CAL POLY

A Hardware-in-the-Loop Star Tracker Test Bed

Ashley Haraguchi

June 10, 2024

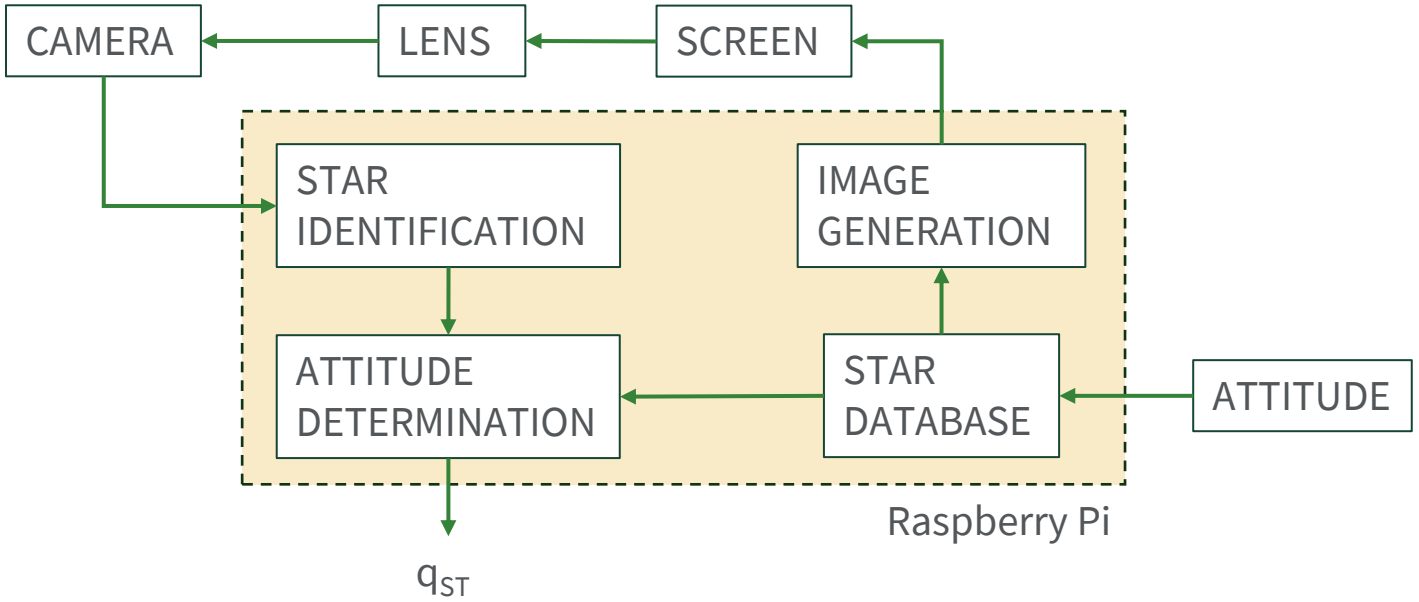
Agenda

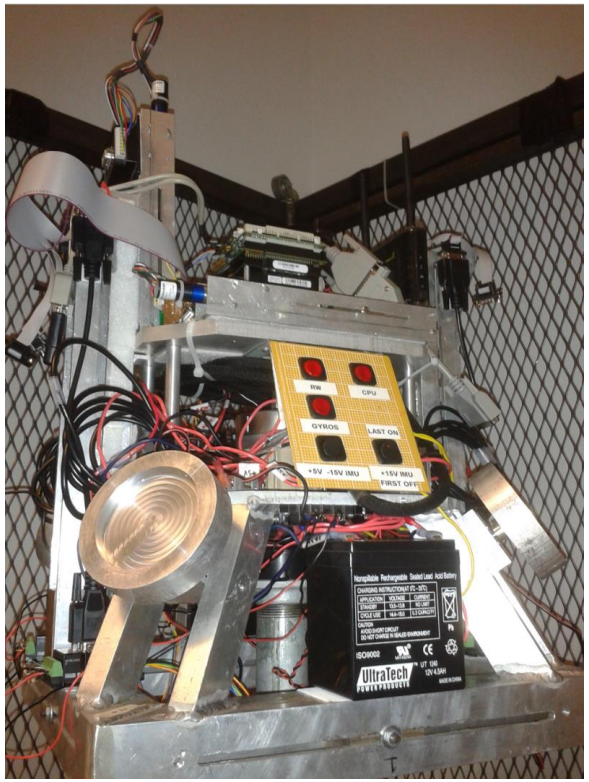
- Introduction
- Background
- Software Design
- Software Validation
- Hardware Design
- Analysis and Results
- Conclusions

Thesis Objectives

Introduction

- Create a hardware-in-the-loop star tracker test bed for eventual integration with Cal Poly's Spacecraft Attitude Dynamics Simulator
- Ensure identification is achieved with a static star field image



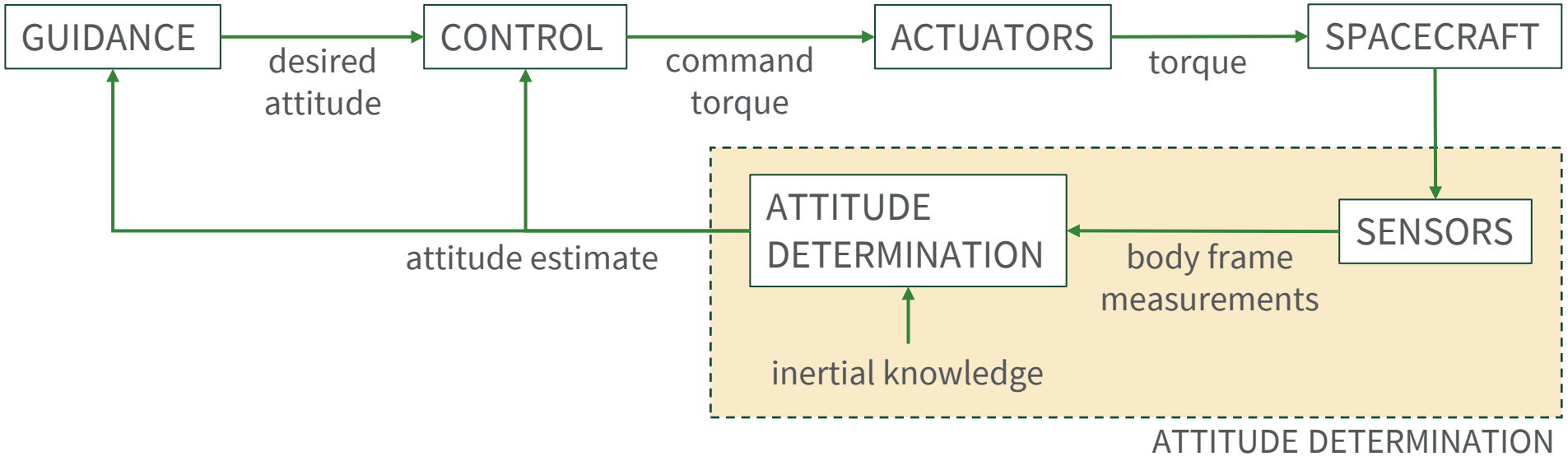


Spacecraft Attitude Dynamics Simulator (SADS) [1]

