}{z}\right)^3 + \beta_4 \left(\frac{y}{z}\right)^4 + \beta_5 \left(\frac{y}{z}\right)^5 \quad (53)$$

Similarly, an i^{th} order equation will have intrinsic parameters till 'α_i' and 'β_i' along with variables 'x/z' and 'y/z' extending to the i^{th} power.

6.2 Simulation results and analysis

Using MATLAB simulation, several graphs have been obtained for various parameters to assist in further non-linear analysis of camera calibration.

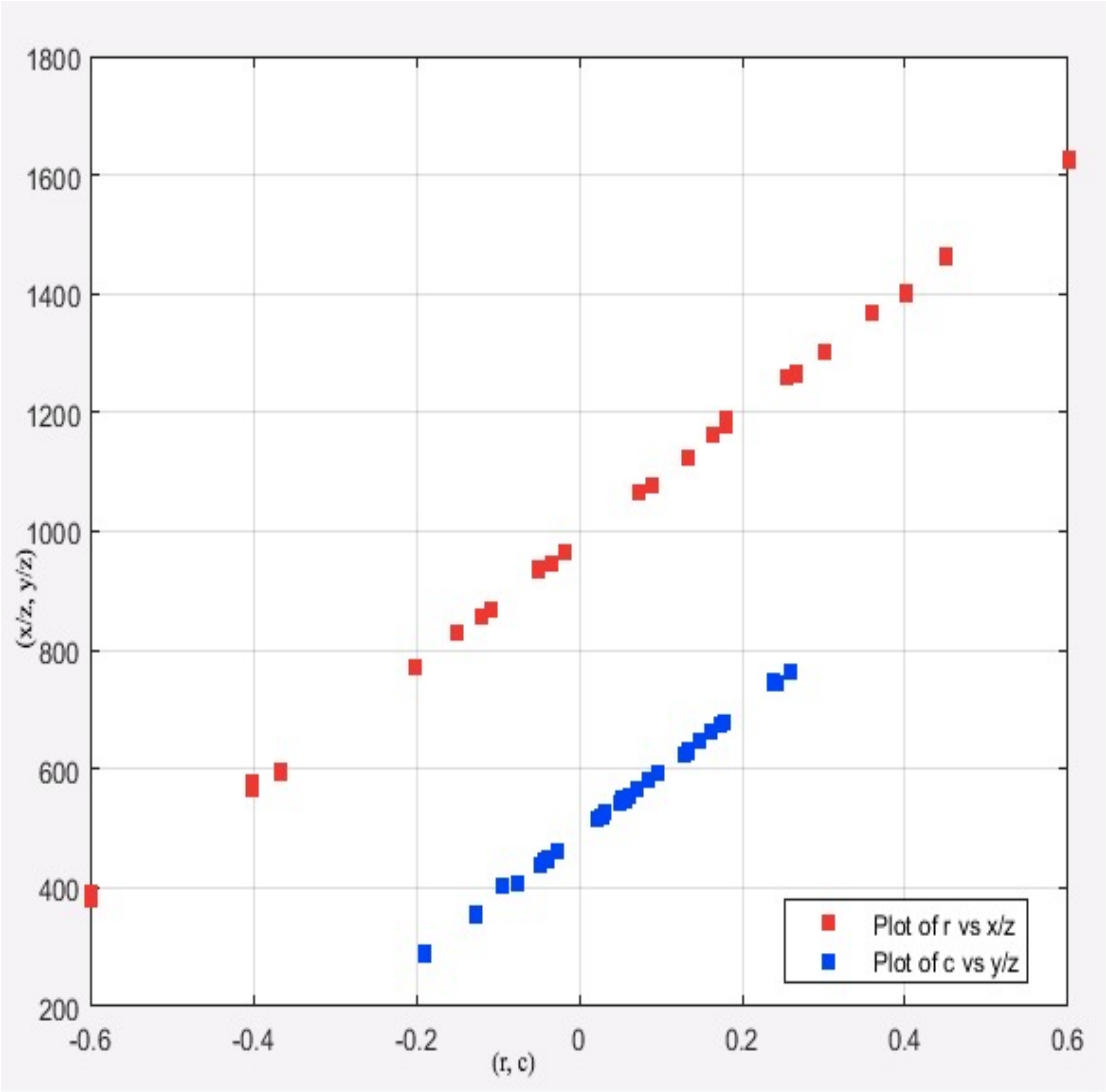

Figure 6: Plot of pixel coordinates against world coordinates

By using least squares method, the error at each point is obtained relative to the pixel coordinates ‘r’ and ‘c’. The graphs below are the simulation results of the approach.

