Summer 8-2023

Mirror Position Detection in a Catoptric Surface

Run Zhang
Washington University in St. Louis

WASHINGTON UNIVERSITY IN ST. LOUIS

McKelvey School of Engineering
Department of Electrical & Systems Engineering

Thesis Examination Committee:
Roger Chamberlain, Chair
Chandler Ahrens
Neal Patwari

Mirror Position Detection in a Catoptric Surface

by
Run Zhang

A thesis presented to
the McKelvey School of Engineering
of Washington University in
partial fulfillment of the
requirements for the degree
of Master of Science

August 2023
St. Louis, Missouri

Table of Contents

List of Figures

List of Tables

Acknowledgments

Thank you to the Washington University Department of Computer Science and Engineering and Department of Electric & Systems Engineering. I would like to acknowledge Roger Chamberlain, Chris Gill, Chandler Ahrens and Neal Patwari for their guidance and support throughout the entire experiment. I would also like to thank Jeremy Manin for the work on applying computer vision to the reflected surface, Samatha Kodali for the work on controlling the natural lighting of environments, and all of the other students who have made contributions to the project.

Run Zhang

Washington University in St. Louis
August 2023

Dedicated to my parents and advisors.

ABSTRACT OF THE THESIS

Mirror Position Detection in a Catoptric Surface

by

Run Zhang

Master of Science in Engineering Data Analytics and Statistics

Washington University in St. Louis, 2023

Professor Roger Chamberlain, Chair

The Catoptric Surface research project is a pioneering exploration of controlling daylight effects within built environments. In this thesis, we focus on the mirror position detection problem, which plays a vital role in achieving dynamic control over the direction of reflected light within a space. To address the challenge of mirror position detection, we employ computer vision techniques, specifically edge detection and the RANdom SAmple Consensus (RANSAC) algorithm. Edge detection is utilized to identify significant changes in intensity or color, corresponding to object boundaries, while RANSAC is applied for ellipse fitting. By iteratively selecting minimal subsets of points and fitting ellipses that meet geometric constraints, we attempt to accurately determine the position and geometry of mirrors in the catoptric array.

We evaluate two different RANSAC libraries for ellipse fitting, and our findings show that the `skimage` library in Python provides superior results compared to other alternatives. Additionally, we leverage the multiprocessing package to enable parallel processing, improving the efficiency of mirror detection.

We conclude that it is possible to detect single steps of mirror movement, however, reliable operation is highly sensitive to parameter settings within the computational pipeline.

Chapter 1

Introduction

1.1 Catoptric Surface Research

The Catoptric Surface research project [1, 2, 6] is an exploration of the control of daylight effects within a built environment. The project seeks to investigate the use of reflected daylight to create image-based patterns of light on interior surfaces, in order to amplify or reduce spatial perception. Quoting from Ahrens et al. [2], this approach "challenges the traditional idea of lighting levels being homogeneous throughout a space, and instead advocates for variegated and heterogeneous conditions that can be specifically targeted for desired lighting conditions, visual effects, and the perception of space."

We are investigating a system of reflecting surfaces that can adjust to redirect light from a source to a target location. The investigation was initially conducted through a digital fabrication studio titled "The Grid, the Cloud, and the Detail," which investigated methods to alter the targeted reflection of daylight as an environmental variable that informs the reading of an existing space.

To test the concept, the research team designed, fabricated, and installed a full-scale prototype, shown in Figure 1.1. The surface was defined by a hyperbolic paraboloid geometry that spanned between the existing columns and beams. The field of custom reflective surfaces acted independently from the twisting surface, thus the system was adaptable to operate on any guiding surface topology [2].

This initial prototype did not support dynamic movement on the part of the individual mirrors. This motivated the design and construction of another prototype.

The installation shown in Figure 1.2 is located in the atrium of Steinberg Hall on the Danforth Campus of Washington University in St. Louis. As with the initial prototype, it needs to

Figure 1.1: Reflective surface prototype installed in the atrium of T-Rex building [2].

a b

Figure 1.2: (a) Image of the entire catoptric surface system. (b) A part of mirrors in the system with a close-up image of an individual module [1].

respond to the existing columns and beams within the atrium space. It is constructed using over 600 individual 3 in. mirrors under pan/tilt control.

1.2 Catoptric Surface Research Aims

The high-level aim of this research project is to explore the generation of visual effects from daylight projected onto architectural surfaces, with the goal of amplifying or reducing the spatial perception of a built environment. By mapping variable organizations of light onto existing or new surfaces, this research project aims to create a condition where the perception of space does not rely solely on form. The resulting effect is a formless atmosphere that can significantly impact the way people use and interact with the space [2].

Again quoting Ahrens et al. [2]:

> The main objective of this research is the production of a unique facade system that is capable of dynamically redirecting daylight to key locations deep within a building. Mirrors in a vertical array are individually adjusted via stepper motors in order to reflect more or less intense daylight into the interior space according to sun position and an image-based map. The image-based approach provides a way

to specifically target lighting conditions, atmospheric effects, and the perception of space.

1.3 Mirror Position Detection in a Catoptric Surface

The Catoptric Surface research project uses computation and robotics to control reflected daylight through a building envelope to form an image-based pattern of light on the building interior's surfaces. The project is an innovative approach to designing interactive architecture that responds dynamically to environmental variables, such as light reflection, to create unique and dynamic spatial experiences.

The pan/tilt mechanisms that make up the catoptric array use a pair of inexpensive stepper motors to control the angle at which each mirror points. Keeping the cost of each module low is important because deploying an array at scale requires many individual modules. This requirement comes with limitations though; the inexpensive stepper motors used for the modules lack any form of encoding. This makes knowing the exact position of the motors – and therefore the angle of the mirrors – impractical. We are investigating several methods for controlling the array with this limitation in mind. The crux of the current research is using computer vision techniques to learn the position of the mirrors in the array and to use that as feedback in a control algorithm. This concept is currently being investigated in two ways: applying computer vision to the mirror position detection and applying computer vision to the reflected surface.

In this thesis, we focus on the mirror position detection approach. We want to use computer vision to detect the positions of the mirrors in the array, ultimately inferring the mirrors' orientation from their positions, which can then be used as feedback in a control algorithm. Image processing with mirrors in the scene is known to be a difficult problem [19, 30]. In our approach, we first use edge detection [5] to recognize the edges in the photos, and then use the RANSAC (RANdom SAmple Consensus) [11] algorithm to pick the ovals based on edges. After that, we try to choose the correct ellipse based on the geometry and use this ellipse to detect the mirror's position. We also execute this program running in parallel, as one approach to diminishing the execution time, because in the complete system there are many mirrors that need to be detected.

1.4 Background and Related Work

Natural light is a crucial aspect of building design [8], as it has a significant impact on the comfort, well-being, and energy efficiency of indoor spaces [3, 7, 13, 23]. Controlling the amount, direction, and distribution of natural light in buildings is a challenging task that requires sophisticated techniques and technologies. The Catoptric Surface research project employs computation and robotics to precisely control the location of reflected daylight through a building envelope, forming an image-based pattern of light on the building interior's surfaces. By utilizing computer vision techniques, such as edge detection, RANSAC [26], and Hough transform, we are able to detect the position and geometry of mirrors in the catoptric array, which is crucial for controlling the direction of reflected light. Additionally, the `multiprocessing` [21] package in Python is employed to enable efficient parallel processing of multiple figures, optimizing the processing time for a large number of mirrors. These advancements [4, 9, 10, 12, 15] in the control of natural light in buildings have the potential to significantly improve the visual comfort and energy performance of indoor spaces, leading to more sustainable and user-friendly built environments.

Mirror detection and segmentation pose significant challenges for computer vision systems, impacting various vision tasks. Recent research has introduced novel computational approaches to address these challenges. One approach, called MirrorNet [30], effectively segments mirrors by modeling semantic and color/texture discontinuities. Another method focuses on content similarity and explicit mirror edge detection [19]. These advancements, along with the construction of diverse benchmark datasets, outperform existing techniques. These findings contribute to improving accuracy in differentiating mirror content from surroundings and provide valuable resources for future research in this domain.

Edge detection is a fundamental step in many computer vision tasks, including object detection, image segmentation, and feature extraction. It involves identifying points in an image where there is a significant change in intensity or color, which often correspond to object boundaries or other important image features. Popular edge detection methods include the Canny edge detector [5], Sobel operator [16], and the Roberts operator [24], which have been widely used in computer vision research and applications.

The Random Sample Consensus (RANSAC) algorithm, proposed by Fischler and Bolles in 1981 [11], is a widely used robust estimation technique for fitting models to noisy data.

RANSAC is particularly suitable for applications where the data may contain outliers or noise, such as in the context of mirror position detection in the Catoptric Surface research project. RANSAC iteratively selects random subsets of data points, fits a model to each subset, and then evaluates the goodness-of-fit of the model. The process is repeated multiple times, and the best-fitting model is selected as the final result. There is another important concept called *connected components* [25]. Connected components are groups of pixels or image regions that are connected to each other based on certain criteria, such as having the same intensity value or being adjacent to each other. It can be used in conjunction with RANSAC to further refine the detection and localization of mirror positions in the Catoptric Surface research project, by grouping nearby edge points or features into connected components to identify coherent structures that correspond to the mirrors.

The Hough Transform, proposed by Paul Hough in 1962 [14] is a popular image processing technique that is commonly used for detecting geometric shapes, such as lines and circles, in images. The Hough Transform is particularly useful for detecting straight lines, which can be relevant in the context of mirror position detection in the Catoptric Surface research project. The Hough Transform works by representing image points in a parameter space, where points that lie on the same line in the image domain will intersect at the same point in the parameter space. This allows for the detection of lines even in the presence of noise and partial occlusion.

We tried two RANSAC libraries to detect the ovals. One is from Emil Tam [27] the other is from the `skimage` package in Python [29]. For the RANSAC library from Tam, some results are shown in Figure 1.3. In this figure, we can easily see the results are not very good. Upon investigation, we determined that we were not properly understanding the scoring function used. So we them moved to the second RANSAC approach which is based on the `skimage` package in Python. As this is the approach we ultimately chose, results will be illustrated in subsequent chapters.

We also tried the Hough Transform method to detect the ellipses, but the algorithm did not recognize the ovals as what we expected. These results are shown in Figure 1.4. In Figure 1.4, the ellipses which are detected by Hough Transform algorithm are the ovals which are drawn in red curves and are not what we want. We suspect that the Hough Transform algorithm is more sensitive to the noise points which looks at the lines in the edge detection figures. So we do not use Hough Transform to detect our mirror in this project.

