# CONSTELLATION DETECTION

*Kartikeya Gupta, Anikait Sahota*


## ABSTRACT

Observing the night sky, pointing out star patterns, looking for planet and celestial bodies, and finding constellations has always attracted astronomers, star gazers and photographers alike. Anyone who is getting into observing the space knows about constellations, and would want to spot them. But identifying constellations in the night sky requires sufficient knowledge about different stars, patterns and where to look in the sky. We developed an algorithm to help detect these constellations. By comparing a test image to our available database of constellation templates, we provide an overlapping plot of the detected constellation and the image on a final plot. Creating templates, processing test images, and comparing these two to identify the constellations are the major steps we perform in this project. We were successfully able to identify the constellation present in a test image.


***Keywords—*** image processing, constellation, template creation, normalization, image matching.

## 1. INTRODUCTION

A constellation is a recognized pattern of stars in the night sky. There are a total of 88 known constellations. The constellations are usually formed due to a group of prominent stars in a simple pattern, which are the first stars that people recognize when looking at it. Constellations were first named by the Greeks and Romans. During the ancient times, certain constellations had acquired special significance over time because of their appearance marking the start of a new season, guiding travelers and people, letting farmers know when to sow or reap a crop, and hunters to tell the best time to hunt. They also played a part in being the first GPS, and still play an important role in satellite placement and positioning [1,2]. A list of constellations has now been published by International Astronomical Union (IAU). While identifying constellations is not very difficult for people who are experience in the activity of stargazing, it can be difficult for newcomers to explore the numerous patterns and the corresponding stars involved in the making of those patters.

We try to apply our knowledge of various image processing techniques and algorithms to help in the detection of these constellations in the night sky. Our main focus is to build a detection tool for naked eye constellations. The main detection tool are the templates we use, against which a test is compared and try to figure out the constellation in that test image. Along with processing the test images, the templates had to be processed as well to make them usable.

There were three important steps in our constellation detection algorithm. The first was the creation of the template database. The original templates were obtained from an image gallery of constellations [3]. We select the 30 largest and most prominent constellations present in the night sky using a list available online [4]. Various image processing techniques are the applied to create the template database which is used in detection algorithm. The template creation is implemented based on an existing set of modified constellation charts [5]. The second was the processing of test images. The test images were obtained from a night sky observation application [6] and then processed to make detection of constellations feasible. The third was creation of a detection algorithm, that would let us detect a constellation irrespective of the way it was present in the image. The performed procedures and techniques are discussed in detail in the following sections.

## 2. METHODOLOGY

As discussed, there were three main procedures that were followed. The creation of template database, the processing of test image, and the detection algorithm.

### 2.1. Template database implementation

The templates originally obtained were very difficult to use to match with test images. An example of the original template [6] for constellation Gemini is shown in Fig. 1. The key feature we need to obtain from the template is the location of the stars (marked in red) and the lines connecting and forming the constellation (marked in blue). These templates play an important role in making the detection better and more accurate, and thus, most information has been extracted from the templates.

We first separated the RGB channels of the original template. As we can see in Fig. 2, on separating the channels, the lines dominantly appear in the blue channel and the stars appear dominantly in the red channel. Continuing with this observation, we used the red channel to identify the stars in the constellation and blue channel to identify the lines connecting them.

