MIRA Image Archive Project

Version 1.1

1 March 2024

Author:

Cole Odegard

INTRODUCTION 3
PROJECT GOALS 3
DELIVERABLES 3
PROJECT PURPOSE 3
PROPOSED IMPROVEMENTS 3
RESULTS 4
CHALLENGES 6
FURTHER ENHANCEMENTS 7

Introduction

The current Mira archive needs an overhaul. Originally it was designed to simply store the Mira images and provide a convenient way to display them. These stored images are very useful for troubleshooting image quality issues involving the primary mirror. Because Mira images show the individual segments, properties of the segments can be tracked and monitored for trends.

Project Goals

- Review the database and identify failed datasets.
- Manually reprocess failed datasets and update database.
- Develop a new algorithm to identify segments in Mira images.
- Use the new algorithm to reprocess the entire database.
- Analyze image quality degradation trends of individual segments.
- Update the nightly ingestion of Mira images and database updates.

Deliverables

- Script with a new method of processing MIRA images and identifying failed segments.
- Written documentation outlining MIRA project results including methodology behind the find_center method.

Project Purpose

To enhance the quality of the mirror segment tracking and mirror segment identification.

Proposed Improvements

- Rotation of the image
    - If the image header does not contain rotation information, then manually find the rotation by trying multiple different rotation angles.
    - The rotation angle is “correct” if all 36 segments are identified.
    - All 36 segments are identified based on the distance from each segment to adjacent segments. This distance should vary by less than 5 pixels.
- Scale of the image
    - The image scale (distance from the center to each of the mirror segments) is calculated using the PMFM and given instrument scale.

    - Minor inconsistencies of scale will still give the correct centroids since we are searching for a box around the expected point but need to be corrected in order to obtain an accurate “expected” segment position.
- Finding the Center
    - Finding the center of the image by blurring the entire image, taking the center, shrinking the image and repeating.
    - Eventually the center will be in the middle of the mirror, but if one side of the mirror is brighter than the other (most KPF images are like this) then the center will be slightly off.
    - Adjust the amount of “shrink” in between centroiding.
    - Adjust the center when you find the mirror manually within a range (take the center then move it in a couple different directions to see if you get a better fit).

Results

The find_center method works based on the following methodology:

1. Remove the edges of the image.
    a. Image edges often contain abnormally bright or dim pixels, and never contain the center of the mirror so they can be removed for the purpose of finding the center.
2. Blur the smaller image.
    a. Blurring the image removes any noise in the image and causes the light around the mirror (which is almost always the brightest spot in the image) to be amplified.
    b. Blurring uses a gaussian blur function.
3. Find the brightest spot on the blurred image.
    a. This spot will become the center of the second, smaller image subset.
4. Take a smaller subset of the image around the new center.
    a. This smaller subset will include more of the actual mirror. The size depends on the size of the mirror, each subset is large enough to include the entire mirror.
5. Blur this subset and find the center again.
    a. By blurring the smaller image that contains more of the actual mirror the effects of any background gradient are reduced.
6. Repeat Steps 4 & 5 for all instruments besides those with “big mirrors” (KCWI and KPF).
7. If the image is KPF or KCWI then take a much smaller subset around the center
    a. KPF and KCWI both have large enough mirror centers that they can be found by finding the dimmest spot, rather than the brightest spot.
8. Lightly blur the smaller subset and find the dimmest spot.

a. Finding the dimmest spot avoids the gradient problem. KPF images often have one side of the mirror that is brighter and skews the center when looking for the brightest spot in a blurred image.
b. This is a much lighter blur in case only one side of the mirror is found, to avoid overcorrection.

9. Move a maximum of one segment length in the direction of the new center.
    a. This step is critical. If only one side of the mirror is found, then the blurred image will be darkest directly opposite of the side of the mirror that is found. By moving only one segment length, the algorithm does not move too far against the gradient.

Here is a visualization of steps 7-9 for a KPF image:

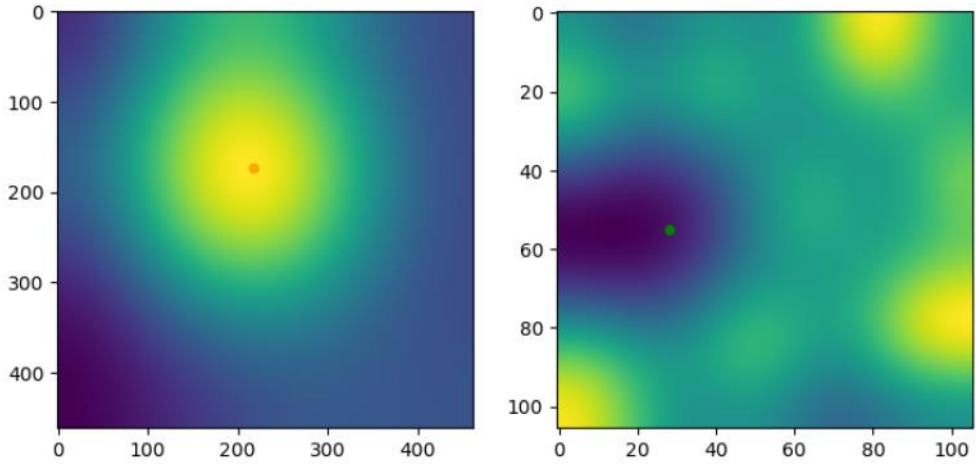

Fig 1.1 & 1.2: The Blur in Steps 5 & 8 of find_center.

