AN ANALYSIS THE POSITIONAL ACCURACY OF A
MULTI-CAMERA BALL TRACKING SYSTEM

by
JIALE ZHU

A thesis submitted to the
Graduate School - Camden
Rutgers, The State University of New Jersey
In partial fulfillment of the requirements
For the degree of
Master of Science
Graduate Program in Mathematical Science

Written under the direction of
Dr. Stephen M. Alessandrini

and approved by

__________________________
Stephen M. Alessandrini

__________________________
Sigi Fu

__________________________
Haydee Herrera-Guzman

__________________________
Haisheng Li

__________________________
Benedetto Piccoli

Camden, New Jersey
May 2021
THESIS ABSTRACT

An analysis the positional accuracy of a multi-camera ball tracking system

By JIALE ZHU
Thesis Director:
Dr. Stephen M. Alessandrini

This paper describes three kinds of pinhole camera models: the camera without lens distortion and calibration, the camera with lens distortion, and the distorted camera calibrated using an inverse lens distortion series of different orders. In this paper, most equations are in matrix form, which makes programming dramatically more comfortable. During the lens distortion process, we used second-order coefficients to simulate the lens distortion. Due to the quantization error when converting image coordinates to pixel coordinates, the actual distortion coefficients are estimated by a least-squares algorithm. The estimated distortion coefficients are applied in the inverse lens distortion series to calibrate the cameras. The position of the ball is triangulated by using a system of eight cameras. The triangulation algorithm will be introduced in Chapter 4. Finally, Chapter 5 compares the performance of the system with and without lens distortion and with various levels of camera calibration to correct for lens distortion.
Acknowledgements

I would like to thank my thesis director Dr. Steve Alessandrini for his patience, detailed instruction, and for providing MATLAB and C++ examples to help to get the modeling of the ball tracking system set up.
Dedication

To my parents whose love, support, and encouragement made this possible.
# Table of Contents

Abstract ................................................................. ii  
Acknowledgements ................................................... iii 
Dedication ............................................................... iv  

1. Introduction ............................................................ 1  

2. Camera Model .......................................................... 3  
   2.1. Transforming from real world coordinates to camera coordinates ... 3  
   2.2. Transforming from camera coordinates to image-plane coordinates ... 4  
   2.3. Transforming from image-plane coordinates to pixel coordinates ... 5  
   2.4. Pinhole camera model without lens distortion ................. 5  
   2.5. Lens distortion ................................................ 6  
   2.6. Pinhole camera model with lens distortion .................. 8  

3. Camera Calibration ..................................................... 9  
   3.1. Calibrate by inverse coefficient series ....................... 9  
   3.2. Least squares to estimate lens distortion coefficients .......... 10  

4. N-Camera Triangulation Algorithm to Determine 3D Position .... 12  
   4.1. The distance between a ray $L$ and a point $P$ ............... 12  
   4.2. Calculate the optimal point $P^*$ .......................... 13  

5. Results ................................................................. 15  
   5.1. Camera model with quantization error and lens distortion .... 15  
   5.2. Camera calibrated by first-order inverse lens distortion series .... 17  
   5.3. Camera calibrated by second-order inverse lens distortion series ... 18
Chapter 1

Introduction

The Hawk-Eye system is a type of computer vision system [1]. In many ball games, for instance tennis, cricket, volleyball, badminton, and Gaelic football, visually tracking the trajectory of the ball is important. The Hawk-Eye system uses several high speed video cameras which provide timing data and visual images to determine the ball track based on triangulation. The system is not only able to build up the trajectory of the ball but even “predict” the future flight path. The US Open started official use of the Hawk-Eye system in 2006. They use 10 high speed cameras, and the average error of that system is 3.6 mm, about 5% error relative to the diameter of the ball.

In early motion capture techniques [2], the 3D position is also calculated by images from multiple cameras. In those multicamera motion tracking systems, we need to find a way to determine the position of the object using several simultaneously taken images. In addition, because there is lens distortion in the images, we need to apply camera calibration.

This paper will use eight cameras and treat direction and position of them as set up in Table 1.1. The pinhole camera model is generally divided into four steps [3], [4], [5]:

Step1: Transform real world coordinates to camera coordinates;
Step2: Transform camera coordinates to image-plane coordinates;
Step3: Add lens distortion;
Step4: Transform distorted image-plane coordinates to pixel coordinates;

In Step 3, this paper will first choose some suitable lens distortion coefficients to simulate the distorted images. Due to the truncation error during quantization in Step 4, this paper will use least-squares algorithm to estimate actual lens distortion coefficients.
Table 1.1: Positions and Directions of Cameras