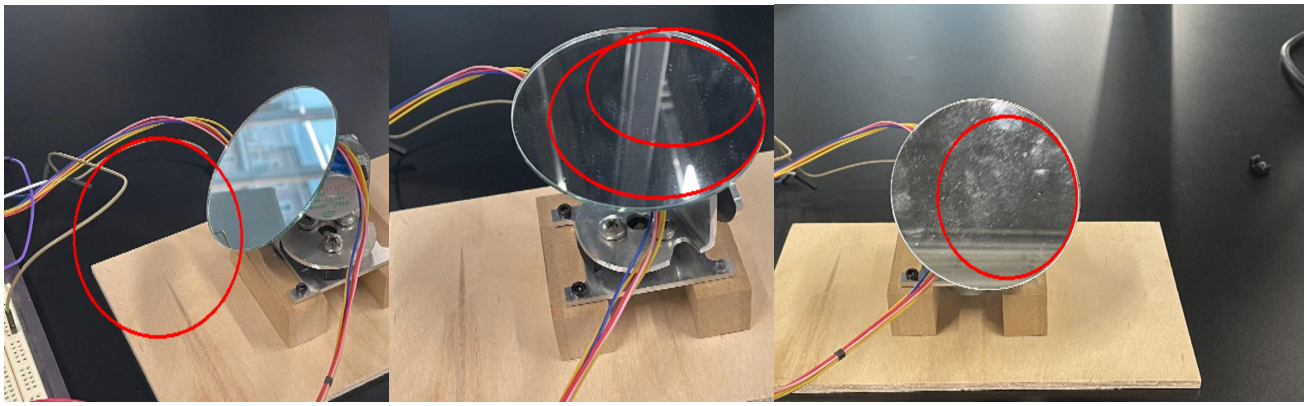

Figure 1.3: GitHub library RANSAC results.

This work is only a portion of the activity present in the catoptric research project. Other recent results include the characterization of the physical properties of the mirror assemblies [28], application of computer vision techniques to the reflected surface [20], investigation of the effectiveness of the surface at changing light intensity [18], and the development of a user-friendly interface [17].

1.5 Contributions and Outline

The contributions of the thesis include the following:

- Experimental setup to catch single and 10 step mirror movement images using a camera. These images were then used to assess the ability to determine mirror position from the imagery.
- Author Python code to invoke edge detection and calculate the connected components in an image.
- Evaluate the effectiveness of 2 RANSAC functions [27, 29] and the Hough transform [22] for the task of mirror position determination.
- Write Python code to determine the implied ellipse position within the image.
- Alter the Python program to execute in parallel.

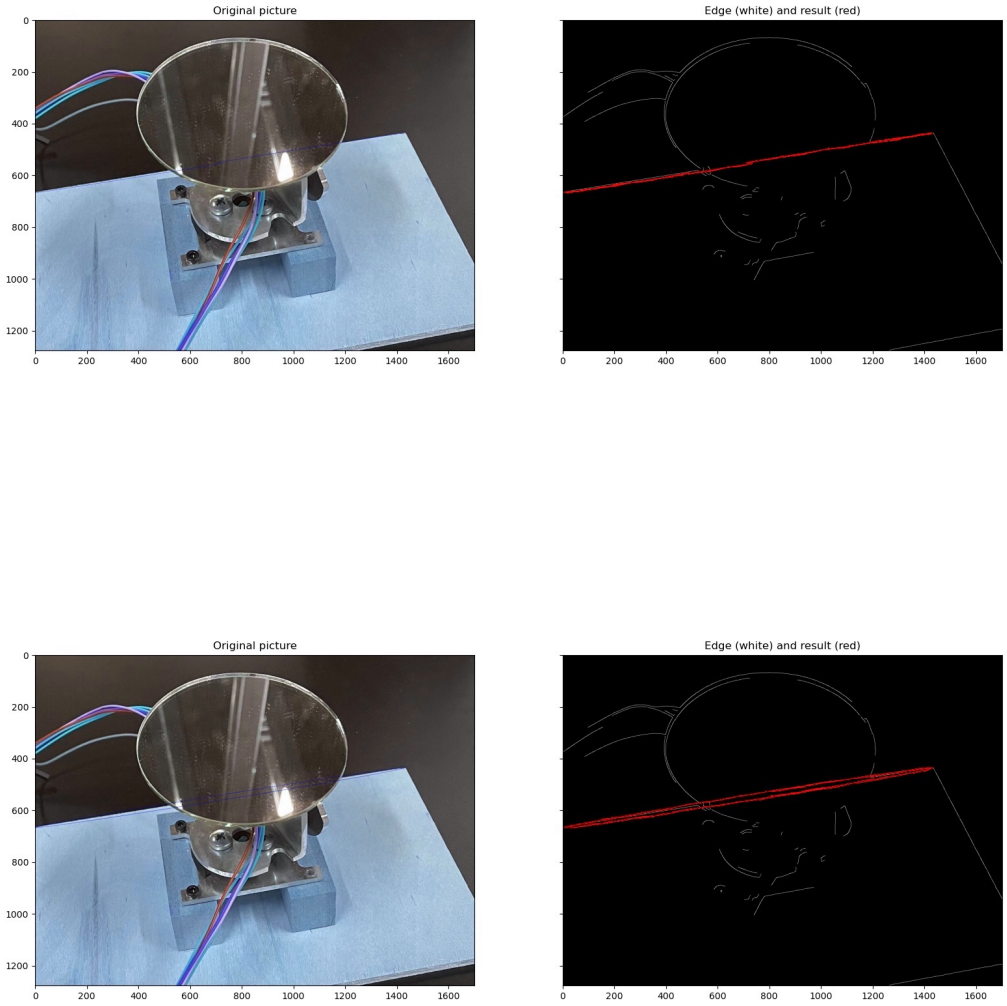

Figure 1.4: Hough transform result.

- Show the changes of ellipses' center points based on mirror movement.

This thesis covers current types of computer vision techniques to learn the orientation of an individual mirror in the array with the intent to use that as feedback in a control algorithm. Chapter 1 discusses the Catoptric Surface research project, using computation and robotics to control reflected daylight through a building envelope to form an image-based pattern of light on the building interior's surfaces. In Chapter 2, we describe the experiment we run to obtain mirror figures in order to use image-based algorithms to detect mirror's positions in those figures. In Chapter 3, we show the results of the experiments, edge detections, RANSAC based ellipses detection and mirror positions, etc. In Chapter 4 we discuss potential future work and conclusions of the thesis.

Chapter 2

Methods

In this chapter, we describe our approach to determining the center positions (and therefore orientations) of individual mirrors. While the system will eventually need to operate on multiple mirrors, for the scope of this thesis, we constrain our focus to an individual mirror.

From Figure 1.2, we abstract a simple model which is shown in Figure 2.1. This 2×2 system is also constructed in the laboratory (see Figure 2.2) and has been used for previous experimental work on the project [20, 28].

2.1 Experiment

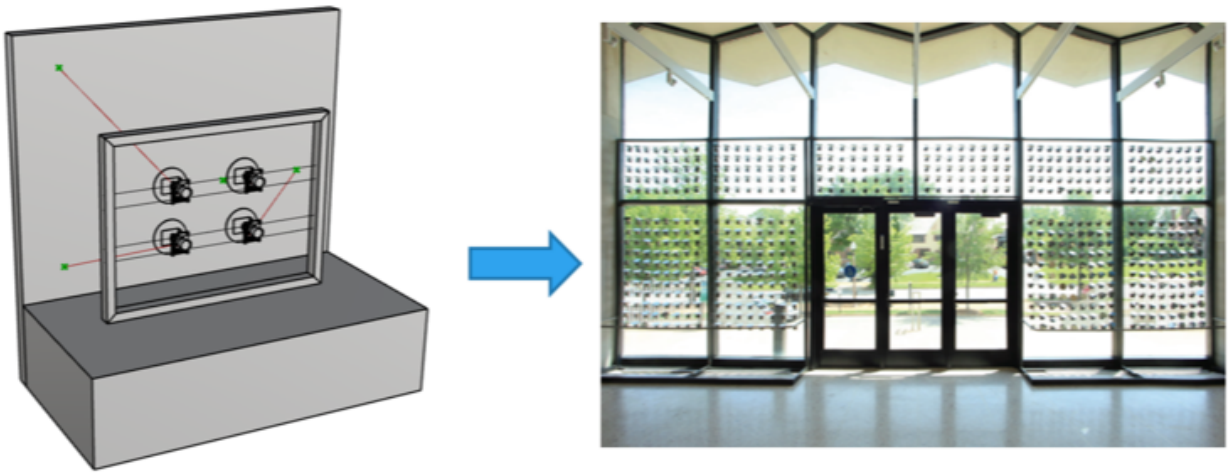

Figure 2.1: A simple model representing the complete system.

In Figure 1.2, the entire catoptric surface system in Danforth Campus of Washington University in St. Louis Steinberg Hall has hundreds of mirrors. It is very hard to do our

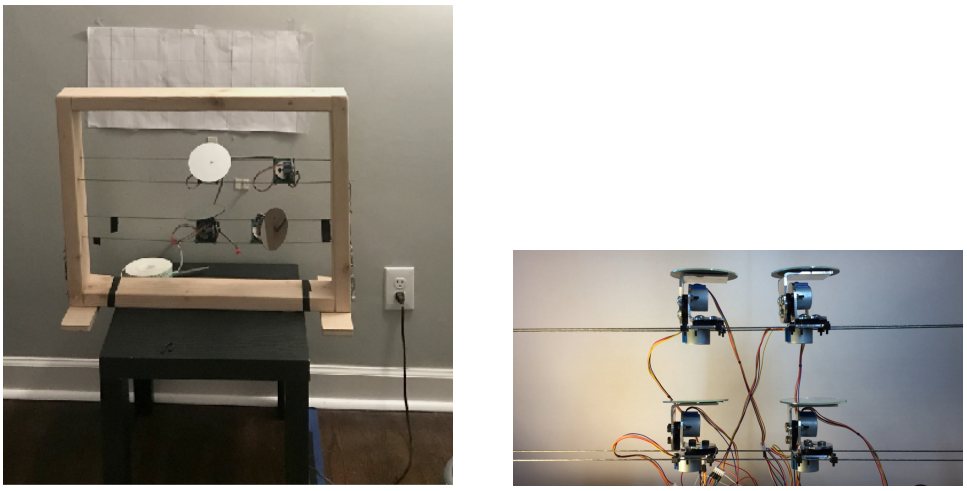

Figure 2.2: 2×2 system constructed in the laboratory [28].

experiment in this big catoptric surface system. So, we generate a prototype which is shown in Figure 2.1 and Figure 2.2. The left image in Figure 2.1 is a concept figure which helps people understand how can we abstract this 2×2 prototype from the whole catoptric surface system. Figure 2.2 is the real prototype in our lab. Based on this prototype, we can choose a single mirror to test our algorithms and tune parameters in the algorithms. When we get the single mirror's data and have tested our algorithms successfully, it will then be time to employ the experiment's algorithms to the whole system. For the lab experiments that have been performed, we give some details below.