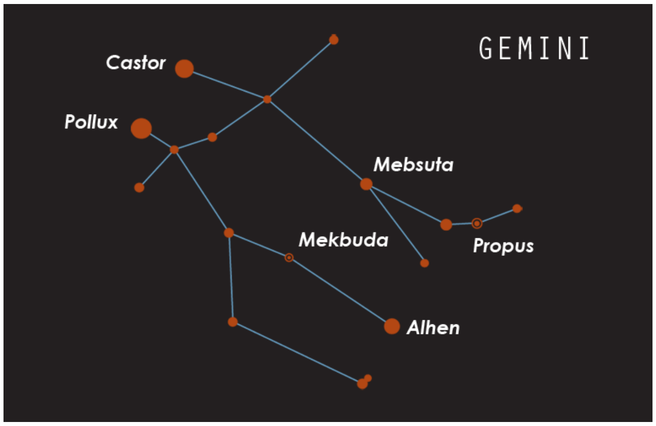



*Fig. 1. The original template of constellation Gemini*

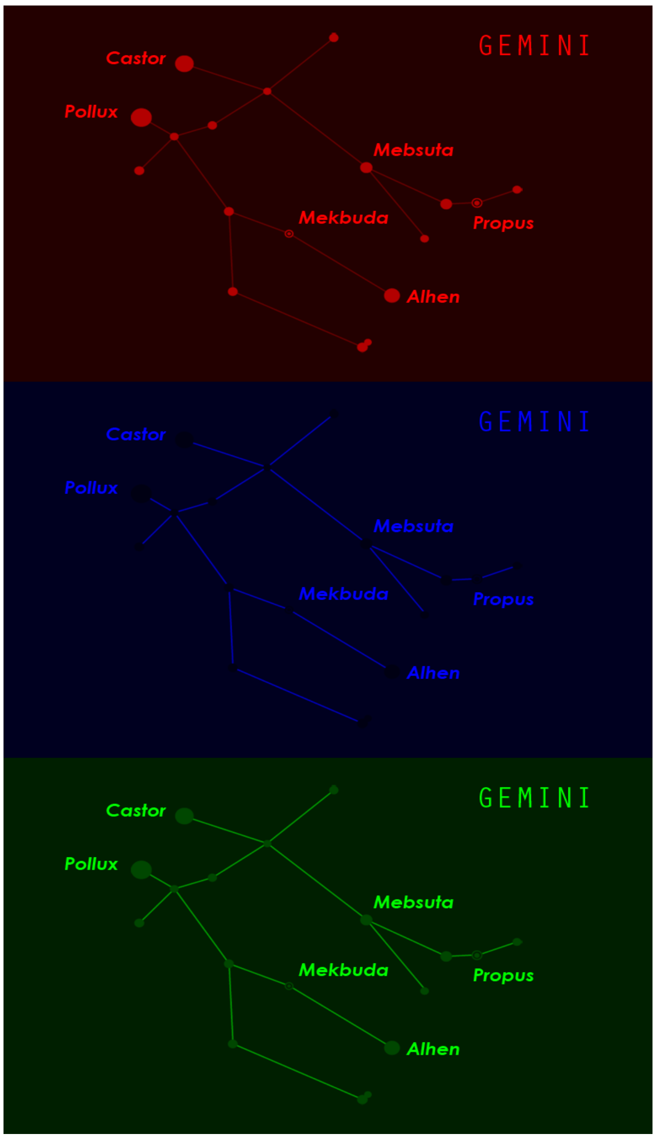



*Fig. 1. The separated RGB channels of template image*

After obtaining these channels, we get rid of the unwanted features from the image like text and leftover lines and stars. We use thresholding to get the brightest component of red and blue channel, which end up being the stars and lines respectively. Even after thresholding, some text components were left, so we applied a median filter of size 3 to get rid of the leftover text. We then binarized the channels separately to obtain a final stars and lines image as see as seen in Fig. 3.

