Constellation Detection

Suyao Ji, Jinzhi Wang, Xiaoge Liu*
*To whom correspondence should be addressed. Electronic mail:liuxg@stanford.edu

***Abstract* — In the night with clear sky, the beautiful star patterns attract lots of star gazers and photographers. Constellation is a convention that people used to partition the stars with myth and stories behind each constellation, making the starry sky more compelling. However, the constellation detection requires complicated knowledge and experience. Our team's goal is to automatically detect the constellations appeared in an image based on the 88 constellations division by International Astronomical Union (IAU)[1]. The name and constellation pattern is plotted onto the original image to directly show the detection results. The project is made of three phases – image preprocessing, template machine learning and constellation pattern detection. Our algorithm shows a good accuracy of 92.8% and fast speed of 85s/image with our 15 test images.**

Keywords*— constellation; constellation detection; star pattern; image processing; template matching

I. INTRODUCTION

Our night sky is divided by 88 modern constellations with the IAU convention. This division method divides the sky into 88 areas. The constellations have served as the 1st global positioning system (GPS) and faithful time determine system in the past and still play an active role in satellite positioning in aerospace [3,4,5,6]. In the meanwhile, due to fascinating myth and story behind every constellation, people always possess lots of enthusiasm for them. However, only skilled and well experienced star gazer with sufficient background knowledge can distinguish the constellation. Numerous guide books, websites and cell phone apps are created to help unprofessional people find the constellations [7]. But they can only provide suggestions and are not able to take environmental factors such as weather into consideration. The accuracy and speed is heavily relying on the watchers.

Here, taking advantage of image processing techniques, we purpose an constellation algorithm detect constellations directly from a photo of the night sky. Since the constellations in the sky have fixed patterns and neighborhood relationships, it is probable to apply out specific template matching techniques to match the constellation templates with the real star patterns on the image.

There are three important steps in our constellation detection approach. First, the image is preprocessed to a binary image, which filters out all the non-star objects and reduces the possible noise to generate a clear image. The preprocessed image will only show the stars with area standing for its visibility magnitude. Second, template images of 88 constellations are used to generate constellation database with the information descriptors. Third, we compare the preprocessed image with constellation templates to search for all possible matching star patterns and then remove all the wrong matching afterwards. The detailed explanation can be found in Part II.

II. IMPLEMENTATION

Constellation detection is a one-to-many template matching problem–Each test image could have multiple constellations. Thus, on one hand, an informative and well constructed template database is built; on the other hand, a precise and efficient detection methodology is developed to detect all the existing constellations in the test image. The implementation flow is shown in Fig. 1.

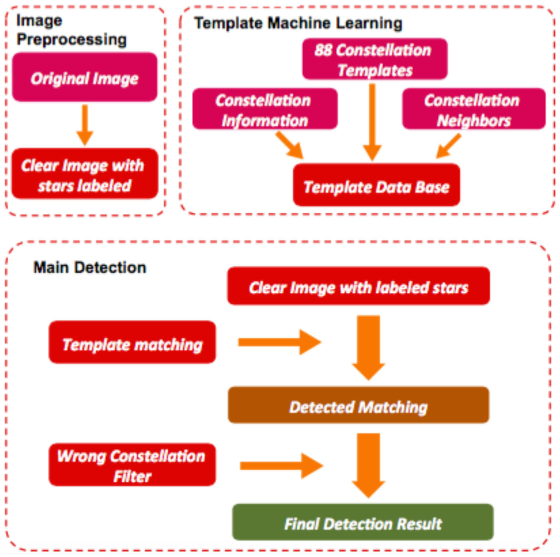
Image Preprocessing
Original Image
Clear Image with stars labeled
Template Machine Learning
88 Constellation Templates
Constellation Information
Constellation Neighbors
Template Data Base
Main Detection
Clear Image with labeled stars
Template matching
Detected Matching
Wrong Constellation Filter
Final Detection Result

Fig. 1 Image Processing Flow

In this section, the photo is firstly preprocessed in part A. In the meanwhile, a database construction is introduced in part B, followed by fundamental detection approach explained in part C. Lastly couple optimization is applied to improve the detection performance in part D.

A. *Image Preprocessing*

It is always easier to process a grey image than a colored one. For constellation detection, we would like to get a binary picture with only all the star information included and foreground objects such as trees and buildings excluded. The preprocessing result is shown in Fig. 2. At night, foreground objects normally have a much weaker illumination compared to stars. By properly choosing a threshold value, a binary image can be generated. Unlike image processing for other applications, we on purpose don't apply further filtering to the binary image; this is because dim stars need to be preserved for detecting all possible constellations.