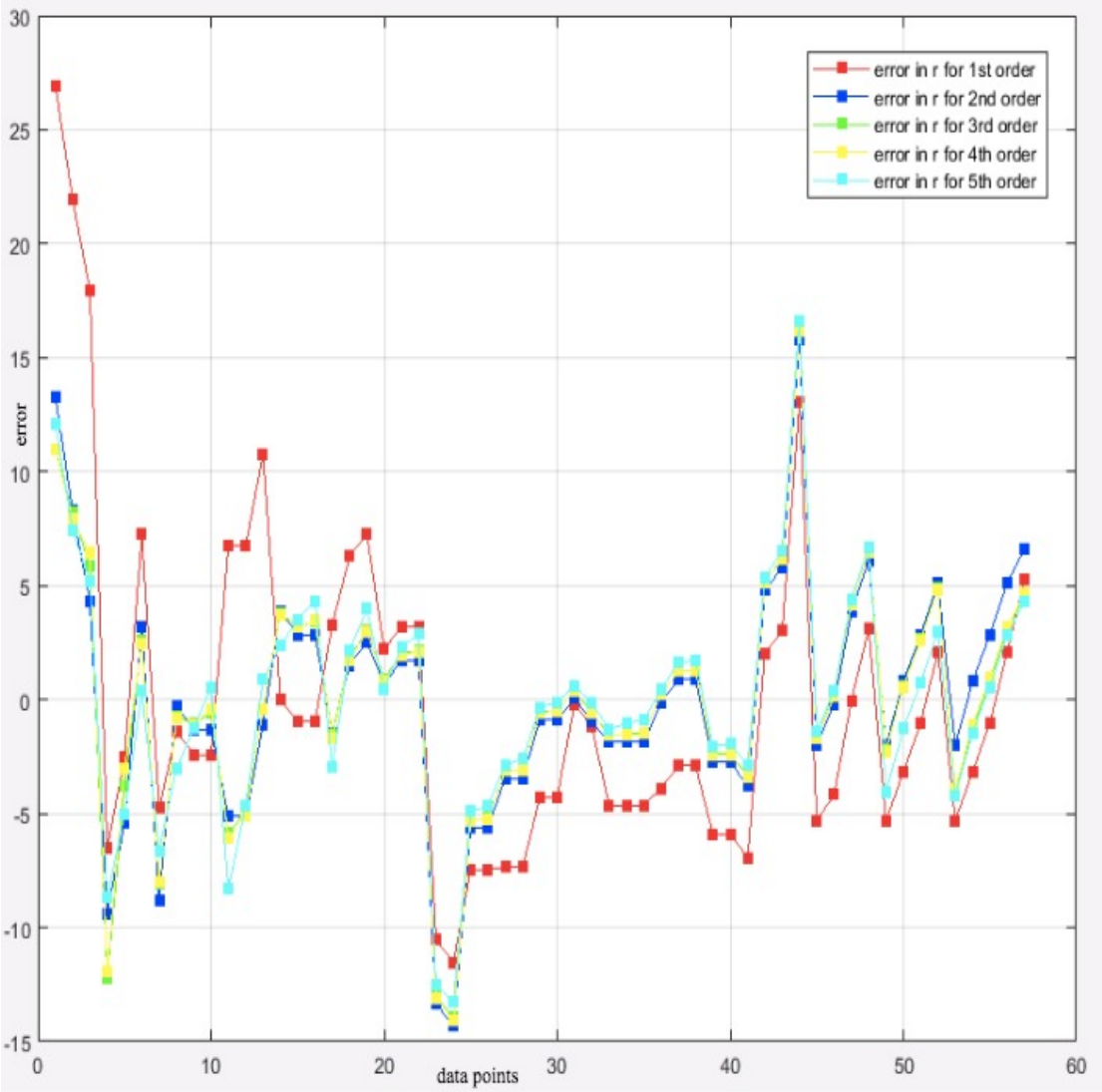

Figure 7: Plot showing error in ‘r’ for different orders

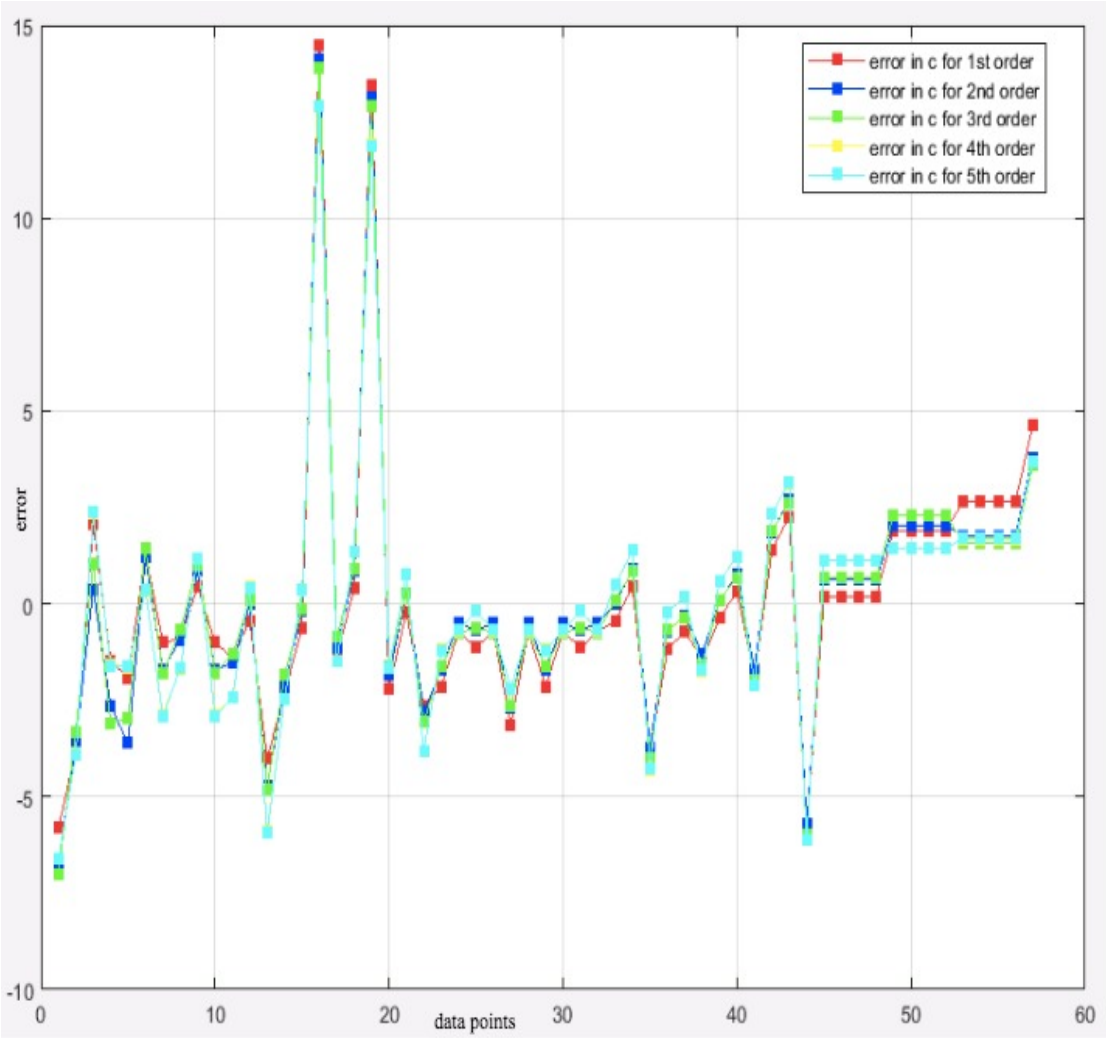

Figure 8: Plot showing error in ‘c’ for different orders

If we carefully study the plot of the error for each order, we notice that the fluctuations have slowly decreased with the increase in the order of the perceptive projection equation. Thus, resulting in gradual increase in the calibration precision.

