Developing optical error software for parabolic trough collectors in concentrated solar power plants

Max Hansen

Office of Science, Science Undergraduate Laboratory Internship Program

Saint John's University, Collegeville, MN

National Renewable Energy Laboratory

Golden, Colorado

August 11, 2023

Prepared in partial fulfillment of the requirement of the Department of Energy, Office of Science's Science Undergraduate Laboratory Internship Program under the direction of Devon Kesseli at the National Renewable Energy Laboratory.

Participant: Max Hansen
Max Hansen

Research Advisor: Devon Kesseli

ABSTRACT

This study focused on the error analysis of parabolic trough collectors (PTC) in concentrated solar power (CSP) plants through advanced image processing techniques. The primary goal was to identify facet corner pixel values on PTC mirrors using computer vision (CV) which can be used as photogrammetry targets to reconstruct troughs in 3D. Additionally, this research sought to connect computer vision with optical accuracy of PTC mirrors. The research started by capturing images of PTC mirrors using drones, which are often affected by noise and unclear imaging. To address this, a denoising pipeline was implemented, involving steps such as grayscale conversion, Gaussian blurring, binarization, and morphological operations. This pipeline effectively reduced noise while preserving essential structures. For point of interest (POI) extraction, two methods are proposed. One method employed known corner locations, dimensions, and camera parameters to predict POIs. Another method used Canny edge detection and the Hough line extraction algorithm to identify intersection points, eliminating the need for manual corner identification. The research also involved region of interest (ROI) extraction, contour and corner detection, facet organization, and corner identification. These steps are designed to accurately locate and represent the corners of PTC facets, considering various orientations and scenarios. Manual validation was employed to verify the accuracy of the CV-based results. The outcomes reveal that the implemented image processing techniques demonstrate promise in PTC mirror error analysis. A comparison of manually selected points and CV selected points had the highest frequency of 4 pixel difference. Challenges like noise and obstructions are addressed. The CV approach is shown to be effective in identifying facet corners. The study suggests future directions, including incorporating multiple perspectives, developing a user-friendly GUI, conducting extensive testing, and implementing confidence interval mechanisms to enhance the algorithm's accuracy and reliability.

I. INTRODUCTION

As the global focus on reducing carbon emissions intensifies, renewable energy sources have emerged as crucial components of a sustainable future. Among these sources, solar energy stands out as a reliable option, providing a promising avenue for clean power generation and industrial process of heat. Traditional photovoltaic (PV) systems have long been the dominant method for capturing solar energy. However, to achieve a more effective and efficient transition towards renewable energy, the exploration of supplemental systems, such as concentrated solar power (CSP), becomes imperative.

CSP systems harness thermal energy by reflecting sunlight onto a receiver system.[1] The receiver system carries this heat through liquid molten salts, thermal oils, or steam. CSP plants utilize the heat for industry purposes, or they transform it into electricity by turning a turbine.[2] CSP offers versatility in capturing and utilizing solar rays effectively. Among these are power tower systems, linear Fresnel reflectors, dish systems, and parabolic trough collectors (PTC), each with its distinct advantages.[2] Unlike PV systems, CSP plants produce energy throughout the night. Thermal energy collected throughout the day maintains at a high temperature, contributing to the grid in high demand hours at night.[3]

PTC mirrors reflect incoming light into a receiver pipe along the focal line of the PTC. If optimal, these troughs can magnify ray intensities from 30 to 100 times their normal magnitude.[4] To optimize energy and thermal output, meticulous attention must be paid to the geometric properties of mirrors, especially in the case of PTC mirrors. These are segmented into square mirror facets.[2] Individual facets are susceptible to deformation and misalignment when exposed to outdoor elements, compromising the overall efficiency of the CSP system.

In pursuit of maximizing CSP efficiency, researchers at the National Renewable Energy Laboratory (NREL) have leveraged cutting-edge computer vision (CV) techniques to gain a comprehensive understanding of mirror properties.[4] Specifically, the Solar Thermal group at NREL focused on parabola-shaped (PTC) mirrors in CSP plants.[43] By employing CV imaging techniques and image clarity algorithms, the group established reliable methods for image processing and error identification. This process is outlined in figure 1. Where the primary goal was to obtain pixel locations of the corners of each facet. These pixel locations allow for ample information to reconstruct the image in post-processing. Image processing yielded 3D information about the troughs. They used software tools to reconstruct the 3D shape and alignment of the troughs. This data provided insights into the alignment of the mirrors, which is a crucial factor in optimizing the performance of CSP plants. This paper presents a different approach to optimizing CSP systems by analyzing drone-captured images using advanced image processing techniques. It delves into the challenges posed by the presence of noise and lack of clarity in the captured images. It discusses the methods employed for corner detection and ideas to isolate the mirror facets effectively.

