



EÖTVÖS LORÁND UNIVERSITY

FACULTY OF INFORMATICS

DEPT. OF ALGORITHMS AND APPLICATIONS

Oriented Ellipse Fitting and 3D Circle Center Estimation

Author:

Xiaohe Du

Computer Science for Autonomous Systems MSc

Supervisor:

Dr. Levente Hajder

Senior Research Fellow

Colleague:

Martin Sallai

Software Engineer

Colleague:

Bandó Kovács

Hardware Engineer

Budapest, 2024

Contents

1	Abstract	3
2	Introduction	4
2.1	Contributions	6
3	Proposed Algorithms	7
3.1	RANSAC-based Oriented Ellipse Fitting	7
3.1.1	RANSAC	7
3.1.2	Oriented Ellipse fitting	9
3.2	3D Circle Center Estimation	13
3.2.1	Circle Center Estimation in 2D	13
3.2.2	Closest point of 3D skew lines	16
4	Implementation	18
4.1	Experimental Setup	18
4.2	Edge Detection	19
4.3	Line and Corner Removal	21
4.4	Edge Segmentation and Contour Detection	21
4.5	Contour Point Down Sampling	23
4.6	Sample Point Filtering	23
4.7	RANSAC Oriented Ellipse Fitting	24
4.8	Camera Calibration	26
4.9	3D Circle Center Estimation	26
5	Experimental Results	28
5.1	Ellipse Fitting Results on Simulated Data	28
5.1.1	Fitting with Partial Ellipse Points	28
5.1.2	Fitting with Noisy Ellipse Data Set	31
5.1.3	Fitting with Noisy Gradients	33

5.2	Ellipse Fitting and Center Estimation Results on Real Images	35
5.2.1	Fitting Results in Different Scenarios	35
5.2.2	Results of Reel Center Estimation in Image	35
5.2.3	Results of 3D Reel Center Estimation	36
5.2.4	Results Comparison of Ellipse Fitting Algorithms	37
6	Conclusion	44
6.1	Future Work	45
	Acknowledgements	46
	References	46
	List of Figures	49
	List of Tables	51
	List of Algorithms	52

Chapter 1

Abstract

In man-made urban environments, objects containing circular surfaces are commonly found in a variety of different scenes, such as wheels, capacitors and round buttons, etc. It is well-known that the projected contour of a 3D circle form an ellipse on the camera image. This paper proposes a novel method to detect the circle by using RANSAC-based oriented ellipse fitting algorithm and determine the center of circle in 3D using two images. The test environment of the method is an experimental automated electronic components unpacking system. In this system the robotic arm first takes two images of a reel from different angles using a camera mounted on the robotic arm. Then applies oriented ellipse fitting algorithm and center estimation on images to locate the center of the reel in 3D. With given 3D location of the reel center, the robotic arm can use special tools to grab the reel through the central hole after that. This paper also introduces the complete implementation process and explain each step in detail. In addition, the proposed algorithm is compared with another ellipse fitting algorithm in terms of accuracy.

Chapter 2

Introduction

This paper starts from a lab project, which is an automated parcel unpacking system. The ultimate goal of the project is to use a robotic arm to open parcel boxes and take out the reels of electronic components from inside automatically.

The part our team needs to complete in this project is to make sure the robotic arm can locate and grab the reels. Figure 2.1 shows three samples of the reel. Considering that reels come in different sizes and styles, including some with hollow or uneven surfaces, it is not practical to use a suction cup tool to grab whole surface of the reel. However, all types of reels have one feature in common, that is, the shape of the central hole is cylindrical. Based on this, the solution we came up with was to position the center hole of the reel so that the reel can be grabbed by the robotic arm with a specific tool (will be explained in chapter 4). And the only sensor we use is a camera mounted on the robotic arm. Therefore, all the information we can take as input is: images, camera intrinsic parameters, rotation and translation parameters of camera, which is enough to extract 3D center of reel surface by processing these data. The brief processes are illustrated in Figure 2.2. The red parts in the flowchart are the novel algorithms mainly introduced in this paper.



Figure 2.1: Samples of the reel

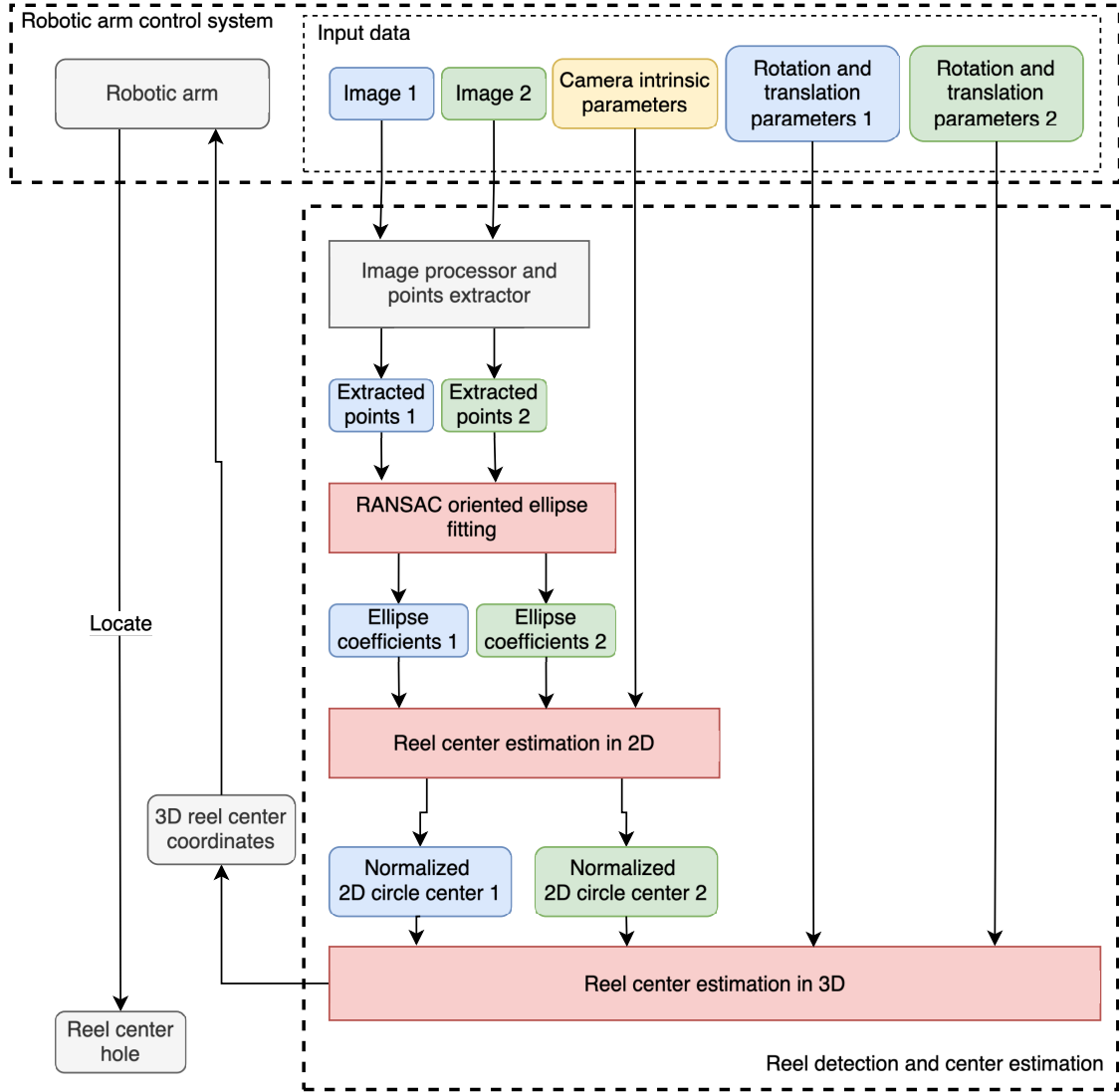


Figure 2.2: Concise flowchart of processes from input data to robot arm positioning the center hole of the reel

2.1 Contributions

Our work presents two primary contributions. Firstly, we introduce a novel algorithm: RANSAC-based oriented ellipse fitting, specifically designed to identify the projection of reel surface edge within an image and extract their corresponding ellipse coefficients. Secondly, we propose an algorithm for estimating the 3D center of circular objects. This algorithm comprises two key components: the first component utilizes ellipse coefficients and camera intrinsic parameters to compute the projected coordinates of the reel center in the image, while the second component leverages the projected center points from two images, obtained through the initial computation, along with camera rotation and translation parameters, to determine the 3D coordinates of the center of reel's surface.

Additionally, several contributions are also made in the image processing part. For binary images, we devise a method that combines probabilistic Hough transform[8] and Harris corner detection[9] to remove straight lines and sharp corners. To enhance overall result accuracy prior to the ellipse fitting process, we exploit the properties of simple polygon approximation algorithm[10, 11] to perform edge segmentation. Furthermore, we introduce a novel down sampling technique for image contours comprising step size indexing and radius point removal, which significantly contributes to accelerating ellipse fitting operations.

Chapter 3

Proposed Algorithms

There are two novel algorithms involved in this project: RANSAC-based oriented ellipse fitting and 3D circle center estimation. The mathematics of both algorithms are explained in this section.

3.1 RANSAC-based Oriented Ellipse Fitting

This algorithm is used for finding ellipse from points extracted from an image and obtaining the coefficients of the ellipse. This algorithm is composed of two sub-algorithms nested in each other.

3.1.1 RANSAC

RANSAC (Random Sample Consensus)[1] is an iterative algorithm used to estimate parameters of a mathematical model from a set of observed data that contains outliers. As a general algorithm framework, RANSAC can embed almost any mathematical model such as line, ellipse, sphere, etc. into itself. Thus, it's commonly used in computer vision, image processing, and machine learning for tasks like fitting models to data points corrupted by noise or outliers.

In this project, we modified RANSAC to better suit our specific needs, necessitating modifications to its conventional steps. Here's the revised pseudocode reflecting these alterations:

Algorithm 1 Modified RANSAC Algorithm

Data: Model function for fitting $\mathcal{M}$, a set of points P , number of sample points used for model fitting in each iteration n , initial maximum number of iterations N , threshold for determining if a point is inlier ϵ , inlier checking function $\mathcal{E}$, extra discriminant function to determine if a fitted model is eligible $\mathcal{D}$, confidence used for recalculating max iteration number c

Result: Best model, best samples, best inliers

```

bestModel  $\leftarrow \emptyset$ 
bestSamples  $\leftarrow \emptyset$ 
bestInliers  $\leftarrow \emptyset$ 
i  $\leftarrow 0$ 
while i < N do
    samples  $\leftarrow n$  random samples from P.
    model  $\leftarrow$  fit a model by  $\mathcal{M}(\textit{samples})$ .
    inliers  $\leftarrow$  get inliers by inlier checking function  $\mathcal{E}(\textit{model}, P, \epsilon)$ .
    if  $\text{length}(\textit{inliers}) > \text{length}(\textit{bestInliers})$  and  $\mathcal{D}(\textit{model})$  then
        bestModel  $\leftarrow \textit{model}$ 
        bestSamples  $\leftarrow \textit{samples}$ 
        bestInliers  $\leftarrow \textit{inliers}$ 
        N  $\leftarrow$  minimum between N and max iteration number from (3.1.1)
    end
    i  $\leftarrow i + 1$ 
end
while  $\text{length}(\textit{bestInliers}) > \text{length}(\textit{bestSamples})$  do
    samples  $\leftarrow \textit{bestInliers}$ 
    model  $\leftarrow \mathcal{M}(\textit{samples})$ 
    inliers  $\leftarrow \mathcal{E}(\textit{model}, P, \epsilon)$ 
    if  $\mathcal{D}(\textit{model})$  then
        bestModel  $\leftarrow \textit{model}$ 
        bestSamples  $\leftarrow \textit{samples}$ 
        bestInliers  $\leftarrow \textit{inliers}$ 
    end
    else
        | Break
    end
end
return bestModel, bestSamples, bestInliers

```

The equation of calculating maximum ransac iteration number is:

$$N = \frac{\log(1 - c)}{\log(1 - \alpha^s)} \quad (3.1)$$

where c is confidence, α is inlier ratio and s is sample size.

It is worth mentioning that in the initial version, another ellipse fitting algo-

rithm(improved Fitzgibbon algorithm[2], is called direct ellipse fitting in the rest of the paper) is nested within RANSAC which requires a minimum of 5 sample points, while only 3 sample points are required for our novel algorithm. By comparing the difference in maximum iterations between 3 and 5 sample points based on (3.1.1), we can get Table 3.1, which demonstrates that the maximum iteration number of RANSAC is significantly lower with 3 sample points compared to 5 across all inlier ratios. This observation underscores the shorter computation time of oriented ellipse fitting when nested within RANSAC compared to the previous algorithm.

Inlier ratio(α)	Max iterations ($s = 3$)	Max iterations ($s = 5$)
0.15	1363	60642
0.3	169	1893
0.45	49	248
0.6	19	57
0.75	9	17
0.9	4	6

Table 3.1: Max iterations between 3 and 5 sample points in different inlier ratio with confidence(c) = 0.99

3.1.2 Oriented Ellipse fitting

In early version we used direct ellipse fitting[2] to get the ellipse coefficients from images. However, this algorithm has the following problems: Since the scroll has two circular planes, it is common for the edges of both circular planes taken in an image, resulting in false fitting; Another problem is that at least 5 sample points are required to fit an ellipse, and more sample points mean that the points randomly sampled from the data set are less likely to belong to the same object, which means more iterations required to fit the best model in RANSAC, resulting in longer calculation time. To overcome these problems, we proposed another novel algorithm, oriented ellipse fitting.

The oriented ellipse fitting algorithm is similar to [2] but contains additional gradient information for calculating ellipse coefficients. Compared with the direct ellipse fitting algorithm, with the additional help of point gradients, oriented ellipse fitting can obtain higher fitting accuracy by checking the error between the fitted ellipse and the gradient of the sample points. Furthermore,

the iterations to find best model in RANSAC is also less due to only 3 samples are required. The following formulas explain the mechanics of oriented ellipse fitting.

The general ellipse equation is as follows:

$$F(x, y) = ax^2 + bxy + cy^2 + dx + ey + f = 0 \quad (3.2)$$

with constraints:

$$b^2 - 4ac = -1. \quad (3.3)$$

The gradients are:

$$\nabla F = \begin{bmatrix} \frac{\partial F}{\partial x} \\ \frac{\partial F}{\partial y} \end{bmatrix} = \begin{bmatrix} 2ax + by + d \\ bx + 2cy + e \end{bmatrix} = \lambda \begin{bmatrix} n_x \\ n_y \end{bmatrix} \quad (3.4)$$

where (n_x, n_y) is normal vector of point (x, y) .