<table>
<thead>
<tr>
<th>Position(m)</th>
<th>Direction(degree)</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>Y</td>
</tr>
<tr>
<td>camera1</td>
<td>28.0</td>
</tr>
<tr>
<td>camera2</td>
<td>28.0</td>
</tr>
<tr>
<td>camera3</td>
<td>-28.0</td>
</tr>
<tr>
<td>camera4</td>
<td>-28.0</td>
</tr>
<tr>
<td>camera5</td>
<td>0.0</td>
</tr>
<tr>
<td>camera6</td>
<td>40.0</td>
</tr>
<tr>
<td>camera7</td>
<td>0.0</td>
</tr>
<tr>
<td>camera8</td>
<td>-40.0</td>
</tr>
</tbody>
</table>

Sheng-Wen Shih and Yi-Ping Hung [6] give two different methods of camera calibration. They use noncoplanar calibration points or coplanar calibration points and formulate the calibration as an eigenvalue problem. However, in their model, the distortion formula is only first-order. Taufiqur Rahman and Nicholas Krouglicof [7] presented a novel calibration system. In their model, both radial distortion and tangential distortion are considered.

Junhee Park and Seong-Chan Byun [9] constructed a model of inverse mapping from distorted coordinate to ideal image coordinate. In this paper, we will only consider radial lens distortion and examine the calibration technique of the inverse radial lens distortion series in Drap [8] to calibrate cameras, and explore whether higher order series will perform a better calibration.

Finally, to determining 3D position, we will apply multicamera triangulation algorithm similar to Slabaugh’s algorithm [10]. This approach is also used by James Black [12], [13].
Chapter 2
Camera Model

2.1 Transforming from real world coordinates to camera coordinates

Assume a ideal pinhole camera without lens distortion.

In the Figure 2.1, WCS is the world coordinate system and CCS is the camera coordinate system. Assume the world coordinate of the camera’s center is:

\[
C = \begin{bmatrix}
x_c \\
y_c \\
z_c 
\end{bmatrix}, \quad (2.1)
\]

and assume the rotation matrix of the camera is:

\[
R = \begin{bmatrix}
r_{11} & r_{12} & r_{13} \\
r_{21} & r_{22} & r_{23} \\
r_{31} & r_{32} & r_{33} 
\end{bmatrix}. \quad (2.2)
\]
Then, the translation vector $T$ is:

$$
T = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} = - \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}.
$$

(2.3)

If $P$ is a point with world coordinates $[x_{wcs}, y_{wcs}, z_{wcs}]^T$, its coordinates $P'$ in CCS are calculated by:

$$
P' = \begin{bmatrix} x_{ccs} \\ y_{ccs} \\ z_{ccs} \end{bmatrix} = \begin{bmatrix} R & T \end{bmatrix} \begin{bmatrix} P \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} x_{wcs} \\ y_{wcs} \\ z_{wcs} \\ 1 \end{bmatrix}.
$$

(2.4)

The $3 \times 4$ matrix in the equation (2.16) is called the extrinsic parameter matrix, because it is only related to position and direction of the camera.

### 2.2 Transforming from camera coordinates to image-plane coordinates

A real pinhole camera will generate $180^\circ$ inverted picture, but the picture shown on its screen is also $180^\circ$ inverted, thus the camera model in this paper ignores such an inversion and assumes the image plane lays on positive $z$-axis in CCS.

![Figure 2.2: From CCS to image plane](image)

In Figure 2.2, $q'$ is the intersection point between the $z$-axis of CCS and the image plane, its coordinates in CCS is $[0, 0, f]^T$, $f$ is the focal length of the camera. The point $p$ is any point $[x_{ccs}, y_{ccs}, z_{ccs}]^T$ in the CCS, and $q$ is the projection of $p$ on the $z$-axis of
CCS, thus \( q = [0, 0, z_{ccs}]^T \). The point \( p' \) is projection of \( p \) on the image plane, then the homogeneous coordinates of \( p' \) are calculated by:

\[
\begin{bmatrix}
    z_{ccs} \cdot x_{image} \\
    z_{ccs} \cdot y_{image} \\
    z_{ccs}
\end{bmatrix}
= \begin{bmatrix}
    f & 0 & 0 \\
    0 & f & 0 \\
    0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
    x_{ccs} \\
    y_{ccs} \\
    z_{ccs}
\end{bmatrix}.
\]

(2.5)

### 2.3 Transforming from image-plane coordinates to pixel coordinates

In Figure 2.3, each small rectangle represents a pixel on the screen. The intrinsic parameters of the camera are also known: \( dx \) is the center-to-center distance of horizontal neighbour pixel; \( dy \) is the center-to-center distance of vertical neighbour pixel; \( resX \) is horizontal resolution in pixels; \( resY \) is vertical resolution in pixels. Assume there is a point \( p \) and its coordinate of the image plane coordinate system is \([x_{image}, y_{image}]\), the coordinates of \( p \) in the pixel coordinate system are calculated by:

\[
\begin{bmatrix}
    p \\
    1
\end{bmatrix}
= \begin{bmatrix}
    1/dx & 0 & resX/2 \\
    0 & 1/dy & resY/2 \\
    0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
    x_{image} \\
    y_{image} \\
    1
\end{bmatrix}.
\]

(2.6)

This step will convert floating-point image plane coordinates to integer pixel coordinates, which will cause quantization error.