Spacecraft Attitude Determination

Background

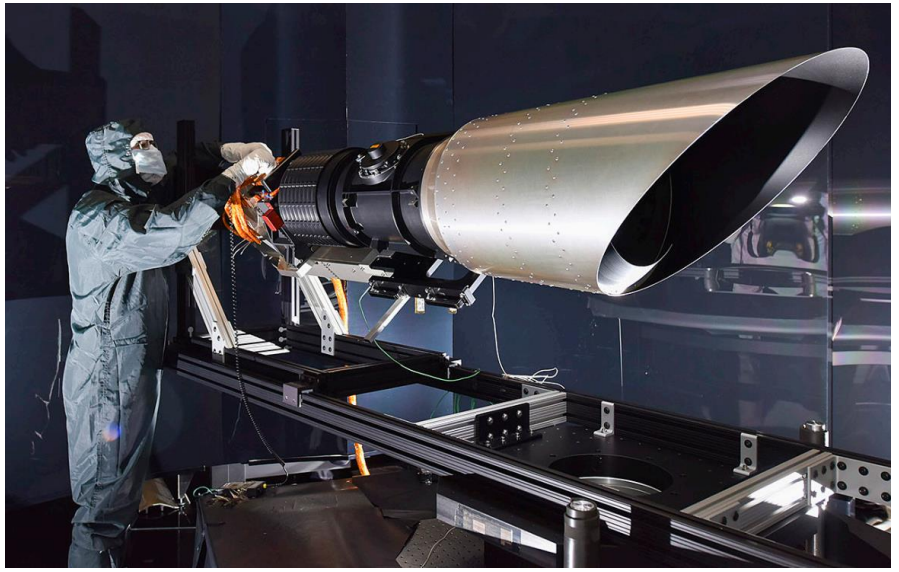
- Many space missions require a high degree of pointing accuracy, which means the spacecraft attitude must be estimated on-board
- Attitude determination deals with obtaining measurements from the spacecraft body and using those to estimate the orientation of the spacecraft with respect to an inertial frame



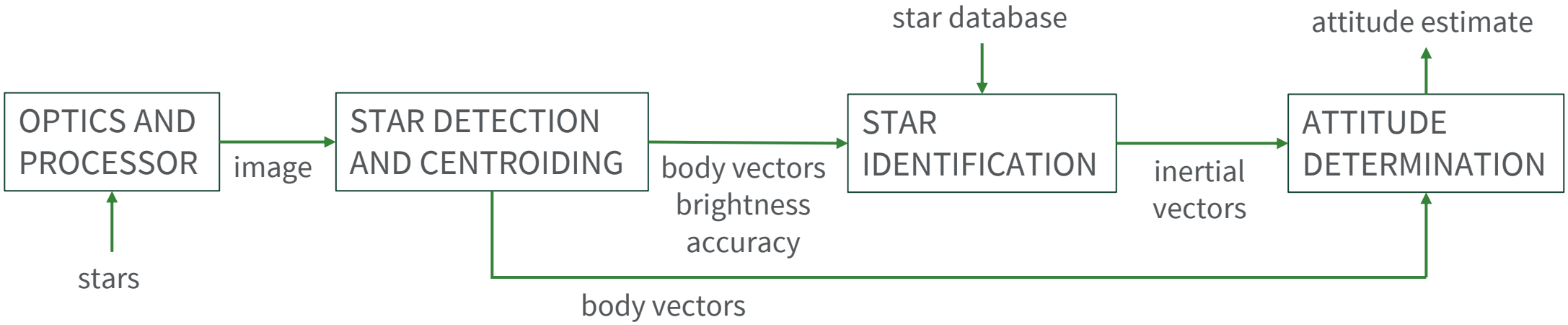
Star Trackers

Background

- Star trackers are the most accurate attitude sensors, able to provide an accuracy on the order of arcseconds
- By observing multiple stars, three axis attitude knowledge can be determined with just one star tracker



Ball Aerospace's High Accuracy Star Tracker [2]



Low-Cost Star Trackers

Background

- Commercial star trackers cost, on average, $33,000
- Cheaper sensors, like magnetometers or sun sensors, have performance an order of magnitude worse
- In-house star trackers are developed by universities and research groups for CubeSats or other small satellite missions
  - Cheaper and smaller than commercial star trackers
  - Use commercial-off-the-shelf components
  - Write their own or use open source software

Developer	Star ID	Accuracy	AD Method
CPCL [3]	Angular	89%	QUEST
TSL [4]	Pyramid	-	SVD
SPEL [5]	Source Extractor/Match	97.3%	-
Sarvi et al. [6]	Pyramid	96.81%	QUEST

Star Field Simulation

Background

- Hardware-in-the-loop testing shows how real hardware responds to simulated input
- For star trackers, an entire star field is simulated to test both star tracker imaging hardware and image processing and attitude determination software
- Two types in the literature
  - Hemispherical dome with LEDs to simulate a star field [7]
  - Screen with collimating lens to display a generated image of a star field [8, 9, 10, 11]

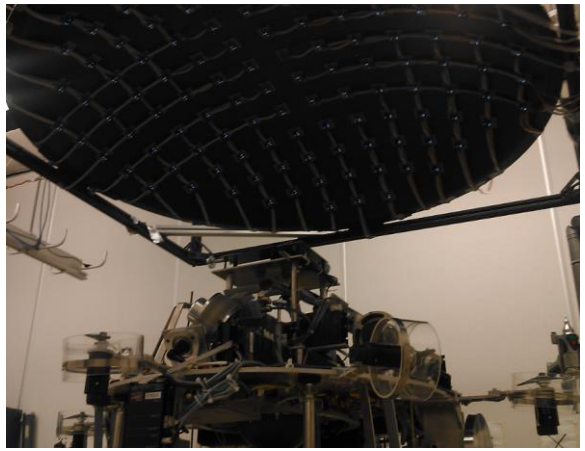


Image source: [7]