Firstly, we take some photos for the single mirror in different random angles. Based on these images we get edge detection results which are shown in the next section and in Chapter 3.

Subsequently, we set up the simple model (Figure 2.1) which is abstracted from the whole system. We let the motors move different steps each time in different pan and tilt angles. Then, we use the following three steps to record the mirror's moving routine.

Step 1. Command the motor to move 10 steps for each image. Part of the images are shown in Figure 2.3. Initially, command the mirror to move all the way to the left (in pan) and keep the mirror in the middle (in tilt). When the motor moves 10 steps in pan, command the motor to keep the position for 5 seconds to get a picture. Continue the mirror's motion until the mirror stops at the far right.

Figure 2.3: 10 steps move samples.

Step 2. Command the motor to move 10 steps for each image. Initially, command the mirror move to the top (in tilt) the mirror in the middle (in pan). When the motor moves 10 steps in the tilt level, command the motor to keep the position for 5 seconds to get a picture. Continue the mirror's motion until the mirror stops at the bottom.

Step 3. Command the motor to move single step each time and repeat the step 1 and step 2 (with the motion in the middle of the range in pan and tilt), the results of which are shown in Figure 2.4.

Due to our focus on on a single mirror, we crop the images manually, and as a result of that the image sizes are a little bit different.

In Figure 2.3, we include all the 10 steps move experiment images. We can easily see the difference between each angle of movement. So we anticipate the 10 steps movement experiment will show the changes between each individual mirror orientation.

In Figure 2.4, we include all the single step move experiment figures. We can easily see there are not very much difference between each angle of movement. So we anticipate the single step movement experiment will have difficulty discerning the change between each individual mirror orientation.

2.2 Edge Detection

Edge detection is a fundamental step in image processing, computer vision, and object recognition. It is used to detect the boundaries of objects within an image and is a critical pre-processing step in many computer vision applications. Edge detection is performed by analyzing the intensity changes in an image to identify areas where the brightness changes abruptly. These areas are referred to as edges.

There are several methods for edge detection in computer vision. One common approach is the Canny edge detector, which was introduced by John F. Canny in 1986 [5]. The Canny edge detector is a multi-stage algorithm that involves smoothing the image with a Gaussian filter, calculating the gradient magnitude and direction, suppressing non-maximum edges,

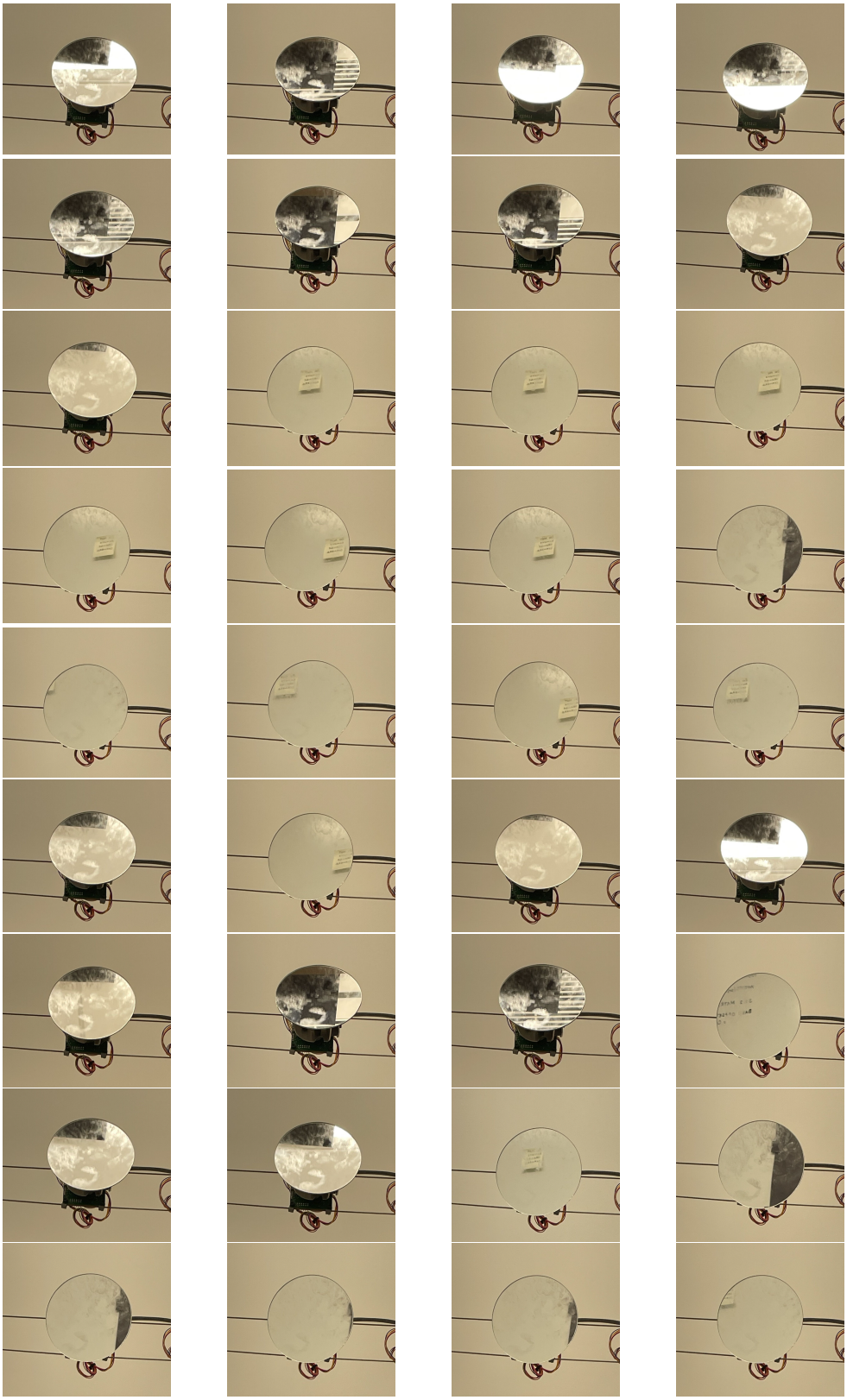

Figure 2.4: Single steps move samples.

and finally, applying hysteresis thresholding. The result of the Canny edge detector is a binary image where the edges are white and the non-edges are black.

Based on the randomly took photos for the single mirror in different angles, we get the edge detection figures by using `cv2` packages in Python.

For get these edges, we firstly convert the image to grayscale. Then, detect edges using edge detector and set the lower and upper threshold of the edge detector. Some edge detection figures are shown in Figure 2.5.

In Figure 2.5, the first row shows the original image used to test edge detection, and the second row gives the edge detection results for a single mirror. Due to our focus on a single mirror, we cropped the figures manually. It is apparent from these images that the ellipse representing the mirror boundary is effectively present in the edge detection results.

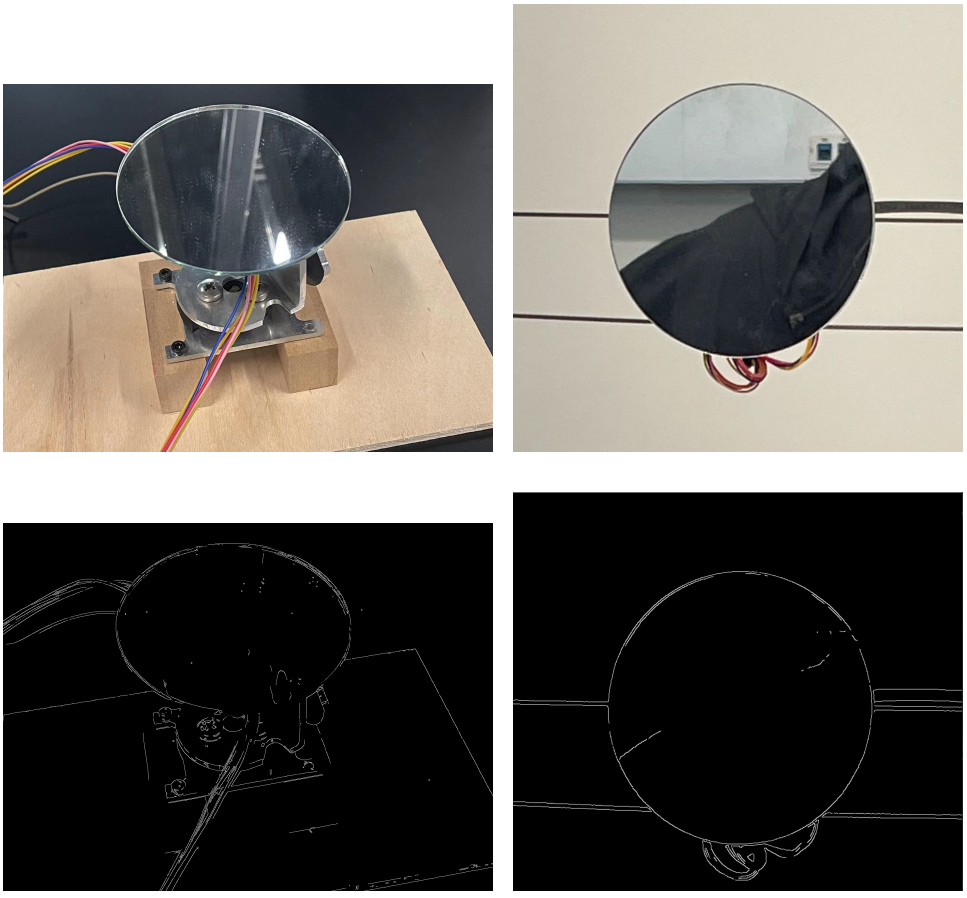

Figure 2.5: Edge detection samples. Top row is a pair of initial images, bottom row is the edge detection output.

2.3 RANSAC (Random Sample Consensus)

RANSAC, which stands for RANdom SAmple Consensus, is a widely used robust method for estimating parameters of a mathematical model from a set of observed data points that may contain outliers. It is first introduced in the computer vision community by Fischler and Bolles in 1981 [11] and has since been applied in various fields such as robotics, geomatics, and computer graphics.

The basic idea behind RANSAC is to randomly sample a subset of data points, and then fit a model to the selected points. Consider the mirror position detection problem, we assume we want to detect the data points from the edge detection results that fit an ellipse. Let S be a subset of k data points selected at random from a set of n data points. Let M be the model that is fitted to the selected points. In line fitting, for example, M might be a line that is parameterized by its slope and y-intercept. Let D be the set of data points that are not in S. The goal of RANSAC is to determine which points in D are consistent with M. A data point x_i is said to be consistent with M if it lies within a certain distance ϵ of the model M. Let C be the set of data points in D that are consistent with M. The goodness-of-fit of M is measured by the size of C, which is called the consensus set.