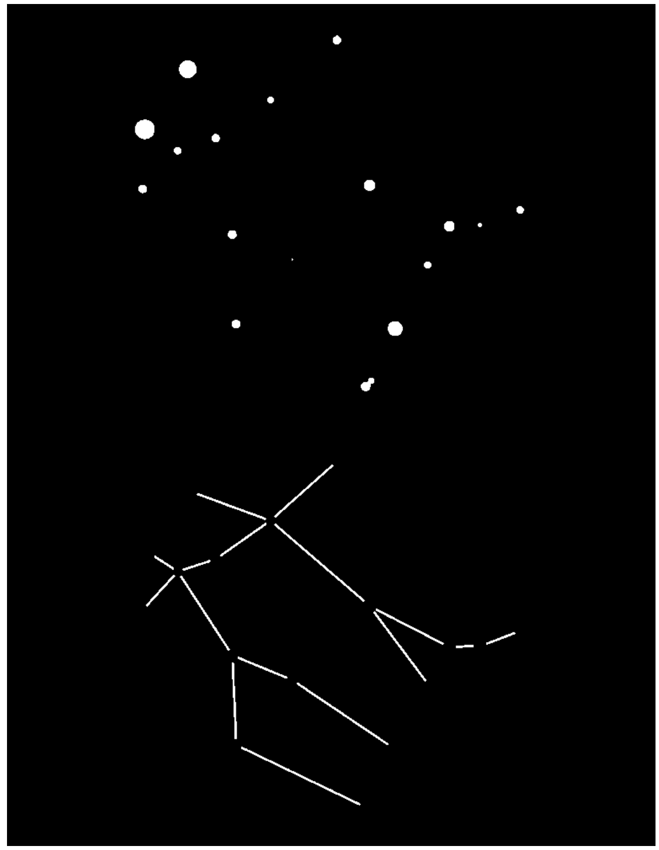


*Fig. 3. The binarized lines and stars image*

To make a template that can be compared to any image, we need to normalize the coordinates of the lines such that it can be used as a universal image of the constellation. We achieve this by using the brightest and second brightest star of the constellation. The area of a star represents its brightness in the sky, so we first apply Canny edge detection and create contours of the stars. Then based on the area (brightness), we rank the stars. We apply translation, scaling and rotation such that the brightest star lies on (0, 0) and the second brightest star lies on (1, 0). All the other stars are the realigned based on these two stars. This results in a normalized plot of the constellation, that can be compares to other test images by ensuring that the transformation is applied in the same way. The lines plot was done in the same way, where lines were detected using Hough lines and the endpoints of the plotted lines was obtained. The final normalized template can be seen in Fig. 4. The same process was repeated for all 30 constellations and a database of normalized templates was created. The normalized coordinates of stars was also stored in a file so that we don't have to create the templates again and again.

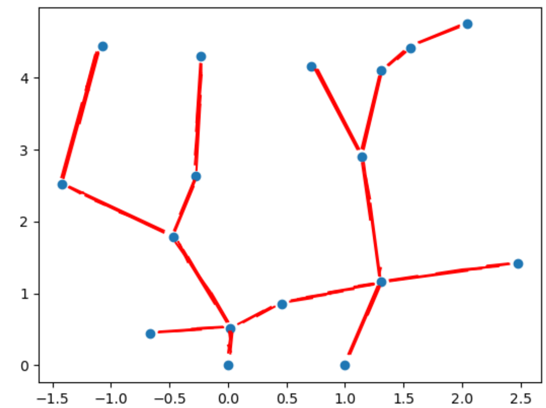



*Fig. 4. The final normalized template*

### 2.2. Test image processing

Due to the absence of a proper database of night sky images, we used an online planetarium to create test images as close to real life images as possible. A sample of this image can be seen in Fig. 5.

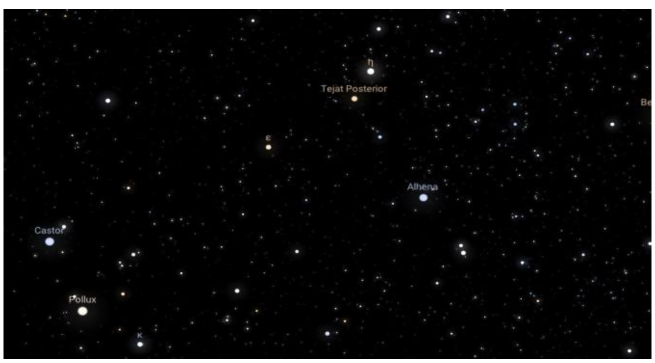



*Fig. 5. A test image obtained from stellarium*

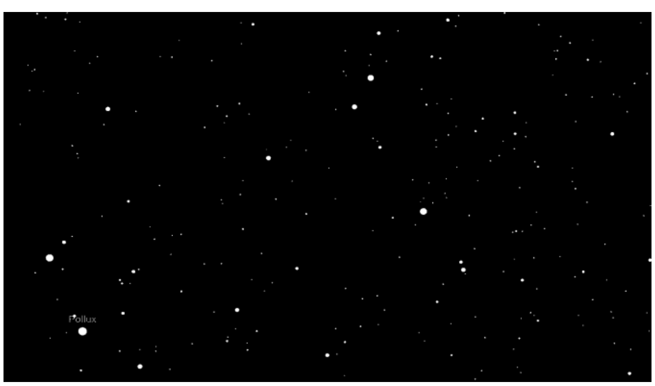



*Fig. 6. The processed and binarized test image*

Since we only require the stars to appear in the image, as is the case with a real image, we binarize the image using an appropriate threshold to keep major in the image. We also ensure to keep a threshold low enough to keep dim stars in the image, since dim stars can also lie in a constellation and will be important to ensure that the detection when comparing to the templates will be accurate. The binarized test image can be seen in Fig. 6.