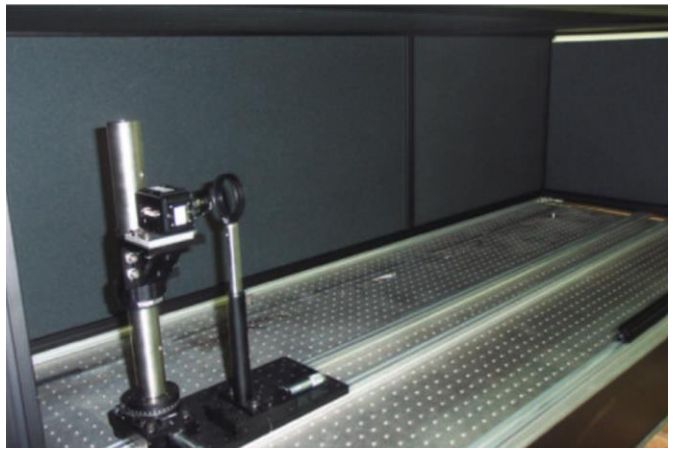
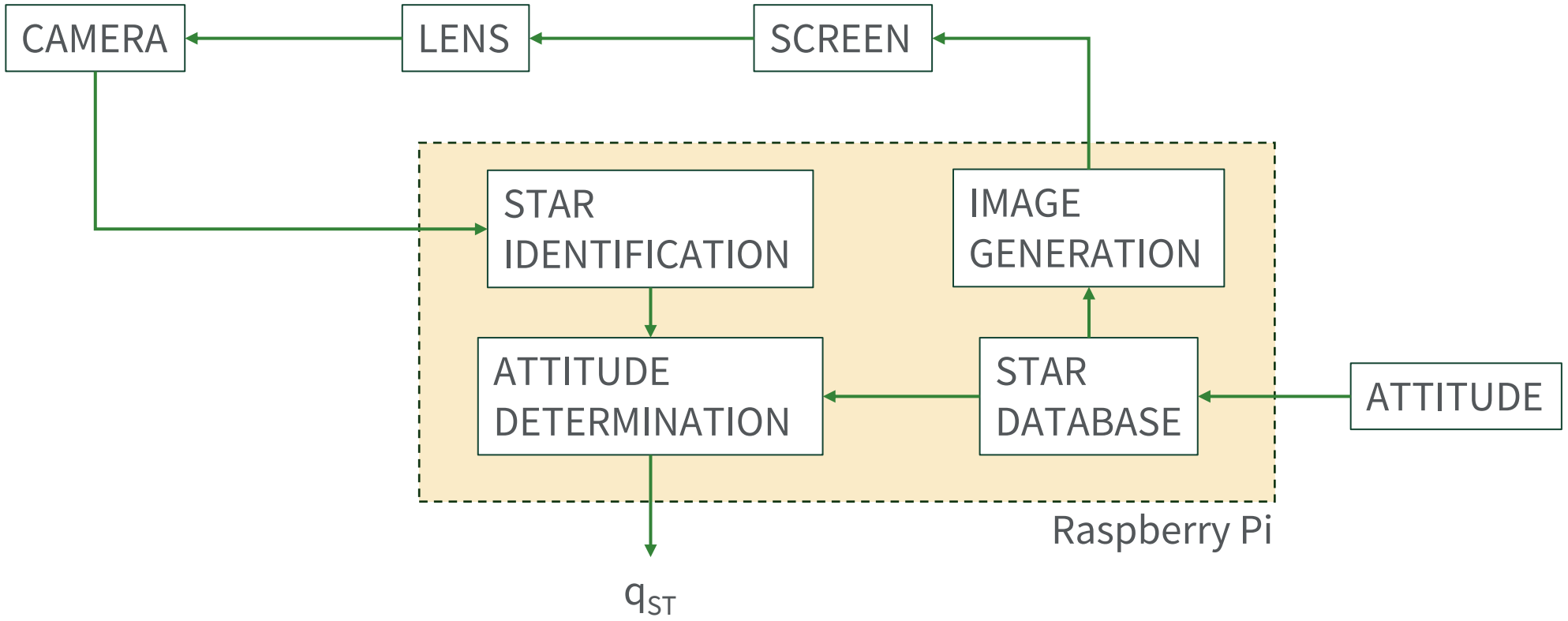


Image source: [9]

Thesis System Design

Background

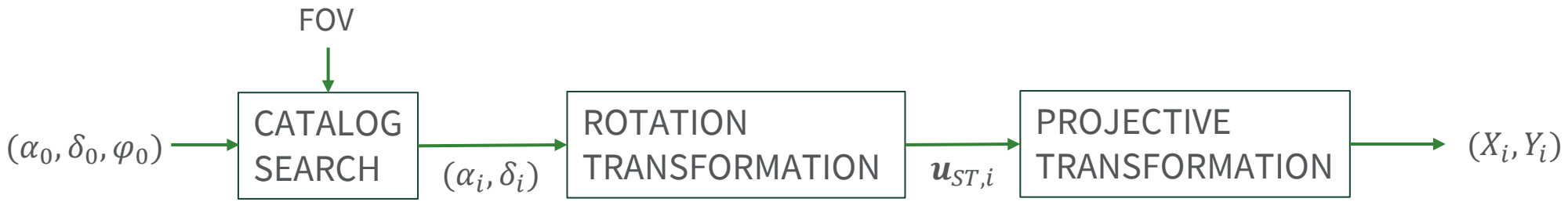


Software Design

Image Generation Software

Software Design

- Provide a sky location to display, given in right ascension, declination, and roll
- Search the star catalog for stars in view
- Rotate the unit vectors of those stars into the star tracker reference frame
- Project the 3D unit vectors onto the 2D image plane



CAL POLY

Star Catalog

Software Design – Image Generation

- Using the Brief Bright Star Catalog (BS5), a pared down version of the Yale Bright Star Catalog (YBSC)
- Contains all 9,110 stars of the YBSC, but with less data columns, and available online in spreadsheet format
- Data used:
  - J2000 right ascension (RA) and declination (DEC)
  - Visual magnitude