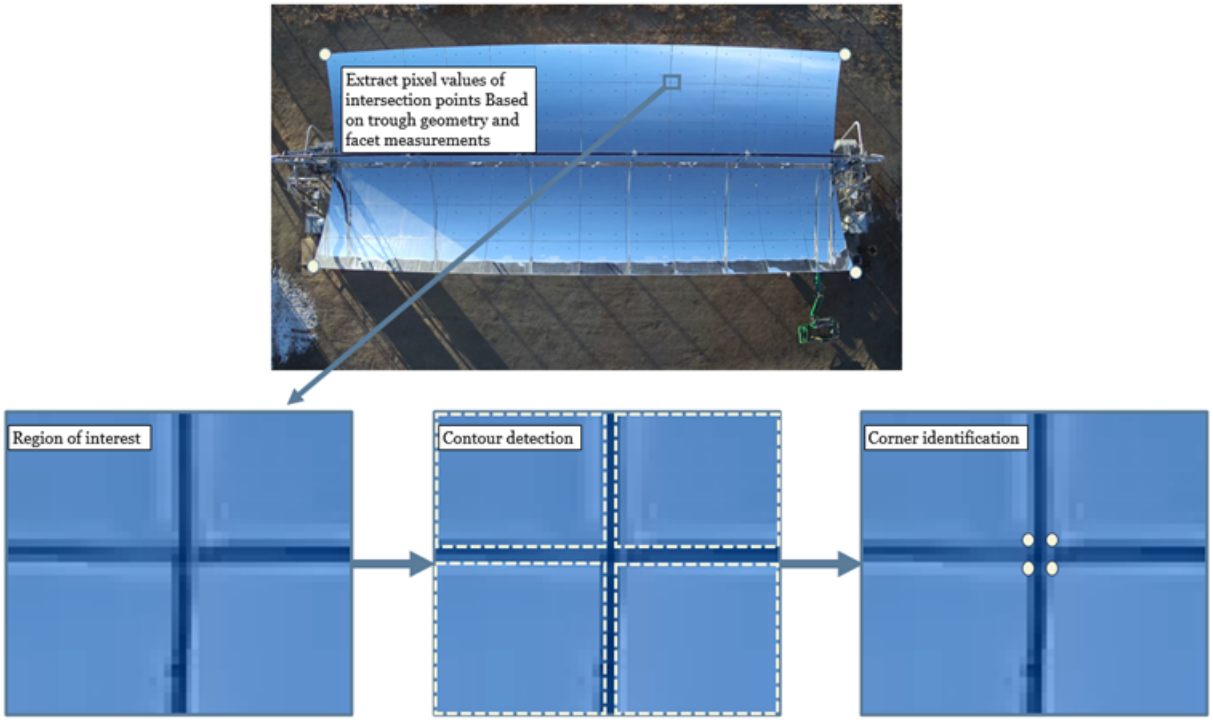


Figure 1. General process for extraction of corner values on a PTC.

II. PROCESS

A. Lens Distortion

In the image acquisition process, lens distortion challenges the validity of pixel locations in a camera reference frame.[5] Lens distortion occurs as a result of light rays entering a non-ideal camera lens, and therefore appearing curved. An ideal camera lens (pin hole) would have all light enter the aperture at one point resulting in accurate pixel representations.[6] However, in practice, pin holes do not allow enough light through. The two major forms of lens distortion are radial and tangential.[6] Figure 2 showcases barrel distortion as a form of radial distortion. The sides of PTC mirrors in drone images are prone to these effects and must be redacted.

In the beginning stages of image processing, the image must be undistorted. The level of image distortion is dependent on properties resulting from physical characteristics of the camera. To acquire camera parameters and the corresponding distortion coefficient, the camera was calibrated. This process was done by translating a point from world coordinates or the global coordinate system (GCS) to the 2D camera plane.[7][18] The relationship between a 3D point and its projection is written as:

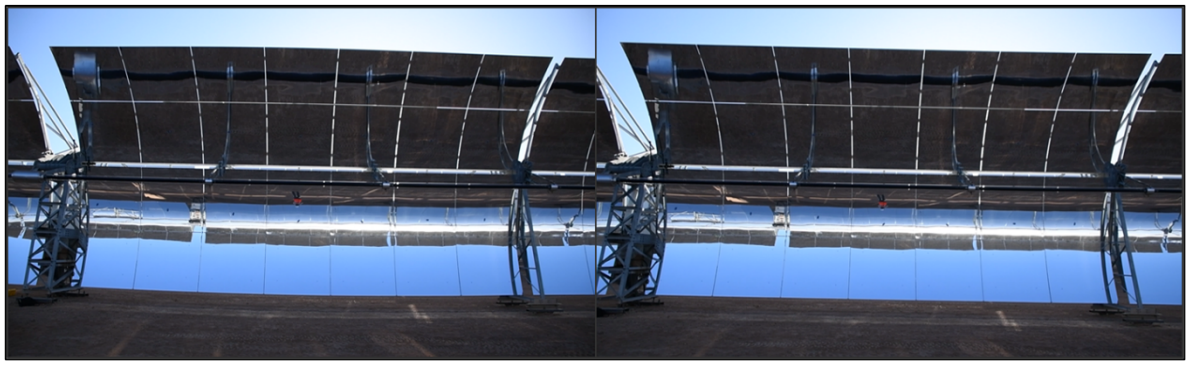


Figure 2. PTC mirror barrel distorted before image processing (left) and undistorted after (right)7