### 2.4 Pinhole camera model without lens distortion

Combining the matrices in transformation from CCS coordinates to image plane
coordinates and the transformation from image plane coordinates to pixel coordinates:

\[
\begin{bmatrix}
\frac{1}{dx} & 0 & \text{resX}/2 \\
0 & \frac{1}{dy} & \text{resY}/2 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
f & 0 & 0 \\
0 & f & 0 \\
0 & 0 & 1
\end{bmatrix}
= \begin{bmatrix}
f/dx & 0 & \text{resX}/2 \\
0 & f/dy & \text{resY}/2 \\
0 & 0 & 1
\end{bmatrix}.
\] (2.7)

The matrix in the right side of equation (2.7) is called the intrinsic parameter matrix. Therefore, the coordinates in WCS can be transformed to its corresponding coordinates by multiplying by the extrinsic parameter matrix and then by multiplying by the intrinsic parameter matrix:

\[
\begin{bmatrix}
z_{ccs} \cdot x_{\text{pixel}} \\
z_{ccs} \cdot y_{\text{pixel}} \\
z_{ccs}
\end{bmatrix}
= \begin{bmatrix}
f/dx & 0 & \text{resX}/2 \\
0 & f/dy & \text{resY}/2 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
r_{11} & r_{12} & r_{13} & t_x \\
r_{21} & r_{22} & r_{23} & t_y \\
r_{31} & r_{32} & r_{33} & t_z
\end{bmatrix}
\begin{bmatrix}
x_{\text{wcs}} \\
y_{\text{wcs}} \\
z_{\text{wcs}}
\end{bmatrix}.
\] (2.8)

After getting the value of \(x_{\text{pixel}}\) and \(y_{\text{pixel}}\), we need to round them to integer values. To determine the direction of the ray, these pixels need to be converted back to image coordinates.

In this paper, a ray \(L = P + kv\) is represented by an initial point \(P\) and a normalized vector \(v\) which represent the ray’s direction. Define a function \(\text{norm}(\text{vector } v)\) which outputs the normalized vector of the input vector \(v\). Let the intrinsic parameter matrix of a camera be \(I\) and the rotation be \(R\). Then, if the pixel coordinates are \([x_{\text{pixel}}, y_{\text{pixel}}]^T\), the ray function from this camera to the point \(P\) in WCS is:

\[
L = C + k \cdot \text{norm}
\left(R^{-1}I^{-1}
\begin{bmatrix}
x_{\text{pixel}} + 0.5 \\
y_{\text{pixel}} + 0.5 \\
1
\end{bmatrix}
\right),
\] (2.9)

where \(C\) is the position of camera’s center in WCS, and a half of a pixel size is added to get the center coordinate of each pixel.

### 2.5 Lens distortion

In the real world, lens distortion is non-negligible. If the camera lens is not perfectly parallel with the image plane, there will be tangential distortion. An optical property
of the lens itself will cause radial distortion. The lens distortion is usually described by [11]:

\[
x_{\text{dist}} = x_{\text{image}} \left(1 + K_1 r^2 + K_2 r^4 + \cdots \right) + P_1 \left(r^2 + 2x_{\text{image}}^2\right) + 2P_2 x_{\text{image}} y_{\text{image}} \left(1 + P_3 r^2 + P_4 r^4 + \cdots \right),
\]

\[
y_{\text{dist}} = y_{\text{image}} \left(1 + K_1 r^2 + K_2 r^4 + \cdots \right) + 2P_1 x_{\text{image}} y_{\text{image}} + P_2 \left(r^2 + x_{\text{image}}^2\right) \left(1 + P_3 r^2 + P_4 r^4 + \cdots \right),
\]

\[
r^2 = x_{\text{image}}^2 + y_{\text{image}}^2.
\]

Here, \(x_{\text{dist}}\) and \(y_{\text{dist}}\) are the distorted image coordinates; \(P_n\) is \(n^{th}\) tangential distortion coefficient and \(K_n\) is \(n^{th}\) radial distortion coefficient. Different values of \(K_1\) and \(K_2\) form different kinds of distortion. A relatively large positive \(K_1\) causes barrel distortion: see Figure 2.4b; a relatively large negative \(K_2\) causes pincushion distortion: see Figure 2.4c; they together causes mustache distortion: see Figure 2.4d.