### 2.3. Detection algorithm

Now we have a database of normalized templates for 30 constellations and a processed test image. We perform the following steps to compare the test image and the templates –

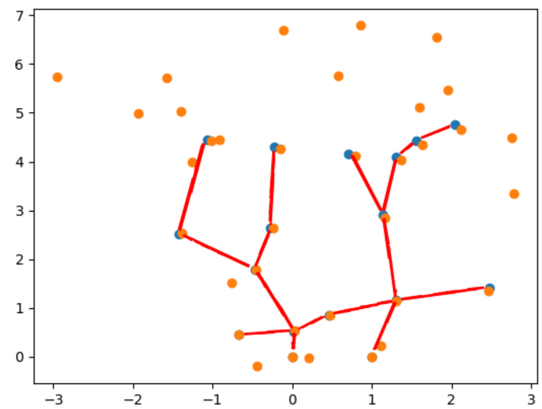
1. Rank the stars in the test image based on brightness.
2. Normalize the test image based on the brightest and second brightest star.
3. Since the normalization is done for both templates and test images, the relative distances can now be compared.
4. Iterating over the 30 templates, we apply two ways to compare the similarity of a constellation in template to the test image, compare the coordinates in the template to the coordinates in test image, or calculate distance of a star in template to the nearest coordinate in the test image.
5. The brightest and second brightest stars are iterated for all stars having an area greater than a specified threshold. This allows us to consider all possible combinations of bright stars present in the test image.

While comparing the coordinates of templates and transformed test image, we allow a small error during each coordinate matching (dx, dy). If a coordinate lies in the tolerant range, we consider as a star corresponding to the constellation as having been detected. If we match at least 50% of the total stars present in the constellation, we conclude that the constellation is detected in the test image.

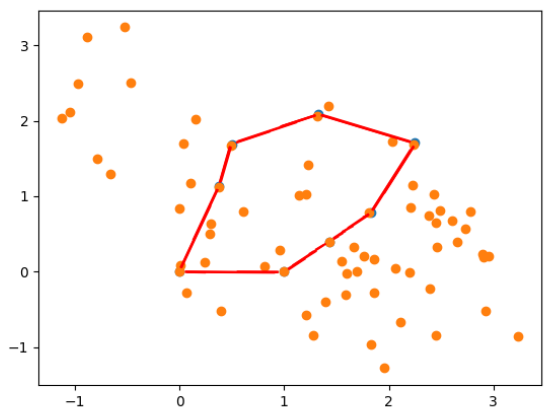
## 3. RESULT

On comparing test images and the 30 templates, we found that the coordinates of the template are overlapping the coordinates in test images, and thus the one with the most overlap will be the constellation present in the test image. We can see in the Fig. 7, the overlap of the template and test image for Gemini and Vela constellations.

Of the given 20 test images, we were able to identify the correct constellations in 10 test images correctly, based on the similarity score of all 30 constellations for each test image. Of the 7 naked eye test images, 5 were correctly classified.




(a)




(b)

*Fig. 7. Matching and overlapped plot of the constellations template and the test image for constellations (a) Gemini and (b)Vela*

## 4. CONCLUSION

Our algorithm was successfully able to detect constellations in the test images using the templates we created. The creation of normalized templates played a big role in the performance of the algorithm. We built the templates based on the key features we wanted from the template, the stars and the lines connecting them, and applied the developed detection algorithm for reliable detection of constellation in a test image. This can help sky gazers and budding astronomers to explore constellation without the frustration of keeping track of various stars and patterns

## 5. FUTURE WORK

While we were able to successfully detect an existing constellation in a test image, we can extend the detection algorithm to consider local matching of constellations, so that we can detect multiple constellations in a single test image. Also, we have set hyperparameters like tolerable error for difference between test and template coordinates, thresholds to binarize test image, and lower threshold of number of stars to classify a constellation as having be detected, but it would be interesting to explore on ways to find the optimal way for setting these parameters instead of hardcoding them.

We can also use the brightness ranking of templates in the detection algorithm as well. We are comparing the relative coordinates of the template and test images, which may lead to detect patterns that are not represented by those stars, which can be somewhat improved by checking if the ranking of brightness in the template is maintained when overlapped with the test image.

## 6. REFERENCES


[1] Rehman, M., etc. “Single Star Identification and Attitude Determination in Tracking Mode” International Conference on Control, Automation and Systems 2008

[2] M. D. Pham, etc. “A Star Pattern Recognition Algorithm for Satellite Attitude Determination” IEEE Symposium on Industrial Electronics and Applications September 23-26, 2012

[3] The night sky – 88 constellations http://astronomyonline.org/Observation/Constellations.asp

[4] List of constellations by their area - https://en.wikipedia.org/wiki/IAU_designated_constellations_by_area

[5] Jiang, M., etc. “A Novel Star Pattern Recognition Algorithm For Star Sensor” Proceedings of the Sixth International Conference on Machine Learning and Cybernetics, 19-22 August 2007

[6] Stellarium, an online planetarium, Matthew Gates, Barry Gerdes. https://stellarium-web.org/