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} p1 & p2 & p3 \\ p4 & p5 & p6 \\ p7 & p8 & p9 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{1}$$

Here, x', y' and z' represent the position on a camera plane where z' is the position along the optical axis. The matrix of "p" values represents the camera matrix. X, Y, and Z represent the position in the GCS. After calibration, this relationship acquired the intrinsic and extrinsic camera parameters with the corresponding distortion coefficients. Intrinsic parameters describe information specific to the camera such as focal length and optical centers.[6] Extrinsic parameters represent translation and rotation vectors.[6] These parameters are encapsulated within a camera matrix. The algorithm fine-tuned this matrix for a particular camera to remove distortion from input images.[78]

B. Image "Denoising"

Drone imaging often captures the mirror along with the surrounding area. The surrounding area coupled with unclear imaging yielded random variation in color intensity within a neighborhood of pixels. As a remedy to this noise, the algorithm employed denoising methods on the images.[9] Images are matrices of pixels each with their own color representation.[92] A red, green, blue (RGB) value accompanies each pixel where each channel has an 8-bit scale (0 – 255).[2] Image processing harnesses these values to dissect and modify images. The algorithm methodology for denoising is found in figure 3.

The program implemented a denoising pipeline that involved several key steps, each contributing to noise reduction and overall image enhancement. Figure 4 showcases this process with a zoomed in picture or region of interest (ROI). This began by converting the color image into a grayscale representation, simplifying the denoising process by eliminating color information that was not essential for noise reduction.[10] This resulted in a single-channel image that focused solely on intensity variations, reducing computational complexity.[10] Next, the algorithm applied Gaussian

blurring, convolving the image with a Gaussian kernel to smooth out noise and irregularities while preserving essential structures and edges. This spatial filtering technique effectively reduced high-frequency noise components, resulting in a cleaner image. After that, it performed binarization to transform the smoothed grayscale image into a binary representation, simplifying the image to only black and white pixels.[10] A clear distinction between foreground and background facilitated subsequent denoising operations. Lastly, it used morphological operations, such as erosion and dilation, to refine the binary image by removing isolated noise pixels and filling gaps in segmented structures, ensuring smooth and coherent regions.[10] By effectively combining these steps, the de-noising pipeline achieved efficient noise reduction and enhanced the quality of the final image, making it more suitable for further analysis and visualization tasks.

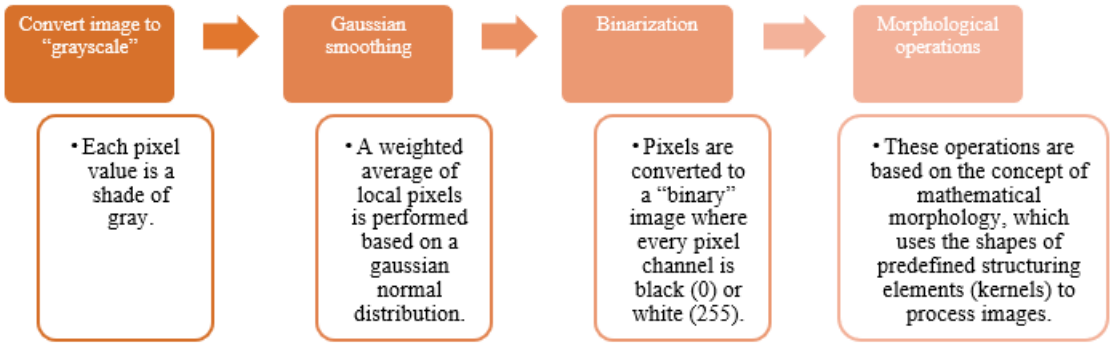


Figure 3. Methodology of denoising an image. Many denoising algorithms exist, but this methodology maintains edges and corners.

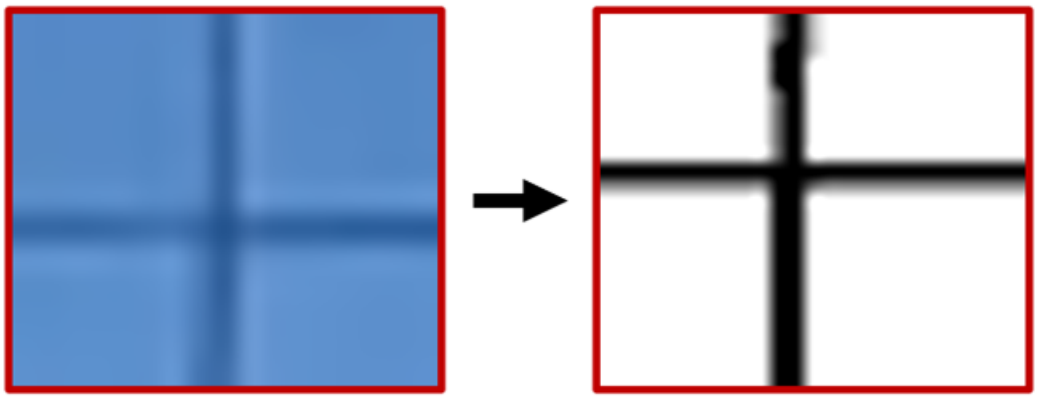


Figure 4. Pre-cleaned image (left) and post-cleaned image (right).