To detect and remove bad data points, the Chi-squared test is used. Below are the two graphs for co-ordinates ‘r’ and ‘c’ with increasing order of polynomial.

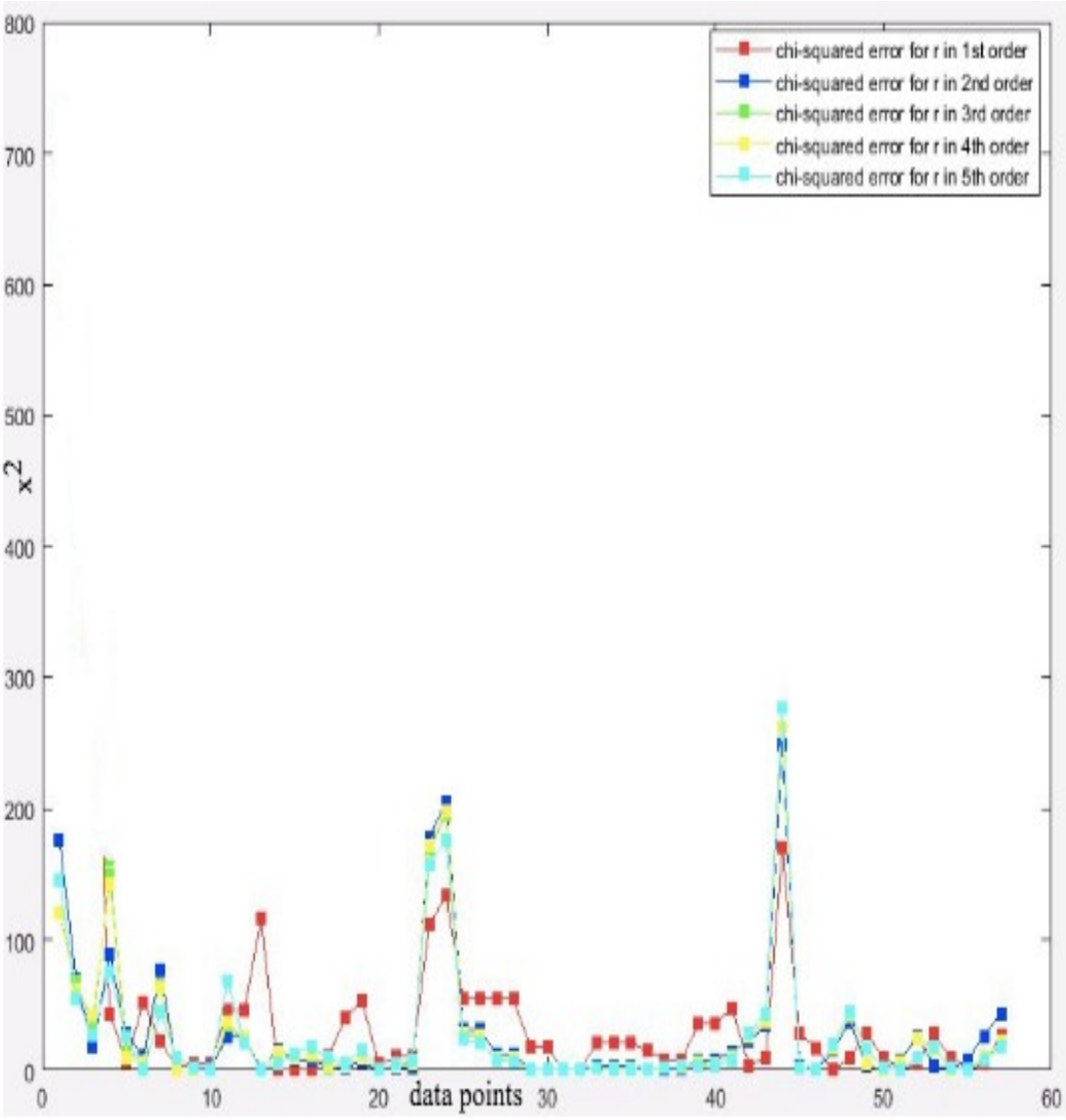

Figure 9: Plot of Chi-squared error for ‘r’