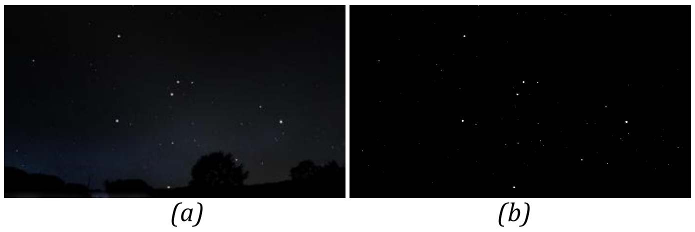

Fig. 2 Image Preprocessing Result. (a) Raw test image; (b) after preprocessing, only stars are preserved

B. *Implementation of constellation database*

The template database is very critical in achieving good accuracy and fast speed. Regular template matching approach does not work well for this application because the constellation template is scatter based. A specific descriptor is built for this project.

The template we use is based on a set of modified constellation charts [2] extracted from IAU standard constellation chart [1]. A sample of the template is shown in Fig. 3 demonstrating the constellation of Gemini.

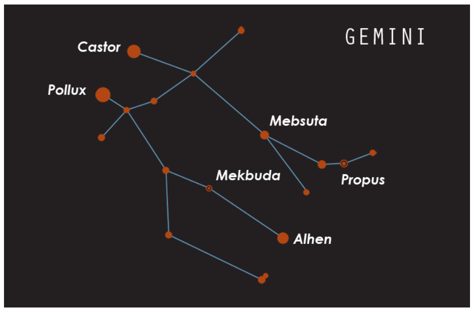

Fig. 3. Original Image of Template of Gemini

We need to process the original template image before we can use it as a template and quantify the template as data structures for further detection. First, Every single template image is first binarized. We pick different colors to efficiently distinguish stars and connection lines (stars are represented by red and connection lines is represented by blue). Second, we filter out unrelated features in the template image. Third, the stars and connection lines are recorded from the template.

To get one step further to the constellation descriptor, for each constellation template, we captured the first and the second brightest star by their area. The brightest star is set as (0, 0) and the second is set as (1, 0), hence the scale is normalized to the distance between these two stars. To extract the constellation scale information, we keep track of the distance between the two brightest stars in number of pixels. All the rest of the stars in the constellation are processed in the coordinate system and their brightness is recorded a cell as our database.

For each constellation, we capture its three brightest stars' visibility magnitude. The test image can be with different scales and brightness configuration, thus we need to keep track of the visibility magnitude and able to reverse the area by the formula defining the visibility magnitude. In approximation, $Area = \left(\frac{1}{2.5}\right)^{Mag}$

For detection optimization discussed in Part D, constellation neighborhood information is utilized. Regardless of different photographs, the relative location between constellations is astronautically invariant. Thus we record the first five nearest constellations as "neighbors" and store them as part of database. The final template of Gemini is shown in Fig. 4., with brightness ranking index as well as normalized brightness information included .

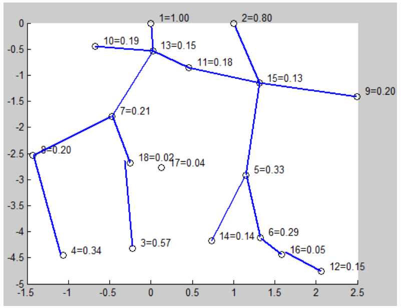

Fig. 4. Visualized Descriptor of Gemini

C. *Detection Algorithm*

Given the preprocessed binary image and constellation templates are available. We follow the steps described below to detect all the constellations in the test image. (1)apply region labeling algorithm to rank the stars based on brightness; (2)ascend the templates based on star number existing in the constellation; (3)pick the brightest two stars in the test image to determine the scale and rotation of the template i (i=1....num_templates); (4)check whether there is a star in the test image match the relative position of the kth (k=3...num_star_in_temp) star in the template; (5)repeat step 4 until NUM_MATCH stars are matched, then we decide there is a constellation matching i in the test image; (6) repeat steps 3-5 to find other constellations.

When iterating over star pairs to determine the template scale and rotation, the rotation angle θ, and transformed template $temp_{rot}$ is calculated as below:

$$\theta = 90 * \left(1 - sign(dx)\right) + \text{atan}\left(\frac{dy}{dx}\right) \qquad (1)$$

$$temp_{rot} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} * \begin{bmatrix} X_T \\ Y_T \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \qquad (2)$$

where the star pairs have coordinates (x_0, y_0) and (x_1, y_1), $dx = (x_1 - x_0), dy = (y_1 - y_0)$, $\begin{bmatrix} X_T \\ Y_T \end{bmatrix}$ is the coordinates of original template.