C. Point of Interest Extraction

To accumulate corner values of each facet on a PTC, the algorithm first approximated the facet intersection points. This approximation allowed for a central intersection point to be examined in a zoomed in ROI. The point of interest (POI) prediction script used known trough

corner locations, trough dimensions, and camera parameters, such as focal length and distortion.[11] With this information, the program undistorted the corner locations using the camera parameters, bringing them onto the ideal pinhole camera sensor. Using these undistorted points and the known 3D positions of the trough corners, the program calculated the camera's external orientation parameters (EOP), providing its position and rotation in the trough's reference frame.[11] Furthermore, the program loaded trough schematic values, extracted the 3D locations of ideal intersection points, and projected them onto the camera sensor. Applying lens distortion, the program mapped the points to their positions on the real camera sensor. The camera's principal point was determined to convert the sensor coordinates into pixels.[11] The output of the program (figure 5) provides the estimated pixel locations of the facet/panel intersection points, identifying the point of interest on the trough image.

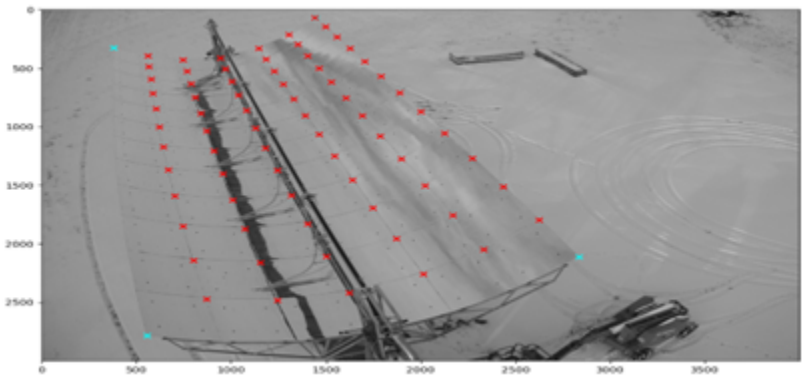


Figure 5. PTC image with POI script output. The "x" marks represent predicted locations of facet intersections based on the known geometry of the trough.[11]

D. Supplemental POI Extraction Method

A method to extract POI positions through computer vision eliminated the need for manual corner identification. This method fails to identify POI positions with an excess of noise in an image. It utilized the Canny edge detection algorithm, a popular method for detecting edges in digital images.[10] It did this by computing the gradients of the smoothed images to identify regions with the most significant changes in intensity, which were potential edges.[10] The gradient of a pixel intensity function $I(x, y)$ is denoted by ∇I and is given by:

$$\nabla I(x, y) = \left[\frac{\partial I}{\partial x}(x, y) + \frac{\partial I}{\partial y}(x, y) \right] \tag{2}$$

The edge magnitude of potential edge pixels becomes zero if the adjacent pixels in the gradient direction have a higher intensity.[10]

After edge detection, the script employed the Hough line extraction algorithm to detect lines in the image. The Hough transform is a robust method for line detection, even in the presence of noise or partial occlusion.[710] It transformed the image space to a parameter space, where each point represented a line (figure 6). By identifying the peaks in the parameter space, it could determine the most prominent lines present in the image.

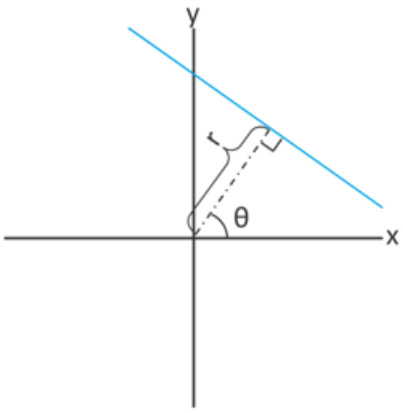


Figure 6. Figure 6. r, θ representation of a Hough line.[7]

Once the lines were extracted, it used intersection algorithms to find the intersection points. The intersection of two lines corresponds to the intersection point in the image space. It leveraged geometric principles, such as solving simultaneous equations for the lines, to find the precise coordinates of the intersection points.[7] These algorithms allowed it to locate the intersection points, providing POI data for subsequent image processing.