RA	DEC	VMAG	INTEN
1.7678	-0.2918	-1.46	255
1.6753	-0.9197	-0.72	178
3.7335	0.3348	-0.04	142

Parsed star catalog snippet

Rotation Transformation

Software Design – Image Generation

- Star i's right ascension and declination can be converted to a unit vector in the celestial sphere reference frame, ICRF, with

$$\boldsymbol{u}_{ICRF} = \begin{bmatrix} cos\delta_i cos\alpha_i \\ cos\delta_i sin\alpha_i \\ sin\delta_i \end{bmatrix}$$

- This is rotated into the star tracker reference frame, ST, with a 3-1-3 active rotation sequence M such that

$$\boldsymbol{u}_{ST} = M^T \boldsymbol{u}_{ICRF}$$

- M is built from the right ascension, declination, and roll provided to the image generation algorithm

Projective Transformation

Software Design – Image Generation

- The coordinates of $\boldsymbol{u}_{ST}$ and the focal length of the system are used to find the (X, Y) coordinates of the star in the ST x-y plane with

$$X = f\frac{x}{z}, \qquad Y = f\frac{y}{z}$$

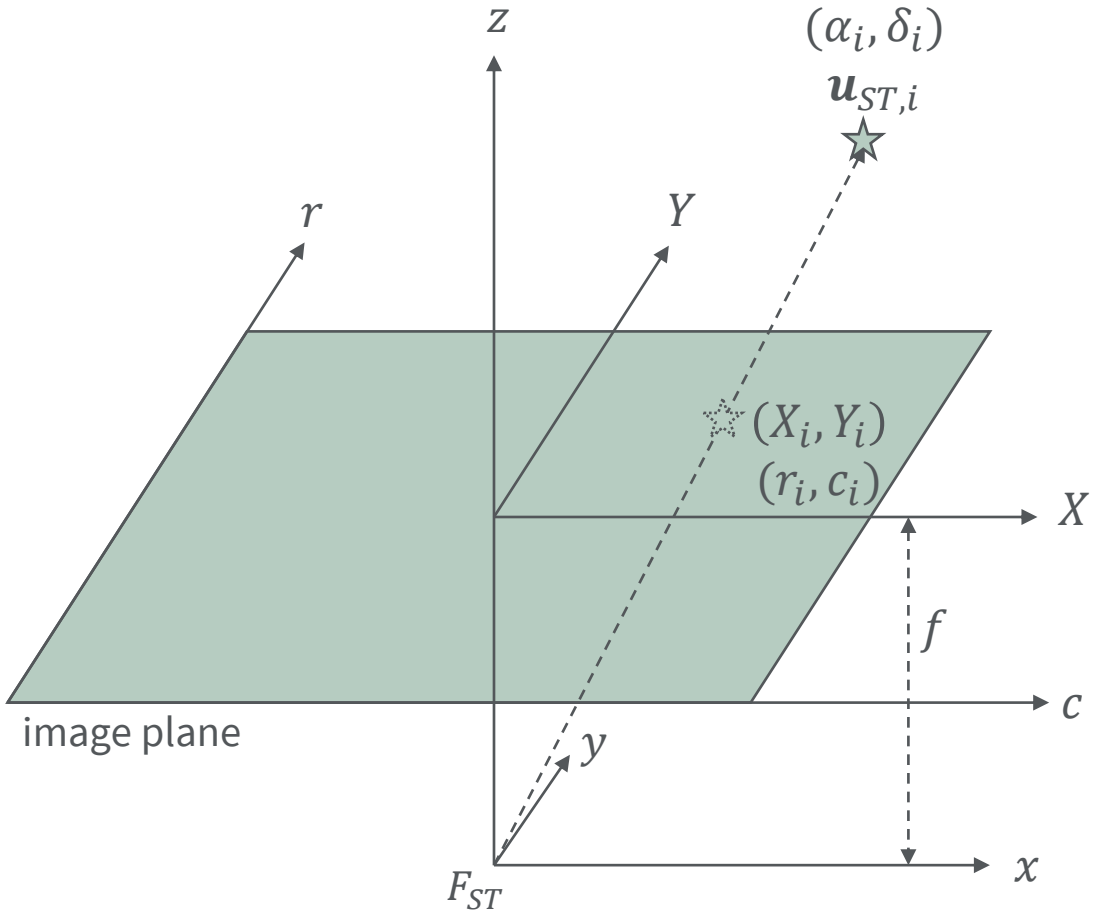
- Converted to raster scanning coordinates, using the desired image width and height, with

$$r = Y + \frac{h}{2}, \qquad c = X + \frac{w}{2}$$

- Raster scanning coordinates are used because the origin is in the top left corner, giving the image the same indexing scheme as a matrix

Coordinate Transformation

Software Design – Image Generation

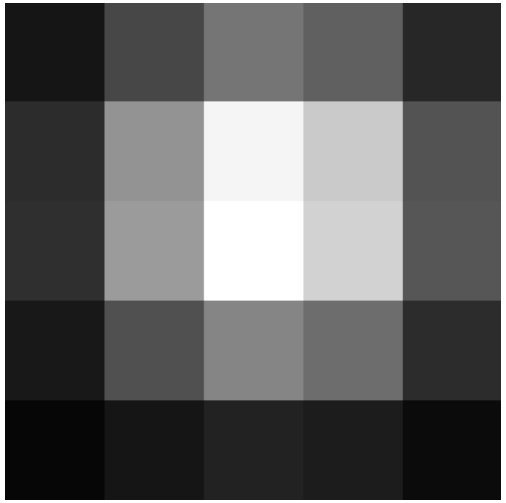


CAL POLY

Star Point Representation

Software Design – Image Generation

- Stars imaged on orbit are considered point sources, but must be represented on screen with at least one pixel [12]
- To increase centroid accuracy, the star is spread over multiple pixels using a Gaussian probability density function
- After trials of a variety of values, a 5x5 pixel star spot with standard deviation $\sigma = 1.2$ was selected
- Central intensity is determined by assigning each representable magnitude a pixel intensity from 0 - 255