The algorithm repeats the sampling and fitting process for a fixed number of iterations t. At each iteration i, a new subset S_i is selected at random from the data set, and a new model M_i is fitted to S_i. The size of the consensus set C_i is then computed by checking which points in D are consistent with M_i. The algorithm returns the model M that has the largest consensus set C, as well as the set C itself. The algorithm can be summarized in the following steps:

1. Select a random subset S of k data points from the data set.
2. Fit a model M to the selected points.
3. Compute the set C of data points in D that are consistent with M.
4. If the size of C is greater than a certain threshold t, re-estimate M using all the points in S and C, and compute the new consensus set C'.
5. Repeat steps 1-4 for a fixed number of iterations, and return the model M that has the largest consensus set C, as well as the set C itself.

For our case, the model M is an ellipse. Some illustrative RANSAC results are shown in Figure 2.6.

2.4 Ellipses Selection

Based on the RANSAC results illustrated in Figure 2.6 we can easily see there are many ovals in the figure. We still need to decide which oval is the one of interest.

In order to select ovals based on their geometry, the eccentricity, e, of the ovals can be calculated using Python. The eccentricity is a measure of how elongated or flattened an oval shape is, with a value ranging from 0 (perfect circle) to 1 (a straight line). The calculated eccentricity can then be compared with the eccentricity of the ovals detected using RANSAC. By comparing the calculated eccentricity with the eccentricity of the RANSAC detected ovals, ovals with similar eccentricity values can be chosen as potential matches. This approach allows for the selection of ovals that exhibit similar geometric characteristics as the real ovals of interest, which can aid in the accurate detection and localization of mirrors in the catoptric surface.

In our project, we calculate the eccentricity plus the length of the X-axis and Y-axis (in pixels). Based on the good oval detection results which from RANSAC algorithm, we calculate the eccentricity and the length of X-axis and Y-axis which are shown in Figure 2.7 and make the Table 2.1 from Figure 2.7. In Table 2.1, the geometry of the oval tell us the eccentricity is 0.7857. So, we set up a range for the eccentricity as (0.78, 0.8) which means if the oval's $e \geq 0.78$ and $e \leq 0.8$, we will let the RANSAC output the oval.

Table 2.1: X-axis and Y-axis lengths and eccentricity.

Ellipse parameters	Value
X-axis length (pixels)	426.6
Y-axis length (pixels)	263.9
Eccentricity, e	0.7857

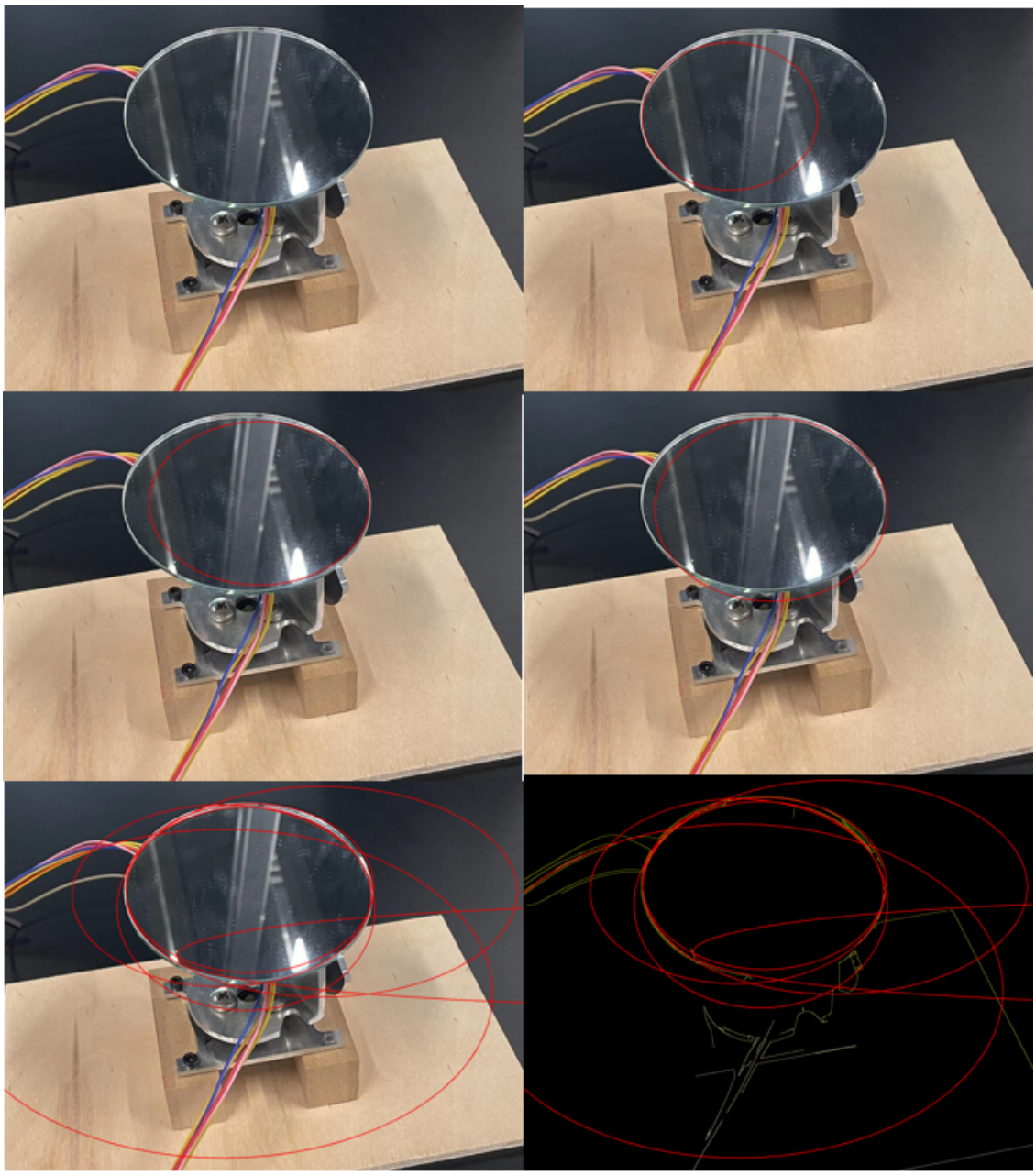

Figure 2.6: A set of images showing the results of the ellipses detection process. The first image is the original image. The second image is after the first iteration the first ellipses detection result by using the RANSAC algorithm. The third image is another ellipse detected by using the RANSAC algorithm. The fourth image is the best fitted result. The fifth image is all the ellipses RANSAC detected. The last image is the ellipses showed in connected component image.

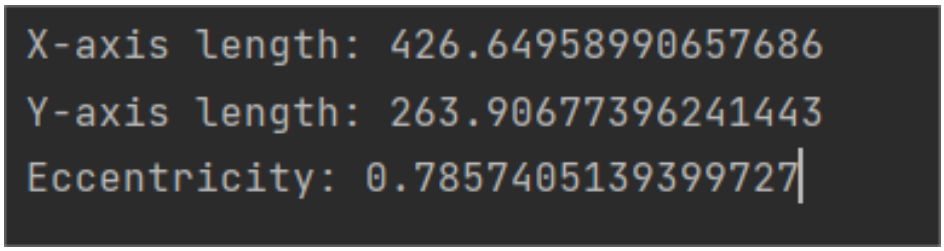

Figure 2.7: Eccentricity and X-axis, Y-axis lengths.

2.5 Position Calculation

Once the ovals with similar eccentricity values have been selected based on the comparison between calculated eccentricity and the RANSAC detected ovals, the center point of the oval can be calculated to represent its position in the images. The center point of an oval can be computed based on its x-axis and y-axis values. In general, the x-axis and y-axis values represent the major and minor axes of the oval, respectively. The center point of the oval can be calculated as the midpoint between the endpoints of the major and minor axes. This can be achieved using simple mathematical formulas to compute the average of the x-axis and y-axis values, which gives the center point of the oval. This approach provides a way to accurately determine the position of the selected ovals in the images, which can be utilized for further analysis or visualization in the catoptric surface research project.

2.6 Program Multiprocessing

To efficiently process a large number of figures in the catoptric surface research project, the `multiprocessing` [21] package in Python is utilized to enable parallel processing. With the `multiprocessing` package, multiple figures can be processed simultaneously, leveraging the available CPU cores for concurrent execution. This allows for significant speedup in the overall processing time, as each figure can be processed independently and in parallel, without waiting for the completion of previous figures. The `multiprocessing` package provides functionalities for creating multiple processes, distributing the workload across the processes, and managing communication and synchronization among them. By utilizing multiprocessing, we are able to significantly reduce the computational time required for processing a large number of figures, improving the efficiency and scalability of the research project. Although

multiprocessing can speed up the oval detection step, it is very hard to speed up the ellipse selection step. Because the use of geometric constraints to restrict the target ovals frequently makes the algorithm go back to the RANSAC step several times. This is a point we need to improve.

Chapter 3

Results

In this chapter, we first illustrate some images from the experiments, then proceed to show results from edge detection, RANSAC, ellipse selection, and position calculation. This is followed by a discussion of the experimental results.

3.1 Experiment

In Figure 3.1 we command the mirror to move once (10 steps) which resulted in these two images. Given the differences in the two images, we anticipate that this level of movement will be detectable by our algorithms.

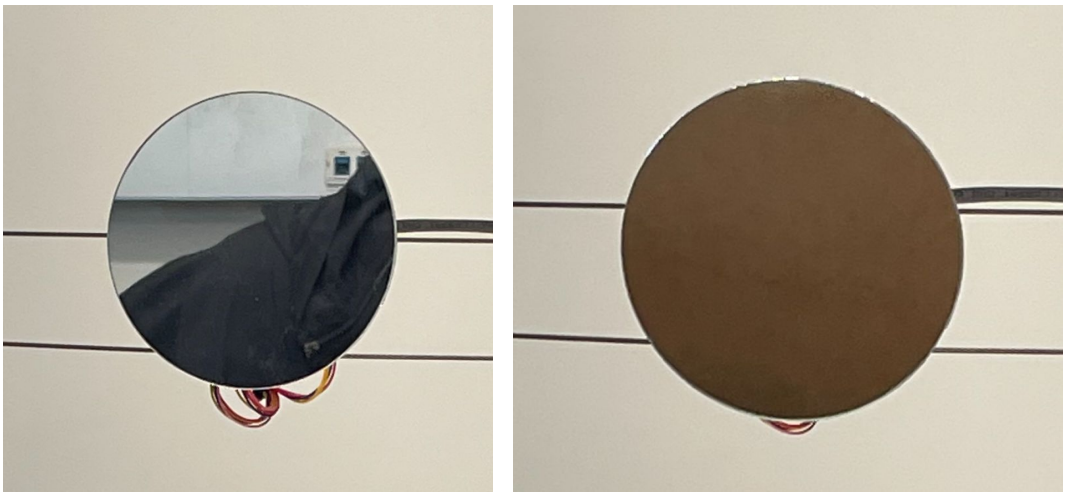

Figure 3.1: 10 steps move experiment.