E. Region of Interest Extraction

Following point approximation, the program extracted regions of interest on each point of facet intersection. CV processing is more effective when evaluating a smaller ROI. In this smaller region, the facet corner values can be found. Variation of square ROI sizes allowed for forgiving point approximation. However, the larger the ROI, the greater the chance of encapsulating unwanted pixels. ROI extraction is visualized through figure 7.

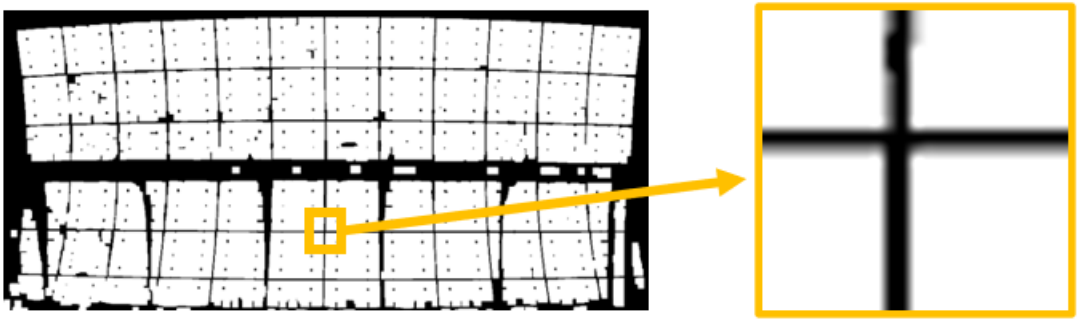


Figure 7. A POI is used as the center of an ROI. This square 50 px by 50 px ROI is to be subsequently processed to identify the corner pixel values in the reference frame of the larger image.

F. Contour and Corner Detection

1. Contour Detection

To extract the corners of each facet in a PTC, the code utilized contour detection. Contours are representative of boundary pixels that have similar color and intensity magnitudes.[12] Within an ROI, four predominant quadrants emerge (figure 11). These quadrants encapsulate a facet corner in each quadrant. In binary representation, it utilized functions that identify contours. This resulted in ideally four contours that contain all edge pixel values within each quadrant.[12] With four contour data structures, it looped through these pixel values finding the pixel furthest from the edges of the ROI. After finding each of the corners, it converted the pixel locations back into the reference frame of the larger image.[12] Figure 8 shows this process through a single calculation. Four contours are detected sectioning off the four quadrants. These contour pixel values are looped through to find the pixel value furthest from the ROI corner the contour has.

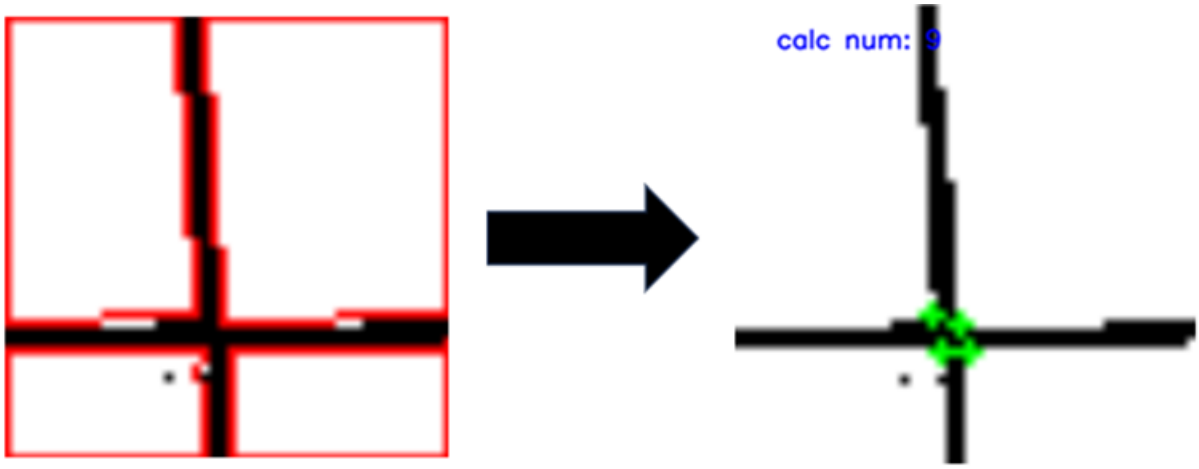


Figure 8. Single frame of corner detection. Contours drawn in red (left), corners drawn in green (right).

2. Edge Cases

POIs along the outside of the trough have only two-pixel locations to be found. This case was handled by detecting contours and finding the two-pixel corners furthest to the side the trough was on. This requires an orientation parameter of the photo taken (figure 9).

G. Facet Organization and Corner Identification

1. Facet and Calculation System

The input trough was treated as a grid. Each facet takes up a position in the grid with a bottom left, bottom right, top right, and top left corner.[13] This system established two 2D arrays that represent the facet value and calculation number. The calculation number was the iteration it was on and the facet value was the facet that a corner calculation belonged to. Representation of this method is in figure 10. The program iterated from the top left of the trough to the right and

Figure 9. Determination of orientation parameter. The orientation parameter is determined by the axis that is closest to parallel with the focal axis. In first orientation, the bottom left corner is determined to be top left of the trough making its facet in the (1,1) position. In orientation two, the top left corner is identified as the top left of the trough making its facet position (1,1).