We will assume the cameras in this paper are “good” enough so that tangential distortion can be ignored. And we can see in Figure 2.4, two radial distortion coefficient \(K_1\) and \(K_2\) are enough to describe those three kinds of radial distortions. Thus, we will use \(K_1\) and \(K_2\) to simulate lens distortion but we will estimate \(K_1\) and \(K_2\) per camera and separately calibrate each camera. The lens distortion equations in this paper are:

\[
x_{\text{dist}} = x_{\text{image}} \left(1 + K_1 r^2 + K_2 r^4 + K_3 r^6 + \cdots \right),
\]
\[ y_{\text{dist}} = y_{\text{image}} (1 + K_1 r^2 + K_2 r^4 + K_3 r^6 + \cdots) . \]  

(2.14)

### 2.6 Pinhole camera model with lens distortion

Because lens distortion is nonlinear, we assume a function of the form:

\[
D \left( \begin{bmatrix} x_{\text{image}} \\ y_{\text{image}} \end{bmatrix} \right) = \begin{bmatrix} f_1 (x_{\text{image}}, y_{\text{image}}) \\ f_2 (x_{\text{image}}, y_{\text{image}}) \end{bmatrix} = \begin{bmatrix} x_{\text{dist}} \\ y_{\text{dist}} \end{bmatrix} .
\]  

(2.15)

The input of function \( D \) are the undistorted image plane coordinates, \( f_1 \) and \( f_2 \) are equations (2.13) and (2.14), and the output of function \( D \) are the distorted image plane coordinates. Thus the model of a pinhole camera with lens distortion is:

\[
\begin{bmatrix} z_{\text{ccs}} \cdot x_{\text{pixel}} \\ z_{\text{ccs}} \cdot y_{\text{pixel}} \\ z_{\text{ccs}} \end{bmatrix} = \begin{bmatrix} 1/\Delta x & 0 & resX/2 \\ 0 & 1/\Delta y & resY/2 \\ 0 & 0 & 1 \end{bmatrix} \cdot D \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & tx \\ r_{21} & r_{22} & r_{23} & ty \\ r_{31} & r_{32} & r_{33} & tz \end{bmatrix} \begin{bmatrix} x_{\text{wcs}} \\ y_{\text{wcs}} \\ z_{\text{wcs}} \\ 1 \end{bmatrix} .
\]  

(2.16)

The process of calculating a ray function \( L = P + kv \) from pixel coordinates \([x_{\text{pixel}}, y_{\text{pixel}}]^T\) is similar to pinhole camera without lens distortion:

\[
L = C + k \cdot \text{norm} \left( R^{-1} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1/\Delta x & 0 & resX/2 \\ 0 & 1/\Delta y & resY/2 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} x_{\text{pixel}} + 0.5 \\ y_{\text{pixel}} + 0.5 \\ 1 \end{bmatrix} \right) .
\]  

(2.17)

Here, \( C \) is the position of camera’s center in WCS.
Chapter 3

Camera Calibration

In this paper, we will assume that the coordinates of camera center and direction are known, which means the extrinsic parameter matrix is known, and the focal length $f$, pixel size $dx$ and $dy$, the screen resolution ratio $resX$ and $resY$ are also known. We will choose $K_1$ and $K_2$ to simulate real lens distortion and generate distorted points. The aim of calibration is to determine $D^{-1}$ in equation (2.17).

For a point $P$, its pixel coordinates $[x_{pixel}, y_{pixel}]^T$ and WCS coordinates $[x_{wcs}, y_{wcs}, z_{wcs}]^T$ are easy to obtain. We will perform camera calibration by using those coordinate pairs. For such $P$, the distorted image plane coordinates are calculated from its pixel coordinates:

$$
\begin{bmatrix}
    x_{dist} \\
    y_{dist} \\
    1
\end{bmatrix} =
\begin{bmatrix}
    1/dx & 0 & resX/2 \\
    0 & 1/dy & resY/2 \\
    0 & 0 & 1
\end{bmatrix}^{-1}
\begin{bmatrix}
    x_{pixel} + 0.5 \\
    y_{pixel} + 0.5 \\
    1
\end{bmatrix}.
$$