Gaussian PDF with $\mu = (2.06, 2.72)$

Sample Images

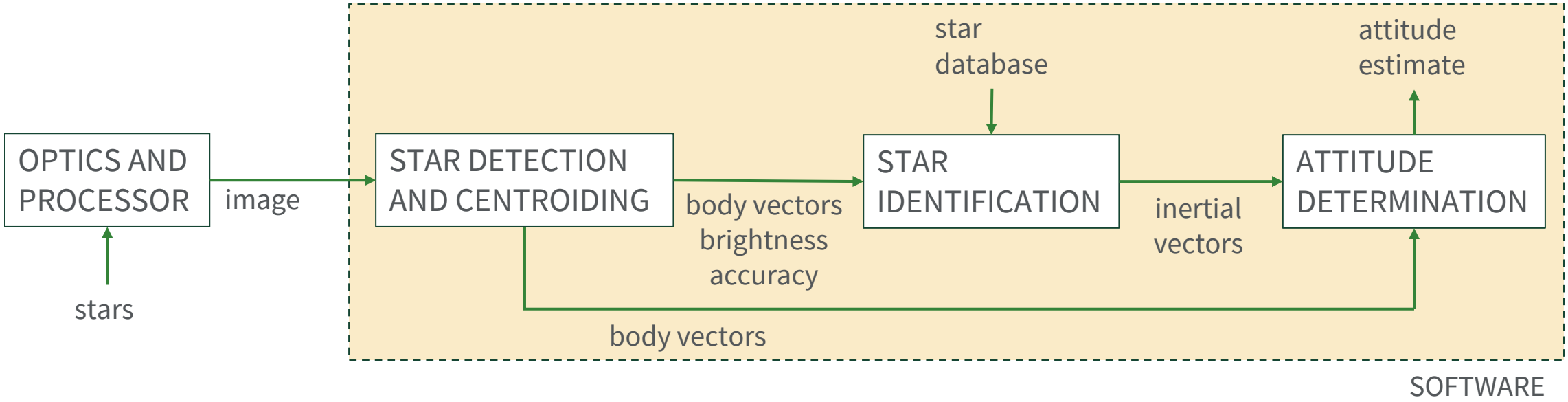
Software Design – Image Generation

Generated Big Dipper with $(\alpha, \delta, \varphi) = (195^o, 55^o, -30^o)$

Generated Orion with $(\alpha, \delta, \varphi) = (82.5^o, 5^o, 0^o)$

Star Tracker Software

Software Design

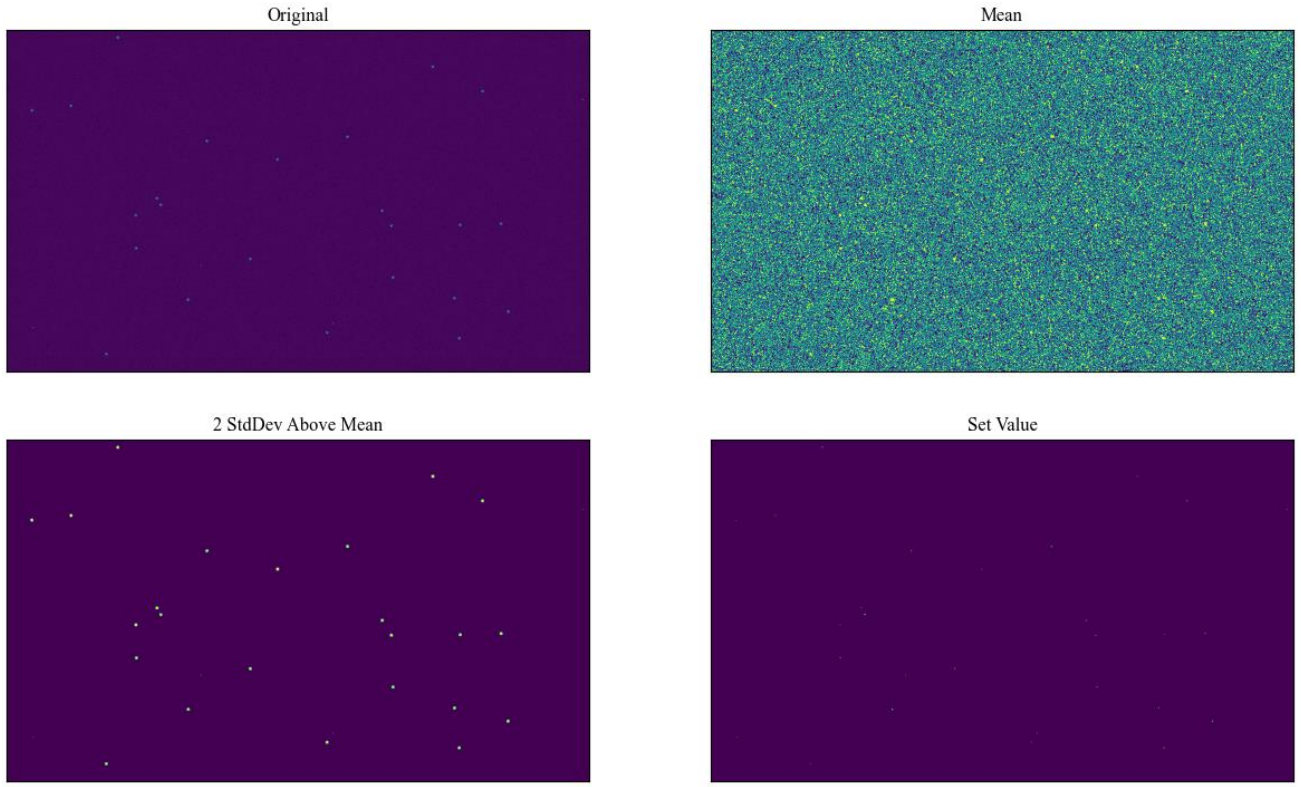


Star Detection and Centroiding

Software Design – Star Tracker

- Image is thresholded to extract star points from noisy background
- Connected component analysis creates groups of star spots, neglecting spots too large or too small to be considered stars
- Centroid of each candidate star spot is computed as the first moment of intensity of the desired region
- Unit vector in the star tracker reference frame is found with

$$\boldsymbol{u}_{ST} = \frac{1}{\sqrt{X^2 + Y^2 + f^2}} \begin{bmatrix} X \\ Y \\ f \end{bmatrix}$$