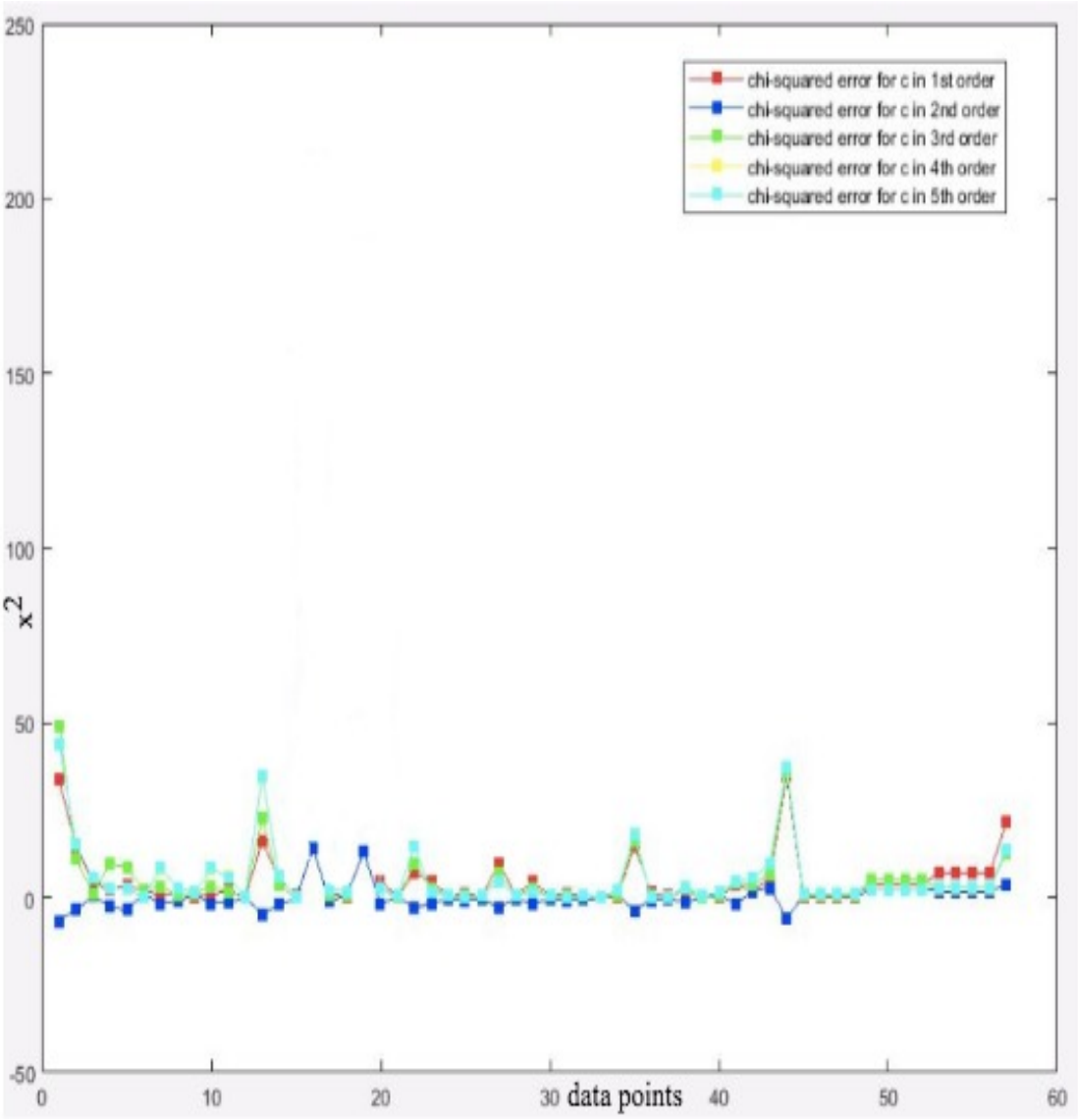

Figure 10: Plot of Chi-squared error for ‘c’

By observing both graphs after the Chi-squared test and comparing them to the corresponding plots for the error before the test, we notice a significant reduction in data point fluctuation resulting in improved calibration of the camera.