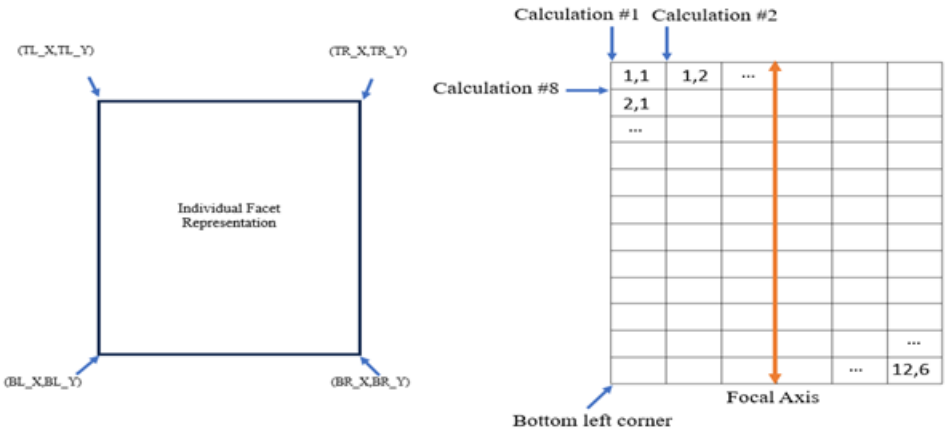


Figure 10. Representation of an individual facet (left) and full trough (right

cycled through positions until the end. Using the two arrays representing the calculation number and facet, it detected which region the corner was a part of.

2. Region Identification and Facet Allocation

After a calculation yielded four-pixel values each representing a corner of a facet, logic based on the orientation parameter allocated a facet to each identified corner pixel.[14] The logic accounted for the center of all four corners and used it as the origin. With the other four pixels, it determined which quadrant it is a part of based on its location.

Another method of region identification utilizes a similar method to the corner values within corner detection. It loops through the values in each contour and finds the value furthest from the intersection point. This value is then used with the same logic as the origin method. This method helps to improve accuracy in image orientations where the trough is not at 90-degree increments.

Figure 11. Region distribution with the center of the intersection acting as the origin.

III. OUTCOMES & DISCUSSION

The core focus of this project was to conduct error analysis on PTC mirrors in CSP plants, with a primary emphasis on the image processing component. Manual pixel values were obtained by carefully clicking through the trough while introducing an uncertainty of ± 5 pixels. These manual points served as a crucial metric to validate the output generated by computer vision techniques.

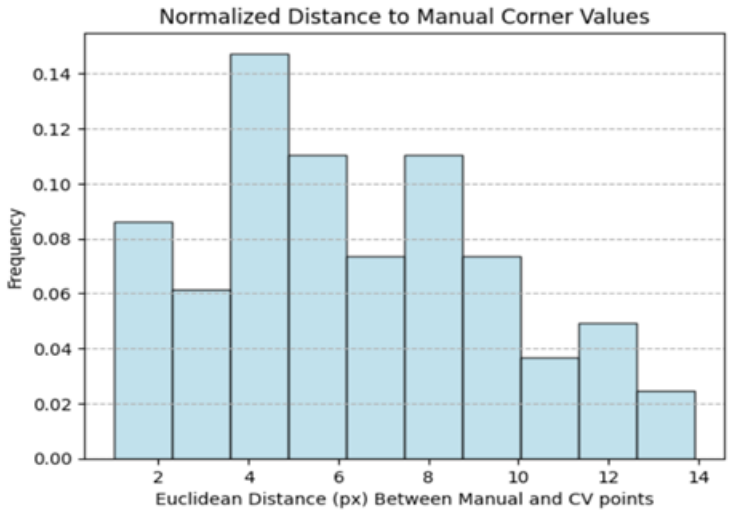


Figure 12. representation of manual points and computer vision points.

Figure 12 displays a normalized pixel difference histogram with frequency peaking at 4 pixels between the manually obtained points and the points selected by the software. This discrepancy can be attributed to image noise and shadows that occasionally appeared throughout the trough, making it challenging for the software to establish distinct edges accurately. Additionally, the focal line of the trough encompassed a receiver tube that obstructed some facet values beneath, further contributing to the deviation. However, the uncertain values, influenced by noise and obstructions, can be effectively filtered out, thus providing a more reliable and clearer portrayal of the points that the software confidently identified. Appendix A showcases valid ROI evaluation and examples of successful corner detection. Appendix B depicts unsuccessful ROI captures where the

ROI is in a shadow, receiver pipe or includes noise. Appendix B.1 showcases improper facet lines in the image. B.2 shows a shadow within the ROI. A.1 and A.2 display successful ROI acquisition with proper corner identification.