In Figure 3.2 we command the mirror to move in single steps and these images result. Based on this, we want to see if the camera can catch the changes of the oval's position. Based on Figure 3.2, these three figures seem to not have any differences, it potentially cannot

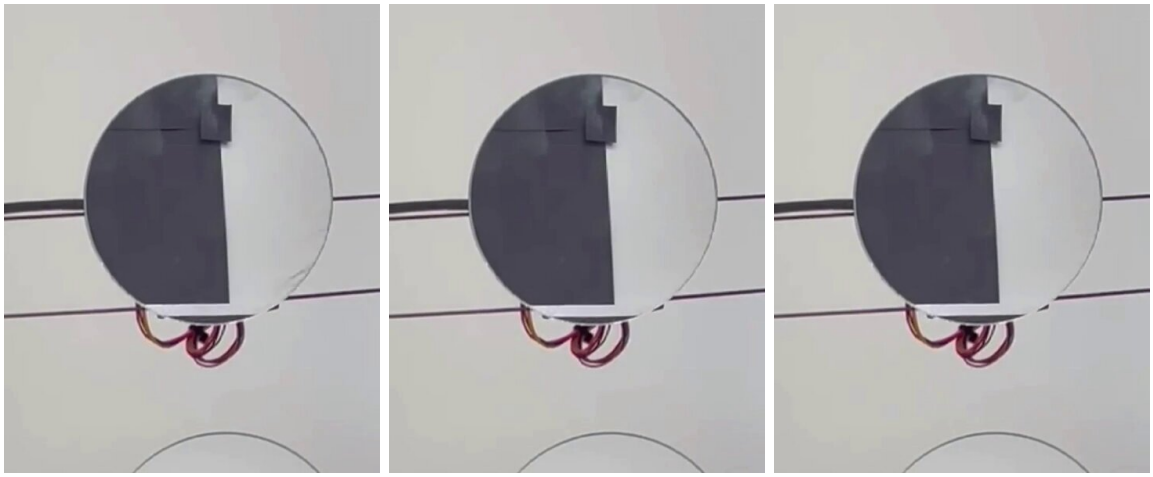

Figure 3.2: Single steps move experiment.

be detected by our algorithms. We address the answer to this question at the end of the chapter.

3.2 Edge Detection

In Figure 3.3, we show one of the edge detection results based on a single mirror. We detect edges using the Canny edge detector and set the threshold of the Canny edge detector be lower threshold = 300 pixels and upper threshold = 400 pixels. As is clear in the figure, in this case it does a very good job of detecting the edge of the mirror. Also, using two thresholds with hysteresis allows more flexibility than single-threshold methods, but the general problem with threshold methods remains. Setting the threshold too high may miss important information. On the other hand, setting the threshold too low can falsely identify irrelevant information (such as noise) as important information. It is difficult to give a general threshold that works for all images. So, when we process different images, it may be necessary to use different thresholds.

We next investigate the effectiveness of edge detection over a wider range of mirror positions. Based on our 10-step experiment, edge detection results with lower threshold of 300 pixels and upper threshold of 400 pixels are shown in Figure 3.4.

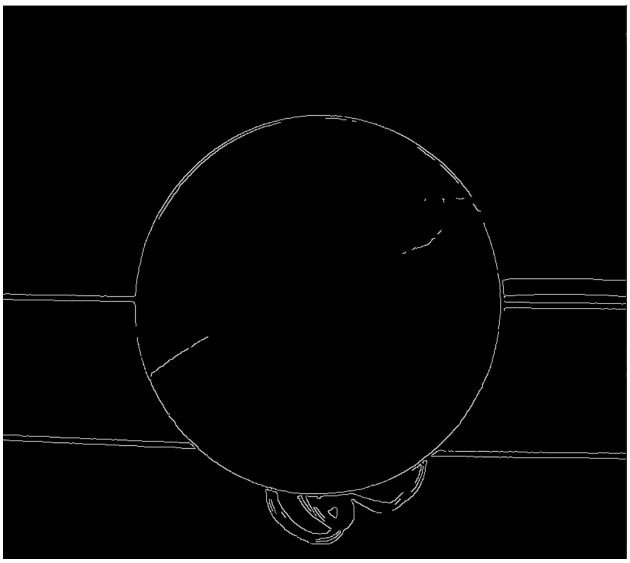

Figure 3.3: Example edge detection output.

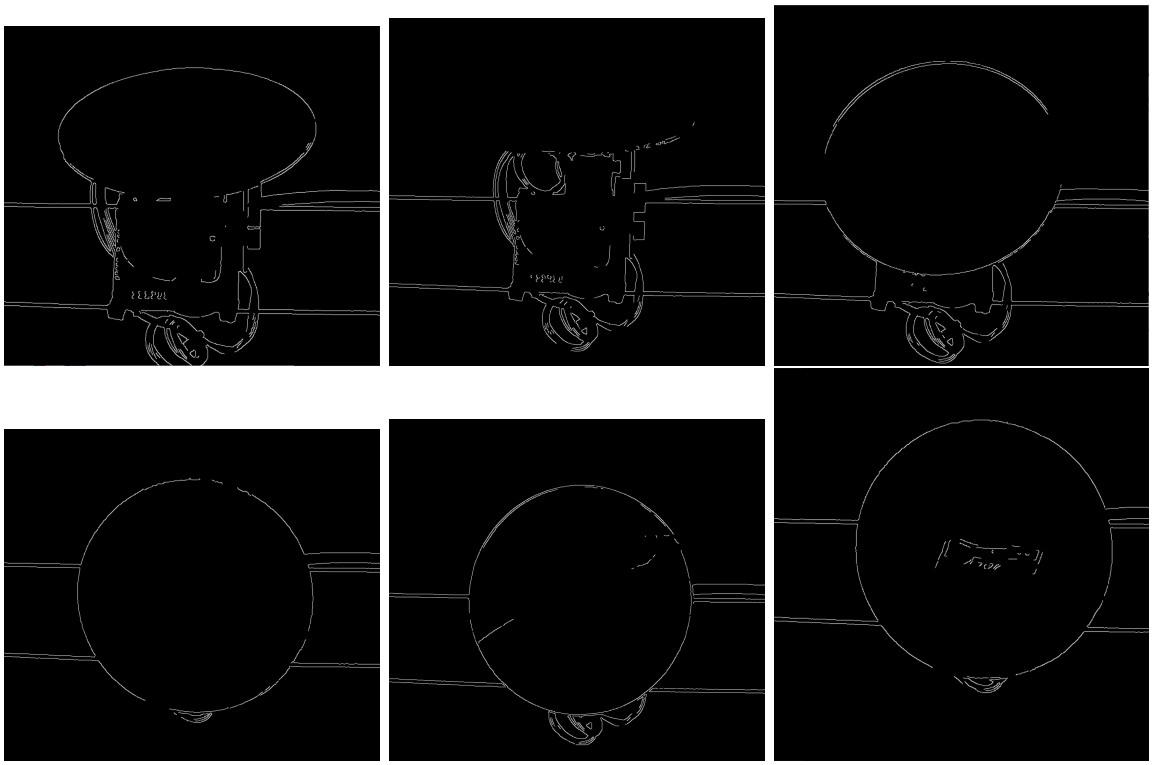

Figure 3.4: Edge detection output for 10-step experiment images.

In Figure 3.4, we can easily see there are differences between each figure. The second figure in the first row misses the oval area, with the mirror pointed up. For the third figure in the second row, there are some edges in the middle of the oval, it probably be noise for RANSAC to detect the correct oval.

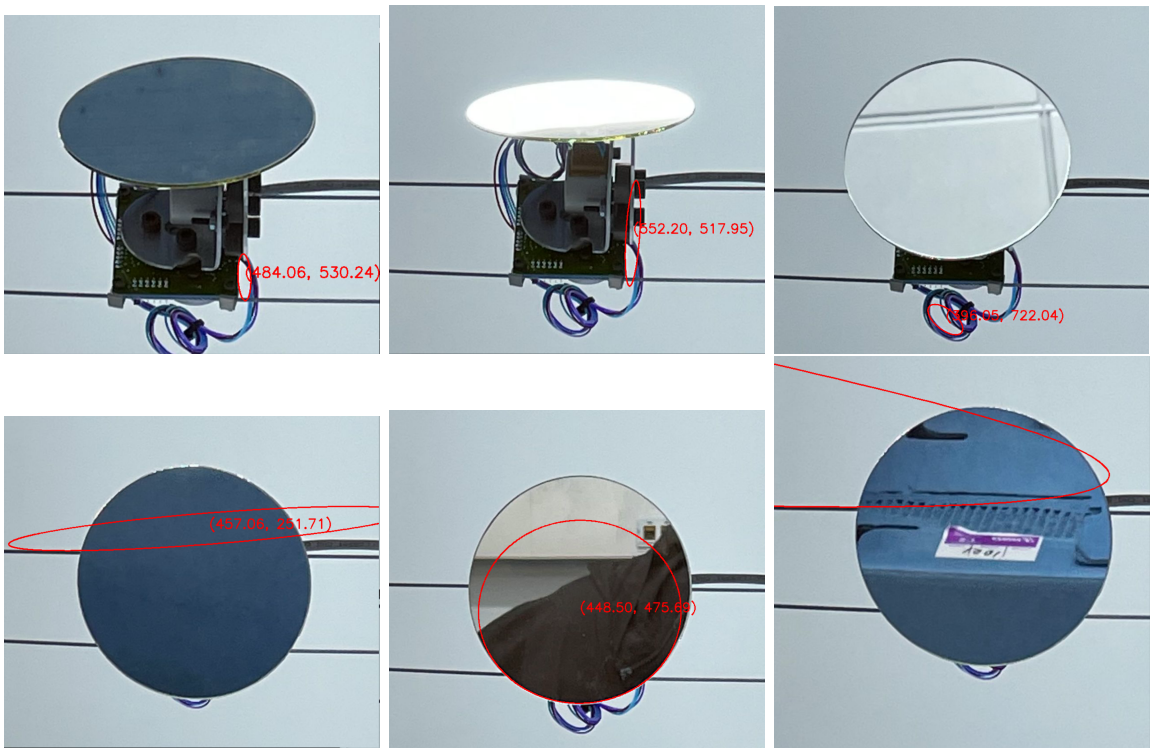

Figure 3.5: RANSAC output for 10-step experiment images.

In Figure 3.5, we show the RANSAC results for these same images. Clearly, the RANSAC results are not very good in part because the edge detection results are not very good. Some mirror edges are not be shown very clear and some area have very big gap so that the connected component area will be not accurate. In this situation, RANSAC will detect the incorrect oval.