Once the template transform is achieved, we then check at each point of $temp_{rot}$, whether there is a star in the test image. Allowing a small offset of the star, we add a tolerance parameter $(\Delta x, \Delta y)$ when doing the detection. There is a possibility part of the constellation stars are missing or out of frame; we set the matching threshold NUM_MATCH to be half of the total star numbers in the constellation template. If the matching number is above the threshold, we decide the constellation is detected in the test image.

Finding the proper scale of the template improves the accuracy and speed tremendously because all constellations should share very close scaling since they are on the same image. The correct scaling could filter out a lot number of wrong matching patterns. Correctly detecting the first constellation is the key to find the proper scale, as it will serve as a reference to filter out detections with wrong scales. Constellations with large number of stars are more likely to be detected correctly. Thus the search list of constellation we follow is with descending star numbers. Once a first constellation is detected, we will use its scale as reference to filter out the rest of the star detections. Fig 5 explains the detection result difference if we don't constrain the scale of the template.

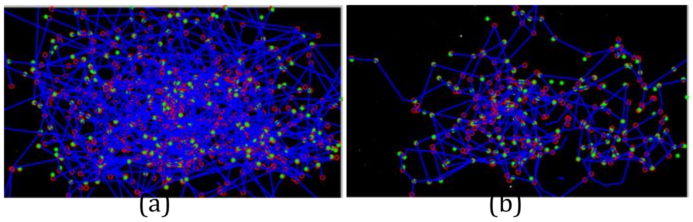

Fig. 5 Detection Result with and without scale filter. (a)Detection results without scale filter, a lot of incorrect constellation matches are generated; (b) Detection result with scale filter, only the matches with proper scale is preserved.

After detected all the correct matching with proper scale, we need to remove all the wrong detections. The first detected constellation with largest number of match stars is correct. In Fig.6, *Gemini* is the first detected constellation with 18 stars in it. Any detection that shares the same stars with *Gemini* is wrong since each star could only belong to one constellation. Then, search the rest of detection for the neighbor of *Gemini* and continuing remove wrong detection with has overlap with those neighbors. Iterate the process until no further neighbor is found.

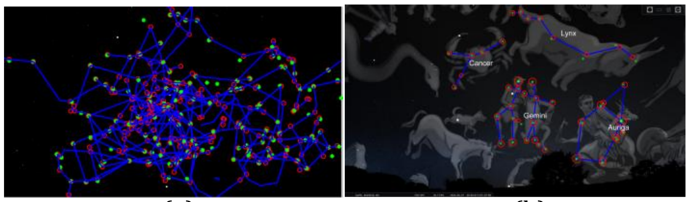

Fig. 6 Detection result with and without filtering incorrect matches with overlapping stars (a) Detection results before removing incorrect detection, (b) Detection result after removing wrong detection.

After detection is done, we plot the detected star patterns together with their connectivity drawn on the raw image, so that the user can identify each detected constellation on graph easily (shown in Fig. 6(b)).

D. Optimization on accuracy and Speed

(a) Brightness ranking order mismatch: In order to improve accuracy, we add star brightness ranking constraints to the detecting approach. We expect the brightness ranking order of the template is very close to the successfully matched stars for that template. The ranking order difference is the norm2 of the brightness rankings of detected constellation and template. If the ranking order is off by a lot — exceeds the threshold we set. Then we still consider it to be an incorrect matching and throw it out of the detected constellation pool.

(b) Total number of stars: The total number of stars of all the constellations is less than 400. When the image has more than 400 stars, it is very likely that there exists very dim stars detected by the camera but not the human eyes. So we limit the total number of stars on the image increases our speed and accuracy.

(c) Undetectable constellations: Constellations with only two or three stars is basically undetectable, because there are too many incorrect matches in the image for them and it is very hard to decide the correct match. Constellation with equal or less than three stars are not in our detection pool.

(d) Neighbor information: We utilize the information of constellation neighborhoods. Once the first constellation is successfully detected, the next templates we are searching for are its astronomical neighbors. This tree search

algorithm shrinks the template searching pool—only possibly existing constellations are detected. Thus, it improves both accuracy and speed performance.

III. RESULT

With the proposed constellation detection approach, we tested with 14 test images with totally 28 constellations on them. Constellations of 10 images are detected correctly with no missing or incorrect detected constellations. Of the 28 constellations, 3 of them are undetected, and there are 2 incorrect detected ones. With different test image size and star information size, the average detection time is 85 seconds. The result is summarized in Table 1.