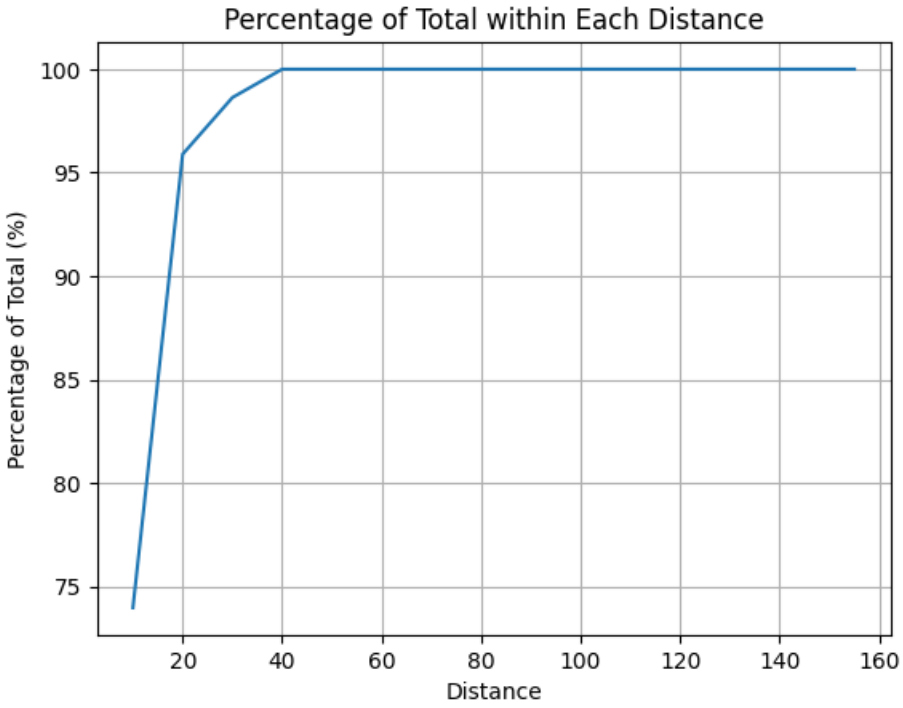


Figure 13. Representation of percentage of points that are within a certain pixel distance from manually selected points. Far distances can be accounted for due to incorrect facet allocation, noise, or shadows within the picture.

An important observation was made that photogrammetry software does not necessitate the acquisition of every pixel corner value; instead, it effectively leverages several pixel values from different perspectives to perform accurate analysis. The employed image processing techniques exhibited promise in the error analysis of PTC mirrors, offering valuable insights for the optimization of CSP systems in the generation of cleaner and more efficient energy. While some challenges persisted, the study demonstrated the potential of computer vision and advanced image processing to contribute to the advancement of sustainable solar energy technologies.

IV. CONCLUSIONS & FUTURE WORK

This project implemented image processing techniques for error analysis on PTC mirrors in CSP plants. The study revealed the feasibility of using computer vision (CV) to identify facet corner pixel values and validate the optics and canting of mirror facets. Manual validation further contributed to the confidence in the software's output, aiding in understanding and interpreting the obtained results.

Moving forward, there are several promising avenues for expanding this research. One significant aspect is to explore multiple perspectives of images to enhance the program's ability to recognize trough corners accurately.[15] By incorporating different viewpoints, the software could

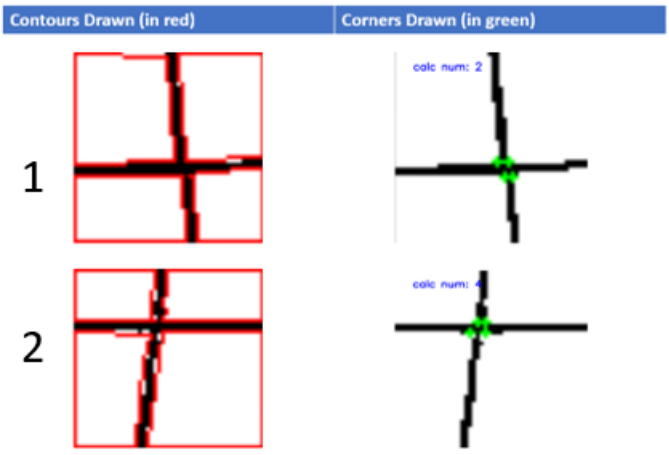
calculate the angle required for image rotation. It could do this while preserving the logical integrity of the data processing. Developing a graphical user interface (GUI) for the image processing software is another crucial aspect of future work. A user-friendly interface would facilitate ease of operation, allowing users to upload images and visualize the results effectively. Additionally, the GUI could include interactive features, enabling users to manually input positions in instances where the algorithm lacks confidence, thus improving overall accuracy. To bolster the algorithm's robustness, further testing with a diverse range of images is crucial.[15] A broader data set would enable a more comprehensive evaluation of the software's performance, ensuring its effectiveness under various conditions and scenarios. Moreover, optimizing the algorithm to incorporate a confidence interval mechanism would enhance reliability. By providing a confidence score for each output, the software could identify instances where uncertainties exist, granting users the option to intervene and validate specific corner positions manually. The combination of multiple perspectives, a user-friendly GUI, extensive testing, and the implementation of a confidence interval would elevate the image processing software to accurate results and further utilization. More research ensures in the advancement of sustainable solar energy technologies and contributes to the ongoing efforts to achieve a cleaner and more sustainable energy future.