Based on our single-step experiment, some edge detection results with lower threshold of 300 pixels and upper threshold of 400 pixels are shown in Figure 3.6. Similarly, RANSAC results are shown in Figure 3.7.

In Figure 3.6 and Figure 3.7, it is hard to catch the changes in each single figure. And this threshold does not work very well on this set of images.

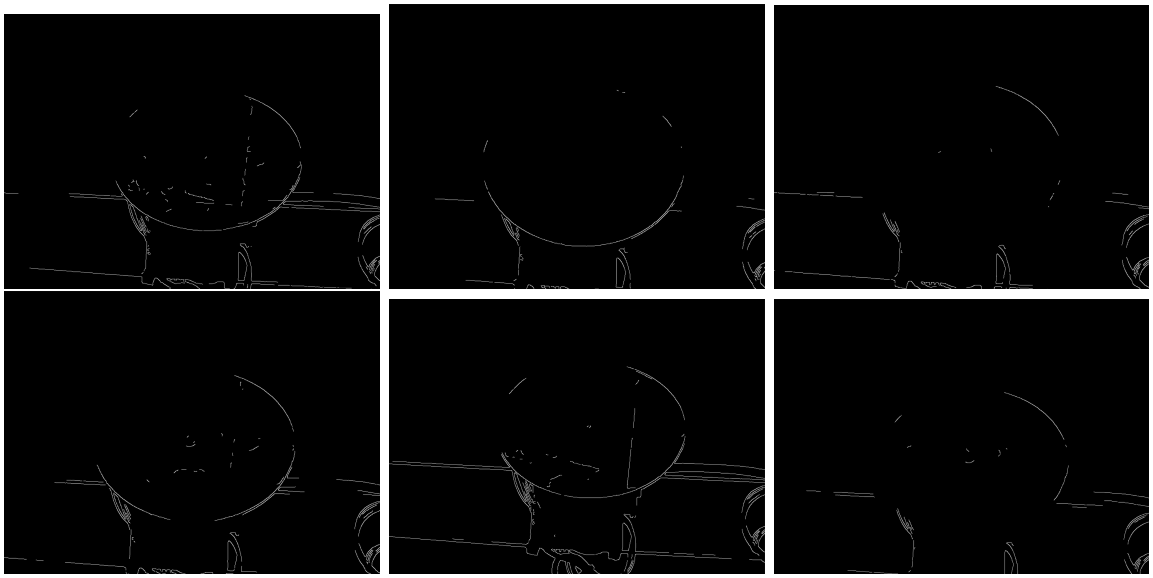

Figure 3.6: Edge detection output for single-step experiment images.

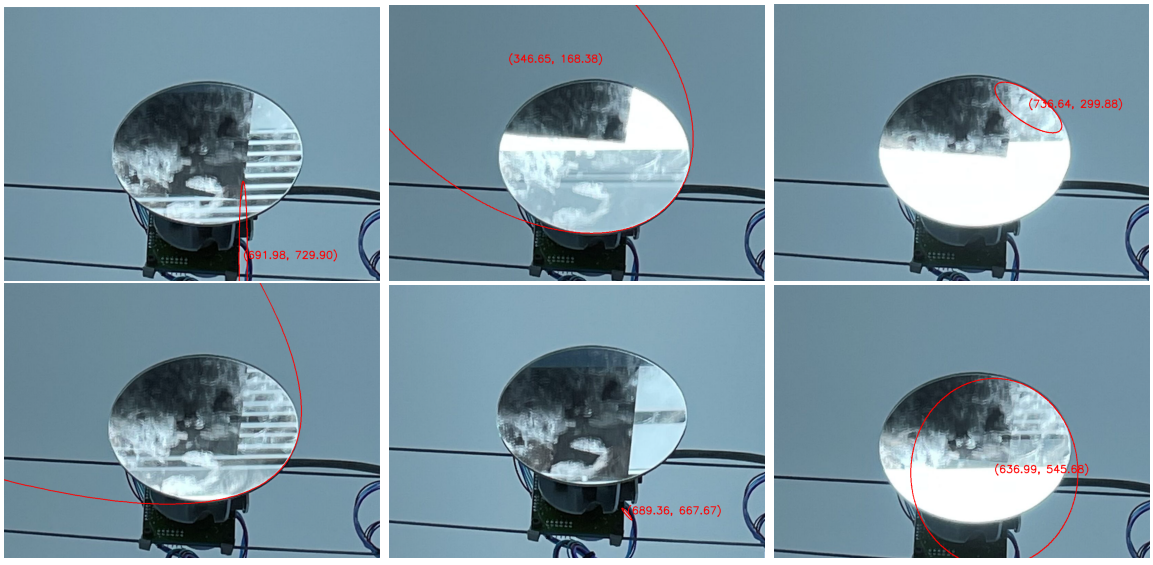

Figure 3.7: RANSAC output for single-step experiment images.

But there still be some problems for edge detection. Based on different figures, we should choose different thresholds to do Canny edge detection. For many of our photos, let lower threshold = 300 pixels and upper threshold = 400 pixels is a good choice. But in some photos like Figure 3.6 and Figure 3.4, this threshold cannot detect all these edges, it has more gaps between each connected component areas. This issue will let us detect some ovals are not what we expected.

3.3 RANSAC (Random Sample Consensus)

Based on an image of the single mirror prototype, the progress of the processing is illustrated in Figure 3.8. Based on the edge detection results, we calculate the connected component areas and highlight them. After that, we try to fit the oval which will overlap as much connected component areas as possible.

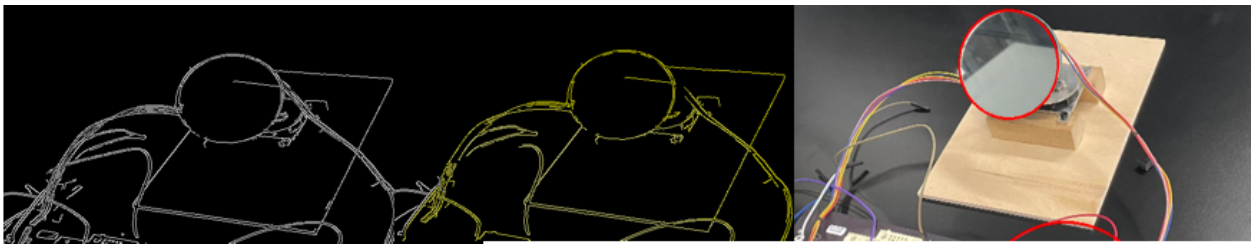

Figure 3.8: A set of images showing the results of ellipse detection. The first image is the edge detection result from the original image. The second image is the connected component result from the edge detection image. The third image is the ellipse detection result by using the RANSAC algorithm on the connected component image.

The RANSAC results from several images in the 10-step experiment are shown in Figure 3.9. While there are clearly differences in the mirror positions between the different images, the ellipse chosen is far from ideal.

Based on both Figure 3.7 and Figure 3.5, what we firstly should change is the edge detection threshold to make the edge detection result include as similar the connected component area as the mirror area as possible, can let RANSAC get better oval detection results. Also, we use the geometry information of the mirror to restrict the oval we detect which is shown below.

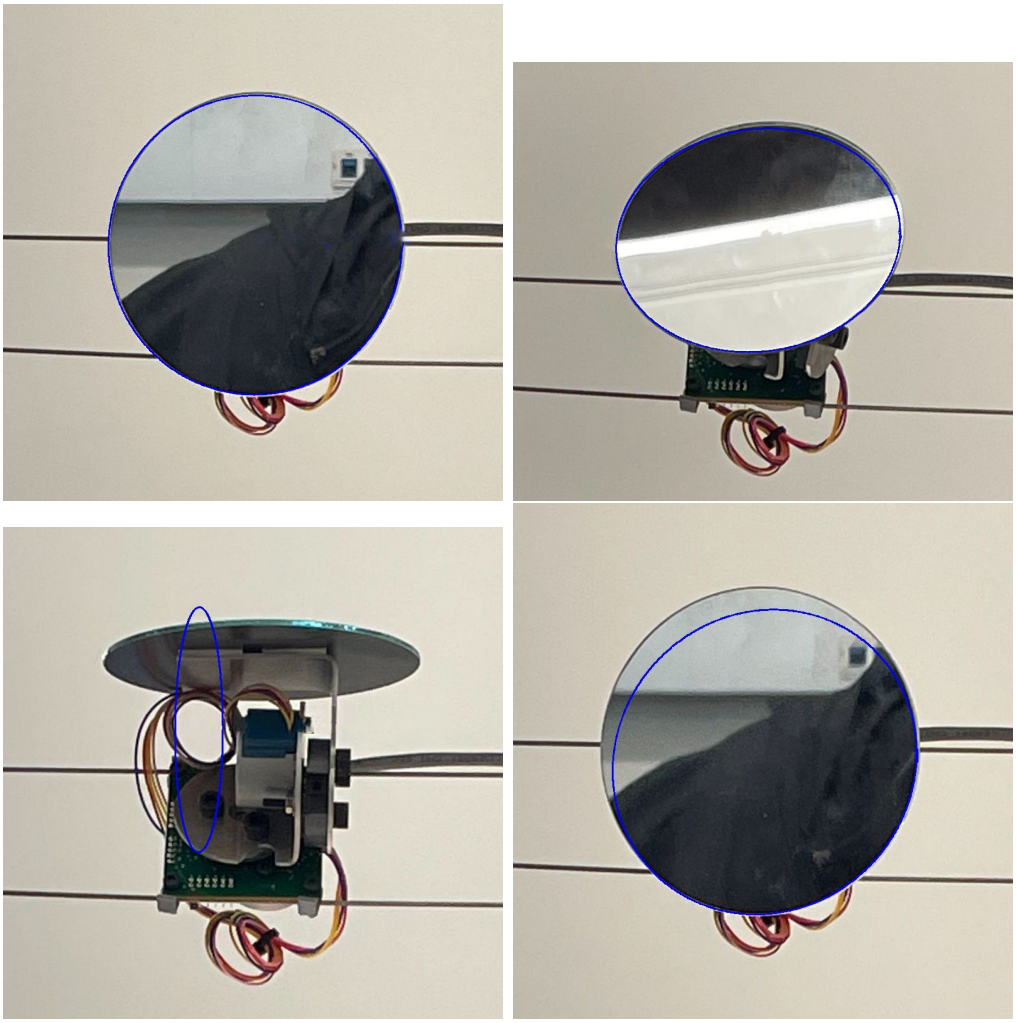

Figure 3.9: Ellipse selection for 10-step experiment images.

