Using OpenCV for Multiple Object Tracking of Sun Imagery (July 2019)

Michael Tinglof, *Research Undergraduate, Georgia State University*, Dustin Kempton, *Postdoctoral Research Advisor, GSU,* and Rafal Angryk, *Facility Advisor, GSU*

The purpose of this project is to design a process that can identify and track dust specs that were on a camera lens at the time an image was taken. Simple image processing techniques prove invaluable as the dust specs move about from image to image. In addition, the images are needed for tracking and studying sunspots, so any removal of the dust specs must be done with the goal of keeping as much information about the sunspots in the image as possible. The challenges faced during this project included creating a system to isolate all spots of interest from the background of the image, tracking spots of interest from frame to frame, and classifying spots of interest based on probability of either being a sunspot or a dust spec. OpenCV was a critical Python software package used for image processing and dust tracking. Using OpenCV, all spots of interest were separated from the background using an image thresholding method, image noise reduction, and contour finding methods. After the image processing was completed, the spots of interest in the photographs were bounded and these regions were passed to object trackers that categorized and updated the movement of each blob from frame to frame. Having a dictionary of movement allowed for k-means cluster of average movement velocity of each blob.

Dust Detection, Image Processing, K-Means Clustering, Object Tracking, OpenCV, Python, Solar Imaging, Sunspots

I. Introduction

THE ability to track an object through temporal space is a useful tool that allows for complex interactions with one's environment. A baseball player can track a flying ball through the air, maneuvering themselves for a game winning catch. A law enforcement agent can track a swiftly moving suspect through a crowd even if line of sight is obstructed or lost for extended periods of time. The realm of object tracking is not just restricted to humans. Computer vision is a critical part of some machine learning algorithms and allows for increasingly complex interactions between machines and the world. By understanding and utilizing different methods for object tracking, computer vision algorithms can become more accurate and precise.

Classifying one hundred animal images can be rather easy for a human user but poses a rather difficult task for a machine. However, if a machine is trained properly, image classification can be done far more quickly than a human, with an acceptable loss of accuracy. For example, assume a photo was taken of a garden and, for whatever reason, some areas of the photo have noise discoloration. A client may want a human user to blend these small points in with the background so that the image may look more complete. The human editor may find most or all the points, and smooth them over properly, but increasing the number of photographs to hundreds, and then thousands, suddenly becomes a daunting and near impossible task, especially when the objects of interest are moving around from frame to frame. This is where a well train machine can step in a takeover.

The goal of this research project is to create a system that can iterate over a large sample of solar images and remove specs of dust from the images. The removal of these dust specs is critical and must be done accurately as the tracking of sunspots in the images are needed in further research by professors at Georgia State University. The system must be implemented in such a way that the dust is removed without affecting the sunspots and preserving as much of the image information as possible. Since the database is quite large (several thousand images), the system put in place must be able to iterate over the database and correct the dust discoloration in all photographs, given just a single initial photograph. The desire would be to have the system maximize accuracy of detection and precision in tracking dust specs from one image to the next.

The intuition for this project beings with understanding the dataset. **Fig 1.1** displays a sample image. Note how close the color gradient dust particles share with the sun background. Indeed, one can see from **Fig 1.2** how small some of these specs can be, a challenge for human eyes to distinguish. The initial steps of this project will be developing a method to separate the dust and other solar features from the background of the sun's surface. This will be accomplished with a pixel threshold feature and noise reduction methods.

Once these features are distinct from the sun itself, tracking can begin. An initial image will be inputted into the system, where by all features on the sun will be cataloged and track through a set sequence of images. As the images progress, the movement of each solar feature will be tracked. Assuming the dust moves about in a small area on the camera lens, the system should observe a rather low average movement of the dust specs from frame to frame and high average movement of solar surface features. Since the sun is rotating, it is assumed that the sunspots will move across the image while the dust specs should remain relatively in the same place.