So one oriented point with (3.2) and (3.4) can form the equation:

$$\begin{bmatrix} x^2 & xy & y^2 & x & y & 1 & 0 & 0 \\ 2x & y & 0 & 1 & 0 & 0 & -n_x & 0 \\ 0 & x & 2y & 0 & 1 & 0 & -n_y & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ \lambda \end{bmatrix} = 0.$$

For multiple points:

$$\begin{bmatrix} x_1^2 & x_1y_1 & y_1^2 & x_1 & y_1 & 1 & 0 & 0 & \dots & 0 \\ 2x_1 & y_1 & 0 & 1 & 0 & 0 & -n_{1x} & 0 & \dots & 0 \\ 0 & x_1 & 2y_1 & 0 & 1 & 0 & -n_{1y} & 0 & \dots & 0 \\ x_2^2 & x_2y_2 & y_2^2 & x_2 & y_2 & 1 & 0 & 0 & \dots & 0 \\ 2x_2 & y_2 & 0 & 1 & 0 & 0 & 0 & -n_{2x} & \dots & 0 \\ 0 & x_2 & 2y_2 & 0 & 1 & 0 & 0 & -n_{2y} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ x_N^2 & x_Ny_N & y_N^2 & x_N & y_N & 1 & 0 & 0 & \dots & 0 \\ 2x_N & y_N & 0 & 1 & 0 & 0 & 0 & 0 & \dots & -n_{Nx} \\ 0 & x_N & 2y_N & 0 & 1 & 0 & 0 & 0 & \dots & -n_{Ny} \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_N \end{bmatrix} = 0. \quad (3.5)$$

With $N = 3$, there are nine unknown elements and nine equations, therefore the problem is solvable.

Moreover, to make the algorithm numerically stable, a similar approach that is used in direct ellipse fitting algorithm[2] is applied. To achieve this, we can mark (3.5) as $\mathbf{D}\mathbf{v} = \mathbf{0}$, then the equation is rewritten as:

$$\mathbf{D}\mathbf{v} = [\mathbf{D}_1 \quad \mathbf{D}_2 \quad \mathbf{D}_3] \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \end{bmatrix} = \mathbf{0}$$

where

$$\mathbf{D}_1 = \begin{bmatrix} x_1^2 & x_1 y_1 & y_1^2 \\ 2x_1 & y_1 & 0 \\ 0 & x_1 & 2y_1 \\ x_2^2 & x_2 & y_2^2 \\ 2x_2 & y_2 & 0 \\ 0 & x_2 & 2y_2 \\ \vdots & \vdots & \vdots \\ x_N^2 & x_N & y_N^2 \\ 2x_N & y_N & 0 \\ 0 & x_N & 2y_N \end{bmatrix}, \mathbf{D}_2 = \begin{bmatrix} x_1 & y_1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ x_2 & y_2 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ \vdots & \vdots & \vdots \\ x_N & y_N & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \mathbf{D}_3 = \begin{bmatrix} 0 & 0 & \dots & 0 \\ -n_{1x} & 0 & \dots & 0 \\ -n_{1y} & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & -n_{2x} & \dots & 0 \\ 0 & -n_{2y} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & -n_{Nx} \\ 0 & 0 & \dots & -n_{Ny} \end{bmatrix},$$

$$\mathbf{v}_1 = \begin{bmatrix} a \\ b \\ c \end{bmatrix}, \mathbf{v}_2 = \begin{bmatrix} d \\ e \\ f \end{bmatrix}, \mathbf{v}_3 = \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_N \end{bmatrix}.$$

With constraints rewritten from (3.3):

$$\mathbf{v}_1^T \mathbf{C} \mathbf{v}_1 = -1 \tag{3.6}$$

where

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & -2 \\ 0 & 1 & 0 \\ -2 & 0 & 0 \end{bmatrix}.$$

Thus the problem is to minimize:

$$\left\| [\mathbf{D}_1 \quad \mathbf{D}_2 \quad \mathbf{D}_3] \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \end{bmatrix} \right\|_2^2 = 0$$

subject to (3.6).

This can be solved by applying Lagrangian multiplier as:

$$J = \mathbf{v}^T \mathbf{D}^T \mathbf{D} \mathbf{v} + \lambda(\mathbf{v}_1^T \mathbf{C} \mathbf{v}_1 + 1). \quad (3.7)$$

Remark that $\mathbf{D}^T \mathbf{D}$ is a symmetric matrix.

Moreover,

$$\begin{aligned} \mathbf{v}^T \mathbf{D}^T \mathbf{D} \mathbf{v} &= \mathbf{v}_1^T \mathbf{D}_1^T \mathbf{D}_1 \mathbf{v}_1 + \mathbf{v}_2^T \mathbf{D}_2^T \mathbf{D}_2 \mathbf{v}_2 + \mathbf{v}_3^T \mathbf{D}_3^T \mathbf{D}_3 \mathbf{v}_3 \\ &\quad + 2(\mathbf{v}_1^T \mathbf{D}_1^T \mathbf{D}_2 \mathbf{v}_2 + \mathbf{v}_1^T \mathbf{D}_1^T \mathbf{D}_3 \mathbf{v}_3 + \mathbf{v}_2^T \mathbf{D}_2^T \mathbf{D}_3 \mathbf{v}_3). \end{aligned}$$

The cost function can be minimized via computing the derivatives:

$$\frac{\partial J}{\partial \mathbf{v}_1} = 2\mathbf{D}_1^T \mathbf{D}_1 \mathbf{v}_1 + 2\mathbf{D}_1^T \mathbf{D}_2 \mathbf{v}_2 + 2\mathbf{D}_1^T \mathbf{D}_3 \mathbf{v}_3 + 2\lambda \mathbf{C} \mathbf{v}_1 = 0 \quad (3.8)$$

$$\frac{\partial J}{\partial \mathbf{v}_2} = 2\mathbf{D}_2^T \mathbf{D}_1 \mathbf{v}_1 + 2\mathbf{D}_2^T \mathbf{D}_2 \mathbf{v}_2 + 2\mathbf{D}_2^T \mathbf{D}_3 \mathbf{v}_3 = 0 \quad (3.9)$$

$$\frac{\partial J}{\partial \mathbf{v}_3} = 2\mathbf{D}_3^T \mathbf{D}_1 \mathbf{v}_1 + 2\mathbf{D}_3^T \mathbf{D}_2 \mathbf{v}_2 + 2\mathbf{D}_3^T \mathbf{D}_3 \mathbf{v}_3 = 0 \quad (3.10)$$

From (3.10), $\mathbf{v}_3$ can be expressed as:

$$\mathbf{v}_3 = -(\mathbf{D}_3^T \mathbf{D}_3)^{-1} \mathbf{D}_3^T (\mathbf{D}_1 \mathbf{v}_1 + \mathbf{D}_2 \mathbf{v}_2)$$

where

$$\mathbf{D}_3^T \mathbf{D}_3 = \begin{bmatrix} \|n_1\|^2 & 0 & \dots & 0 \\ 0 & \|n_2\|^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \|n_N\|^2 \end{bmatrix}.$$

If the normals $n_1, n_2, \dots, n_N$ have unit length, this is an identity matrix. In this case,

$$\mathbf{v}_3 = -\mathbf{D}_3^T (\mathbf{D}_1 \mathbf{v}_1 + \mathbf{D}_2 \mathbf{v}_2). \quad (3.11)$$

By combining (3.8), (3.9), (3.11) we can get equations:

$$\begin{aligned} \lambda \mathbf{C} \mathbf{v}_1 &= \mathbf{D}_1^T [\mathbf{L} \mathbf{D}_2 (\mathbf{D}_2^T \mathbf{L} \mathbf{D}_2)^{-1} \mathbf{D}_2^T - \mathbf{I}] \mathbf{L} \mathbf{D}_1 \mathbf{v}_1 \\ \mathbf{v}_2 &= -(\mathbf{D}_2^T \mathbf{L} \mathbf{D}_2)^{-1} \mathbf{D}_2^T \mathbf{L} \mathbf{D}_1 \mathbf{v}_1 \end{aligned} \quad (3.12)$$

Where

$$\mathbf{L} = \mathbf{I} - \mathbf{D}_3 \mathbf{D}_3^T.$$

Finally, the coefficients of ellipse $\mathbf{v}_1, \mathbf{v}_2$ can be yield by solving the system of equations (3.12).

3.2 3D Circle Center Estimation

The function of this algorithm is to derive the 3D coordinates of the circle center by inputting the two ellipse coefficients obtained from RANSAC oriented ellipse fitting and the intrinsic and extrinsic parameters of the camera. The algorithm also consists of two parts: circle center estimation in 2D and closest point of 3D skew lines.

3.2.1 Circle Center Estimation in 2D

This algorithm is capable of estimating projected circle center from the image, which is based on the work of Alvarez et al[3]. With extra constraints which is derived from the following equations, the projection of a 3D circle center on image plane can be decided within two candidate points if the coefficients of ellipse and the intrinsic parameters of camera are known. Moreover, if the relative rotation angle from camera to circle plane is known. The true projection of circle center can be determined.

Based on Alvarez algorithm, a unit circle can be transformed and projected into an ellipse on camera image with homography matrix:

$$\mathbf{H} = \mathbf{O}^T \mathbf{\Lambda} \mathbf{R}(\alpha) \mathbf{B}(a) \mathbf{R}(\beta) \quad (3.13)$$

where $\mathbf{O}$ is an orthogonal matrix obtained from the eigendecomposition of matrix built from ellipse coefficients (3.2):

$$\begin{bmatrix} a & c/2 & d/2 \\ c/2 & b & e/2 \\ d/2 & e/2 & f \end{bmatrix} = \mathbf{O}^T \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \mathbf{O}.$$

Moreover,

$$\mathbf{\Lambda} = \begin{bmatrix} \frac{1}{\sqrt{\lambda_1}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{\lambda_2}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{\lambda_3}} \end{bmatrix},$$

$$\mathbf{R}(z) = \begin{bmatrix} \cos(z) & -\sin(z) & 0 \\ \sin(z) & \cos(z) & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{B}(a) = \begin{bmatrix} \sqrt{1+a^2} & 0 & a \\ 0 & 1 & 0 \\ a & 0 & \sqrt{1+a^2} \end{bmatrix}.$$

As a 3D circle is a planar object, the projection matrix can also be represented as:

$$\mathbf{H} = \mathbf{K}[s\tilde{\mathbf{R}}|\mathbf{t}],$$

where $\mathbf{K}$ is the intrinsic matrix of calibrated camera. $\tilde{\mathbf{R}}$ and $\mathbf{t}$ are rotation and translation from circle plane to camera coordinates respectively. Since the camera is calibrated, the image coordinates can be normalized by multiplying $\mathbf{K}^{-1}$ with homogeneous form of image points. Therefore based on this, we assume that $\mathbf{K} = \mathbf{I}$. Then

$$[s\tilde{\mathbf{R}}|\mathbf{t}] = \mathbf{O}^T \mathbf{\Lambda} \mathbf{R}(\alpha) \mathbf{B}(a) \mathbf{R}(\beta).$$

By left-multiplying $\mathbf{O}$ and right-multiplying $\mathbf{R}^T(\beta)$ on both sides, we can get:

$$[s\mathbf{O}\tilde{\mathbf{R}}|\mathbf{O}\mathbf{t}] \begin{bmatrix} \cos(\beta) & \sin(\beta) & 0 \\ -\sin(\beta) & \cos(\beta) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \mathbf{\Lambda} \mathbf{R}(\alpha) \mathbf{B}(a).$$

The first two columns of the matrix on the left is as follows:

$$[s\mathbf{O}\tilde{\mathbf{R}}|\mathbf{O}\mathbf{t}] \begin{bmatrix} \cos(\beta) & \sin(\beta) \\ -\sin(\beta) & \cos(\beta) \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} sr_1 & sr_4 & a_x \\ sr_2 & sr_5 & a_y \\ sr_3 & sr_6 & a_z \end{bmatrix} \begin{bmatrix} \cos(\beta) & \sin(\beta) \\ -\sin(\beta) & \cos(\beta) \\ 0 & 0 \end{bmatrix}.$$

The result is not influenced by the vector (a_x, a_y, a_z) , therefore $[s\mathbf{O}\tilde{\mathbf{R}}|\mathbf{O}\mathbf{t}]$ is also a scaled rotation. In this case, the rest two columns of original equation becomes

$$s\mathbf{R}_2 = \mathbf{\Lambda} \mathbf{R}(\alpha) \tilde{\mathbf{B}}(a),$$

where

$$\tilde{\mathbf{B}}(a) = \begin{bmatrix} \sqrt{1+a^2} & 0 \\ 0 & 1 \\ a & 0 \end{bmatrix},$$

and sR_2 is a scaled orthogonal matrix. Thus

$$s\mathbf{R}_2 = \mathbf{\Lambda}\mathbf{R}(\alpha)\tilde{\mathbf{B}}(a) = \begin{bmatrix} \frac{\cos(\alpha)}{\sqrt{\lambda_1}}\sqrt{1+a^2} & -\frac{\sin(\alpha)}{\sqrt{\lambda_1}} \\ \frac{\sin(\alpha)}{\sqrt{\lambda_2}}\sqrt{1+a^2} & \frac{\cos(\alpha)}{\sqrt{\lambda_2}} \\ \frac{a}{\sqrt{\lambda_3}} & 0 \end{bmatrix}. \quad (3.14)$$

This is a scaled orthonormal matrix. Thus the two columns should be perpendicular to each other and the lengths should be the same. Based on perpendicular constraint we have the equation:

$$\frac{\cos(\alpha)\sin(\alpha)}{\lambda_2}\sqrt{1+a^2} = \frac{\cos(\alpha)\sin(\alpha)}{\lambda_1}\sqrt{1+a^2}. \quad (3.15)$$

Generally, $\lambda_1 \neq \lambda_2$ as the length of ellipse major axis and minor axis are not equal. Therefore the equation holds only if $\cos(\alpha) = 0$ or $\sin(\alpha) = 0$. However, consider the case of $\cos(\alpha) = 0$ or $\alpha = \{\pi/2, 3\pi/2\}$:

$$\mathbf{R}\left(\frac{\pi}{2}\right) = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{R}\left(\frac{3\pi}{2}\right) = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Both of these two matrices will result in permutation of coordinate axes and break the correspondence between $\mathbf{O}^T$ and $\mathbf{\Lambda}$ in (3.13). Thus the equation holds only if $\alpha = \{0, \pi\}$.

There is another constraint, the lengths of the columns in matrix (3.14) should be equal to each other. For the case of $\alpha = 0$ and $\alpha = \pi$, the final solution of a are both the same:

$$a = \pm \sqrt{\frac{\lambda_3(\lambda_1 - \lambda_2)}{\lambda_2(\lambda_3 + \lambda_1)}}.$$

Based on all the conditions above, there are two possible homography matrices that project the transformed center of unit circle (0,0,1) to image plane:

$$\begin{aligned} \mathbf{H}_1 &= \mathbf{O}^T \mathbf{\Lambda} \mathbf{R}(\alpha) \mathbf{B}(a) \mathbf{R}(\beta) \\ \mathbf{H}_2 &= \mathbf{O}^T \mathbf{\Lambda} \mathbf{R}(\alpha) \mathbf{B}(-a) \mathbf{R}(\beta) \end{aligned} \quad (3.16)$$

Where $\alpha = 0$ or $\alpha = \pi$, both get the same result. β can be any angle. Then

$$a = \sqrt{\frac{\lambda_3(\lambda_1 - \lambda_2)}{\lambda_2(\lambda_3 + \lambda_1)}}.$$

Thus, the difference between $\mathbf{H}_1$ and $\mathbf{H}_2$ is determined by $\mathbf{B}(a)$.