3.4 Ellipses Selection

In an attempt to improve the inaccurate results, we use some geometric constraints to let the program know what kind of ovals are what we want. So, based on the geometry of the mirror, we set the eccentricity be in the range (0,78, 0.8) which shows in Table 2.1. Based on this condition, we will spend more time to let the program to select the correct oval which eccentricity is between 0.78 and 0.8. And get the results shown in Figures 3.9 and 3.10.

Compare Figure 3.10 and Figure 3.9 with Figure 3.5 and Figure 3.7, we can easily see if we use the eccentricity to restrict the RANSAC algorithm to detect mirrors, it detects the desired ovals more accurately.

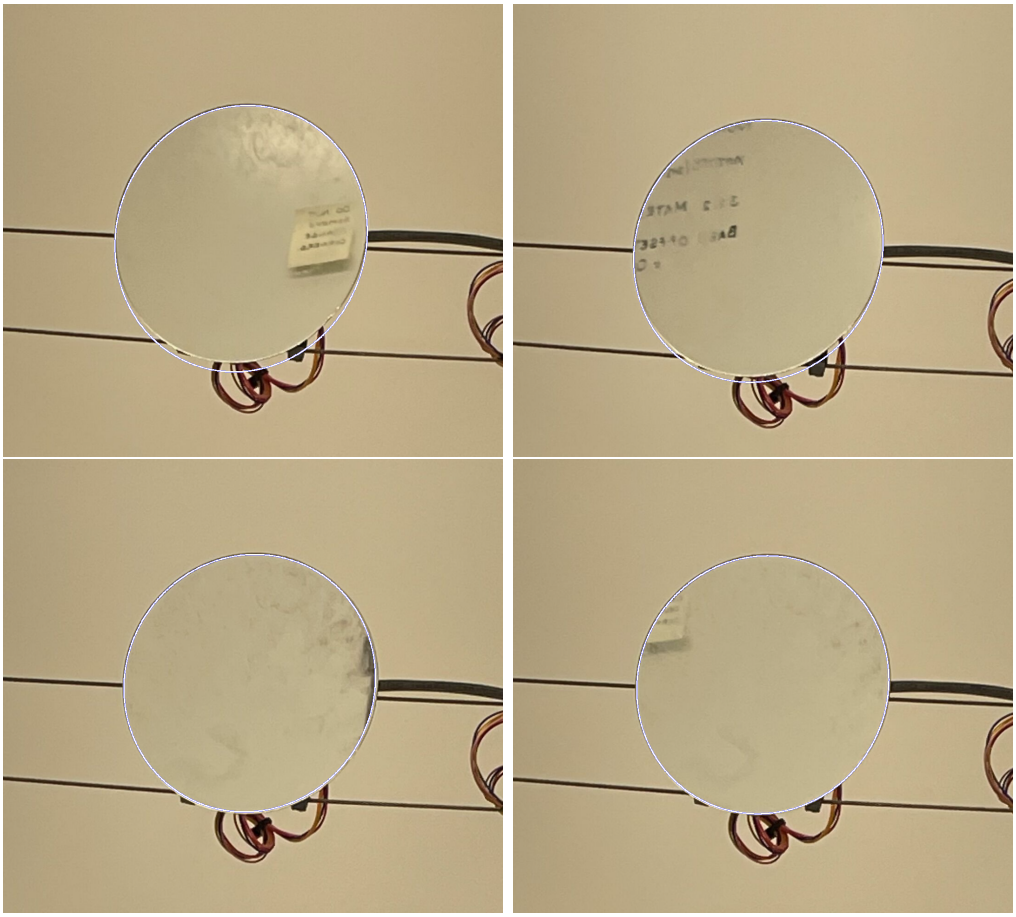

Figure 3.10: Revised ellipse selection for single-step experiment images.

3.5 Position Calculation

After the above improvements in the oval selection, the next issue we need to address is the degree to which these selected ovals can identify the orientation of the mirrors themselves. For example, when a mirror moves a small amount, can this be detected in the imagery.

To assess this, we calculate the center points of the detected ellipses, identify reference base points (manually) and compute the distance from the center points the base points for each output oval. The results of this exercise is shown in Figures 3.11 and 3.12.

Figure 3.11 shows the 10-step experiments results. We choose the same point which have the same position on the real mirror as the base point. Then, we mark the positions of the center points and base points in the figure and calculate the distance between these two points which shows in red in the figures. Figure 3.12 shows the single-step experiment results.

The coordinates of the center points and reference bases, plus the computed distance are repeated in Tables 3.1 and 3.2 for easier viewing.

Table 3.1: Ellipses selection for 10 steps move figures center points table.

Figure 3.11 image (row, column)	Center points (pixels)	Base points (pixels)	Distance (pixels)
(1,1)	(445.83, 426.05)	(450, 762)	336.0
(1,2)	(441.45, 321.13)	(442, 742)	420.9
(2,1)	(312.00, 322.42)	(349, 727)	412.8
(2,2)	(412.50, 424.91)	(410, 744)	319.1

Table 3.2: Ellipses selection for single step move figures center points table.

Figure 3.12 image (row, column)	Center points (pixels)	Base point (pixels)	Distance (pixels)
(1,1)	(498.77, 484.66)	(641, 769)	317.9
(1,2)	(526.40, 527.22)	(666, 802)	308.2
(2,1)	(530.55, 481.17)	(650, 756)	299.7
(2,2)	(537.91, 493.92)	(657, 758)	289.7

Based on these center point positions, base point positions and distances between center points and base points, we can easily see, the first figure of the second row in Figure 3.11 shows, when the mirror move to this angle, our algorithm will not find the mirror correctly

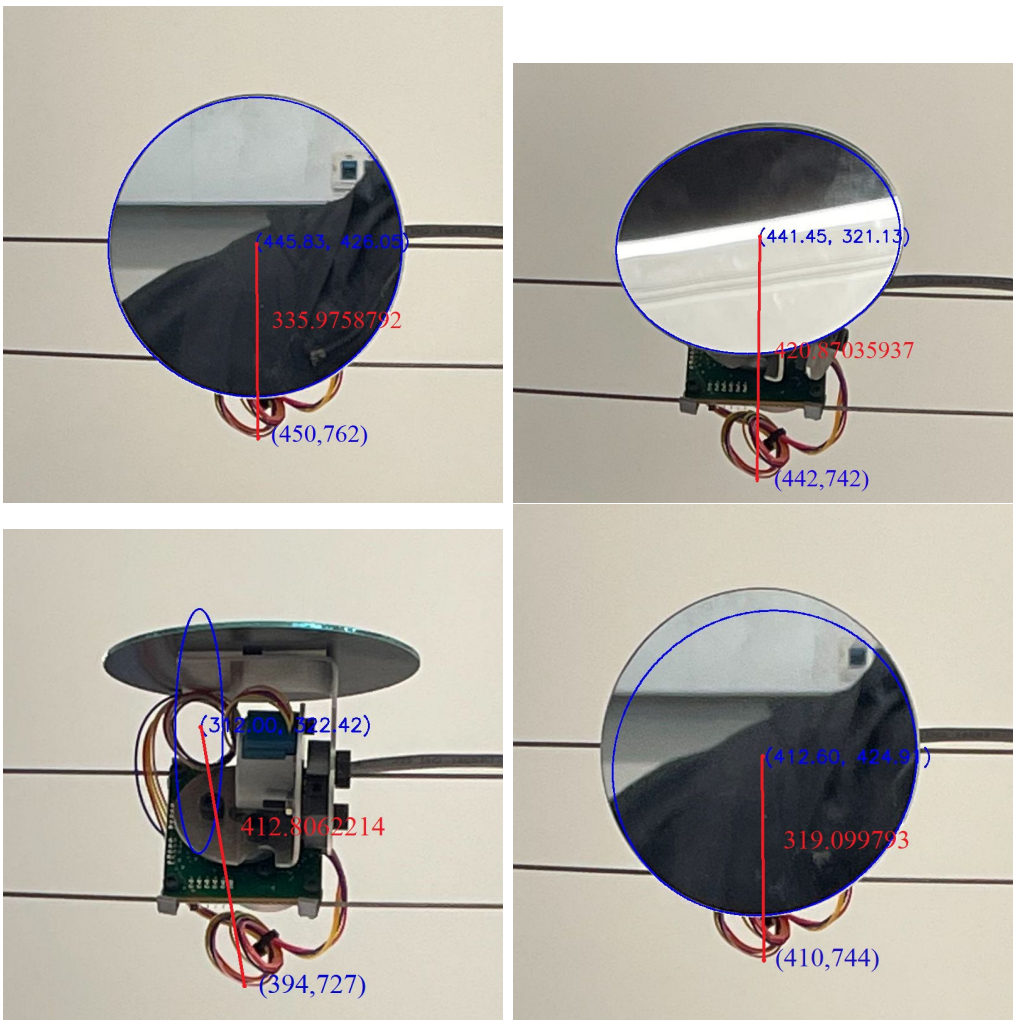

Figure 3.11: In this figure, it shows single mirror move to different angles' position. For the first figure in the second row, it is easy to see that when the mirror moves at this angle, it is hard to catch the correct oval and calculate the distance we expect.