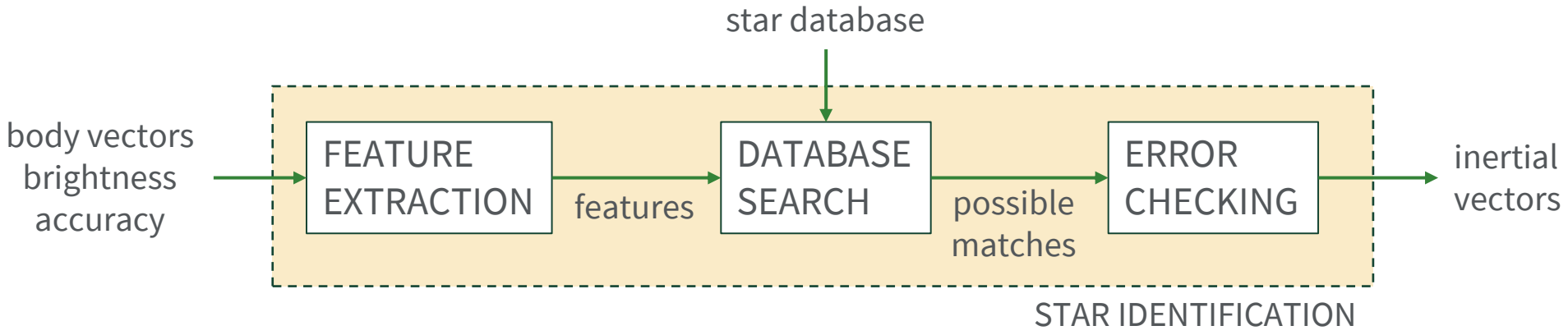


CAL POLY

Star Identification

Software Design – Star Tracker

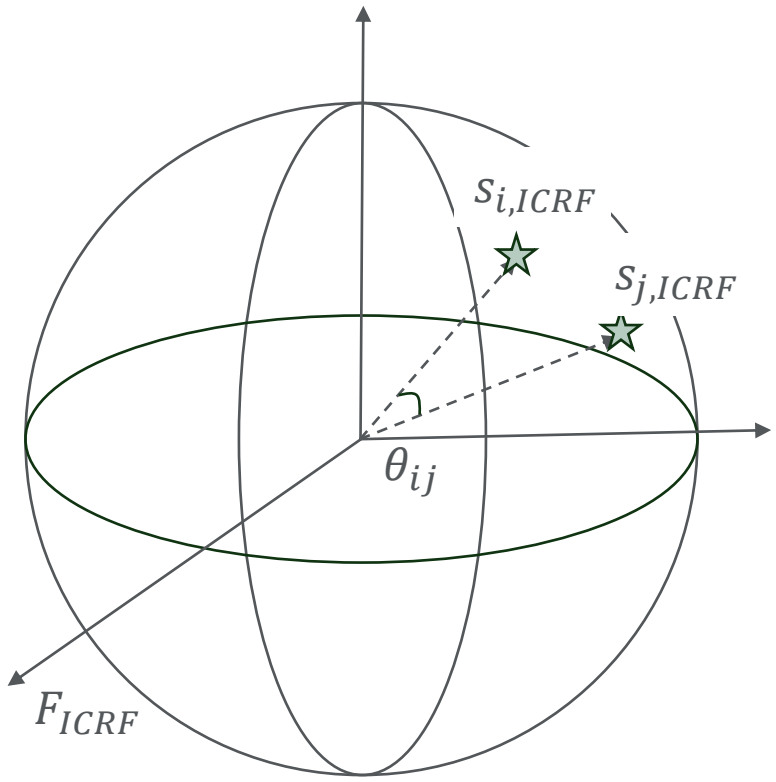
- Star identification algorithms use star features to match stars in the image with stars in the database
- The on-board star database is generated from the BS5 and contains unit vectors of stars in the inertial frame and the same identifying features

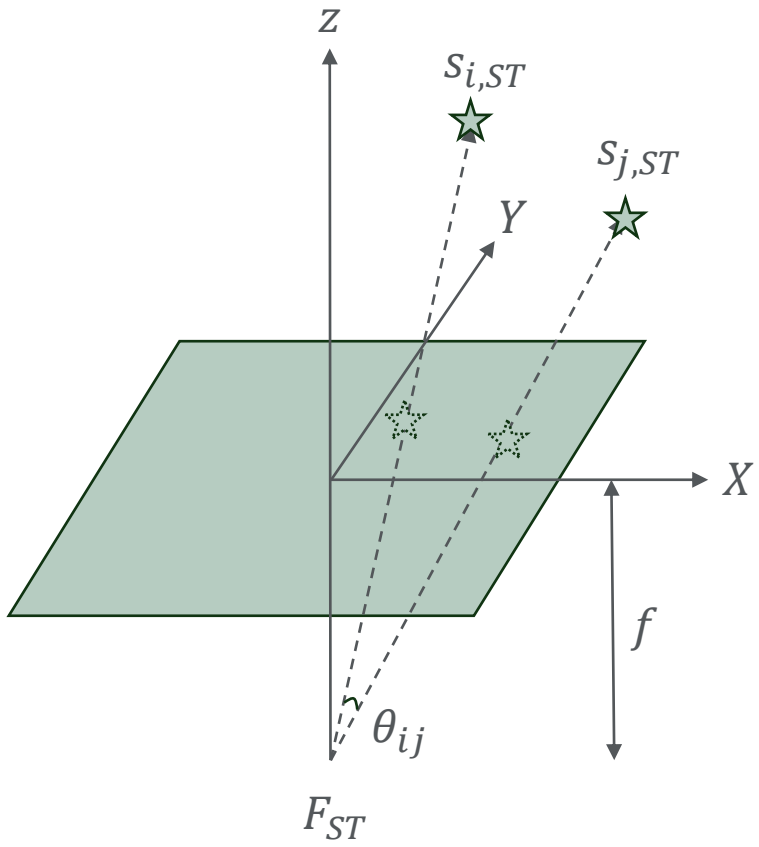


Identifying Features

Software Design – Star Tracker

- The pyramid algorithm identifies stars by the angular distance between a pair, $d_{ij} = u_i^T u_j = cos\theta_{ij}$