The reason why the center of a circle cannot be uniquely determined can be explained by introducing four reference points $(1, 0, 1), (0.5, 0, 1), (0, 0, 1), (-1, 0, 1)$ on the unit circle coordinate system. The same result will be obtained by calculating the cross ratio after transforming the points through $\mathbf{H}_1$ and $\mathbf{H}_2$. Combined with the principle of projection invariance, the matrices $\mathbf{H}_1$ and $\mathbf{H}_2$ can be regarded as transforming the unit circle into two different planes in 3D but have same ellipse on camera view plane.

Figure 3.1 demonstrates the situation when the unique circle center is determined. Suppose there exists a transformation matrix $\mathbf{H}$ that transforms the unit circle to a circle that perfectly parallels to image plane. In this case the projection on image plane is also a circle and therefore no center ambiguity which means $a = 0, \mathbf{H}_1 = \mathbf{H}_2 = \mathbf{H}$. However, it is almost impossible for this situation to happen in reality. The common situation is shown in Figure 3.1b. Same circle with two different orientations can produce the same elliptical projection on the camera image plane but with different centers. The green and red line segments represent the projection of the reference points on the YZ plane after transformed by $\mathbf{H}_1$ and $\mathbf{H}_2$ respectively. Since we only consider YZ plane in camera coordinate system. If the bottom of camera image plane is closer to circle plane then the normalized center projection of the circle usually has lower y value, i.e. the right image of Figure 3.1b. If the top of camera image plane is closer to the surface of circle then the other candidate center should be chosen. Based on this, the center of circle can be uniquely determined if the angle of the image plane relative to the circular plane is known.

3.2.2 Closest point of 3D skew lines

By calculating the homography matrix on minimum two images, combined with corresponding extrinsic matrices of the camera, the circle center in 3D is able to be recovered.

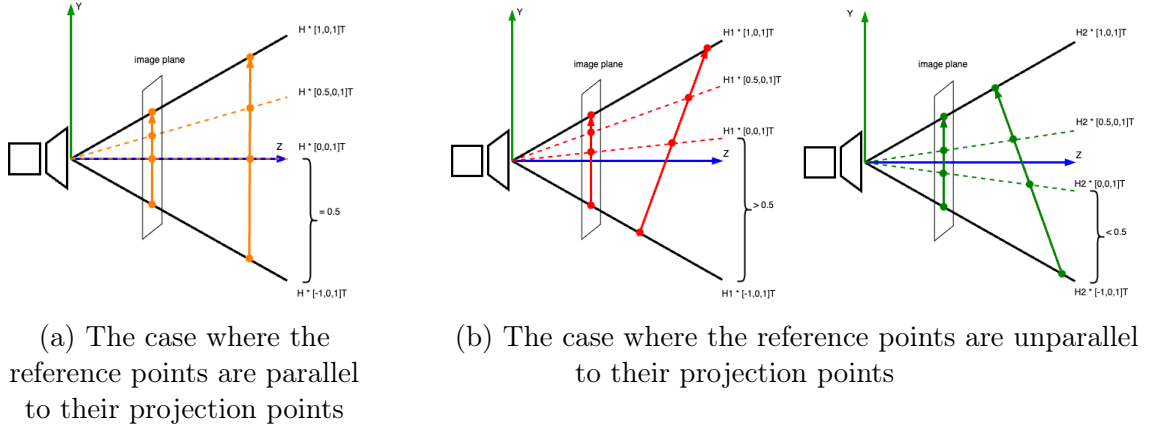


Figure 3.1: Possible projection results of the reference points

Suppose homography matrices calculated from two images are $\mathbf{H}_a, \mathbf{H}_b$. Then the normalized circle center projection points are:

$$\mathbf{p}_1 = \mathbf{H}_a \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \mathbf{p}_1 = \frac{\mathbf{p}_1}{\|\mathbf{p}_1\|}, \quad \mathbf{p}_2 = \mathbf{H}_b \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \mathbf{p}_2 = \frac{\mathbf{p}_2}{\|\mathbf{p}_2\|}.$$

The corresponding 4x4 world to camera transformation matrices $\mathbf{T}_1, \mathbf{T}_2$ can be received from robotic arm system. With multiplication of focal point $(0, 0, 0, 1)$ and transformation matrices we can get focal point in the world coordinate system:

$$\mathbf{a}_1 = \mathbf{T}_1 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad \mathbf{a}_2 = \mathbf{T}_2 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}. \quad (3.17)$$

Also the projected circle center in the world coordinate system:

$$\mathbf{p}'_1 = \mathbf{T}_1 \begin{bmatrix} \mathbf{p}_1 \\ 1 \end{bmatrix}, \quad \mathbf{p}'_2 = \mathbf{T}_2 \begin{bmatrix} \mathbf{p}_2 \\ 1 \end{bmatrix}. \quad (3.18)$$

Two lines can be built from (3.17) and (3.18):

$$\mathbf{L}_1 = \mathbf{a}_1 + t_1 \frac{\mathbf{p}'_1 - \mathbf{a}_1}{\|\mathbf{p}'_1 - \mathbf{a}_1\|}, \quad \mathbf{L}_2 = \mathbf{a}_2 + t_2 \frac{\mathbf{p}'_2 - \mathbf{a}_2}{\|\mathbf{p}'_2 - \mathbf{a}_2\|}. \quad (3.19)$$

Finally, by using the algorithm in [4] which allows for the computation of the point that has the minimum sum of perpendicular distances to all skew lines, the 3D center of the circle can be obtained.

Chapter 4

Implementation

This chapter introduces the experimental setup of the project and the entire process of the program. To enable the application of our algorithms to images, a multi-step processing procedure is initially conducted on the image to extract sample points for subsequent RANSAC oriented ellipse fitting and latter steps.

4.1 Experimental Setup

The hardware setup is shown as Figure 4.1 below. The image source is the pre-calibrated RGB-D camera mounted on the robotic arm. The program is packaged as an independent module and communicates with other modules to receive extrinsic and intrinsic parameters of the camera and image data through ROS2 (Robot Operating System). The transformation matrix T_2 is pre-calibrated. Everytime when taking an image. The rotation and translation from origin to tool can be obtained from robotic arm system. Thus matrix T_1 can be generated. And by multipling T_1 and T_2 we can get the transformation from origin to camera.

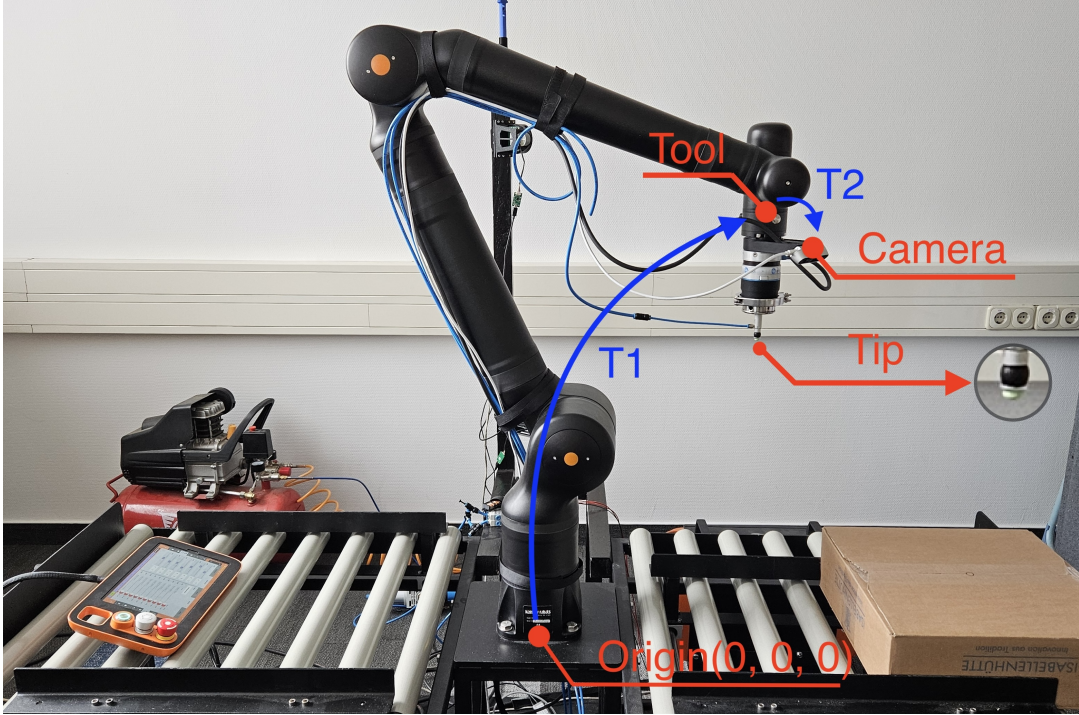


Figure 4.1: Hardware setup for the experiment

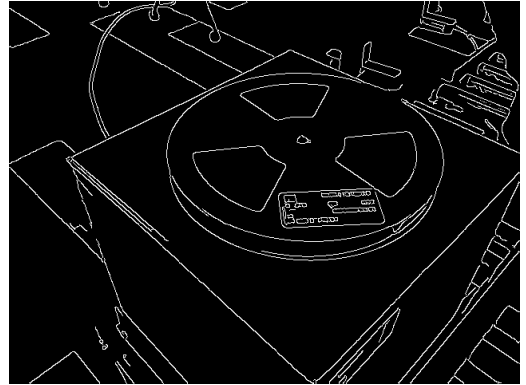
4.2 Edge Detection

Edge detection is the fundamental step in whole process. Both Canny edge detection[5] and Sobel edge detection[6] techniques are used to extract oriented edge points from the image.

Figure 4.2 demonstrates the results of edge detection. The image is firstly smoothed by Gaussian blurring to reduce the effect of pixel noises on edge detection and gradients calculation. Then the Canny edge detection algorithm is utilized to obtain binary edge image (Figure 4.2b), while the Sobel operator is leveraged to compute the gradient image. Here the pixel values in Figure 4.2c indicate the radians of pixels in the range of $\{-\pi, \pi\}$. By looking for the detected edges on binary image with corresponding value on the gradient image, the oriented edge points can be obtained (Figure 4.2d).



(a) Original image



(b) Edge image after Canny edge detection



(c) Radian map after Sobel edge detection



(d) Oriented edge image

Figure 4.2: The process of edge detection

4.3 Line and Corner Removal

Since straight lines are nothing related to ellipse fitting. They can be removed in advance by applying probabilistic Hough transform[8] on edge image. And the purpose of removing corners is to separate curves and straight lines as much as possible considering that the detected edges may have curves connected to straight lines. This step not only lower the possibility of errors in subsequent ellipse fitting, but also reduces the number of RANSAC iterations required to achieve the best fitting results, thereby reducing computation time. Harris corner detection[9] is used for corner removal.

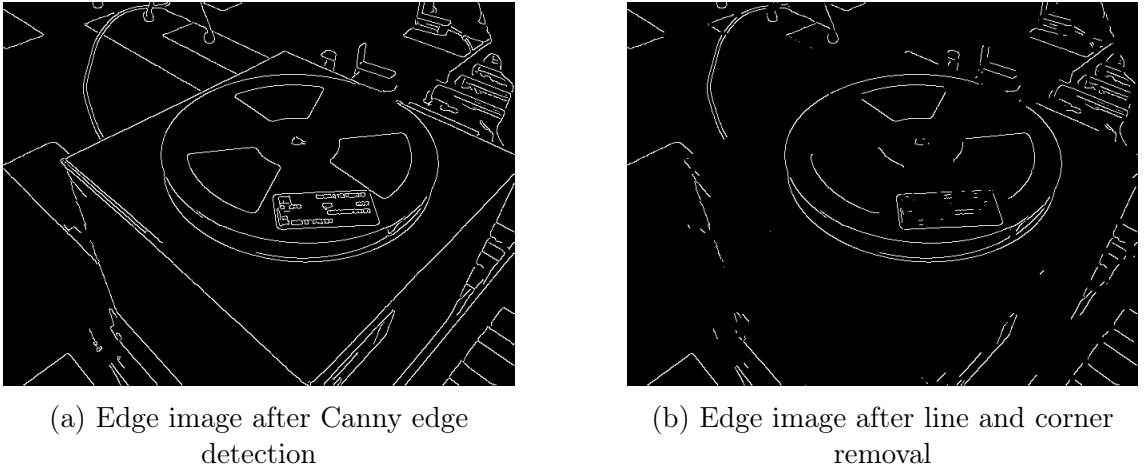


Figure 4.3: The process of line and corner removal

4.4 Edge Segmentation and Contour Detection

The purpose of edge segmentation is to split continuous non-arc edges into several smaller edge segments. First, the simple polygon approximation(Ramer–Douglas–Peucker algorithm)[10, 11] is applied on the edge image (Figure 4.4a). Then the vertices of polygons are removed from the edge image (Figure 4.4b). As a result, some small gaps will appear on the originally continuous edges. Next, use contour detection[12] to segment the edge points into different clusters.

The reason why non-arc edges need to be split into more segments is related to the latter filtering process, in which only arcs are considered. The example in

Figure 4.5 well presents the purpose of this step. In Figure 4.5c, the reel edge will be disregarded during the filtering step since it has spiral shape and cannot fit an ellipse well. However, if the edge of the reel is split into several arc segments in advance, they can then be retained.

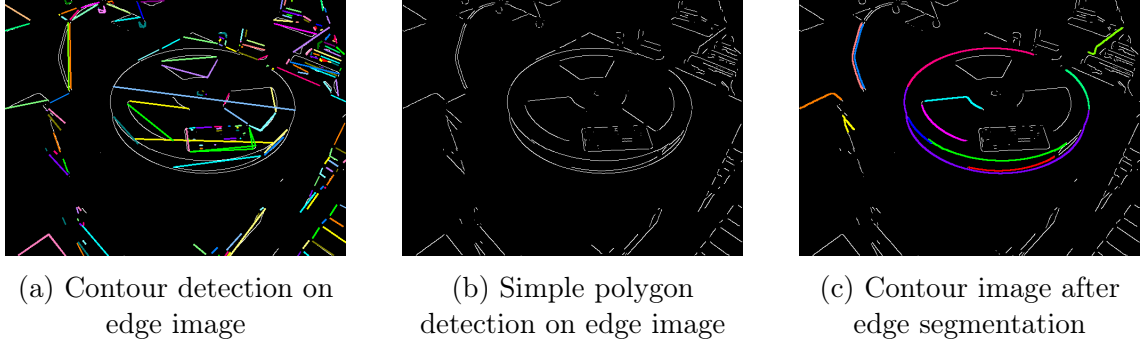


Figure 4.4: The process of edge segmentation, each color stands for one contour.