Having the movement of each feature cataloged, the average velocity of each feature can be mapped in two-dimensional space. From this mapping a k-means algorithm can be applied to cluster those objects that have the perceived movement of dust and those that are just solar features. Once an accurate catalog of dust features is established, the system will then be able to iterate through the image database and begin tracking the dust through each image, smoothing over the dust pixels using surrounding pixel information.

Fig 1.1

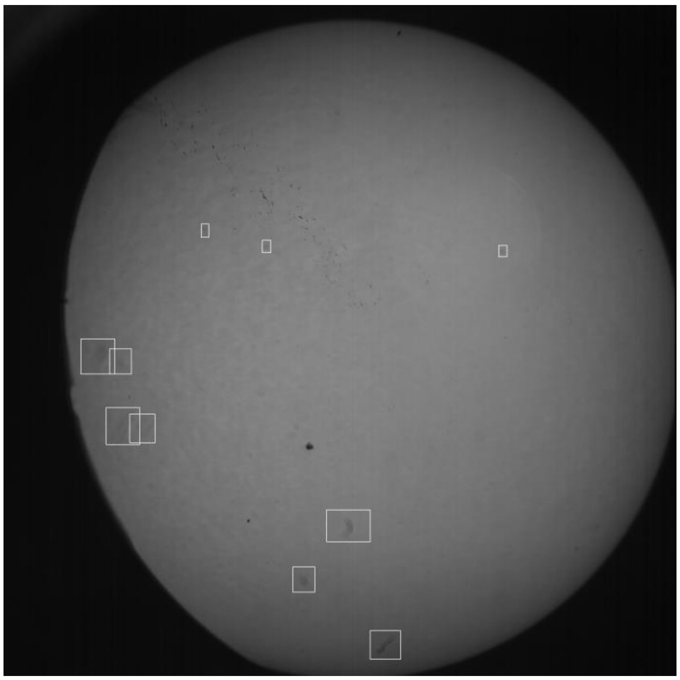

Fig 1.2

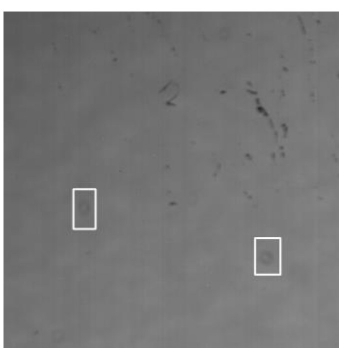

Fig 1.3

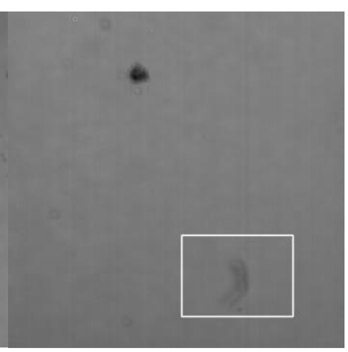

Fig 1.1 shows a sample image from the database of photos. Boxed in white are areas of interest in this project, specifically dust specs in variety of sizes, shapes, and brightness. **Fig 1.2** and **Fig 1.3** show close-up visuals of **Fig 1.1**, specifically dust specs next to solar features.

Once implemented, the system has shown promise in processing images for thresholding solar features and dust specs from the solar surface. However, it has been found that a long sequence of consistent images is needed for an accurate cluster of dust specs to form. In addition, the sequence of images needs to span several hours of solar observation time for accurate clustering to occur. Even once this clustering is created, a small percentage of dust specs have shown a large variety of movement, and a more complex algorithm (nonlinear classification or a machine learning technique) may be needed for greater dust classification accuracy. Further, due to limits on time, the system may need further testing to confirm that results are consistent across multiple sequences of images.

In the end, the hope for this project is that future collaborators may first draw upon the image processing techniques and develop a more complex and accurate classification tool.

