

---

# Amaterasu

---

Mohamad El Moussawi

[mhd2002mswi@gmail.com](mailto:mhd2002mswi@gmail.com)



# Amaterasu

Amaterasu is a project designed to extract celestial data from 3D fly through space videos frames. Drawn from Japanese mythology, Amaterasu is the goddess of the sun, symbolizing light, warmth, and radiance.



”SHEDDING LIGHT ON THE MYSTERIES OF THE UNIVERSE”



## Origin

It was created as the core of project (stellar tone) made for nasa space app challenge 2023 (Immersed in the Sounds of Space).

Where the challenge was to design a method to create sonifications of 3D NASA space datasets. The project (stellar tone) lead the team to the top 20 teams in the local judging in Lebanon.

Amaterasu is a python code that contain some data extraction function using image segmentation, build up using OpenCv library.

## Important concpets

- A mask: a mask is essentially a binary image where each pixel can be either 0 (black) or 1 (white). The pixel is set to 1 in case of satisfying certain constraint and 0 when not.
- Contouring refers to the process of detecting and representing the boundaries of objects in an image. Contours are essentially the outlines or curves that join continuous points along the boundary of an object with the same color or intensity.
- Finding the center and size of the contoured area:  
The contour is a sum of points, doing some math we can get the average position of those points. Or by drawing the smallest box (rectangle) that contains the contoured object and finding its one center.
- Opening Operation:  
Used to remove small objects and smooth boundaries while preserving the larger structures in an image.  
It's a two-step process consisting of an erosion followed by dilation.
- Closing Operation:  
Used to close small holes, gaps, or breaks within objects and smoothen boundaries.  
It's a dilation followed by an erosion.

The following is some of the technical detail of the project.

## Data extraction points

- Color Distribution.
- Detecting bright spots position and size.
- Detecting Dust position.
- Detecting Stars position and size.
- Detecting gazes or galaxies.



## Colors Distribution

Extracting the color distribution in the frame and finding the contribution of each color in it.

The steps:

- converting the frame from BGR to HSV.
- Projecting the frame into its 3 colors, (creating Red, Green and Blue masks).
- In each mask we contour the spots, finding their area and center.
- Calculating the average color distribution position in the mask taking into account the area of each spot as a weight.

## Bright spots position and size

Extracting the position and size of the bright spots in the frame.

The steps:

- Converting the frame from BGR to HLS
- Creating mask according to the lightness factor (we are setting a lightness threshold on the frame).
- Contouring the bright spots on the mask and getting their center and area.

## Detecting Stars position and size

Extracting the position and size of the stars in the frame.

The logic is to find a polygone with 8 edges and a bright spot in it.

The steps:

- Converting the frame from BGR to HLS
- Creating mask according to the lightness factor (we are setting a lightness threshold on the frame).
- Contouring the bright spots on the mask and getting their center and area.
- Putting the bright spots centers in a list.
- Applying Canny edge detector method to find the edges in the frame after somme Gaussian blur.
- Doing closing operation (by that we connect neighborhood edges together).
- Having the results of canny edge and closing methods we got a mask.
- Contouring the areas in the mask
- Applying polygon approximation to the contour points
- now that we got for each area a polygon we can count it's edges.
- We select the areas where the polygon have 8 edges (star shape).
- Checking if those areas have a bright spot from our list.



IMPORTANT NOTE: A polygon with 8 edges can be some kind of an octagon rather than a star. The key is the area of that polygon, we can identify a star shape from an octagon shape by comparing it to the smallest rectangle that can fit the area in it.

## Detecting Dust position

The elimination of the dust is way easier than detecting it, so the idea is to get a dust free frame then subtract it from the original frame.

### Dust Elimination

A dust particle in the frame it's a lonely small bright spot.

The idea to eliminate them is : Gathering the neighborhood objects together in the frame and eliminating the isolated ones (dust particles).

The steps:

- Creating structure elements (think of it as we are defining new elemental object that can be added to form the frame)
- The structure is a somme of pixels.
- Doing Close morphography
- Then an Opening morphography
- We got a dust free frame

### Dust Detection

- We turn the dust free frame and the original one from BGR to GRAY
- Turning the gray frames into binary by applying a lightness threshold
- applying bitwise operations to subtract
- the dust only frame now is some kind of mask, we extract the position of the on pixels in the frame

## Galaxies or gazes detection

The idea is to detect all none (dust particles, stars, bright spots ) objects.

The steps:

- Turning the original frame into gray and applying a lightness threshold so that we got a binary frame.
- getting dust,stars and bright spots binary frames (frame that have an on pixels only where at the object positions)
- By somme bitwise operations we subtract the frames that contain the stars,bright spots and dust particles.
- Contouring the rest objects (galxies or gazes) and finding their own center and size.

Till now (7 december 2023), the code need more optimizations.