7. CONCLUSIONS

The importance of camera calibration has reached new heights due to the various applications such as robotics and navigation systems. This project solves the problem of camera calibration. The least squares method is commonly used to solve linear and nonlinear approximation problems. In this project we apply it for non-linear analysis of camera calibration. Through careful observation of the results obtained during simulation, we can conclude that the non-linear least squares approach helps to improve calibration precision of a camera. Also, the accuracy will be much better by further subjecting those results to Chi-squared test which eliminates inaccurate data points.

Appendix A. Data points of camera parameter for input

S=150

P1= [[1*S, 130, 2*S]]

P2= [[1*S, 96, 2*S]]

P3= [[1*S, 0, 2*S]]

P4= [[3*S, 135, 2*S]]

P5= [[3*S, 0, 2*S]]

P6= [[1*S, 135, 3*S]]

P7= [[1*S, 0, 3*S]]

P8= [[3*S, 130, 3*S]]

P9= [[3*S, 95, 3*S]]

P10= [[3*S, 0, 3*S]]

P11= [[4*S, 137, 3*S]]

P12= [[4*S, 89, 3*S]]

P13= [[4*S, 0, 3*S]]

P14= [[2*S, 137, 4*S]]

P15= [[2*S, 89, 4*S]]

P16= [[2*S, 0, 4*S]]

P17= [[4*S, 137, 4*S]]

P18= [[4*S, 89, 4*S]]

P19= [[4*S, 0, 4*S]]

P20= [[4*S, 130, 5*S]]

P21= [[4*S, 96, 5*S]]

P22= [[4*S, 0, 5*S]]

P23= [[0*S, 108, 6*S]]

P24= [[0*S, 32, 6*S]]

P25= [[1*S, 108, 6*S]]

P26= [[1*S, 32, 6*S]]

P27= [[2*S, 108, 6*S]]

P28= [[2*S, 32, 6*S]]

P29= [[3*S, 108, 6*S]]

P30= [[3*S, 32, 6*S]]

P31= [[4*S, 108, 6*S]]

P32= [[4*S, 32, 6*S]]

P33= [[1*S, 131, 8*S]]

P34= [[1*S, 96, 8*S]]

P35= [[1*S, 0, 8*S]]

P36= [[3*S, 131, 9*S]]

P37= [[3*S, 96, 9*S]]

P38= [[3*S, 0, 9*S]]

P39= [[1*S, 131, 10*S]]

P40= [[1*S, 96, 10*S]]

P41= [[1*S, 0, 10*S]]

P42= [[4*S, 137, 10*S]]

P43= [[4*S, 89, 10*S]]

P44= [[4*S, 0, 10*S]]

P45= [[1*S, 1*S, 11*S]]

P46= [[2*S, 1*S, 11*S]]

P47= [[3*S, 1*S, 11*S]]

P48= [[4*S, 1*S, 11*S]]

P49= [[1*S, 2*S, 11*S]]

P50= [[2*S, 2*S, 11*S]]

P51= [[3*S, 2*S, 11*S]]

P52= [[4*S, 2*S, 11*S]]

P53= [[1*S, 3*S, 11*S]]

P54= [[2*S, 3*S, 11*S]]

P55= [[3*S, 3*S, 11*S]]

P56= [[4*S, 3*S, 11*S]]

P57= [[5*S, 3*S, 11*S]]

S1= [[390, 337]]

S2= [[385, 455]]

S3= [[381, 788]]

S4= [[1400, 315]]

S5= [[1404, 792]]

S6= [[579, 404]]

S7= [[567, 724]]

S8= [[1266, 418]]

S9= [[1265, 499]]

S10= [[1265,724]]

S11= [[1622, 402]]

S12= [[1622, 514]]

S13= [[1626, 727]]

S14= [[937, 450]]

S15= [[936, 533]]

S16= [[936, 675]]

S17= [[1462, 449]]

S18= [[1465, 532]]

S19= [[1466, 676]]

S20= [[1367, 488]]

S21= [[1368, 534]]

S22= [[1368, 672]]

S23= [[596, 531]]

S24= [[595, 619]]

S25= [[773, 530]]

S26= [[773, 619]]

S27= [[947, 532]]

S28= [[947, 619]]

S29= [[1124, 531]]

S30= [[1124, 619]]

S31= [[1302, 530]]