II. Related Work

Many methods for dust detection were considered before deciding on a combination image processing, k-means clustering solution. In the paper "Dust particle detection in surveillance video using salient visual descriptors", a "multi-scale approach to learn the discriminative salient features for dust particle detection" was proposed including "salient shape feature from the candidate dust particle regions in addition to the isolation of color contrast and texture saliency features from the scene context." Once these features of the dust were isolated, the research team decided upon using a Support Vector Machine to help in the classification process. While this method would yield higher accuracy in most circumstances, the team was looking at low resolution, security camera footage, which is much smaller in size than the 4K images this project is focusing on understanding.

In the paper "Comprehensive Solutions for Removal of Dust and Scratches from Images", we see the use of credibility measurements in order to detect deficient pixels, "The candidate pixels are those that are much lighter (or darker) than their neighbors and are therefore suspected to be parts of defects." The paper goes on to state that "one simple and efficient option for generating a detail-less image is a median filter… a detail-less image is created using a median filer. The algorithm then uses the gray image and the detail-less image to compute two measure at each pixel, where each measure indicates how likely this pixel is to be defective." This method may prove to be effective in the future, as comparing different pixel values would lead to promising results with object detection on the surface of the sun. However, the method may fail with discoloration on the surface of the sun.

The paper "Digital Single Lens Reflex Camera Identification From Traces of Senor Dust" describes a system that uses a similar technique to the one created for this project, "a binary dust template is generated by thresholding the correlation values such that values smaller than a preset value are set to zero and others to one. In the binary dust map, each binary object, obtained by combining together neighboring binary components, is indexed and a list of dust-spot candidates is formed." This process closely mirrors the system this paper will describing; however an adaptive thresholding method was used instead of a global threshold. It should also be mentioned that the writers of the paper developed a secondary step, "after binary map analysis, all detected dust spots are re-evaluated by analyzing the values in the NCC output. For actual dust spots, NCC values are expected to monotonically decrease around the center of the dust spot." This understanding of dust spots would be useful for later research as sunspots can be observed to be a consistent shade while dust spots decrease or increase in shade as one moves from the center of the spot to the outside.

Adrian Rosebrock's article on detecting bright objects on dark backgrounds lead to another portion of methodology for this paper. However, his article does not go into the practice of detecting objects and object movements. Rosebrock's article should be considered as a starting point for the image analysis in this paper.

To understand multitracking in OpenCV, this project turns to Mallick's article. In his introduction, he explains the different styles and version of a multitracker and how it is simply a container for many single object trackers. His article will provide the basis for working with object tracking in this project.

III. Proposed System

As stated earlier, the main goal of this project will be to identify and track dust specs in a database of solar images, eventually removing them from each image. To begin, the database includes around sixteen thousand images. However, some of these images were corrupted, failed to capture the entire sun in the view of the camera, or were too dark to be used for processing. These image discrepancies will lead to problems in the future of this project since most object trackers available rely on mostly consistent image views. Once the dataset is cleaned, an initial picture will be chosen for processing. The system will threshold the image allowing for darker spots to become white and the light surface of the sun to become black. However, a difficulty in this task would be that the space around the sun is the same color as some of the dust. The thresholding method will need to keep the space around the sun black, while allowing for the dark dust specs to become white. Having the dust specs and other solar features separated from the solar image, each feature will need to be cataloged in a system and given a bounding box for tracking. Then a set sequence of images will be fed to the object tracker, with each frame updating the movement of each object in the system. When the sequence terminates, the average velocity change of each object can be mapped and a k-means clustering algorithm can be applied to differentiate between dust objects and other solar features.

A. Data Collection and Cleaning

Like all good datasets, the data needs to be cleaned and organized. A directory will be created that will host all the information needed for each image. This includes a unique identification number, dates and times of when the picture was taken, and the path to the image array in the server. All image data is hosted on an external server at Georgia State University and can only be accessed through verified credentials. The image directory will be ordered by capture data and time. Once the data is sorted, each image is filtered for being corrupted (no image is present), lacking the entire sun in the frame of the image, or being too dim for processing. In order to quickly differentiate what is in frame for each image, a summation of the image's pixel values will be taken. Each image is 4k in resolution, with each pixel having a 16bit value. Images too dim or not having the sun completely in the view of frame have drastically lower pixel sum values than other complete images. So, images with a sum less than five billion would be removed from the dataset.