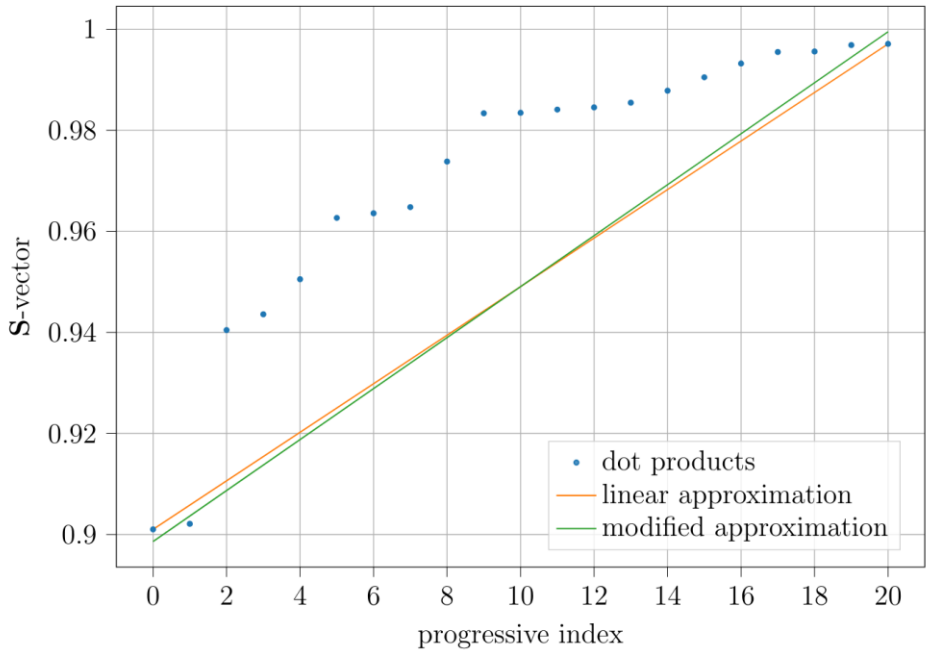


CAL POLY

K-Vector Searching

Software Design – Star Tracker

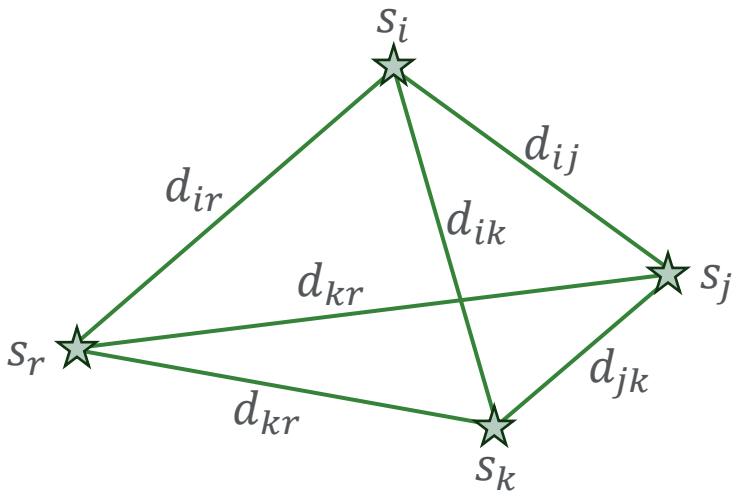
- Method of searching through ordered data in $O(k)$ time, where k is a small subset of elements in the database
- Detailed in "Search-Less Algorithm for Star Pattern Recognition" [13]
  - The **P**-vector is created as the value of the dot product of every star pair that could be in the field of view at any given time, $u_i^T u_j \geq cos\theta_{FOV}$, with **I** and **J** containing the associated indices
  - **S** is **P** sorted in ascending order, with **I** and **J** rearranged correspondingly
  - The **K**-vector is created using a modified linear approximation of **S**, with each element representing the number of elements in **S** below the approximation line
  - Values in **K** index into **I** and **J** to find the id of stars a certain distance apart



Pyramid Algorithm

Software Design – Star Tracker

- Developed in “The Pyramid Star Identification Technique” [14]
  - Select a star, s_i and find the three closest stars, s_j, s_k, s_r
  - Use **K**-vector searching to generate hash tables for the distances d_{ik}, d_{ir}, d_{ij}
  - For each pair in the ij hash table, search the other two tables using i as the key
  - If unique matches are found, check the six angular distances
  - Return the inertial vectors for stars s_i, s_j, s_k, s_r



QUEST

Software Design – Star Tracker

- Quaternion method for finding the optimal rotation between the inertial and body frame
  - Most common attitude determination algorithm because of its robustness and computational simplicity
- Solves Wahba's problem, which looks to find the least squares solution for rotation matrix A

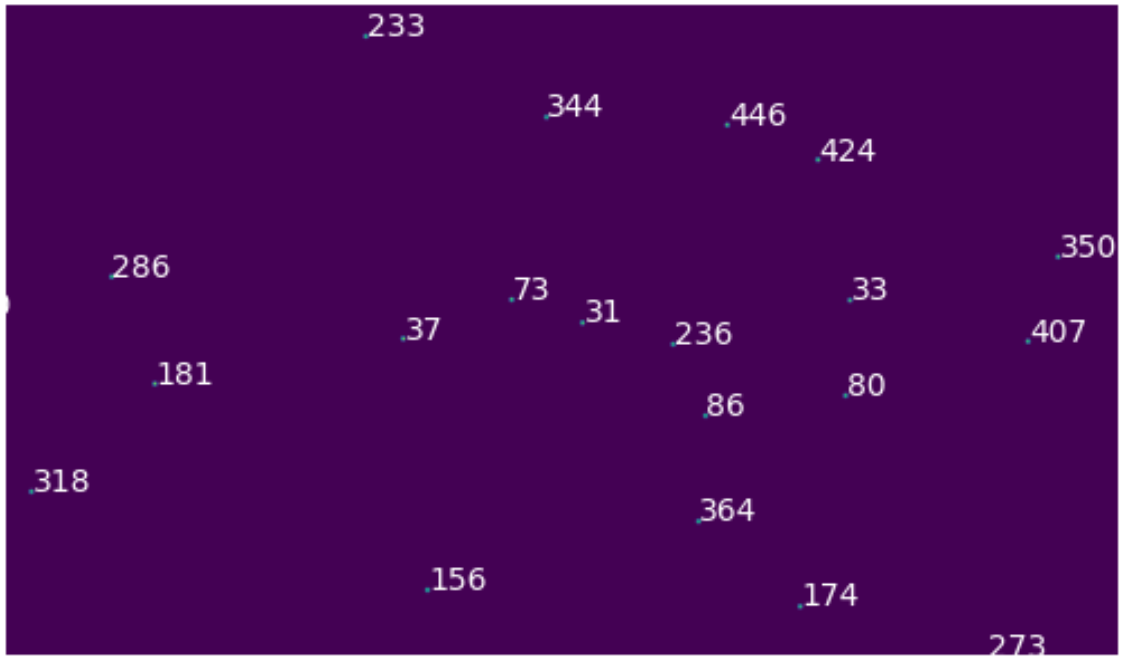
$$\sum_{j=1}^{n} \left\| b_j - A r_j \right\|^2$$

- Accepts the four inertial vectors, r_j, found with the pyramid algorithm and corresponding four body vectors, b_j, found from star detection and centroiding
- Numerically solves a polynomial expression for the maximum eigenvalue of the system, which can be used to construct the quaternion representation of the equation