S32= [[1301, 619]]

S33= [[828, 524]]

S34= [[828, 554]]

S35= [[828, 643]]

S36= [[1078, 532]]

S37= [[1079, 559]]

S38= [[1079, 635]]

S39= [[858, 537]]

S40= [[858, 561]]

S41= [[857, 631]]

S42= [[1179, 531]]

S43= [[1180, 564]]

S44= [[1190, 635]]

S45= [[870, 529]]

S46= [[966, 529]]

S47= [[1065, 529]]

S48= [[1163, 529]]

S49= [[870, 431]]

S50= [[967, 431]]

S51= [[1064, 431]]

S52= [[1162, 431]]

S53= [[870, 334]]

S54= [[967, 334]]

S55= [[1064, 334]]

S56= [[1162, 334]]

S57= [[1260, 332]]

Appendix B. Matrix generation and plots

```
Cdd=[P1;P2;P3;P4;P5;P6;P7;P8;P9;P10;P11;P12;P13;P14;P15;P16;P17;P18;P19;P20
;P21;P22;P23;P24;P25;P26;P27;P28;P29;P30;P31;P32;P33;P34;P35;P36;P37;P38;P3
9;P40;P41;P42;P43;P44;P45;P46;P47;P48;P49;P50;P51;P52;P53;P54;P55;P56;P57];
R=[S1;S2;S3;S4;S5;S6;S7;S8;S9;S10;S11;S12;S13;S14;S15;S16;S17;S18;S19;S20;S
21;S22;S23;S24;S25;S26;S27;S28;S29;S30;S31;S32;S33;S34;S35;S36;S37;S38;S39;
S40;S41;S42;S43;S44;S45;S46;S47;S48;S49;S50;S51;S52;S53;S54;S55;S56;S57];
x= Cdd(:,1)-330;
y= Cdd(:,2)-57;
z= Cdd(:,3)-0;
r= R(:,1);
c= 1080-R(:,2);
one= ones(57,1);
ls= linspace(1,57,57);

%FIRST ORDER
W1= x./z;
U1= y./z;
X= [one W1];
A= inv(X'*X)*X'*r;
Y= [one U1];
B= inv(Y'*Y)*Y'*c;

r1= A(1)+A(2)*W1;
r_err = r- r1;
%plot(ls,r_err,'rs-','MarkerFaceColor', 'r');                    %error plot for r
r_error2= (r- r1).^2;
plot(ls,r_error2,'rs-','MarkerFaceColor', 'r');          %plot of chi-squared error for r

c1=B(1)+B(2)*U1;
c_err = c-c1;
```

```
%plot(ls,c_err,'rs-','MarkerFaceColor', 'r');                    %error plot for c
%c_error= (c-c1).^2;
%plot(ls,c_error,'rs-','MarkerFaceColor', 'r');          % plot of chi-squared error for c
hold on;

%SECOND ORDER
W2= (x./z).^2;
U2= (y./z).^2;
X2= [one W1 W2];
Asec= inv(X2'*X2)*X2'*r;
Y2= [one U1 U2];
Bsec= inv(Y2'*Y2)*Y2'*c;

r2=Asec(1)+Asec(2)*W1+Asec(3)*W2;
r_err2 = r-r2;
%plot(ls,r_err2,'bs-','MarkerFaceColor', 'b');                    %error plot for r
r_error2= (r-r2).^2;
plot(ls,r_error2,'bs-','MarkerFaceColor', 'b');          %plot of chi-squared error for r

c2=Bsec(1)+Bsec(2)*U1+Bsec(3)*U2;
c_err2 = c-c2;
%plot(ls,c_err2,'bs-','MarkerFaceColor', 'b');                    %error plot for c
%c_error2= (c-c2).^2;
%plot(ls,c_error2,'bs-','MarkerFaceColor', 'b');          %plot of chi-squared error for c
hold on;


%THIRD ORDER
W3= (y./z).^3;
U3= (y./z).^3;
X3= [one W1 W2 W3];
Athi= inv(X3'*X3)*X3'*r;
```

```
Y3= [one U1 U2 U3];
Bthi= inv(Y3'*Y3)*Y3'*c;

r3= Athi(1)+Athi(2)*W1+Athi(3)*W2+Athi(4)*W3;
r_err3 = r-r3;
%plot(ls,r_err3,'gs-','MarkerFaceColor', 'g');                    %error plot for r
r_error3= (r-r3).^2;
plot(ls,r_error3,'gs-','MarkerFaceColor', 'g');          %plot of chi-squared error for r