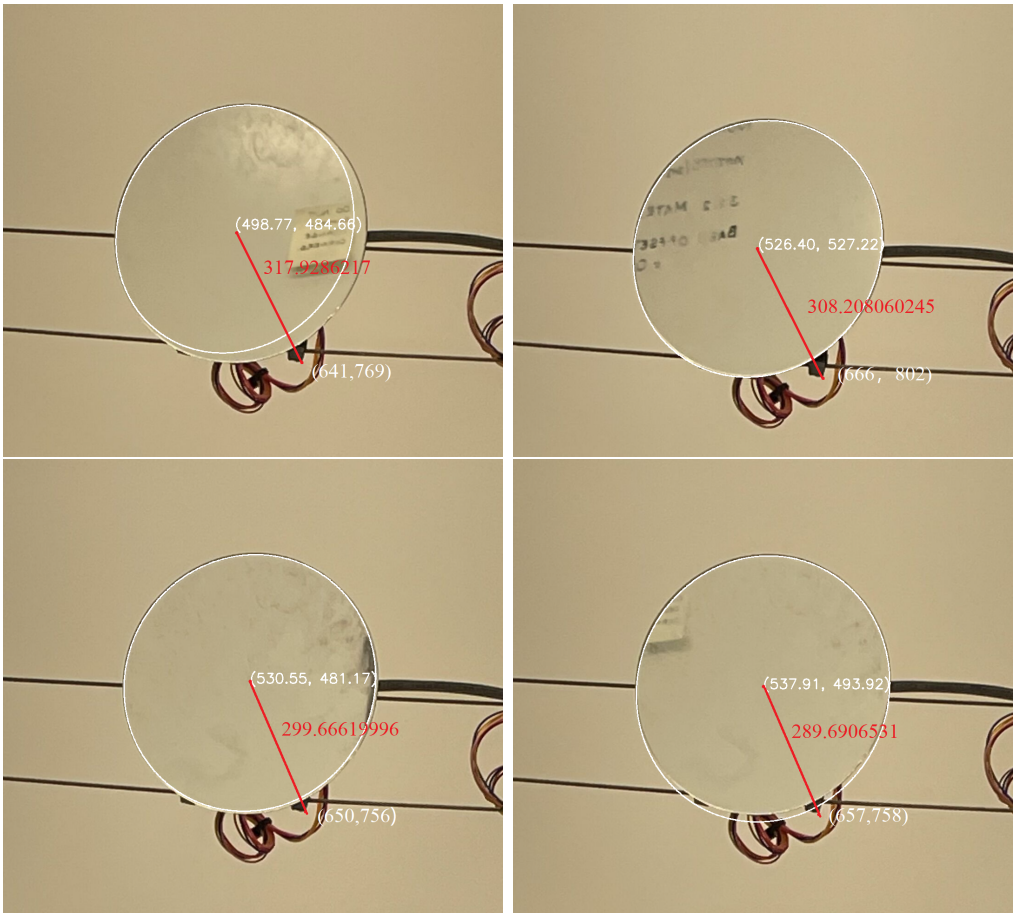

Figure 3.12: For the images in the first row, we show two not continue different single step movement result and choose the same point which have the same position on the real mirror as the base point. Then, we mark the positions of the center points and base points in the image and calculate the distance between these two points which shows in red in the figures. For the second row, it is the single move 2 single steps' results. One of the distance is 299.7 pixels, the other is 289.7 pixels. The difference between these two single steps are 10 pixels. So, one single steps' distance change will be 5 pixels. Think about the pixels' steps are too small to see, it is very hard to see the differences between these two figures in eyes. And think about some random errors, it is also hard to see, this program can detect the differences between the single steps movement.

because this angel is very hard to detect. For the other figures (including Figure 3.11 and Figure 3.12), our algorithm works very good to detect the oval and find the center points, base points and distances from center points and base points in the figure.

3.6 Discussion

In a nutshell, this is a computer vision-based approach for accurately detecting a single mirror's position in the Catoptric Surface research project. The method combines edge detection and the RANSAC algorithm, incorporating geometric restrictions to improve the mirror position prediction accuracy. The use of multiprocessing enables more efficient processing of large-scale mirror arrays. The findings contribute to advancements in controlling natural light in buildings, enhancing visual comfort and energy performance. While we are able to conclude that it is possible to resolve mirror position to within a single step for the stepper-motors driving the mirror's motion, reliable operation is highly sensitive to parameter settings within the compute pipeline.

Not only focusing on a single mirror's position, we also want to know the positions of all the mirrors in the Catoptric Surface. So, future work will need to focus on ensuring this algorithm can work in multi-mirror system. And optimize this algorithm to make the detection be more accurate and faster.

Based on our whole Catoptric Surface systems, we should also pay more attention on the camera's position and whether should we zoom in or zoom out the photos. Also, calculate the real position in geometry is also important for our whole Catoptric Surface systems.

Chapter 4

Conclusions and Future Work

The approach of using edge detection and RANSAC, combined with the restriction of geometric properties, is a promising method for accurately detecting the position of mirrors in the Catoptric Surface research project. By applying edge detection, the algorithm can identify the edges of the ovals formed by the mirrors' reflections, providing a reasonable starting point for the RANSAC algorithm. The RANSAC algorithm can then estimate the ellipse parameters by iteratively selecting a minimal subset of points and fitting an ellipse to those points. The geometric restrictions further improve the accuracy of the ellipse fitting by discarding ellipses that do not meet the desired geometric properties.

When selecting the correct oval using these methods, the algorithm can accurately determine the position of the mirror within the array. Calculating the center points of these ovals allows for a precise representation of the position of the mirrors within the building's interior. By comparing these calculated positions with the actual position of the mirrors, the accuracy of the algorithm can be evaluated and further refined.

Moreover, the use of multiprocessing allows for the processing of multiple images in parallel, increasing the speed and efficiency of the algorithm. This approach provides a promising method for detecting the position of mirrors in the Catoptric Surface research project and has potential applications in other fields that require the localization of objects based on natural light reflections.

The major challenge that is still unresolved in this approach is that the reliable detection of a mirror's position is highly dependent on the parameter settings in the various steps in the process. As a result, many images result in incorrect position information if the parameters are not well tuned.

In summary, the combination of edge detection, RANSAC, and geometric restrictions provides a method for accurately detecting the position of mirrors in the Catoptric Surface research project. This approach, coupled with the calculation of center points and multiprocessing, enables a fast and efficient algorithm for precise mirror localization. With further refinement and testing, this method has potential applications in other fields that require accurate localization of objects based on natural light reflections.

Future work to be accomplished comes in several forms. First, methods need to be formulated for reliably tuning the parameter settings within the algorithms. One approach to consider here is further geometric constraints (e.g., maximum ellipse dimension, upper bound on ellipse area, etc.). Second, this thesis only investigated approaches for a single mirror. In practice, a set of mirrors will be within the field of view of the camera, and the resulting images will need to be segmented and the algorithms applied to multiple sub-images concurrently. Also, for the photos we take in this project, the background of the 2×2 prototype is a white wall. In addition, our mirror's reflected light is always very bright, so that some edges are bright enough that the algorithm cannot detect it clearly. We should also think about how can we make the edges in the photos are easier to detect. Due to our project being based on 2D pictures, we can also think about how can we imply our oval detection in 3D space or make it as a real-time oval detection algorithm.

References

[1] C. Ahrens, R. Chamberlain, S. Mitchell, A. Barnstorff, and J. Gelbard. Controlling daylight reflectance with cyber-physical systems. In *Proc. of 24th International Conference on Computer-Aided Architectural Design Research in Asia (CAADRIA)*, volume 1, pages 433–442, 2019.

[2] C. Ahrens, R. D. Chamberlain, S. Mitchell, and A. Barnstorff. Catoptric surface. In *Proc. of 38th Conference of Association for Computer Aided Design in Architecture (ACADIA)*, pages 216–225, 2018.

[3] M. Alrubaih, M. Zain, M. Alghoul, N. Ibrahim, M. Shameri, and O. Elayeb. Research and development on aspects of daylighting fundamentals. *Renewable and Sustainable Energy Reviews*, 21:494–505, Feb. 2013.

[4] M. Bodart and A. De Herde. Global energy savings in offices buildings by the use of daylighting. *Energy and Buildings*, 34(5):421–429, 2002.

[5] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, 1986.

[6] R. D. Chamberlain, C. Ahrens, C. Gill, and S. A. Mitchell. Work-in-progress: Hierarchical control of a catoptric surface. In *Proc. of International Conference on Embedded Software (EMSOFT)*. IEEE, 2018.

[7] J. Christoffersen and K. Johnsen. An experimental evaluation of daylight systems and lighting control. In *Proc. of Right Light 4, 4th European Conference on Energy-efficient Lighting*, volume 2, pages 245–254, 1997.

[8] L. Edwards and P. Torcellini. Literature review of the effects of natural light on building occupants. Technical Report NREL/TP-550-30769, National Renewable Energy Lab., Golden, CO, USA, July 2002.

[9] M. Figueiro, M. Kalsher, B. Steverson, J. Heerwagen, K. Kampschroer, and M. Rea. Circadian-effective light and its impact on alertness in office workers. *Lighting Research & Technology*, 51(2):171–183, 2019.

[10] M. Figueiro and M. Rea. Office lighting and personal light exposures in two seasons: Impact on sleep and mood. *Lighting Research & Technology*, 48(3):352–364, 2016.

[11] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[12] A. Galatioto and M. Beccali. Aspects and issues of daylighting assessment: A review study. *Renewable and Sustainable Energy Reviews*, 66:852–860, Sept. 2016.

[13] F. Harb, M. P. Hidalgo, and B. Martau. Lack of exposure to natural light in the worskpace is associated with physiological, sleep and depressive symptoms. *Chronobiology International*, 32(3):368–375, 2015.

[14] P. V. Hough. Method and means for recognizing complex patterns, Dec. 18 1962. US Patent 3,069,654.

[15] P. Ihm, A. Nemri, and M. Krarti. Estimation of lighting energy savings from daylighting. *Building and Environment*, 44(3):509–514, 2009.

[16] N. Kanopoulos, N. Vasanthavada, and R. L. Baker. Design of an image edge detection filter using the Sobel operator. *IEEE Journal of Solid-state Circuits*, 23(2):358–367, 1988.

[17] Q. Khuc. User-friendly interface for catoptric project. Master's project, Dept. of Computer Science and Engineering, Washington University in St. Louis, May 2023.

[18] S. Kodali. Measuring the effectiveness of light concentration with the catoptric surface. Master's thesis, Dept. of Computer Science and Engineering, Washington University in St. Louis, Dec. 2022.

[19] J. Lin, G. Wang, and R. W. Lau. Progressive mirror detection. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3697–3705, 2020.

[20] J. Manin. Control of catoptric arrays by reflected surface analysis. Master's project, Dept. of Computer Science and Engineering, Washington University in St. Louis, Aug. 2022.

[21] Python `multiprocessing` — Process-based parallelism. https://docs.python.org/3/library/multiprocessing.html.

[22] P. Nair and A. Saunders Jr. Hough transform based ellipse detection algorithm. *Pattern Recognition Letters*, 17(7):777–784, 1996.

[23] A. Pellegrino, S. Cammarano, V. R. L. Verso, and V. Corrado. Impact of daylighting on total energy use in offices of varying architectural features in Italy: Results from a parametric study. *Building and Environment*, 113:151–162, 2017.

[24] L. G. Roberts. *Machine Perception of Three-dimensional Solids*. PhD thesis, Massachusetts Institute of Technology, 1963.

[25] C. Ronse and P. A. Devijver. *Connected Components in Binary Images: The Detection Problem*. John Wiley & Sons, Inc., 1984.

[26] A. M. Tabar, M. Moradi, and R. Fayaz. Effects of interior architecture for optimal use of natural light and electrical energy saving. *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects*, 44(1):2415–2429, 2022.

[27] E. Tan. Ellipse Detection using RANSAC. https://github.com/emiltan97/ellipse-detection.

[28] L. Torres-Escobedo. C^3: Characterization and Control of a Catoptric Surface. Master's project, Dept. of Computer Science and Engineering, Washington University in St. Louis, Jan. 2021.

[29] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, and the scikit-image contributors. scikit-image: image processing in Python. *PeerJ*, 2:e453, June 2014.

[30] X. Yang, H. Mei, K. Xu, X. Wei, B. Yin, and R. W. Lau. Where is my mirror? In *Proc. of IEEE/CVF International Conference on Computer Vision*, pages 8809–8818, 2019.