3.1 Calibrate by inverse coefficient series

In this section, we give a brief overview of the technique used in [8]. The lens
distortion and inverse distortion equation are written as:

\[
D : \begin{bmatrix} x_{\text{image}} \\ y_{\text{image}} \end{bmatrix} \rightarrow \begin{bmatrix} x_{\text{dist}} \\ y_{\text{dist}} \end{bmatrix} = P(r) \begin{bmatrix} x_{\text{image}} \\ y_{\text{image}} \end{bmatrix}, \quad (3.3)
\]

\[
D^{-1} : \begin{bmatrix} x_{\text{dist}} \\ y_{\text{dist}} \end{bmatrix} \rightarrow \begin{bmatrix} x_{\text{image}} \\ y_{\text{image}} \end{bmatrix} = Q(r') \begin{bmatrix} x_{\text{dist}} \\ y_{\text{dist}} \end{bmatrix}, \quad (3.4)
\]

where \( r^2 \) is from equation (2.12) and \( r'^2 = x_{\text{dist}}^2 + y_{\text{dist}}^2 \). \( P(r) \) and \( Q(r') \) are the lens distortion and inverse distortion series:

\[
P(r) = \sum_{i=0}^{+\infty} K_i r^{2i}, \quad (3.5)
\]

\[
Q(r') = \sum_{i=0}^{+\infty} K_i' r'^{2i}, \quad (3.6)
\]

where \( K_0 = 1 \) and \( K_0' = 1 \). Real \( P(r) \) is an infinite series, if only \( K_1, K_2 \) and \( K_3 \) are given, we could derive a simpler relationship between \( K \) and \( K' \) by dynamic programming:

\[
K_1' = -K_1, \quad (3.7)
\]

\[
K_2' = 3K_1^2 - K_2, \quad (3.8)
\]

\[
K_3' = -12K_1^3 + 8K_1K_2 - K_3. \quad (3.9)
\]

Then, we can calibrate the cameras using:

\[
x_{\text{image}} = x_{\text{dist}} \left( 1 + K_1' r'^2 + K_2' r'^4 + K_3' r'^6 + \cdots \right), \quad (3.10)
\]

\[
y_{\text{image}} = y_{\text{dist}} \left( 1 + K_1' r'^2 + K_2' r'^4 + K_3' r'^6 + \cdots \right). \quad (3.11)
\]

In the next section, we will introduce the method of estimating the lens distortion coefficients.

### 3.2 Least squares to estimate lens distortion coefficients

Rearranging equations (2.13) and (2.14), we have:

\[
Y_1 = \frac{x_{\text{dist}} - x_{\text{image}}}{x_{\text{image}} r^2} = K_1 + K_2 r^2 + K_3 r^4 + \cdots, \quad (3.12)
\]
\[ Y_2 = \frac{y_{\text{dist}} - y_{\text{image}}}{y_{\text{image}}r^2} = K_1 + K_2r^2 + K_3r^4 + \cdots. \] (3.13)

Known points on the court are a suitable choice to perform the camera calibration. Assume there are \( N \) such data pairs measured. Each coordinates pair generates two equations as above. Write those \( r \) and \( Y_1, Y_2 \) in matrix form:

\[
X = \begin{bmatrix}
1 & 1 & 1 & 1 & \cdots & 1 & 1 \\
r_1^2 & r_1^2 & r_2^2 & \cdots & r_N^2 & r_N^2 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots
\end{bmatrix}^T,
\]

(3.14)

\[
Y = \begin{bmatrix}
Y_1^{(1)} & Y_2^{(1)} & Y_1^{(2)} & Y_2^{(2)} & \cdots & Y_1^{(N)} & Y_2^{(N)}
\end{bmatrix}^T
\]

(3.15)

Then, we apply linear regression to calculate actual distortion series \( K \):

\[
\begin{bmatrix}
K_1 \\
K_2 \\
\cdots
\end{bmatrix} = (X^TX)^{-1}X^TY.
\]

(3.16)

Here are coordinates of those court points used in this paper:

**Table 3.1: Court Coordinates**

<table>
<thead>
<tr>
<th>X(m)</th>
<th>0.0</th>
<th>0.0</th>
<th>0.0</th>
<th>-4.115</th>
<th>4.115</th>
<th>-4.115</th>
<th>4.115</th>
<th>-4.115</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y(m)</td>
<td>0.0</td>
<td>-6.4</td>
<td>6.4</td>
<td>6.4</td>
<td>6.4</td>
<td>-6.4</td>
<td>-6.4</td>
<td>11.885</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>X(m)</th>
<th>-4.115</th>
<th>4.115</th>
<th>4.115</th>
<th>5.485</th>
<th>5.485</th>
<th>-5.485</th>
<th>-5.485</th>
</tr>
</thead>
</table>
Chapter 4
N-Camera Triangulation Algorithm to Determine 3D Position

4.1 The distance between a ray $L$ and a point $P$

As defined in section 2.6, in the equation of a ray $L = C + kv$ $C$ represents the WCS coordinates of the camera’s center and $v = [v_x, v_y, v_z]^T$ is its direction. In Figure 4.1, the WCS coordinates of the point $P$ are $[x_0, y_0, z_0]^T$, and assume $PR$ perpendicular to the ray $L$, thus the length of $PR$ is:

$$\|PR\|_2 = \frac{\|CR \times CP\|_2}{\|CR\|_2} = \|v \times CP\|_2.$$  \hspace{1cm} (4.1)
4.2 Calculate the optimal point $P^*$