B. Initial Dust Identification

The dust identification process begins with a specific sequence of images to be tested on. The first image in the sequence will be used as the initial starting point for dust identification. Using the software package OpenCV, the Adaptive Threshold method will be used to separate image features from the sun's surface. (It should be noted that although the images are 16bit, all images will be converted to 8bit images for processing with a maximum pixel value of 255.) Simple thresholding takes a single global pixel value and compares it to all other pixels in an image. If the compared pixel value is higher than the threshold value, the compared pixel value is set to the image extreme (in this case 255) which would turn it white. If it is below, the compared pixel value is set to the opposite extreme (in this case 0) and turn it black. Since the project requires the sunspots and dust to stand out against the background of the sun, any pixel value below the threshold value will be inversely changed, so as dark pixels become white. However, this leads to problems with this specific dataset since objects of interest are very close in pixel value to the black space around the sun. To circumvent his problem, Adaptive Thresholding will be used. Adaptive Thresholding differs from simple thresholding in that instead of comparing all pixel values to one global value, the image is divided into different areas and each region has an optimal threshold limit set based on the lighting in that area of the photo. The photo can then be divided in such a way that one can threshold the space around the sun separately from the sunspots.

Once the initial image has an inverted color threshold applied, noise on the image will be reduced using an erosion method. From the OpenCV documentation, "This operation consists of convoluting an image *A* with some kernel (*B),* which can have any shape or size, usually a square or circle. The kernel *B* has a defined anchor point, usually being the center of the kernel… What this does is to compute a local minimum over the area of the kernel. As the kernel *B* is scanned over the image, we compute the minimal pixel value overlapped by *B* and replaced the image pixel under the anchor point with that minimal value." Having a satisfactory level of noise reduction reached, a dilation method will be applied. This is the opposite of the erode method and will expand the points of interest to allow for sizing of bounding boxes.

Finally, our initial image will consist of only bright spots indicating areas we would like to track. Since the points of interest will be bright against a dark background, the OpenCV method Find Contours can be applied to bound each bright spot. This bounding can be passed to another OpenCV helper method that will return a bounding box for each contour. Armed with these bounding boxes, a directory will be created that will hold each object's specific information. This includes a unique identification, initial bounding box coordinates, the centroid of the current bounding box, previous bounding box coordinates, and movement of each centroid per frame along the x and y axis of the image.

C. Blob Tracking

With the initial photo processed, and all objects of interest categorized, a set sequence of proceeding images can be fed to the system. At the initial start of the sequence, a multitracker object is created. A multitracker object is a class from OpenCV that takes in individual tracker classes. Each object in the dictionary is assigned one tracker, which is responsible for tracking that object from frame to frame. Each image in the sequence is also processed using the OpenCV CLAHE method. The CLAHE is a histogram equalization method that makes it easier for the tracker to follow objects of interest from one image to the next. A simple histogram equalization method involves transforming the global pixel values of an image to create a more normal distribution of values. However, this can lead to information lose in bright areas of the image. In addition, we would still want to keep the black area around the sun black to aid the object trackers in following objects from frame to frame. In order to solve these problems, we use the CLAHE method which is much like the Adaptive Thresholding method in that CLAHE will divide the target image into a set number of regions, and then equalize each region based on the local pixel values histograms.