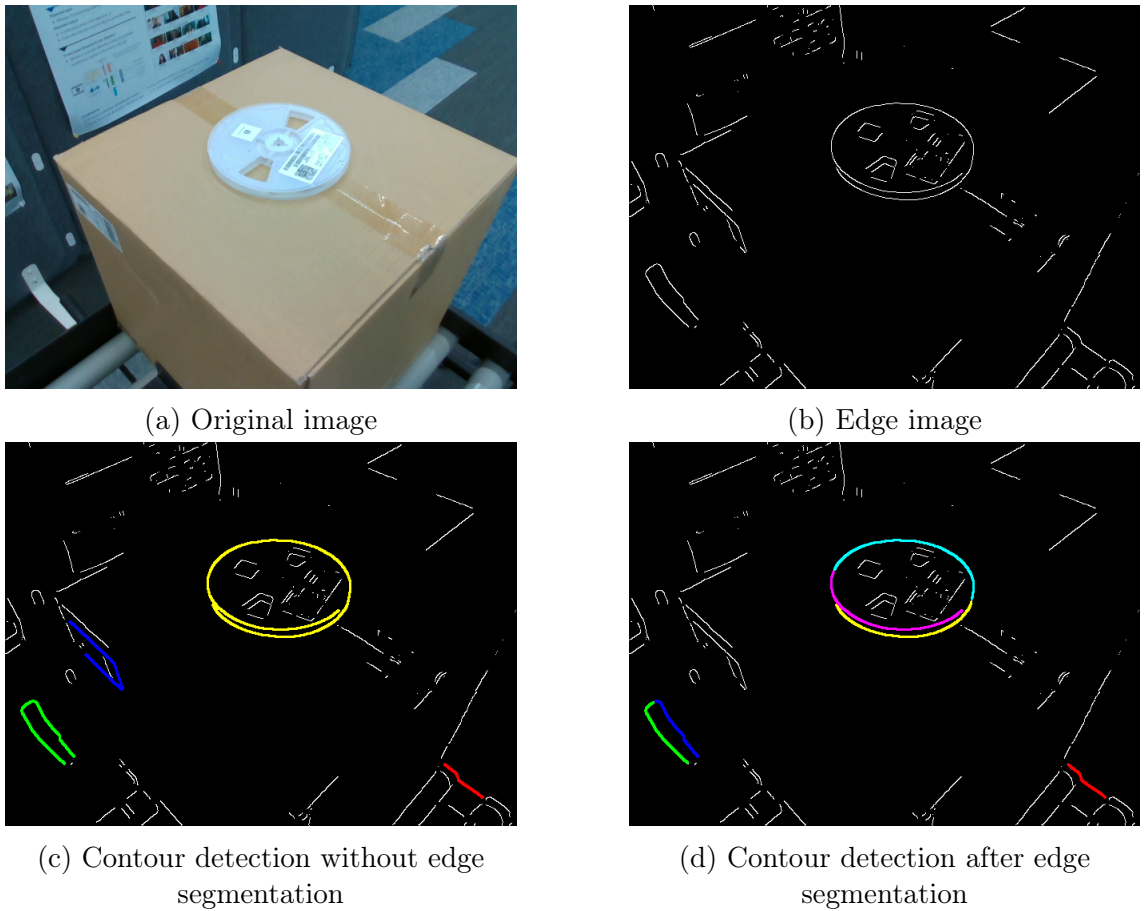


Figure 4.5: An example that requires edge segmentation, each color stands for one contour.

4.5 Contour Point Down Sampling

Discrete points can be obtained by using contour detection on the edge image(Figure 4.6a). However, too many sample points are not only unhelpful for ellipse fitting, but also reduce the fitting speed and accuracy. Therefore, the down sampling is necessary. Since the points after contour detection are arranged in order, they can first be roughly down sampled by choosing indices of the list with custom step size(Figure 4.6b). Then apply the radius sample removal to eliminate redundant points that are too close to each other and get the sample points used for subsequent step(Figure 4.6c).

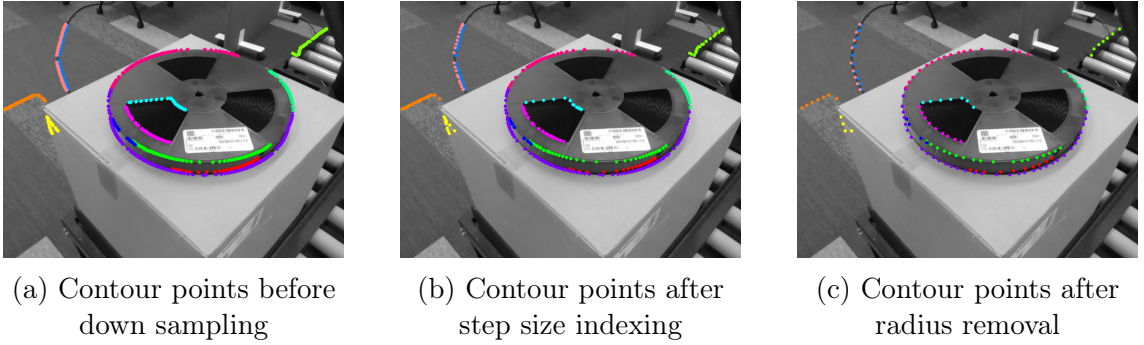


Figure 4.6: The process of point down sampling, which is applied once on each contour.

4.6 Sample Point Filtering

Ellipse fitting is performed on all sample points of each contour in order to filter out sample points that do not belong to the ellipse. The final set of sample points is then determined by checking whether the gradient error and distance error between each fitted ellipse and the corresponding sample points are below thresholds. All of the oriented sample points in Figure 4.7c are then merged into one set for the next ellipse fitting step.

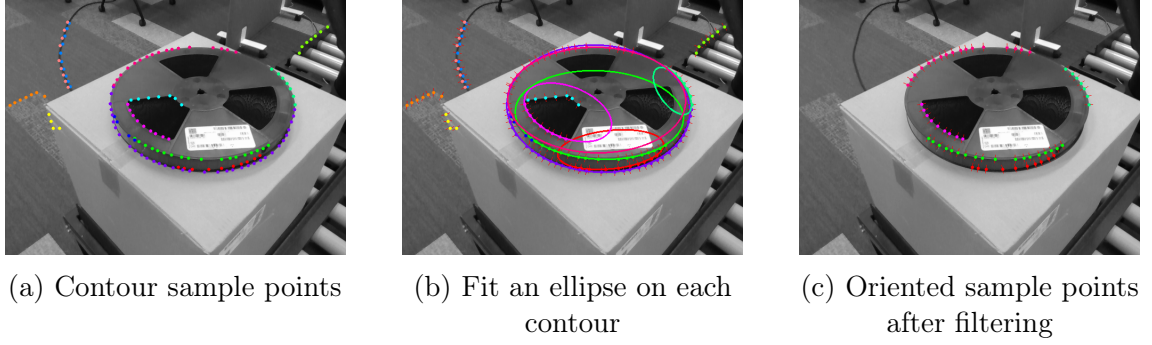
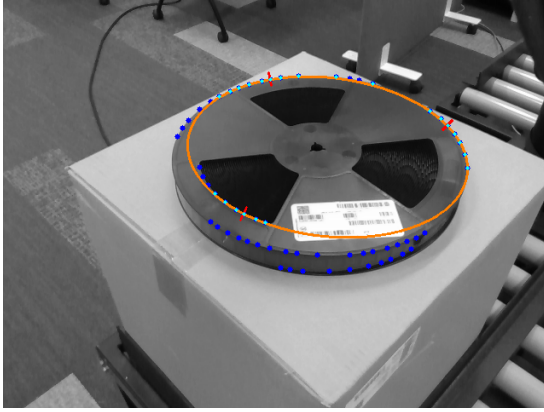


Figure 4.7: The process of sample point filtering

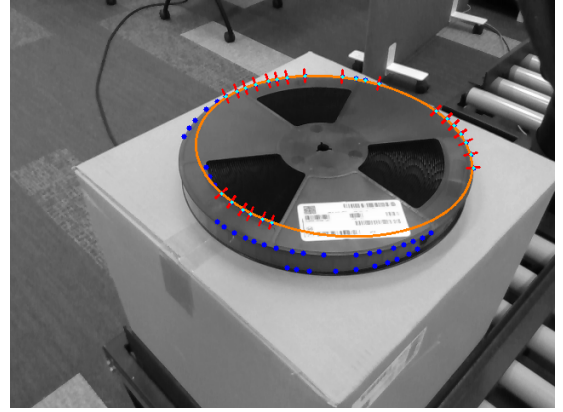
4.7 RANSAC Oriented Ellipse Fitting

The ellipse can finally be fitted by applying RANSAC algorithm embedded with oriented ellipse fitting algorithm on filtered sample points.

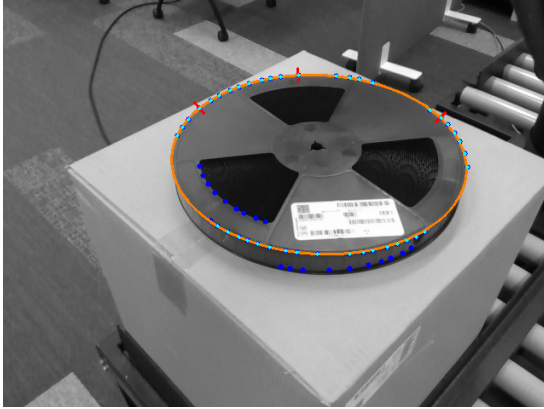
For each iteration, three sample points will be chosen for ellipse fitting. If a better ellipse is found, i.e. an ellipse passes both gradient error and distance error checking and has higher inlier ratio(Figure 4.8a, Figure 4.8c). The inliers of the ellipse are then turned into sample points to fit another ellipse. And it will be considered as current best ellipse if it has higher inlier ratio(Figure 4.8b, Figure 4.8d). With this extra step, the RANSAC can terminate in a lower number of iterations. In addition, to get a more accurate ellipse after the RANSAC algorithm terminates, the final loop will occur and keep converting the best inliers into sample points to fit an ellipse until the inlier ratio no longer increases(Figure 4.8e).



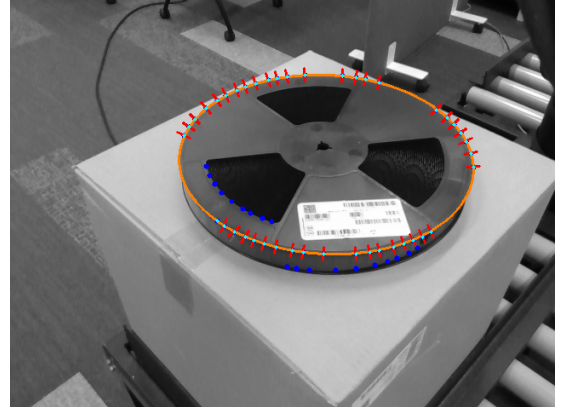
(a) Fit an ellipse with 3 sample points



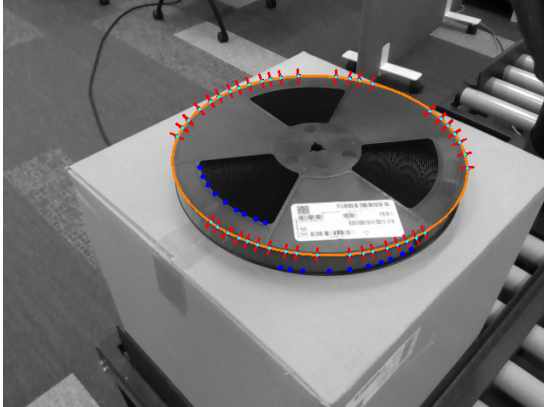
(b) Fit an ellipse with inliers from (a)



(c) Find an ellipse better than (b)



(d) Fit an ellipse with inliers from (c)



(e) Fit the final ellipse with best inliers from (d)

Figure 4.8: The process of RANSAC oriented ellipse fitting. The fitted ellipses are showed in orange. Blue points are outliers while cyan points are inliers. Red line segments are orientation of point. Points with red line are sample points for ellipse fitting.

4.8 Camera Calibration

This step is prerequisite of center estimation if camera is uncalibrated. The camera can be calibrated by using Zhang's method [7], which is by taking multiple images of a chessboard with different angles to calculate the intrinsic parameters of the camera (Figure 4.9). However, this step is not necessary in the experimental environment since the camera mounted on robotic arm is pre-calibrated.

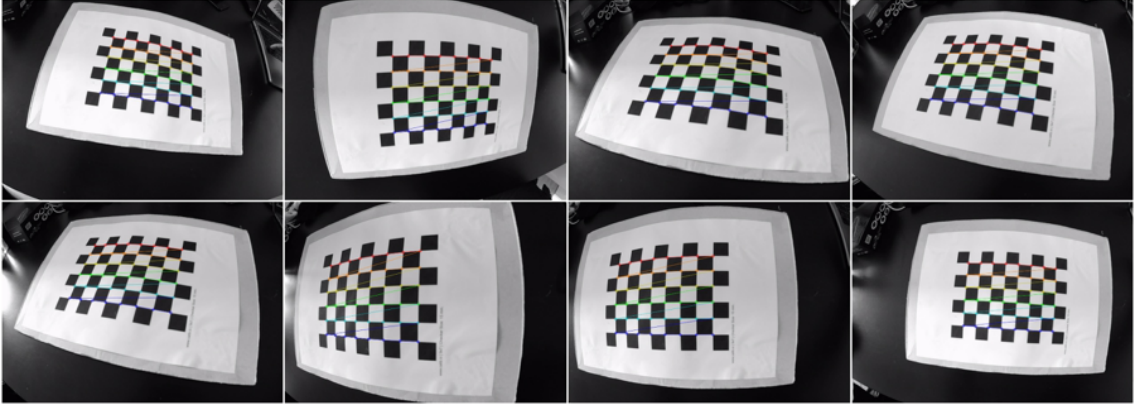
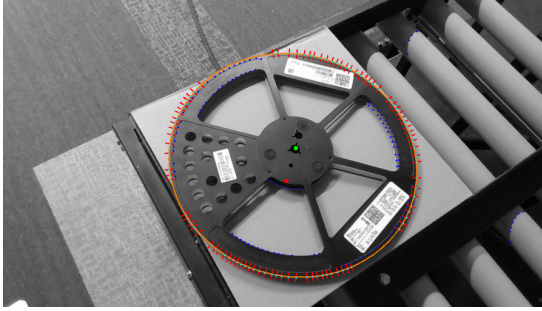


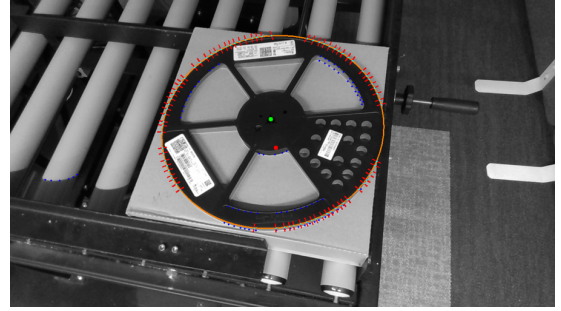
Figure 4.9: Example of using a chessboard for camera calibration with Zhang's method.