In Figure 4.2, $k$ cameras simultaneously produce an image of the point $P$, but the rays generated from those images do not necessarily intersect at a single point. Thus finding the optimal point $P^*$ which minimize the square sum of distance between $P^*$ and all those rays $L_i$ is needed. Assume the coordinates of the optimal point $P^*$ are $[x^*, y^*, z^*]^T$. The square of the distance between any $P$ and each ray $L_i$ is $\|PR_i\|_2^2$, the square sum of them is:

$$S(P) = \sum_{i=1}^{k} \|PR_i\|_2^2. \tag{4.2}$$

Therefore, when the $P$ is optimal, its coordinates $[x, y, z]^T$ should satisfy:

$$\frac{\partial S(P)}{\partial x} = 0, \tag{4.3}$$

$$\frac{\partial S(P)}{\partial y} = 0, \tag{4.4}$$

$$\frac{\partial S(P)}{\partial z} = 0. \tag{4.5}$$
Simplifying these three equations and writing in the form \( Ax = b \) [10]:

\[
\begin{bmatrix}
\sum_{i=1}^{k} (1 - v_x^{(i)}) & -\sum_{i=1}^{k} v_x^{(i)} v_y^{(i)} & -\sum_{i=1}^{k} v_x^{(i)} v_z^{(i)} \\
\sum_{i=1}^{k} v_x^{(i)} v_y^{(i)} & \sum_{i=1}^{k} (1 - v_y^{(i)}) & -\sum_{i=1}^{k} v_y^{(i)} v_z^{(i)} \\
-\sum_{i=1}^{k} v_x^{(i)} v_z^{(i)} & -\sum_{i=1}^{k} v_y^{(i)} v_z^{(i)} & \sum_{i=1}^{k} (1 - v_z^{(i)})
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z
\end{bmatrix}
= 
\begin{bmatrix}
\sum_{i=1}^{k} (1 - v_x^{(i)}) x_c^{(i)} - v_x^{(i)} v_y^{(i)} y_c^{(i)} - v_x^{(i)} v_z^{(i)} z_c^{(i)} \\
\sum_{i=1}^{k} v_x^{(i)} v_y^{(i)} x_c^{(i)} + (1 - v_y^{(i)}) y_c^{(i)} - v_y^{(i)} v_z^{(i)} z_c^{(i)} \\
-\sum_{i=1}^{k} v_x^{(i)} v_z^{(i)} x_c^{(i)} - v_y^{(i)} v_z^{(i)} y_c^{(i)} + (1 - v_z^{(i)}) z_c^{(i)}
\end{bmatrix}.
\]

(4.6)

By solving the linear equation above, the optimal point \( P^* \) is found.
Chapter 5
Results

In this paper, the camera models are implemented in Java, and the ball tracks are plotted in MATLAB. The related code and data of this paper are at: https://github.com/JLZhu2020/GraduateThesis.

5.1 Camera model with quantization error and lens distortion

![Graphs](figure5.1.png)

Figure 5.1: Real ball track

Figure 5.1 shows the real ball track in the world coordinate system. The data is in ball.dat. Next, we set the frame rate = 340 Hz to simulate the high speed camera. At each time step, we will get the ball position in WCS coordinates by cubic Hermite interpolation.
Figure 5.2: Error of quantized camera model

Figure 5.2 is error between real ball track and the ball track triangulated from the quantized camera model. The aim of this step is to see how the quantization error introduced by converting from image coordinates to pixel coordinates affects the ball track. The total error is the square root of the sum of the squares of the three individual errors. This gives a baseline for the accuracy if we just use the measurement of the system produced by the triangulation. The real system would also have a process to further reduce the error possibly by using a Kalman filter.

Figure 5.3: Error of distorted camera model
Figure 5.3 is error between real ball track and the ball track triangulated from quantized and distorted camera model. This is the worst case since no calibration is performed.

5.2 Camera calibrated by first-order inverse lens distortion series

We use $K_1 = -1.28 \times 10^{-4} \times \text{mm}^{-2}$ and $K_2 = 1.61 \times 10^{-6} \times \text{mm}^{-4}$ in equations (2.13) and (2.14) to simulate the lens distortion. We use the known court points in Table 3.1 to perform the calibration.

Using the least-squares algorithm to estimate $K_1$ and calculate inverse coefficients $K_1'$ by equation (3.7) for each camera, we have the results given in Table 5.1.