Fig 3.1

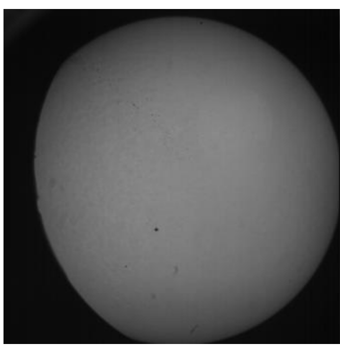

Fig 3.2

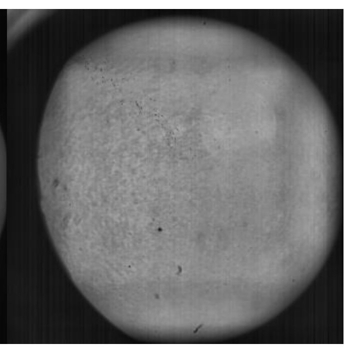

Fig 3.3

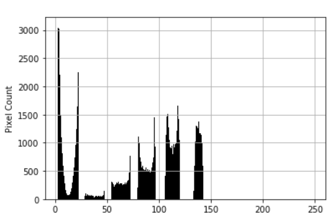

Fig 3.4

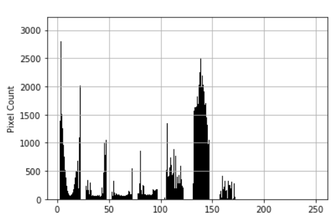

Fig 3.1 is a sample image from the database where as **Fig 3.2** is the same image after a CLAHE method has been used on the image. **Fig 3.3** and **Fig 3.4** are the image pixel value histograms for **Fig 3.1** and **Fig 3.2** respectfully. Notice an increase in higher pixel vales from **Fig 3.3** to **Fig 3.4**. Viewing **Fig 3.1** and **Fig 3.2**, one can see an increase in visibility of dust spots and sun spots. However, the sun's surface texture increases greatly, and may pose to be a problem for our object tracker.

Once the image has been normalized, each singleton tracker will update the dictionary with the newest position of their specific object. The dictionary will also be updated with the centroid location as well as the movement along the x and y axis of the object. This process will continue until the end of the sequence is reach.

Each frame called will return the object directory, bounding boxes, and processed image. These items can be used for further analysis outside the proposed method.

D. Object Clustering

Once the sequence of images terminates, the final object dictionary will be returned. The dictionary will be iterated over, and the average velocity along the x and y axis will be calculated for each object. These averages will serve as the coordinates in a two-dimensional scatter plot mapping for each object. Once the objects are mapped, a k-means clustering algorithm will be implemented to try and classify which average velocities belong to dust objects and which belong to solar phenomenon.

IV. Implementation and Evaluation

A. Implementation

The project began with implementing a data cleaning process. As mentioned, a simple benchmark of a summation of an image's pixel values were used, with each image sum needing to be greater than five billion. From the 18,673 images, 2,194 images failed to meet this mark, leaving 16,479 images.

Using the software package created for this project, an initial image was picked as the starting point, and objects of interest were extrapolated (**Fig 4.1**). The Adaptive Thresholding method was used with a block size of 201 with a factor of 3 subtracted from each calculation (**Fig 4.2**). The image was eroded twice, before each object was dilated twelve times (**Fig 4.3**). Once completed, objects of interest were identified (**Fig 4.4**) and an object dictionary was initialized.

Having the initial starting objects cataloged, tests could begin. **Table 4.1** shows the number of frames, time and date length of each trial sequence. Once the sequence of frames terminated, the average velocity movement of each object in the object dictionary is calculated. This dictionary is then passed to the software package and a k-means algorithm is implemented to try and classify which group of objects belong where. Currently the system requires a user to set how many different clusters they would like to see, then visually determine what objects are in the cluster. If the cluster is deemed to contain all dust particles, the object identification numbers can be recorded and stored for later analysis. If it is determined that the cluster contains only non-dust objects, the cluster can be dropped from future iterations. If, however, the cluster contains both dust and non-dust objects, the cluster is kept in the processes and a new k-means iteration occurs. This process assumes that the user would continually iterate over the dictionary, dropping and recording the needed objects, until all objects were properly classified.