V. ACKNOWLEDGEMENTS

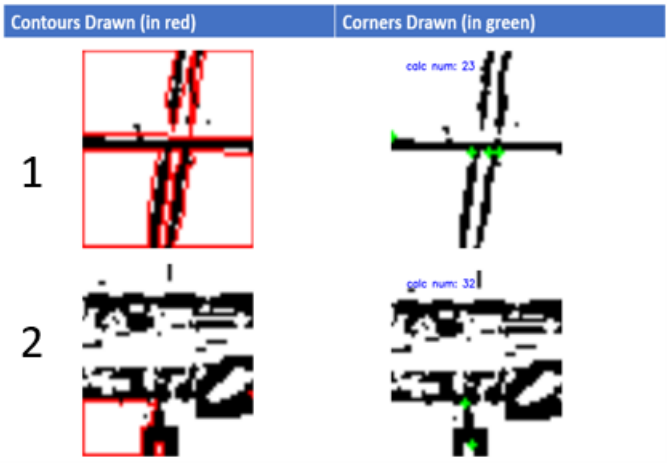
I would like to thank my mentor Devon Kesseli for giving me the opportunity to work at NREL and helping me every step of the way. "This work was supported in part by the U.S. Department of Energy, Office of Science, Office of Workforce Development for Teachers and Scientists (WDTS) under the Science Undergraduate Laboratory Internship (SULI) program".

VI. APPENDICES

Appendix A: Examples of successful ROI extraction



Appendix B: Examples of noisy ROI extraction



VII. REFERENCES

[1] *Concentrating Solar-Thermal Power Basics*, Energy.gov, https://www.energy.gov/eere/solar/concentrating-solar-thermal-power-basics (visited on 08/08/2023).

[2] *Private communication*, in collab. with D. Kesseli, July 25, 2023.

[3] E. T. Tchao, S. A. Asakipaam, Y. A. K. Fiagbe, B. Yeboah-Akowuah, E. Ramde, A. S. Agbemenu, and B. Kommey, "An Implementation of an optimized dual-axis solar tracking algorithm for concentrating solar power plants deployment," Scientific African **16**, e01228 (2022).

[4] *Solar thermal power plants - U.S. Energy Information Administration (EIA)*, https://www.eia.gov/energyexplained/solar/solar-thermal-power-plants.php (visited on 08/08/2023).

[5] *Understanding Lens Distortion | LearnOpenCV #*, https://learnopencv.com/understanding-lens-distortion/ (visited on 08/08/2023).

[6] J. Steward, *Camera Modeling: Exploring Distortion and Distortion Models, Part I*, Tangram Vision, https://www.tangramvision.com/blog/camera-modeling-exploring-distortion-and-distortion-models-part-i (visited on 08/08/2023).

[7] Z. Guangdong, *NREL Performance Progress Report* (National Renewable Energy Laboratory, Dec. 31, 2021).

[8] D. Bhatt, *What is Camera Calibration in Computer Vision?* (June 13, 2023) https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-for-camera-calibration-in-computer-vision/ (visited on 08/08/2023).

[9] D. Kesseli, V. Chidurala, R. Gooch, and G. Zhu, "A Combined Computer Vision and Deep Learning Approach for Rapid Drone-Based Optical Characterization of Parabolic Troughs," Journal of Solar Energy Engineering **145**, 10.1115/1.4055172 (2022).

[10] *OpenCV: Image Denoising*, https://docs.opencv.org/3.4/d5/d69/tutorial_py_non_local_means.html (visited on 08/08/2023).

[11] D. Kesseli, *Area of Interest*, version 1, NREL, Golden, CO, 2023.

[12] *Contour Detection using OpenCV (Python/C++)*, https://learnopencv.com/contour-detection-using-opencv-python-c/#What-are-Contours (visited on 08/08/2023).

[13] *OpenCV: Shi-Tomasi Corner Detector & Good Features to Track*, https://docs.opencv.org/3.4/d4/d8c/tutorial_py_shi_tomasi.html (visited on 08/08/2023).

[14] M. B. Elbeh and A. K. Sleiti, "Analysis and optimization of concentrated solar power plant for application in arid climate," Energy Science & Engineering **9**, 784–797 (2021).

[15] *OpenCV: Canny Edge Detection*, https://docs.opencv.org/3.4/da/d22/tutorial_py_canny.html (visited on 08/08/2023).