4.9 3D Circle Center Estimation

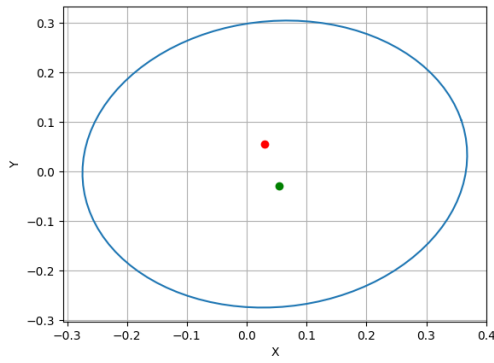
The ellipse coefficients are first normalized with camera intrinsic matrix K . Subsequently, the homography matrices H_1, H_2 are calculated based on the normalized coefficients. Since the bottom of the camera is closer to reels when taking images, the homography matrix with lower y value is chosen. In Figure 4.10, the green dots indicate the chosen projection of circle centers. After finding the centers in images, two 3D lines are generated in the world coordinate system. Finally, the 3D center point is derived by calculating the closest point of the lines (the red dot in Figure 4.10e).



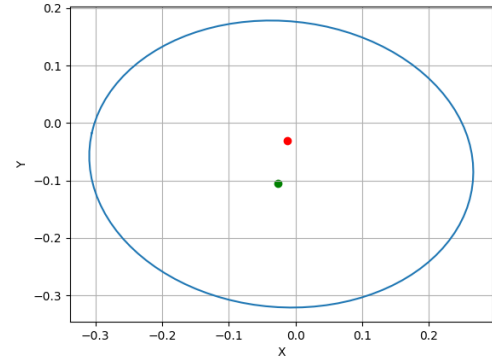
(a) Circle center projection in image 1



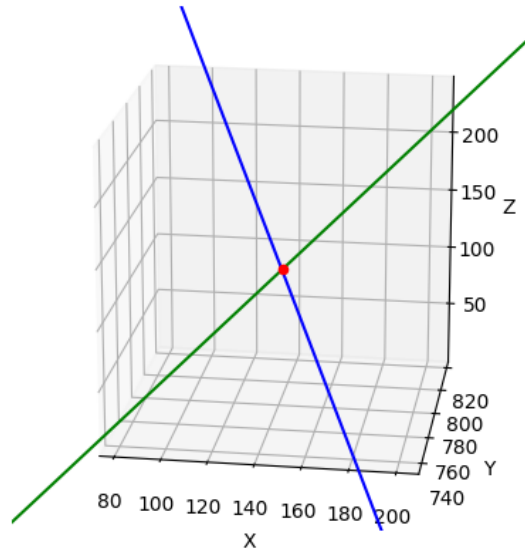
(b) Circle center projection in image 2



(c) Corresponding normalized circle center of image 1



(d) Corresponding normalized circle center of image 2



(e) Closest point between two lines

Figure 4.10: Process of 3D circle center estimation

Chapter 5

Experimental Results

5.1 Ellipse Fitting Results on Simulated Data

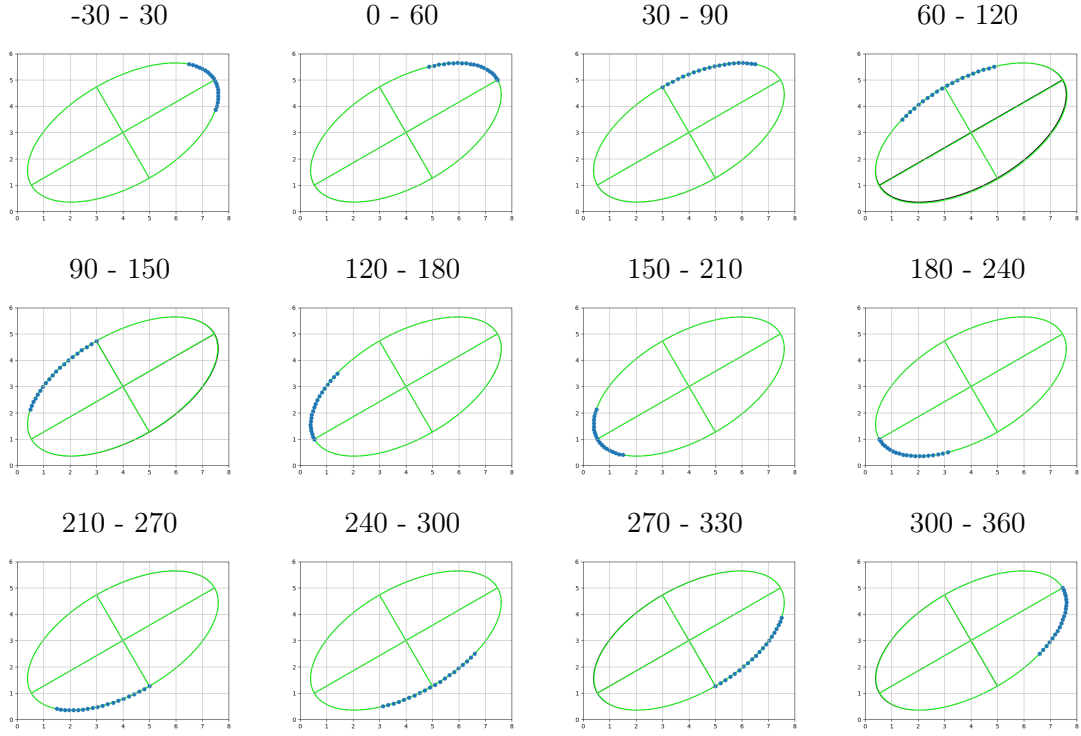
In this section, the performance of oriented ellipse fitting is tested on simulated data. Moreover, direct ellipse fitting algorithm[2], which is used in the early stage of our project, is compared with oriented ellipse fitting in terms of accuracy. The main idea of direct ellipse fitting is to fit an ellipse with minimum 5 points by using least square method. It is similar to oriented ellipse fitting but without gradient part in the formula (3.5)

5.1.1 Fitting with Partial Ellipse Points

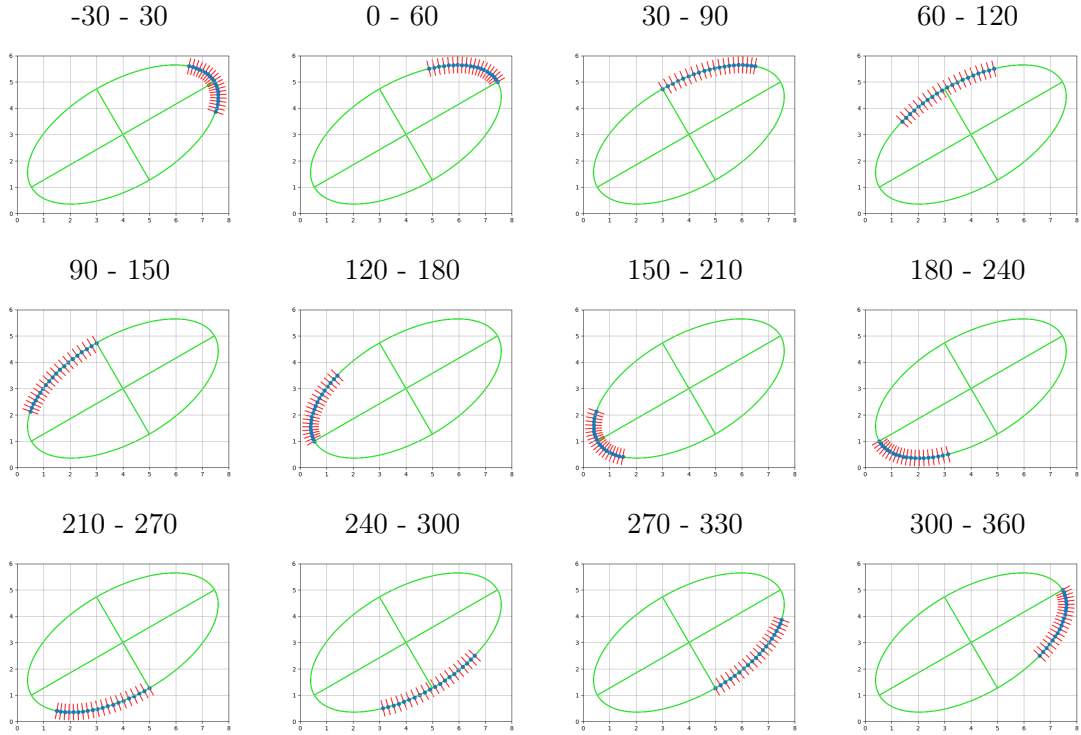
In this test both algorithms are fitted with partial data set of ground truth ellipse without noise. Each data set holds 20 sample points of a small portion of the ellipse (60 degrees). To make the results easier to compare, in Figure 5.1 the ground truth ellipse is set to black and it is one layer behind the fitted ellipse in green, which means if the fitting result is perfect then there should be no black color edge. Both Figure 5.1 and Table 5.1 indicate that with the advantage of extra gradient data, oriented ellipse fitting has higher accuracy in terms of partial fitting compare to direct ellipse fitting.

L2 Error of Fitted Centers		
	Direct Ellipse Fitting	Oriented Ellipse Fitting
-30 - 30	0.0004	9.5477×10^{-10}
0 - 60	0.0031	6.8790×10^{-9}
30 - 90	0.0029	1.2241×10^{-8}
60 - 120	0.0217	1.4247×10^{-9}
90 - 150	0.0098	1.2665×10^{-10}
120 - 180	0.0027	6.3032×10^{-12}
150 - 210	0.0003	4.3169×10^{-13}
180 - 240	0.0007	1.2490×10^{-12}
210 - 270	0.0028	4.6986×10^{-11}
240 - 300	0.001	8.8061×10^{-10}
270 - 330	0.0079	4.7012×10^{-10}
300 - 360	0.0077	1.0215×10^{-8}
Avg. Error	0.005	2.7706×10^{-9}
Total Error	0.0608	3.3247×10^{-8}

Table 5.1: Comparison of the L2 error between the centers fitted by the two algorithms and the true center



(a) Direct ellipse fitting on simulation data



(b) Oriented ellipse fitting on simulation data

Figure 5.1: Results of both fitting algorithms on data sets of 20 points which represent different elliptical arcs of the same ellipse with the center $(4, 3)$, semiaxes $(4, 2)$ and tilt 30 degrees. The appropriate arc section interval in degrees is given above each figure.

5.1.2 Fitting with Noisy Ellipse Data Set

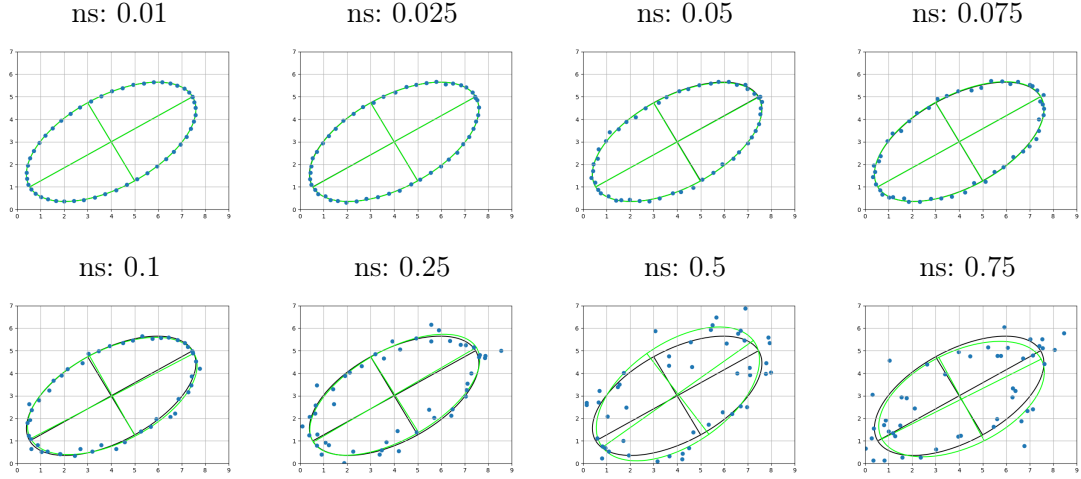
In this test the sample points are added with different scales of Gaussian noise. The results of direct ellipse fitting and oriented ellipse fitting are similar in Figure 5.2. However, by comparing Table 5.2 and Table 5.3 we can find that oriented ellipse fitting performs slightly better than direct ellipse fitting.

Results of Direct Ellipse Fitting						
Noise Scale	Center (4, 3)		Semi Axes (4, 2)		Tilt (30°)	
	Result	L2 Error	Result	L2 Error	Result	Abs. Error
0.0100	(3.9985, 2.9981)	0.0024	(4.0000, 1.9990)	0.0010	29.9870	0.0130
0.0250	(4.0031, 2.9971)	0.0042	(3.9916, 1.9985)	0.0085	30.1242	0.1242
0.0500	(3.9912, 2.9862)	0.0163	(3.9929, 1.9968)	0.0078	29.8202	0.1798
0.0750	(4.0068, 3.0097)	0.0118	(4.0024, 2.0178)	0.0179	30.0188	0.0188
0.1000	(4.0489, 2.9930)	0.0494	(3.9581, 2.0274)	0.0500	28.5969	1.4031
0.2500	(4.0275, 3.0471)	0.0546	(4.0540, 1.9555)	0.0700	31.1596	1.1596
0.5000	(4.0312, 3.0841)	0.0898	(3.9882, 2.2267)	0.2270	36.3971	6.3971
0.7500	(4.1008, 2.8543)	0.1772	(3.8216, 2.0864)	0.1982	27.6102	2.3898
Mean Error		0.0507		0.0726		1.4607
Std.		0.0556		0.0841		2.0288

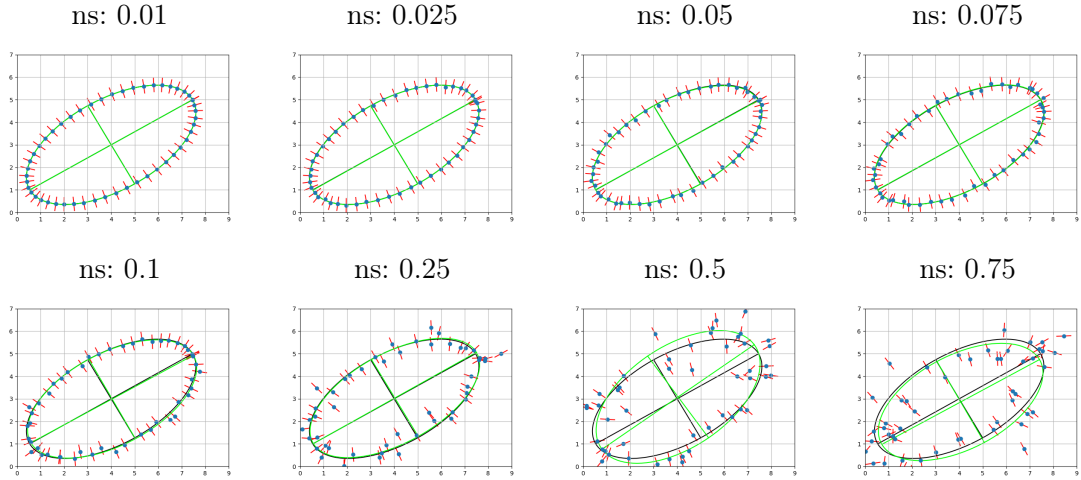
Table 5.2: Direct ellipse fitting results of sample points with different noise levels