Fig 4.1

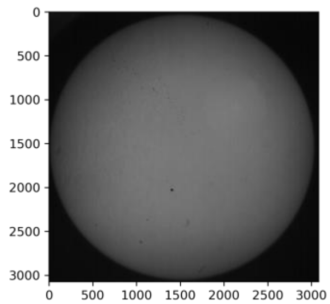

Fig 4.2

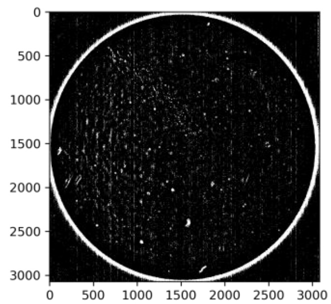

Fig 4.3

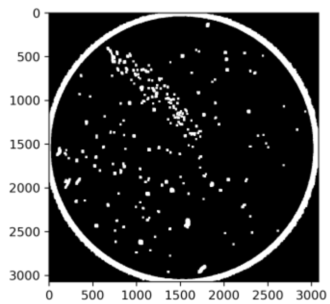

Fig 4.4

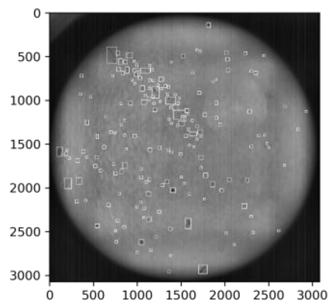

Fig 4.1 displays a sample image (index 33) with no processing done to it. **Fig 4.2** displays adaptive thresholding being applied to the image. Take note of the vertical lines of noise in the image. **Fig 4.3** displays the image after the erosion and dilation processes. The erosion has removed most of the noise and the dilation has increased the size of the points of interest. **Fig 4.4** is a continuation of the image process, with bounding boxes around all points that have been revealed from the image processing. **Fig 4.4** is the initial image after having a CLAHE method applied.

B. Evaluation

Early investigation reveals that the system holds some promise for classifying and tracking the dust across images. **Fig 4.5** and **Fig 4.6** shows the clustering of confirmed dust particles of images for Trial 2 and Trial 3 respectively. Trial 1 was deemed inconclusive as dust and non-dust objects shared too similar average movement velocity and could therefore not be separated from each other using clustering.

Fig 4.5 and **Fig 4.6** show that 52.1% and 41.5% of classified dust had no average movement in either the x or y direction. If further testing reveals that these statistics hold for most sequences of images, this would be a critical step in removing a large majority of the dust from images. For Trial 2, it can be shown that 28.8% of additional dust can be found within one velocity difference of the (0, 0) x-y mark. In Trial 3, an additional 47.3% of classified dust can be found within one velocity difference of the (0, 0) x-y mark. Together, 80.9% and 88.8% of all confirmed dust objects for Trial 2 and Trial 3 can be found within one velocity mark of the (0, 0) x-y mark.

Fig 4.5

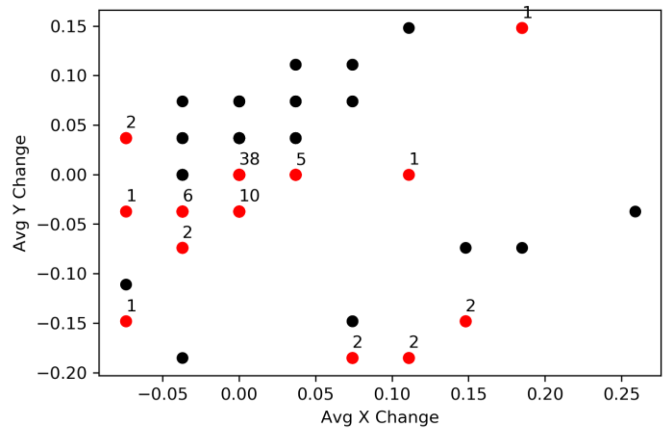

Fig 4.5 displays the clustering results for Trial 2. The x and y values refer to the average change in pixel values between each frame of the object. The red points indicate confirmed dust-object locations and the black points indicate confirmed non-dust objects. Above the confirmed dust locations are counts for the number of dust object with those velocity values.