c3=Bthi(1)+Bthi(2)*U1+Bthi(3)*U2+Bthi(4)*U3;
c_err3 = c-c3;
%plot(ls,c_err3,'gs-','MarkerFaceColor', 'g');                    %error plot for c
c_error3= (c-c3).^2;
%plot(ls,c_error3,'gs-','MarkerFaceColor', 'g');          %plot of chi-squared error for c
hold on;

%FOURTH ORDER
W4= (y./z).^4;
U4= (y./z).^4;
X4= [one W1 W2 W3 W4];
Afou= inv(X4'*X4)*X4'*r;
Y4= [one U1 U2 U3 U4];
Bfou= inv(Y4'*Y4)*Y4'*c;

r4=Afou(1)+Afou(2)*W1+Afou(3)*W2+Afou(4)*W3+Afou(5)*W4;
r_err4 = r-r4;
%plot(ls,r_err4,'ys-','MarkerFaceColor', 'y');                    %error plot for r
r_error4= (r-r4).^2;
plot(ls,r_error4,'ys-','MarkerFaceColor', 'y');          %plot of chi-squared error for r

c4=Bfou(1)+Bfou(2)*U1+Bfou(3)*U2+Bfou(4)*U3+Bfou(5)*U4;
c_err4 = c-c4;
```

```
%plot(ls,c_err4,'ys-','MarkerFaceColor', 'y');                    %error plot for c
c_error4= (c-c4).^2;
%plot(ls,c_error4,'ys-','MarkerFaceColor', 'y');          %plot of chi-squared error for c
hold on;

%FIFTH ORDER
W5= (y./z).^5;
U5= (y./z).^5;
X5= [one W1 W2 W3 W4 W5];
Afif= inv(X5'*X5)*X5'*r;
Y5= [one U1 U2 U3 U4 U5];
Bfif= inv(Y5'*Y5)*Y5'*c;

r5=Afif(1)+Afif(2)*W1+Afif(3)*W2+Afif(4)*W3+Afif(5)*W4+Afif(6)*W5;
r_err5 = r-r5;
%plot(ls,r_err5,'cs-','MarkerFaceColor', 'c');                    %error plot for r
r_error5= (r-r5).^2;
plot(ls,r_error5,'cs-','MarkerFaceColor', 'c');          %plot of chi-squared error for r

c5=Bfif(1)+Bfif(2)*U1+Bfif(3)*U2+Bfif(4)*U3+Bfif(5)*U4+Bfif(6)*U5;
c_err5 = c-c5;
%plot(ls,c_err5,'cs-','MarkerFaceColor', 'c');                    %error plot for c
c_error5= (c-c5).^2;
%plot(ls,c_error5,'cs-','MarkerFaceColor', 'c');          %plot of chi-squared error for c
hold off;
end
```

REFERENCES

[1] Zhang, Z. “A Flexible New Technique for Camera Calibration.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 22, No. 11, 2000, pp. 1330–1334

[2] Charnes, A.; Frome, E. L.; Yu, P. L. (1976). "The Equivalence of Generalized Least Squares and Maximum Likelihood Estimates in the Exponential Family". *Journal of the American Statistical Association*. **71** (353):169–171.

[3] Pearson, Karl (1900). "On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling" . Philosophical Magazine. Series 5. 50: 157–175. doi:10.1080/14786440009463897.

[4] Bagdonavicius, V.; Nikulin, M. S. (2011), "Chi-squared goodness-of-fit test for right censored data", *The International Journal of Applied Mathematics and Statistics*, pp. 30–50

[5] C.L. Lawson and R.J. Hanson, Solving Least Squares Problems, Prentice–Hall, 1974

[6] “What is Simple Linear Regression?” Pennsylvania State University. Retrieved 2016-10-17

[7] *Q.T. Luong (1992). Matrice fondamentale et auto-calibration en vision par ordinateur. PhD Thesis, University of Paris, Orsay.*