Table 1. Constellation Detection Statistics

	Value
Correct Constellation detection percentage	92.8%
Correct Test Image detection percentage	71.4%
Average speed	85s

In Fig. 7 We show some typical test image's results. Our algorithm can achieve quite high accuracy with relative short time. Even some constellations such as *Orion* and *Taurus* (Fig. 7) with a few stars blocked by the tree can still be detected.

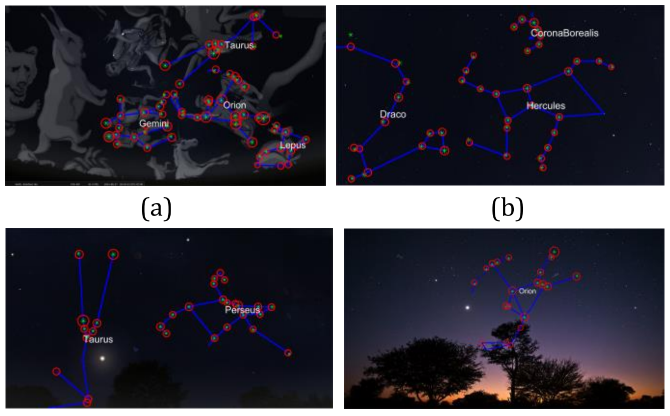

Fig. 7 Final results after detection. (a)TestImg2, (b)TestImg5, (c)TestImg13, (d)TestImg14.

It is noted that for some special cases, the detection is still not perfect. (1) When the constellation is cut by the image, showing only small portion of it. (2) We didn't detect constellation only has 2 or 3 stars in it such as *Canis Minor* (3) when the constellation only has very dim stars, the accuracy drop very fast.

IV. CONCLUSIONS AND FUTURE WORK

Our algorithm successfully achieves accurate and relative fast constellation detection. Unlike regular template matching problems, constellations are scatter-based, so existing techniques such as SIFT detector or SURF detector cannot be applied. We build the database based on the information we need for constellation detection; then we apply the proposed algorithm to detect star patterns we are interested in. This enables the sky gazers and amateur photographers to extract star patterns of interest conveniently from their photos.

Since the constellation can be used to determine the location and time. An inspiration of our future work would be telling where and when the image is taken. In the meanwhile, the parameters such as error tolerance, scale tolerance will still depend on the cameras. Automatically choose the optimized parameters will also be very interesting.

ACKNOWLEDGEMENT

We would like to greatly acknowledge Prof. Bernd Girod and Gordon Wetzstein for their enlightening course. We would also like to thank Huizhong Chen, Jean-Baptiste and Jason Chavez for their helpful and kind mentorship. Special thanks to Hang Li who is auditing this course and contributed to the project.

REFERENCE

[1] Liebe, C. "Pattern Recognition of star Constellations for Spacecraft Applications", *IEEE AES Systems Magazine*, Jan 1993
[2] 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
[3] Rehman, M., etc. "Single Star Identification and Attitude Determination in Tracking Mode" *International Conference on Control, Automation and Systems* 2008
[4] M. D. Pham, etc. "A Star Pattern Recognition Algorithm for Satellite Attitude Determination" *IEEE Symposium on Industrial Electronics and Applications* September 23-26, 2012,
[5] Stellarium, Star Mapping Software, Matthew Gates, Barry Gerdes
[6] The constellations. *International Astronomical Union*. 5 June 2015. Retrieved from http://www.iau.org/public/themes/constellations/
[7] The Night Sky – 88 Constellations. *AstronomyOnline.org*. 5 June 2015. Retrieved from http://astronomyonline.org/Observation/Constellations.asp?Cate=Observation&SubCate=MP07&SubCate2=MP0801

APPENDIX A
CONTRIBUTIONS

The constellation detection algorithm emerged through group brainstorm. All design decisions, software architecture, algorithm selection and evaluation and coding were group efforts. In that context, the following are of concentration were claimed by individual group members to allow periods of uninterrupted individual effort.

Suyao Ji:
• Image preprocessing algorithm investigation, modeling and development
• Detection algorithm investigation, modeling and development

Jinzhi Wang:
• Template database study and implementation
• Constellation background research and algorithm investigation
• Poster design

Xiaoge Liu:
• Come up with the project idea and frame of the algorithm
• Template database study and implementation
• Constellation background research
• Detection algorithm development
• Detection accuracy and speed optimization
• Test implementation