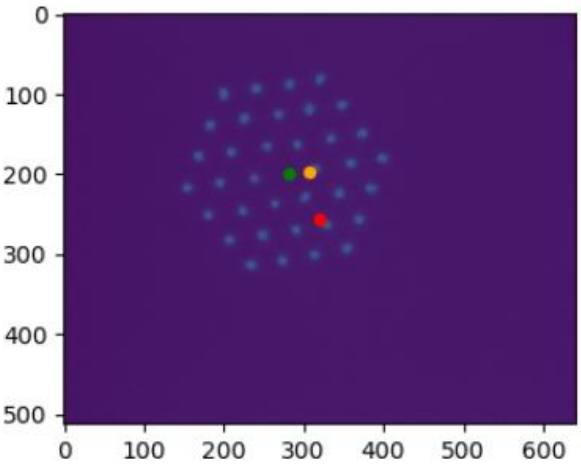

Figure 1.3: The first (red), second (orange) and third (green) center estimates of find_center.

Figure 1.1 shows the gradient problem; in Figure 1.1 a blur is applied to the mirror, finding the orange center. The orange center is located on top of a KPF segment, roughly one segment length away from the actual image center (Figure 1.3). Figure 1.2 depicts steps 7-9, where a light blur is applied to a small subset that contains the actual center, and the dimmest point is located. If step 9 was excluded the dimmest point would be located at roughly (0,50), which would move the center ~20 pixels too far in the opposite direction. Step 9 limits the center movement enough that our final center estimate is located within 10 pixels of the center as shown in Figure 1.3. In this case the gradient is so strong that without proper image cropping the original blurred center of the image would fall almost two segments away from the actual center of the image (red point in Figure 1.3).

Challenges

- Gradients
    - It is difficult to find the center of the mirror when there is a gradient. Even now there is somewhat of a "guessing how far away the center is" process that occurs.
- File rotation angles.
    - Most files have correct rotation angles. Some do not, or their rotation angle is not properly extracted from the file which makes finding each of the 36 segments more difficult.
- Coordinates
    - Image data is plotted with the y-axis flipped. This makes it difficult to keep track of movements in the y direction.
- Visualizing
    - It is impossible to tell if the center has been found without being able to visualize the movements the algorithm is making. It is similarly difficult to visualize the mirror segment acquisition. Visualizing is tricky because of the coordinate problem but also because to blur and find the center you take different sized subsets of the image, each with a different coordinate scale. So, 100,100 on one image might be 250,350 on a different image and each of these coordinates must be accounted for when plotting.
- Class structure
    - While it was simple to break the different pieces of the mirror up, it was difficult to envision what each piece of the mirror would need to be able to do in relation to the other pieces. The mirror is composed of mirror segments, which have centers, brightness, coordinates, etc., but the MIRA image must store the file, image data and attributes about

the image and the mirror to be used by all the pieces of the mirror. For example, to find the segment brightness, which is useful in determining if a segment is a mirror segment or just a random point on the image, the image data is needed. At the same time each file corresponds to a certain image, which it is stored with. So, to access the image data you must first access the file.

Further Enhancements

- The find_center algorithm struggles when the mirror is further from the actual center of the image. This is because it is impossible to take a smaller subset that goes off the mirror, which makes the ability to capture a uniform area around the mirror much more difficult.
- The program also struggles when the mirror segments are not bright, either when the entire mirror is not much brighter or when certain segments of the mirror are not as bright.
- Another issue that the program runs into is when the center of the mirror is found by simply blurring the entire image. This means that as subsets of the image are blurred, they will remove either too much background or the mirror itself, which skews the blurred center away from the actual center.
- Fine tuning the amount of blur for each instrument and based on the results of the first few blurs would improve accuracy. Once it is clear the center of the mirror will be further away from the center of the image it can be assumed that the subsequent blurs will be less accurate once they get too small or too close to the edge of the image. It is also more important to have a strong blur the bigger the mirror is, because larger mirrors have more prominent gradients and do not have a large area of background that will make the brightness of the mirror stand out.

Find_Center vs Current Program

Current MIRA Program

- MIRA files from 1/1/24 through 3/19/24 on K1 and K2: 94/128 = 73% Accuracy

New MIRA Program with updated find_center

- MIRA files from 1/1/24 through 3/19/2024 on K1 and K2: 97/128 = 76% Accuracy
- Correctly found center 108/128 = 84% Accuracy
- Files with correct center but not all 36 segments found: 11/108 = 10%

The new MIRA program makes clear improvements when analyzing KPF and KCWI images which are subject to gradients which often throw off the old program. However, based on over 100 images analyzed,

it does not appear that there is a significant difference between the new MIRA program and the old program for the other instruments. This does not matter much, since both the old and new MIRA programs usually fail on KPF images and less on other instruments like MOSFIRE or DEIMOS. The simplest upgrade of the current system would be adjusting the current program that finds the image center to look for the darkest spot rather than the brightest spot in the center once it has gotten close to where the actual mirror center is. This method will make the current program more accurate on images with a gradient, which is usually the issue with the program being able to find the center.