Results of Oriented Ellipse Fitting						
Noise Scale	Center (4, 3)		Semi Axes (4, 2)		Tilt (30°)	
	Result	L2 Error	Result	L2 Error	Result	Abs. Error
0.0100	(3.9986, 2.9985)	0.0021	(4.0002, 1.9989)	0.0011	29.9961	0.0039
0.0250	(4.0021, 2.9977)	0.0031	(3.9951, 1.9966)	0.0059	30.1579	0.1579
0.0500	(3.9921, 2.9893)	0.0133	(3.9893, 1.9985)	0.0108	29.7646	0.2354
0.0750	(4.0028, 3.0067)	0.0073	(4.0096, 2.0140)	0.0169	29.9012	0.0988
0.1000	(4.0413, 2.9920)	0.0421	(3.9657, 2.0226)	0.0411	29.3224	0.6776
0.2500	(4.0294, 3.0365)	0.0469	(4.0151, 1.9741)	0.0300	30.2718	0.2718
0.5000	(4.0408, 3.0822)	0.0918	(3.9829, 2.2286)	0.2293	35.6009	5.6009
0.7500	(4.0648, 2.8602)	0.1541	(3.8451, 2.0596)	0.1660	29.2890	0.7110
Mean Error		0.0451		0.0626		0.9697
Std.		0.0502		0.0804		1.7670

Table 5.3: Oriented ellipse fitting results of sample points with different noise levels



(a) Direct ellipse fitting on noisy simulation data



(b) Oriented ellipse fitting on noisy simulation data

Figure 5.2: Results of both fitting algorithms on noisy data sets of 50 points which represent whole ellipse with the center $(4, 3)$, semiaxes $(4, 2)$ and tilt 30 degrees. The noise scale is given above each figure.

5.1.3 Fitting with Noisy Gradients

In order to test how biased gradients affect the fitting result of oriented ellipse fitting. The noise is added to gradients while keep the points noiseless. Due to the fact that direct ellipse fitting is not affected by noisy gradients. Two algorithms won't be compared in this test.

From Figure 5.3 we can notice that when the noise scale exceeds 0.1, as the noise increases, the fitting result will be significantly affected. This shows oriented ellipse fitting is sensitive to the gradient of points, for point sets with large gradient errors, the additional gradient information can become a disadvantage of the algorithm.

Noise Scale	Avg. Dist
0.0100	0.0009
0.0250	0.0022
0.0500	0.0044
0.0750	0.0069
0.1000	0.0096
0.2500	0.0308
0.5000	0.0780
0.7500	0.1190
1.0000	0.1406
1.2500	0.1509
1.5000	0.1547
1.7500	0.1547

Table 5.4: Oriented ellipse fitting results with noisy gradients. The corresponding average distance of each noise scale is the average value after 1000 times repeated tests.

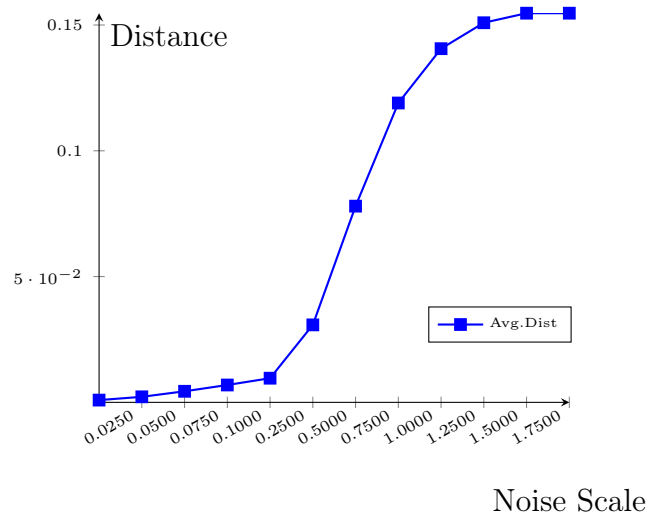


Figure 5.3: Corresponding chart of Table 5.4

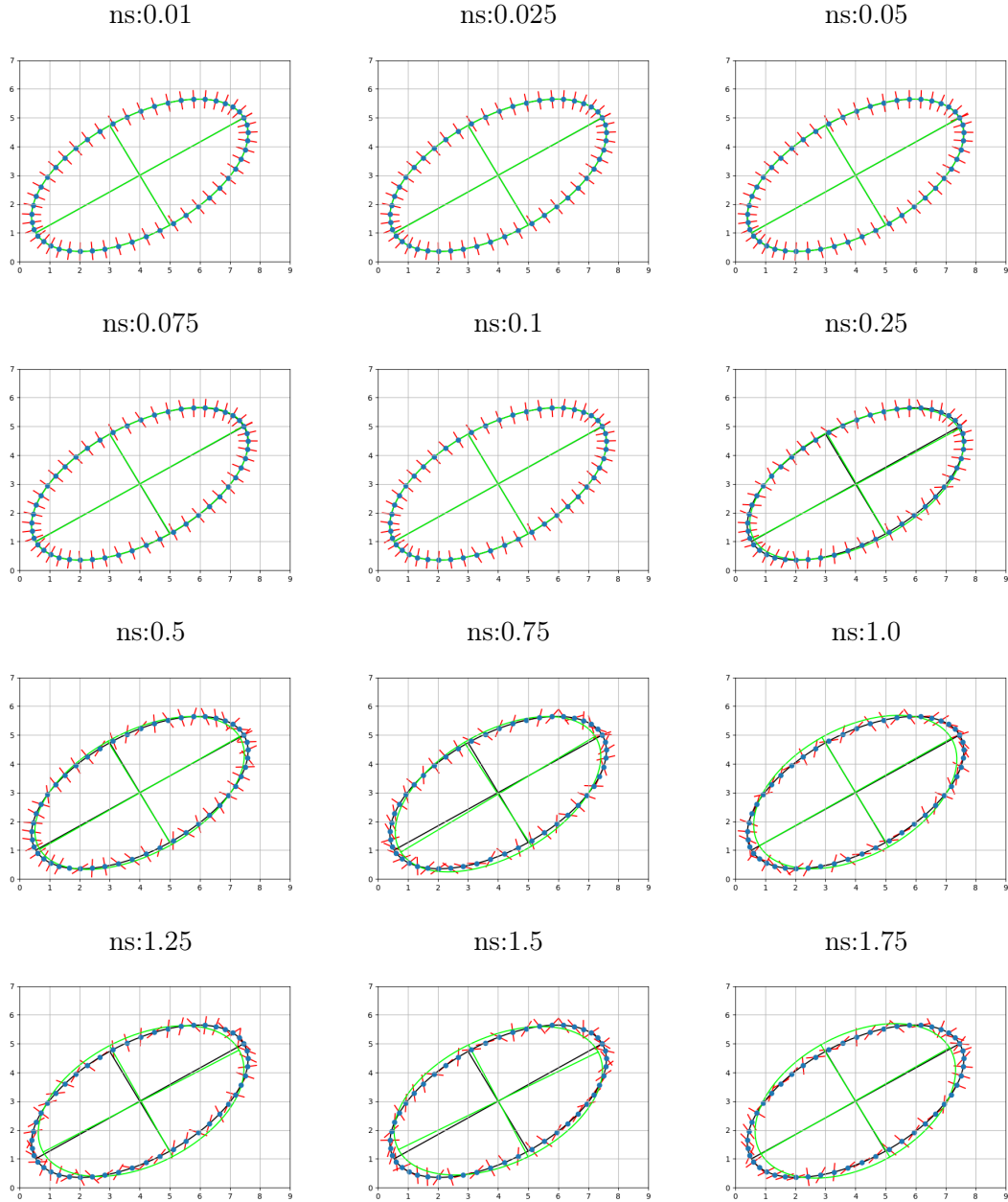
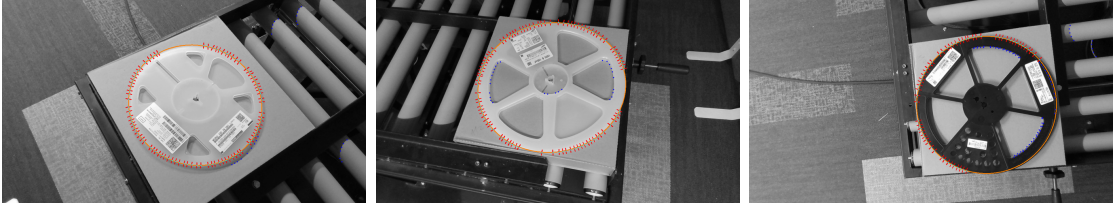


Figure 5.4: Results of the fitting algorithm on data sets of 50 points which represent whole ellipse with the center $(4, 3)$, semiaxes $(4, 2)$ and tilt 30 degrees. The noise scale of gradient is given above each figure.

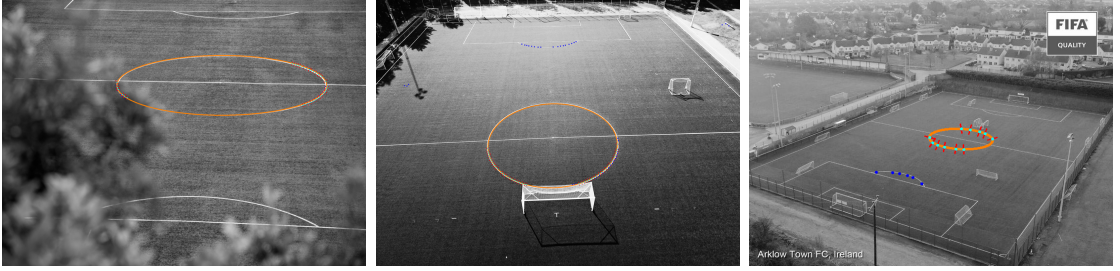
5.2 Ellipse Fitting and Center Estimation Results on Real Images

5.2.1 Fitting Results in Different Scenarios

Figure 5.5 demonstrates the results of oriented ellipse fitting algorithm on different type of objects. Here reels for electronic components, center circle of soccer fields and car wheels are taken as examples.



(a) Oriented Ellipse Fitting on Reels



(b) Oriented Ellipse Fitting on Soccer Fields



(c) Oriented Ellipse Fitting on Wheels

Figure 5.5: Fitting results for different types of objects with circular parts. Blue points are outliers, orange lines are fitted ellipses and red line segments are point orientations.

5.2.2 Results of Reel Center Estimation in Image

Different types of reels are used for demonstration in this section since center estimation in image only works for images taken by calibrated camera. In Figure 5.6, it is obvious that the bottom of the camera is closer to reel surface in this way the

correct center can be chosen. As for the bottom right image, the reel surface is almost parallel to the camera image plane. In this case, two center candidates are overlapping to each other, thus whichever can be correct center.

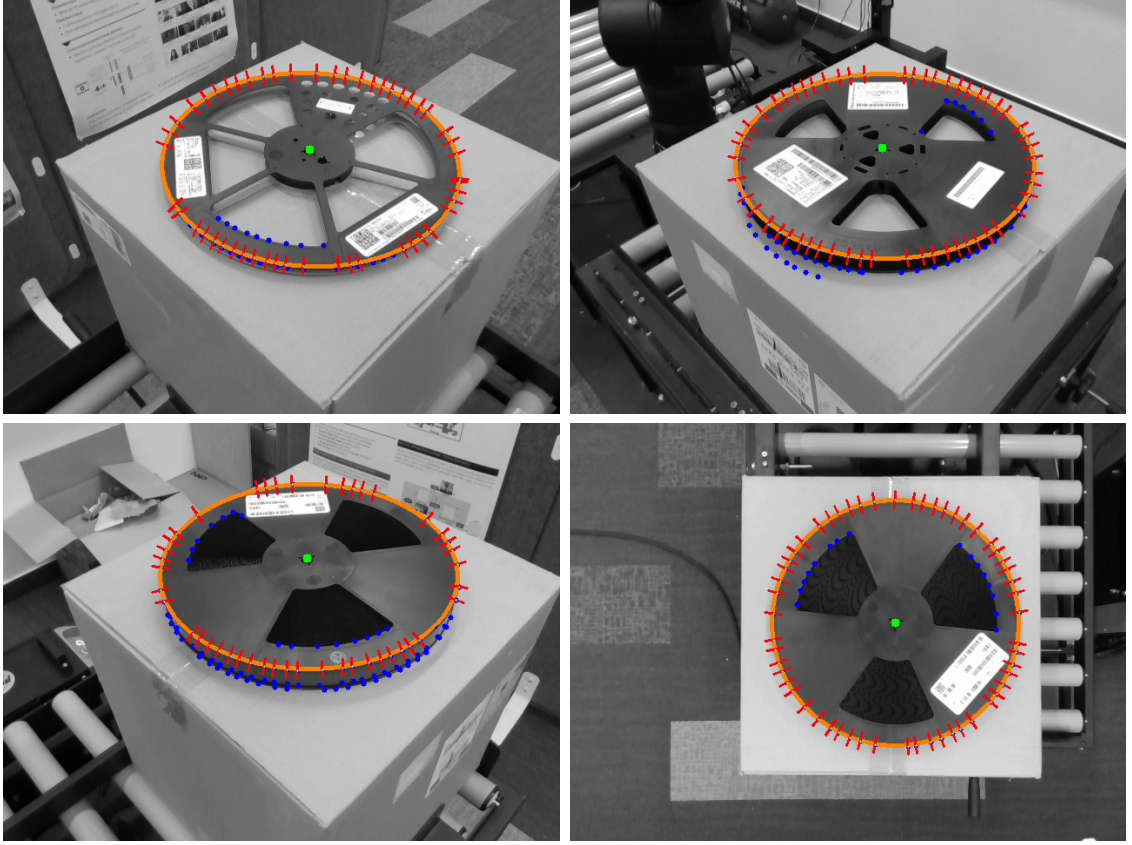


Figure 5.6: Results of Reel Center Estimation in Image. The additional green points are estimated circle center.

5.2.3 Results of 3D Reel Center Estimation

The videos below demonstrate the 3D reel center estimation process on the robotic arm system. The robotic arm first takes two images of a reel at different angles. Next, two images are packed with rotation and translation data and passed to the program to estimate the 3D center of the reel. Finally, the robotic arm receives the location of the reel center and move its tool to pick up the reel.