There is a decrease in average velocity among classified dust as well as a movement away from the (0, 0) x-y mark by non-dust points of interest when comparing **Fig 4.5** and **Fig 4.6**. There also seems to be a relationship to quality of clustering and quantity of frames used for initial analysis. Where Trial 3 used almost twice as many frames as Trial 2, Trial 3 had much more distinct clustering formations. If this trend holds true, this would explain the inconclusive results for Trial 1, which used half as many frames as Trial 2.

Fig 4.6

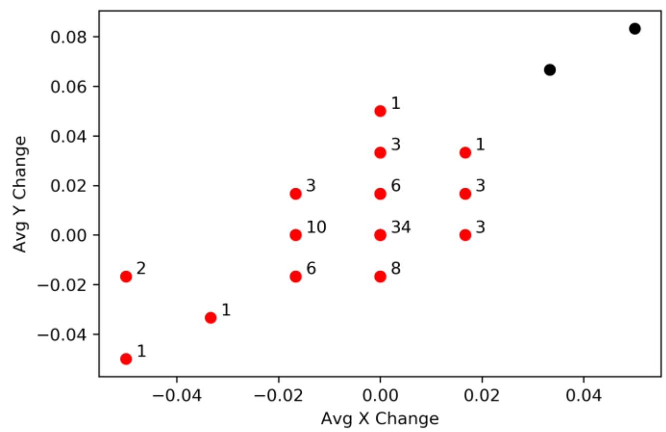

Fig 4.6 displays the clustering results for Trial 3. The x and y values refer to the average change in pixel values between each frame of the object. The red points indicate confirmed dust-object locations and the black points indicate confirmed non-dust objects. Above the confirmed dust locations are counts for the number of dust object with those velocity values.

Table 4.1

Trial	Date Range	Time Length	Frames
1	Start: Jan 5, 2017 __9:09:00 End: Jan 6, 2017 _ 23:06:23	1 day, 8 hours, 15 minutes, 23 seconds	17
2	Start: Jan 7, 2017 __7:28:13 End: Jan 7, 2017 9:31:00	2 hours, 2 minutes, 47 seconds	27
3	Start: Jan 20, 2017 _ 22:29:38 End: Jan 21, 2017 __3:04:45	4 hours, 34 minutes, 7 seconds	60

Table 4.1 shows the three different test trials conducted during this research. The data also displays date range, time length, and number of frames used for each trial.

V. Discussion and Conclusion

A. Time Limitations

A large component of this project was understanding and fine tuning the image processing methods. With a robust system now in place, further experimenting can, and should, continue. Given more time, it is this researcher's opinion that a better clustering and more trial samples could have been developed as only three test trials were conducted. With more trials, more conclusive results can be derived from this method.

B. Project Setbacks

From the preliminary test trials, it was deduced that higher accuracy level clusters may come from longer sequences of images. This process may be impeded by a few factors, and researchers in the future should be made aware of them.

1) Image Brightness

A common issue encountered would be a sudden dimming of the solar images (**Fig 5.2**). This would lead to individual trackers losing "sight" of their assigned blobs and begin tracking other blobs around the sun. Researchers should either removed these dimmed images from the dataset or develop tools to detect and brighten these images.

2) Image Shift

This problem was observed to be common among the dataset. Sun shifting is defined as when the sun stays completely within the frame but suddenly moves to a different position within the frame. In other words, the sun would shift several hundred pixels in either direction on the x or y axis. This sudden movement would completely throw off individual trackers and cause inaccuracies in the object tracking. Likewise, a monitoring tool should be developed to alert when this shift takes place.

3) Solar Eclipsing

This does not refer to the actual solar phenomenon, but a data discrepancy observed during the time of research. In certain frames of the database, the sun will begin to eclipse and grow smaller (**Fig 5.1**). Again, a tool should be developed to determine when this happens so these images can be removed from the dataset, as this eclipsing will hamper the use of the multitracker.