<table>
<thead>
<tr>
<th>Camera</th>
<th>Estimated $K_1 (\times \text{mm}^{-2})$</th>
<th>$K_1' (\times \text{mm}^{-2})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$-8.05 \times 10^{-5}$</td>
<td>$8.05 \times 10^{-5}$</td>
</tr>
<tr>
<td>2</td>
<td>$-8.05 \times 10^{-5}$</td>
<td>$8.05 \times 10^{-5}$</td>
</tr>
<tr>
<td>3</td>
<td>$-8.05 \times 10^{-5}$</td>
<td>$8.05 \times 10^{-5}$</td>
</tr>
<tr>
<td>4</td>
<td>$-8.05 \times 10^{-5}$</td>
<td>$8.05 \times 10^{-5}$</td>
</tr>
<tr>
<td>5</td>
<td>$-1.05 \times 10^{-4}$</td>
<td>$1.05 \times 10^{-4}$</td>
</tr>
<tr>
<td>6</td>
<td>$-1.22 \times 10^{-4}$</td>
<td>$1.22 \times 10^{-4}$</td>
</tr>
<tr>
<td>7</td>
<td>$-1.05 \times 10^{-4}$</td>
<td>$1.05 \times 10^{-4}$</td>
</tr>
<tr>
<td>8</td>
<td>$-1.22 \times 10^{-4}$</td>
<td>$1.22 \times 10^{-4}$</td>
</tr>
</tbody>
</table>

Figure 5.4 is error between real ball track and the ball track triangulated from cameras which are calibrated by first-order inverse lens distortion series.
Using the least-squares algorithm to estimate $K_1$ and $K_2$ and calculate inverse coefficients $K'_1$ and $K'_2$ by equation (3.7) to (3.8) for each camera, we have the results given in Table 5.2.

Table 5.2: Second-order distortion coefficients

<table>
<thead>
<tr>
<th>Camera</th>
<th>$K_1(\times mm^{-2})$</th>
<th>$K_2(\times mm^{-4})$</th>
<th>$K'_1(\times mm^{-2})$</th>
<th>$K'_2(\times mm^{-4})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$-1.29 \times 10^{-4}$</td>
<td>$2.40 \times 10^{-6}$</td>
<td>$1.29 \times 10^{-4}$</td>
<td>$-2.35 \times 10^{-6}$</td>
</tr>
<tr>
<td>2</td>
<td>$-1.29 \times 10^{-4}$</td>
<td>$2.40 \times 10^{-6}$</td>
<td>$1.29 \times 10^{-4}$</td>
<td>$-2.35 \times 10^{-6}$</td>
</tr>
<tr>
<td>3</td>
<td>$-1.29 \times 10^{-4}$</td>
<td>$2.40 \times 10^{-6}$</td>
<td>$1.29 \times 10^{-4}$</td>
<td>$-2.35 \times 10^{-6}$</td>
</tr>
<tr>
<td>4</td>
<td>$-1.29 \times 10^{-4}$</td>
<td>$2.40 \times 10^{-6}$</td>
<td>$1.29 \times 10^{-4}$</td>
<td>$-2.35 \times 10^{-6}$</td>
</tr>
<tr>
<td>5</td>
<td>$5.46 \times 10^{-4}$</td>
<td>$-6.27 \times 10^{-5}$</td>
<td>$-5.46 \times 10^{-4}$</td>
<td>$6.37 \times 10^{-5}$</td>
</tr>
<tr>
<td>6</td>
<td>$1.84 \times 10^{-6}$</td>
<td>$-6.14 \times 10^{-6}$</td>
<td>$-1.84 \times 10^{-6}$</td>
<td>$6.14 \times 10^{-6}$</td>
</tr>
<tr>
<td>7</td>
<td>$5.46 \times 10^{-4}$</td>
<td>$-6.27 \times 10^{-5}$</td>
<td>$-5.46 \times 10^{-4}$</td>
<td>$6.37 \times 10^{-5}$</td>
</tr>
<tr>
<td>8</td>
<td>$1.84 \times 10^{-6}$</td>
<td>$-6.14 \times 10^{-6}$</td>
<td>$-1.84 \times 10^{-6}$</td>
<td>$6.14 \times 10^{-6}$</td>
</tr>
</tbody>
</table>
Figure 5.5: Error of camera calibrated by second-order series

Figure 5.5 is error between real ball track and the ball track triangulated from cameras which are calibrated by second-order inverse lens distortion series.

5.4 Camera calibrated by third-order inverse lens distortion series

Using the least-squares algorithm to estimate $K_1$, $K_2$ and $K_3$ and calculate inverse coefficients $K_1'$, $K_2'$ and $K_3'$ by equation (3.7) to (3.9) for each camera, we have the results given in Table 5.3.