[Demo1](#)¹ [Demo2](#)²

¹Demo1 url: <https://1drv.ms/v/s!Avm-qDf6hSdOhpA-JSeo5IGdT1SM1A>

²Demo2 url: https://1drv.ms/v/s!Avm-qDf6hSdOhpA_UHEhV3qYoSepbw

5.2.4 Results Comparison of Ellipse Fitting Algorithms

Both correct and wrong fitting results by oriented ellipse fitting are presented respectively, while the corresponding results by direct ellipse fitting[2] are taken as reference.

Figure 5.7 and Figure 5.8, Figure 5.9 and Figure 5.10, Figure 5.11 and Figure 5.12 are three result pairs in different scenarios. The left images stand for direct ellipse fitting[2], while right images are the results by oriented ellipse fitting.

In terms of ellipse fitting on wheel images, oriented ellipse fitting outperforms direct ellipse fitting most of the time due to extra constraints on point orientations and less sample points required for model fitting (minimum sample points: 3 vs 5), which means it is more likely to choose sample points from same object. In the 62 wheel images we tested, none of the results from direct ellipse fitting is better than those from oriented ellipse fitting. If direct ellipse fitting yields good result from an image, oriented ellipse fitting also produces the correct one, but the reverse is not always true. On the other hand, whenever oriented ellipse fitting fits wrong ellipse, direct ellipse fitting also fail to find the correct one. Moreover, for some false results from direct ellipse fitting, oriented ellipse fitting corrects them.

As for the scenario of reel detection, both algorithms fit the correct circular plane of the reel most of the time. In images where the shooting angle is low so that the double-layer structure of the reel can be seen, direct ellipse fitting can fail to fit the correct circular surface while orient ellipse fitting is able to find the correct one. But sometimes the result is reversed.

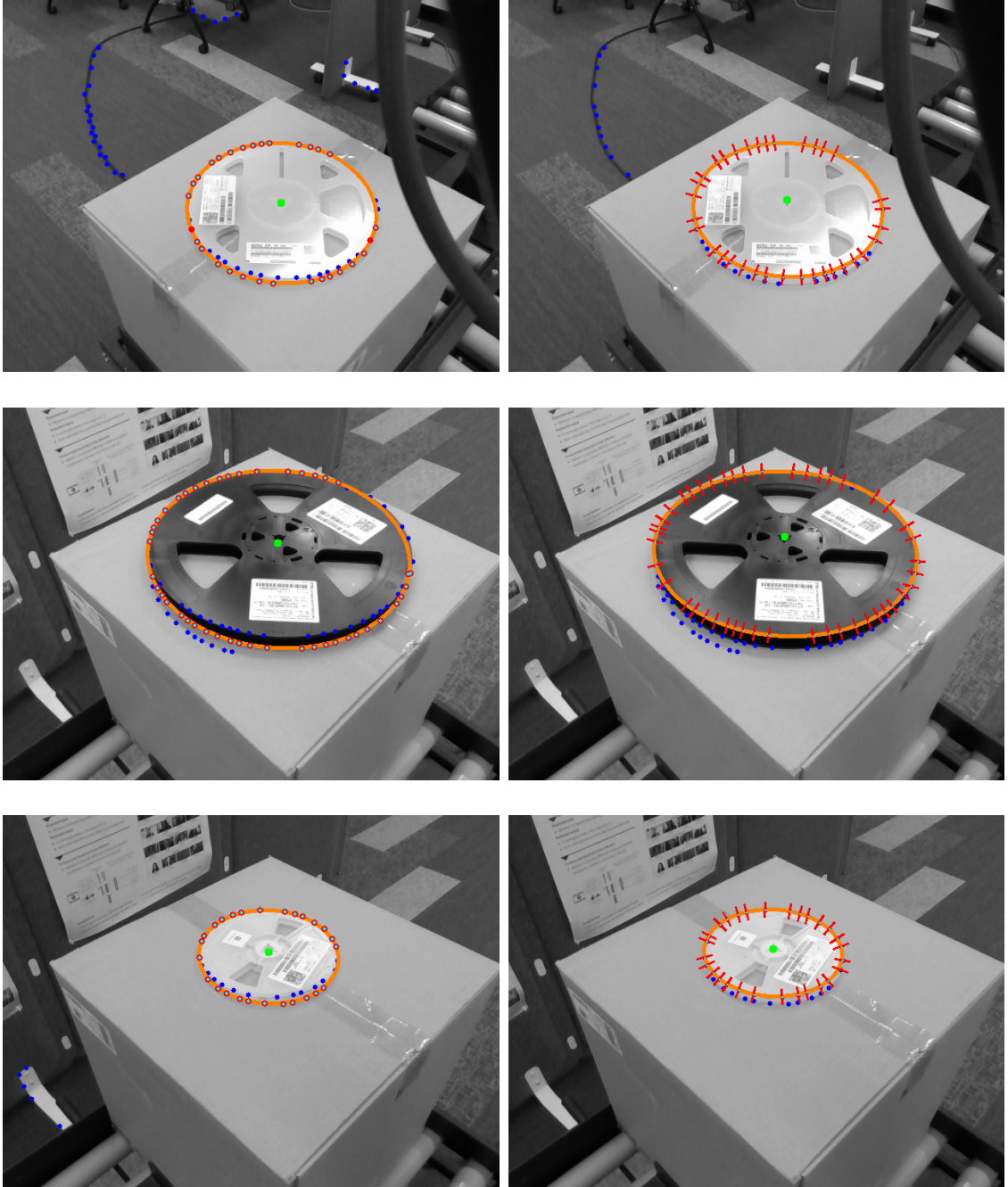


Figure 5.7: Good results fitted by oriented ellipse fitting on reel images(right) and corresponding results of direct ellipse fitting(left). Blue points are outliers, orange lines are fitted ellipses, red line segments in right images stand for point orientations and green points are estimated circle center.

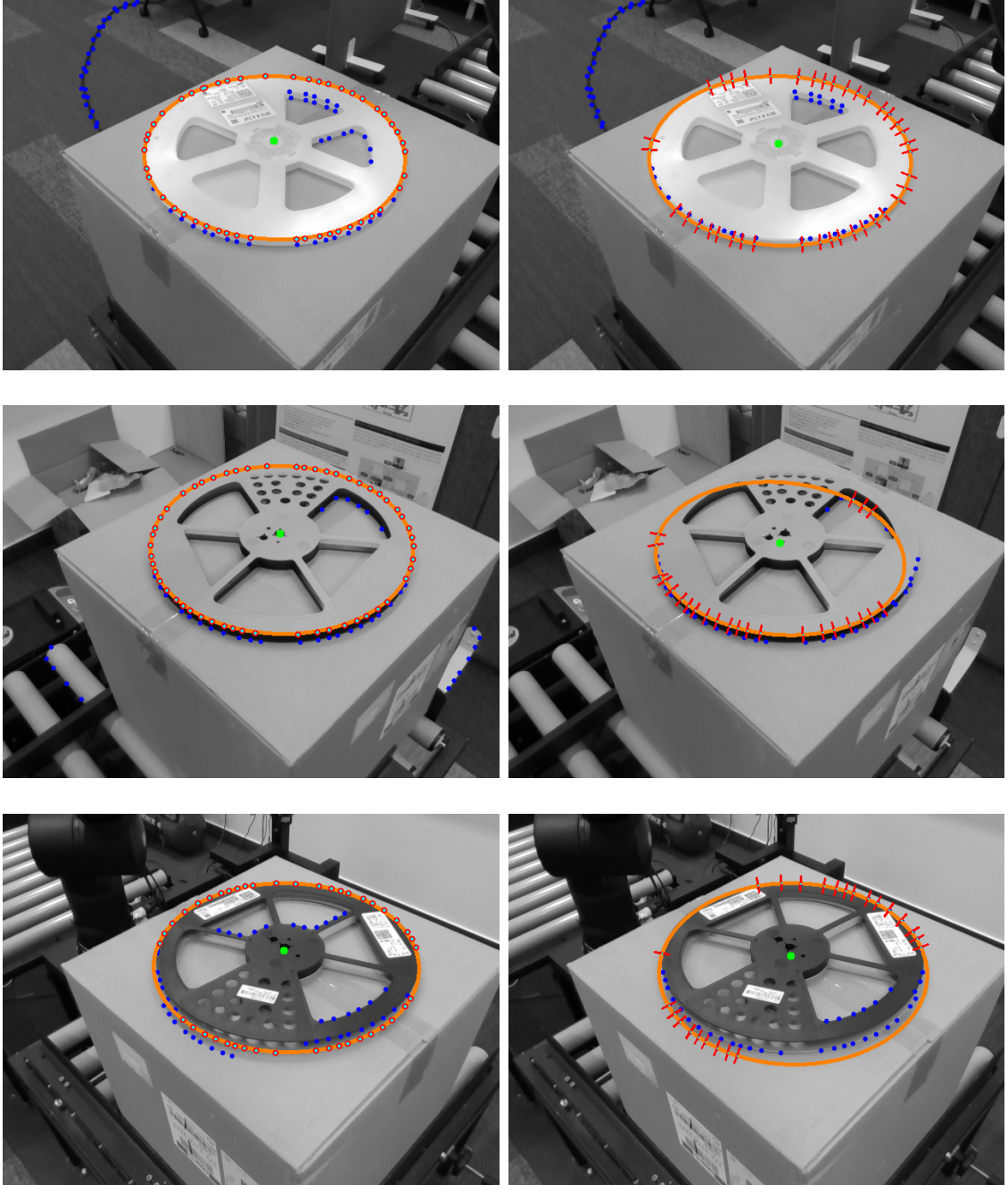


Figure 5.8: Incorrect results from oriented ellipse fitting on reel images(right) and corresponding results of direct ellipse fitting(left).



Figure 5.9: Good results fitted by oriented ellipse fitting on wheel images(right) and corresponding results of direct ellipse fitting(left). There's no indicate points for center estimation since the images are taken by unknown cameras.

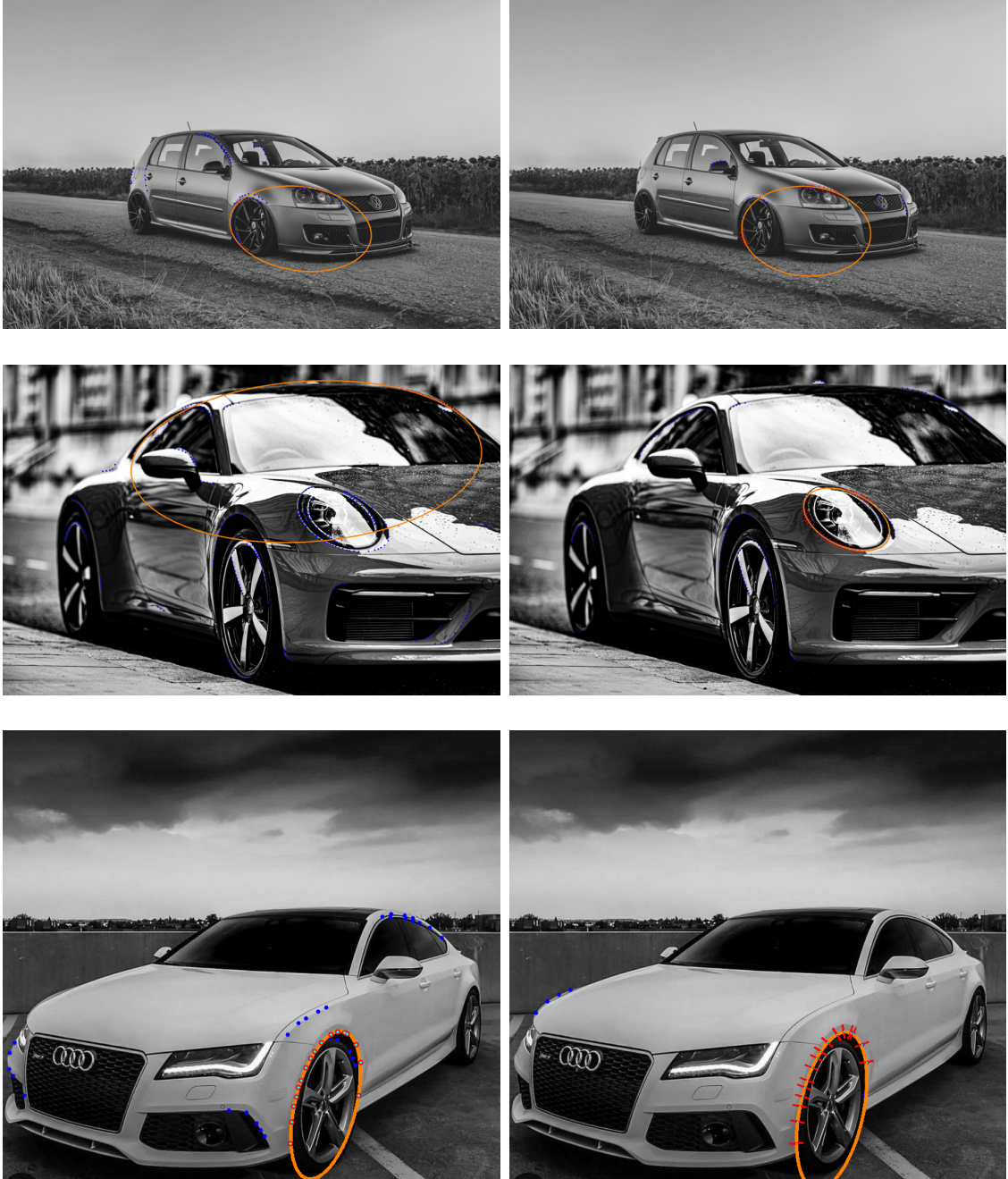


Figure 5.10: Incorrect results from oriented ellipse fitting on wheel images(right) and corresponding results of direct ellipse fitting(left).