4) Timing

As mentioned in the *Evaluation* of this paper, quantity of frames is preferred for creating more tight forming clusters of objects. However, it has been observed that several hundred frames of solar imaging can exist in sequence, with the solar observation time being less than half an hour. In this case, even though hundred frames are analyzed, there would exist inconclusive results as there would not be a significant shift of movement to be studied. It is recommended that future researchers should consider not only the amount and quality of frames being analyzed, but the time span these frames cover as well.

Fig 5.1

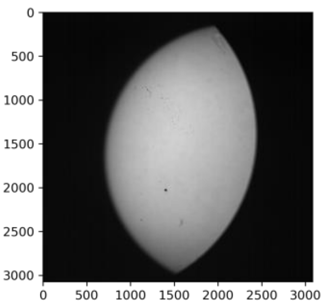

Fig 5.2

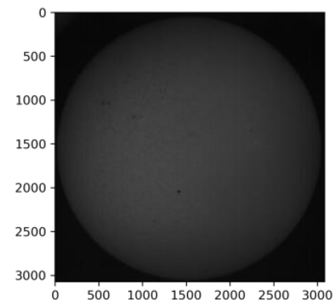

Fig 5.1 shows an image with solar eclipsing occurring. The sun in the frame will progressively get more and more small until it is lost from view. It will then begin to eclipse back to normal size. **Fig 5.2** displays an image of the sun but with drastically reduced visual levels. A more in-depth study should be done of these images to determine if another minimum pixel value should be set for removing these images, or if a different course of action should be taken.

C. Future Improvements

Future researchers should take the work done here and extrapolate more effective tools when it comes to both processing image sequences and classifying object clusters. More advanced tools are needed to make sure image sequences are consistent to allow for object trackers to properly work and adapt to different image qualities. In addition, understanding object clustering and developing a more complex classification tool will aid in future advances. It should also be noted that the final step, dust removal, still needs to be developed. The details of that tool should be discussed with those who intend to use the solar images once they have been processed.

D. Conclusion

This project has laid the foundation for future improvements and developments require to finish this experiment. This researcher has shown that dust specs can successfully and efficiently be separate from the solar background, allowing for either the techniques listed in this paper or other techniques deemed more accurate by future researchers, to be used for the image cleaning processes.

References

[1] Bergman, Ruth & Maurer, Ron & Nachlieli, Hila & Ruckenstein, Gitit & Chase, Patrick & Greig, Darryl. "Comprehensive Solutions for Removal of Dust and Scratches from Images." *Journal of Electronic Imaging*, vol. 17, no. 013010, 2008.

[2] Dirik, Ahmet Emir & Sencar, Husrev Taha & Memon, Nasir. "Digital Single Lens Reflex Camera Identification From Traces of Senor Dust." *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 3, 2018, pp. 539-552.

[3] doxygen. "Image Thresholding." *OpenCV*, 24 Oct. 2017, https://docs.opencv.org/3.3.1/d7/dd0/tutorial_js_thresholding.html

[4] doxygen, "Histograms – 2: histogram Equalization." *OpenCV*, 18 Dec. 2015, https://docs.opencv.org/3.1.0/d5/daf/tutorial_py_histogram_equalization.html

[5] Mallick, Satya. "MultiTracker : Multiple Object Tracking using OpenCV (C++/Python)." *Learn OpenCV*, 5 Aug. 2018,

https://www.learnopencv.com/multitracker-multiple-object-tracking-using-opencv-c-python/

[6] opencv dev team. "Eroding and Dilating." *OpenCV,* 1 Aug. 2019, https://docs.opencv.org/2.4/doc/tutorials/imgproc/erosion_dilatation/erosion_dilatation.html

[7] Rosebrock, Adrian. "Detecting multiple bright spots in an image with Python and OpenCV." *pyimagesearch,* 31 Oct. 2016, https://www.pyimagesearch.com/2016/10/31/detecting-multiple-bright-spots-in-an-image-with-python-and-opencv/

[8] Yang, Ping & Chen, Li & Tian, Jing. "Dust particle detection in surveillance video using salient visual descriptors." *Computers & Electrical Engineering*, vol. 62, no. 1, 2017, pp. 224-231.