Table 5.3: Third-order distortion coefficients

<table>
<thead>
<tr>
<th>Camera</th>
<th>$K_1(\times mm^{-2})$</th>
<th>$K_2(\times mm^{-4})$</th>
<th>$K_3(\times mm^{-6})$</th>
<th>$K_1'(\times mm^{-2})$</th>
<th>$K_2'(\times mm^{-4})$</th>
<th>$K_3'(\times mm^{-6})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$-2.53 \times 10^{-4}$</td>
<td>$1.91 \times 10^{-5}$</td>
<td>$-4.96 \times 10^{-7}$</td>
<td>$2.53 \times 10^{-4}$</td>
<td>$-1.89 \times 10^{-5}$</td>
<td>$4.57 \times 10^{-7}$</td>
</tr>
<tr>
<td>2</td>
<td>$-2.53 \times 10^{-4}$</td>
<td>$1.91 \times 10^{-5}$</td>
<td>$-4.96 \times 10^{-7}$</td>
<td>$2.53 \times 10^{-4}$</td>
<td>$-1.89 \times 10^{-5}$</td>
<td>$4.57 \times 10^{-7}$</td>
</tr>
<tr>
<td>3</td>
<td>$-2.53 \times 10^{-4}$</td>
<td>$1.91 \times 10^{-5}$</td>
<td>$-4.96 \times 10^{-7}$</td>
<td>$2.53 \times 10^{-4}$</td>
<td>$-1.89 \times 10^{-5}$</td>
<td>$4.57 \times 10^{-7}$</td>
</tr>
<tr>
<td>4</td>
<td>$-2.53 \times 10^{-4}$</td>
<td>$1.91 \times 10^{-5}$</td>
<td>$-4.96 \times 10^{-7}$</td>
<td>$2.53 \times 10^{-4}$</td>
<td>$-1.89 \times 10^{-5}$</td>
<td>$4.57 \times 10^{-7}$</td>
</tr>
<tr>
<td>5</td>
<td>$-6.22 \times 10^{-4}$</td>
<td>$3.09 \times 10^{-4}$</td>
<td>$-2.57 \times 10^{-5}$</td>
<td>$6.22 \times 10^{-4}$</td>
<td>$-3.08 \times 10^{-4}$</td>
<td>$2.42 \times 10^{-5}$</td>
</tr>
<tr>
<td>6</td>
<td>$-3.67 \times 10^{-4}$</td>
<td>$4.03 \times 10^{-5}$</td>
<td>$-1.35 \times 10^{-6}$</td>
<td>$3.67 \times 10^{-4}$</td>
<td>$-3.99 \times 10^{-5}$</td>
<td>$1.23 \times 10^{-6}$</td>
</tr>
<tr>
<td>7</td>
<td>$-6.22 \times 10^{-4}$</td>
<td>$3.09 \times 10^{-4}$</td>
<td>$-2.57 \times 10^{-5}$</td>
<td>$6.22 \times 10^{-4}$</td>
<td>$-3.08 \times 10^{-4}$</td>
<td>$2.42 \times 10^{-5}$</td>
</tr>
<tr>
<td>8</td>
<td>$-3.67 \times 10^{-4}$</td>
<td>$4.03 \times 10^{-5}$</td>
<td>$-1.35 \times 10^{-6}$</td>
<td>$3.67 \times 10^{-4}$</td>
<td>$-3.99 \times 10^{-5}$</td>
<td>$1.23 \times 10^{-6}$</td>
</tr>
</tbody>
</table>

Figure 5.6 is error between real ball track and the ball track triangulated from
cameras which are calibrated by third-order inverse lens distortion series.

![Graphs of error for different calibration orders](image)

**Figure 5.6: Error of camera calibrated by third-order series**

### 5.5 Conclusion

As a final comparison of the different order calibrations, we compare the maximum error for each order.

<table>
<thead>
<tr>
<th>Camera</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Only quantized</td>
<td>15.23</td>
<td>14.07</td>
<td>11.51</td>
<td>17.41</td>
</tr>
<tr>
<td>Distorted</td>
<td>15.23</td>
<td>21.97</td>
<td>12.83</td>
<td>22.32</td>
</tr>
<tr>
<td>First-order calibrated</td>
<td>15.23</td>
<td>13.51</td>
<td>12.09</td>
<td>17.40</td>
</tr>
<tr>
<td>Second-order calibrated</td>
<td>15.23</td>
<td>13.66</td>
<td>12.87</td>
<td>17.41</td>
</tr>
<tr>
<td>Third-order calibrated</td>
<td>15.23</td>
<td>13.41</td>
<td>11.83</td>
<td>17.39</td>
</tr>
</tbody>
</table>

Table 5.4: Maximum error between triangulated track with real track

In Table 5.4, obviously all of calibrated camera models almost eliminate the lens distortion. However, cameras calibrated by different order series perform nearly the same. We can also see this looking at Figures 5.4 to 5.6. Therefore, camera calibration could not reduce the error which caused by quantization. We did not expect the
calibration to correct this error since it is not due to lens distortion. Also, the lens distortion modeled here is small. If the distortion were larger, then we would probably see a bigger difference between the different order calibrations.
References