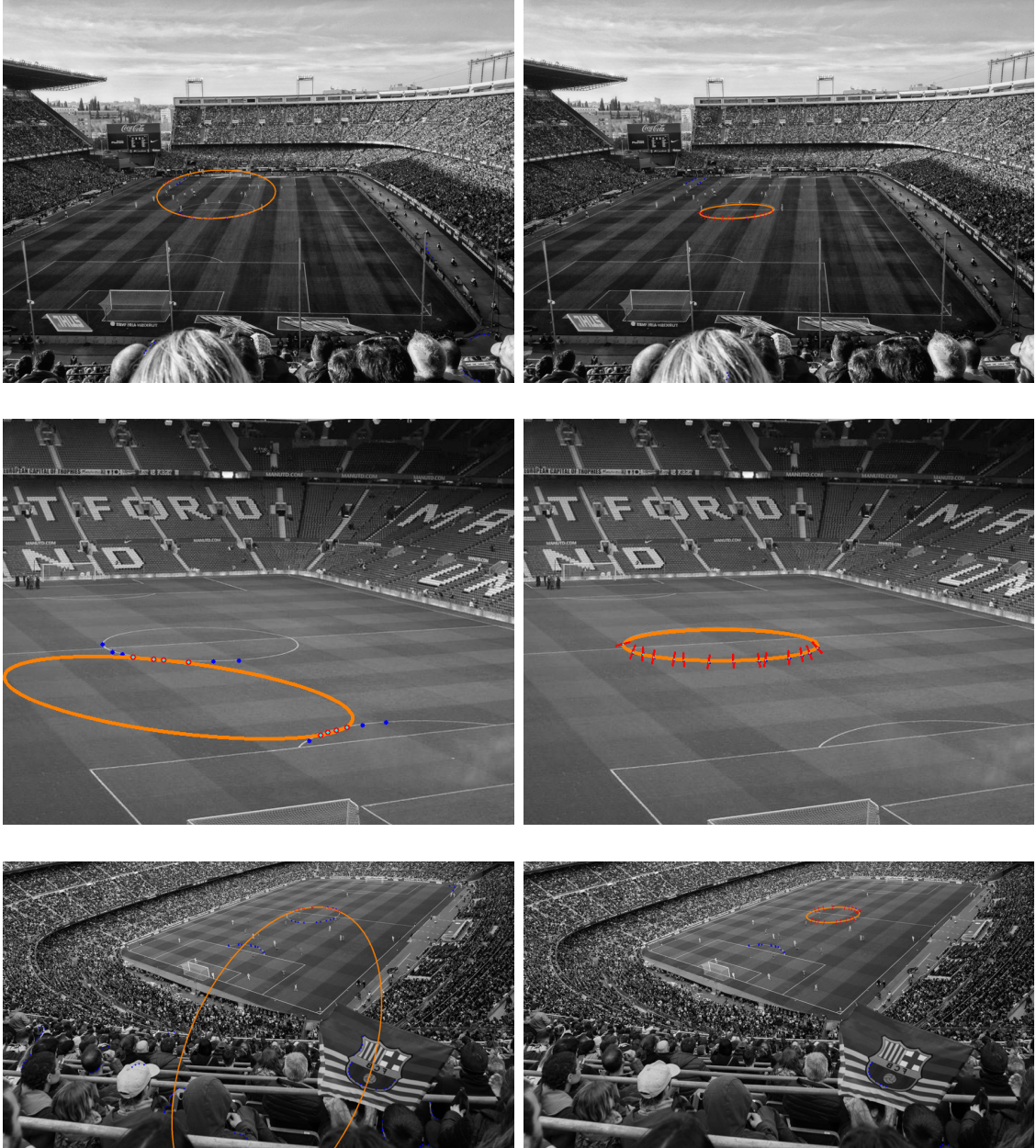


Figure 5.11: Good results fitted by oriented ellipse fitting on soccer field images(right) and corresponding results of direct ellipse fitting(left).

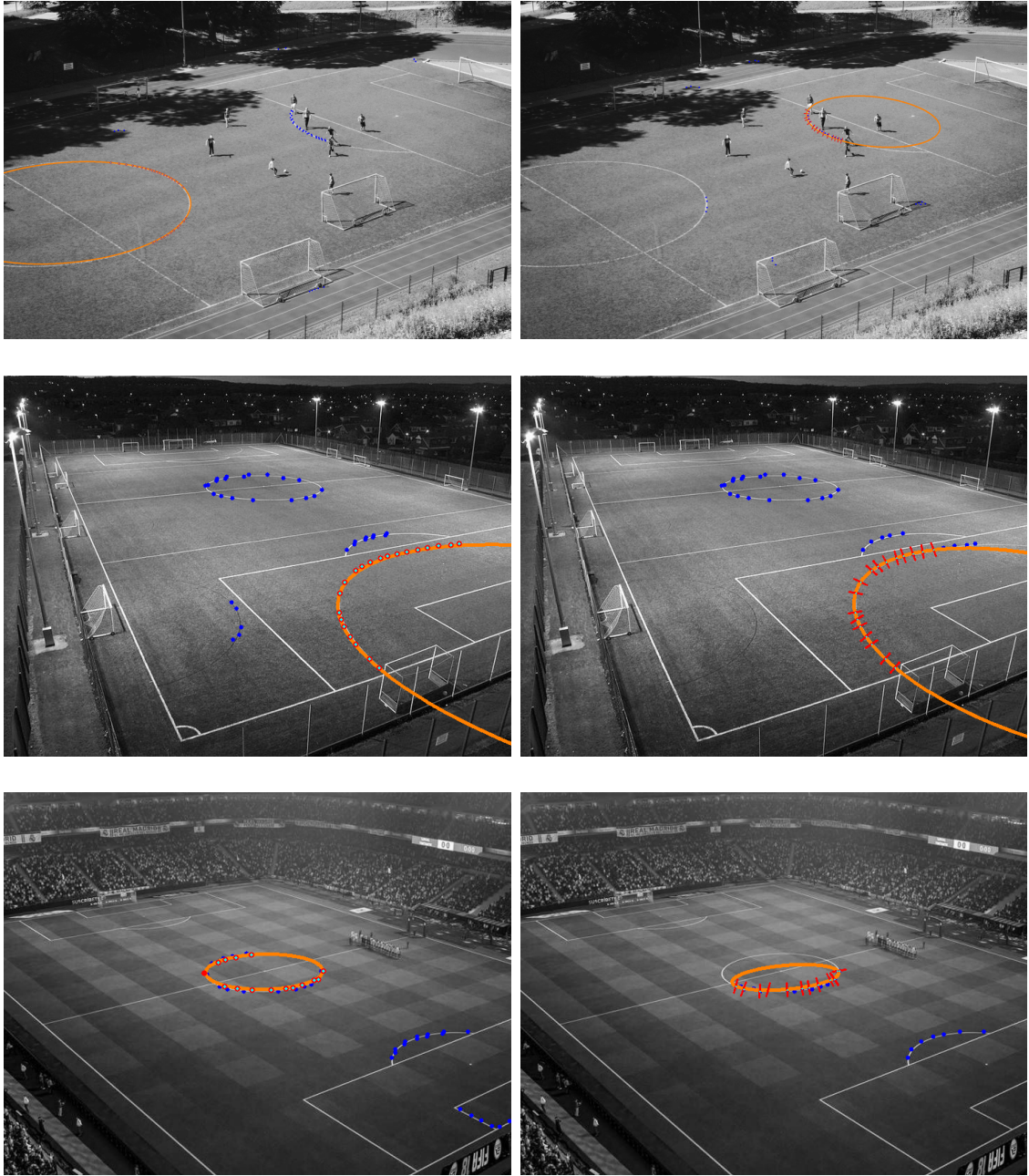


Figure 5.12: Incorrect results from oriented ellipse fitting on soccer field images(right) and corresponding results of direct ellipse fitting(left).

Chapter 6

Conclusion

At the conclusion of our project, we successfully implemented a program designed to locate the center of a reel hole using a combination of novel algorithms and methods. The process begins with image processing part, where the Canny edge detection[5] and Sobel operator[6] are employed for oriented edge detection. Subsequently, we applied a combination algorithm of probabilistic Hough transform[8] and Harris corner detection[9] to remove straight lines and corners, followed by edge segmentation to split continuous curve edges into arc segments. Simple polygon approximation[10, 11] is utilized in this step. Next, contour detection[12] converts edge pixels to a coordinate array, followed by point down sampling, where points are initially down sampled based on indexing with a step size and then points near to each other are removed using radius point removal. The final point set is extracted after sample point filtering, where the oriented ellipse fitting algorithm is employed for the first time in the entire process to filter out points that did not belong to arcs.

At this point, the image processing step concludes, followed by the reel detection and center estimation processes. The point set is served as input for RANSAC-based oriented ellipse fitting algorithm to identify the reel in an image and obtain the ellipse coefficients. The coefficients, along with camera intrinsic parameters, are then utilized for 2D circle center estimation. Finally, by running the aforementioned procedure twice on two images and combining it with camera rotation and translation parameters, the 3D reel center is derived through closest point of skew lines algorithm[4].

As one of the main contributions of our work, oriented ellipse fitting demonstrates significant potential for outperforming direct ellipse fitting algorithm in terms of accuracy when applied to images. When gradients of points are accurate or have low errors, oriented ellipse fitting not only provides higher accuracy than direct ellipse fitting but also operates relatively faster when nested within the RANSAC algorithm. While our other novel algorithm: 3D circle center estimation provides a way to locate the center of objects with circular surfaces, which is useful in the field of automation.

6.1 Future Work

Despite the great potential of RANSAC-based oriented ellipse fitting, many problems remain to be solved, such as how to obtain a sample point set with higher gradient accuracy through image processing before fitting, or to filter out points with large gradient errors. Multiple ellipses detection in one image is also considered in our future work. Moreover, our recent research has uncovered that 3D circle center estimation possesses the capability to not only locate the center of a circle in three-dimensional space but also to estimate the normal vector of the circle's surface, which will be one of our primary extended research in the future.

Acknowledgements

I would like to express my sincere thanks to Robert Bosch Kft for providing me with the opportunity to participate in this research, as well as to the Department of Artificial Intelligence of Eötvös Loránd University for generously providing laboratories and supporting software and hardware facilities. I am very grateful to my supervisor Dr. Levente Hajder for his active guidance and various support and help on algorithms and documentation. Special thanks go to Martin Sallai for his dedicated help with the control of the robotic arm and software-system communication throughout the research process. Additionally, I would like to acknowledge Bandó Kovács for his significant contributions to the hardware implementation and support.

References

- [1] Martin A Fischler and Robert C Bolles. “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography”. In: *Communications of the ACM* 24.6 (1981), pp. 381–395.
- [2] Radim Hahř and Jan Flusser. “Numerically stable direct least squares fitting of ellipses”. In: *Proc. 6th International Conference in Central Europe on Computer Graphics and Visualization. WSCG*. Vol. 98. Citeseer. 1998, pp. 125–132.
- [3] Luis Alvarez and Vicent Caselles. “Homography estimation using one ellipse correspondence and minimal additional information”. In: *2014 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2014, pp. 4842–4846.
- [4] Johannes Traa. “Least-squares intersection of lines”. In: *University of Illinois Urbana-Champaign (UIUC)* (2013).
- [5] John Canny. “A computational approach to edge detection”. In: *IEEE Transactions on pattern analysis and machine intelligence* 6 (1986), pp. 679–698.
- [6] Irwin Sobel and G Feldman. “An isotropic 3x3 image gradient operator for image processing”. In: *Mach. Vis. Three-Dimens. Scenes* June (1968), pp. 376–379.
- [7] Zhengyou Zhang. “A flexible new technique for camera calibration”. In: *IEEE Transactions on pattern analysis and machine intelligence* 22.11 (2000), pp. 1330–1334.
- [8] Nahum Kiryati, Yuval Eldar, and Alfred M Bruckstein. “A probabilistic Hough transform”. In: *Pattern recognition* 24.4 (1991), pp. 303–316.

- [9] Chris Harris, Mike Stephens, et al. “A combined corner and edge detector”. In: *Alvey vision conference*. Vol. 15. 50. Citeseer. 1988, pp. 10–5244.
- [10] Urs Ramer. “An iterative procedure for the polygonal approximation of plane curves”. In: *Computer graphics and image processing* 1.3 (1972), pp. 244–256.
- [11] David H Douglas and Thomas K Peucker. “Algorithms for the reduction of the number of points required to represent a digitized line or its caricature”. In: *Cartographica: the international journal for geographic information and geovisualization* 10.2 (1973), pp. 112–122.
- [12] Satoshi Suzuki et al. “Topological structural analysis of digitized binary images by border following”. In: *Computer vision, graphics, and image processing* 30.1 (1985), pp. 32–46.
- [13] Andrew W Fitzgibbon, Maurizio Pilu, and Robert B Fisher. “Direct least squares fitting of ellipses”. In: *Proceedings of 13th international conference on pattern recognition*. Vol. 1. IEEE. 1996, pp. 253–257.

List of Figures

2.1	Samples of the reel	4
2.2	Concise flowchart of processes from input data to robot arm positioning the center hole of the reel	5
3.1	Possible projection results of the reference points	17
4.1	Hardware setup for the experiment	19
4.2	The process of edge detection	20
4.3	The process of line and corner removal	21
4.4	The process of edge segmentation, each color stands for one contour.	22
4.5	An example that requires edge segmentation, each color stands for one contour.	22
4.6	The process of point down sampling, which is applied once on each contour.	23
4.7	The process of sample point filtering	24
4.8	The process of RANSAC oriented ellipse fitting. The fitted ellipses are showed in orange. Blue points are outliers while cyan points are inliers. Red line segments are orientation of point. Points with red line are sample points for ellipse fitting.	25
4.9	Example of using a chessboard for camera calibration with Zhang's method.	26
4.10	Process of 3D circle center estimation	27
5.1	Results of both fitting algorithms on data sets of 20 points which represent different elliptical arcs of the same ellipse with the center (4, 3), semiaxes (4, 2) and tilt 30 degrees. The appropriate arc section interval in degrees is given above each figure.	30

5.2	Results of both fitting algorithms on noisy data sets of 50 points which represent whole ellipse with the center (4, 3), semiaxes (4, 2) and tilt 30 degrees. The noise scale is given above each figure.	32
5.3	Corresponding chart of Table 5.4	33
5.4	Results of the fitting algorithm on data sets of 50 points which represent whole ellipse with the center (4, 3), semiaxes (4, 2) and tilt 30 degrees. The noise scale of gradient is given above each figure.	34
5.5	Fitting results for different types of objects with circular parts. Blue points are outliers, orange lines are fitted ellipses and red line segments are point orientations.	35
5.6	Results of Reel Center Estimation in Image. The additional green points are estimated circle center.	36
5.7	Good results fitted by oriented ellipse fitting on reel images(right) and corresponding results of direct ellipse fitting(left). Blue points are outliers, orange lines are fitted ellipses, red line segments in right images stand for point orientations and green points are estimated circle center.	38
5.8	Incorrect results from oriented ellipse fitting on reel images(right) and corresponding results of direct ellipse fitting(left).	39
5.9	Good results fitted by oriented ellipse fitting on wheel images(right) and corresponding results of direct ellipse fitting(left). There's no indicate points for center estimation since the images are taken by unknown cameras.	40
5.10	Incorrect results from oriented ellipse fitting on wheel images(right) and corresponding results of direct ellipse fitting(left).	41
5.11	Good results fitted by oriented ellipse fitting on soccer field images(right) and corresponding results of direct ellipse fitting(left). . .	42
5.12	Incorrect results from oriented ellipse fitting on soccer field images(right) and corresponding results of direct ellipse fitting(left). . .	43

List of Tables

3.1	Max iterations between 3 and 5 sample points in different inlier ratio with confidence(c) = 0.99	9
5.1	Comparison of the L2 error between the centers fitted by the two algorithms and the true center	29
5.2	Direct ellipse fitting results of sample points with different noise levels	31
5.3	Oriented ellipse fitting results of sample points with different noise levels	31
5.4	Oriented ellipse fitting results with noisy gradients. The correspond- ing average distance of each noise scale is the average value after 1000 times repeated tests.	33

List of Algorithms

1	Modified RANSAC Algorithm	8
---	-------------------------------------	---