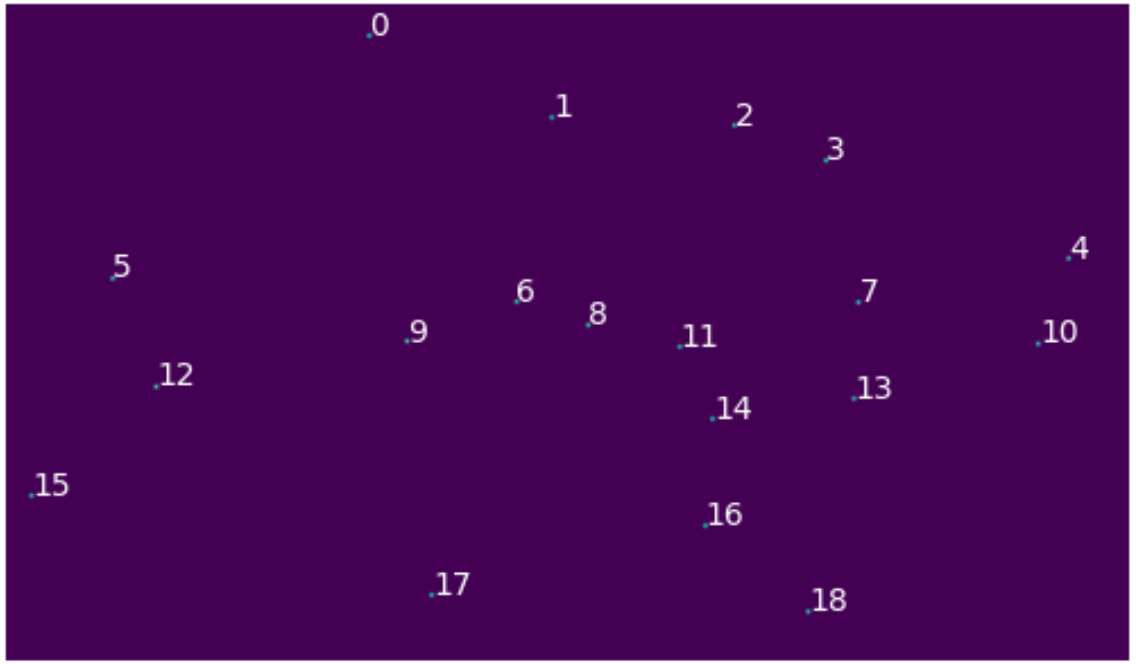
Software Validation

Star Tracker Manual Validation

Software Validation



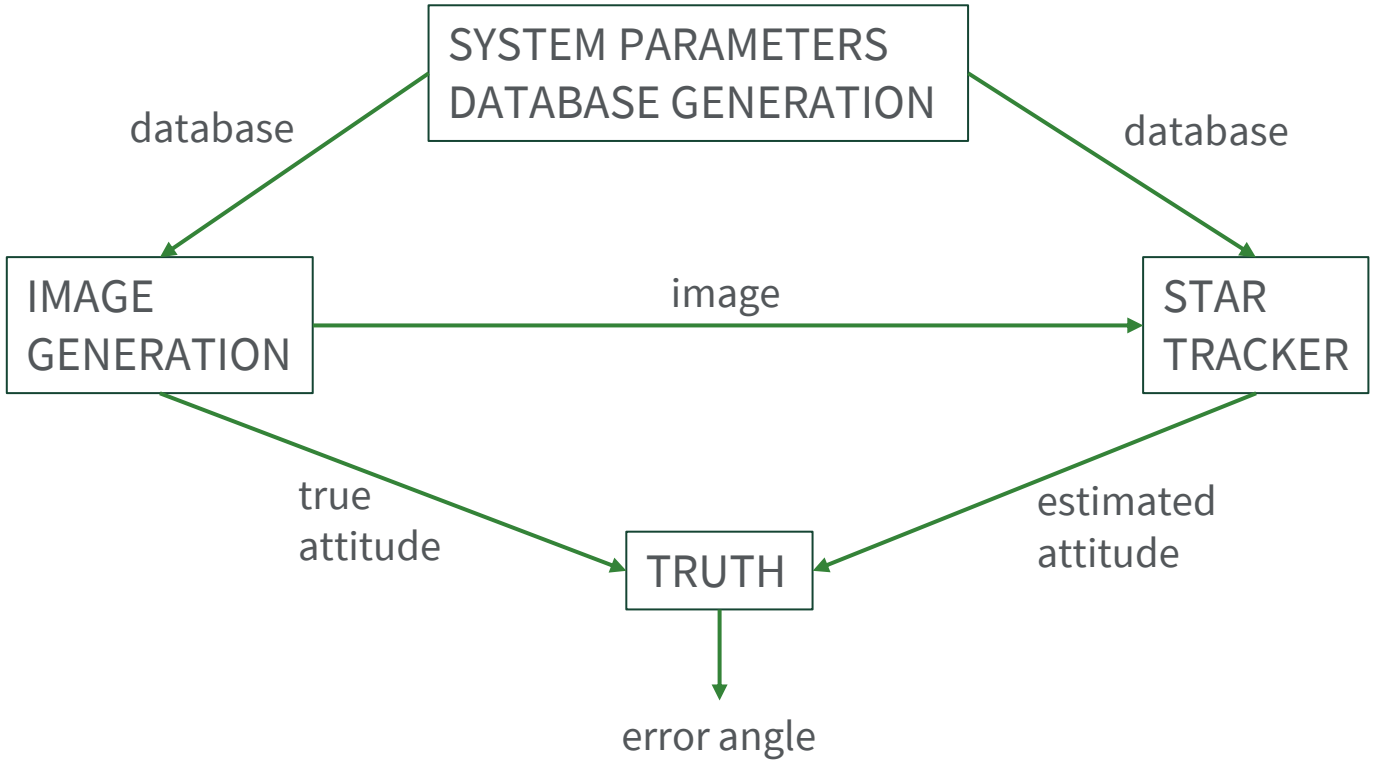
Global labels from image generation star catalog



Local labels from star detection and centroiding

Software Testing Closed Loop

Software Validation



Star Tracker Random Trials

Software Validation

- Error is calculated as the rotation angle associated with the error rotation matrix, given the attitude estimate produced by the star tracker and the true attitude supplied to the image generation algorithm

$$C_{error} = C_{estimated} C_{true}^{T}$$

$$\varphi_{error} = cos^{-1}(\frac{1}{2}(tr(C_{error}) - 1))$$

10,000 Random Trials	
Average Error	0.0489 deg
Standard Deviation	0.0113
Failed Identifications	2

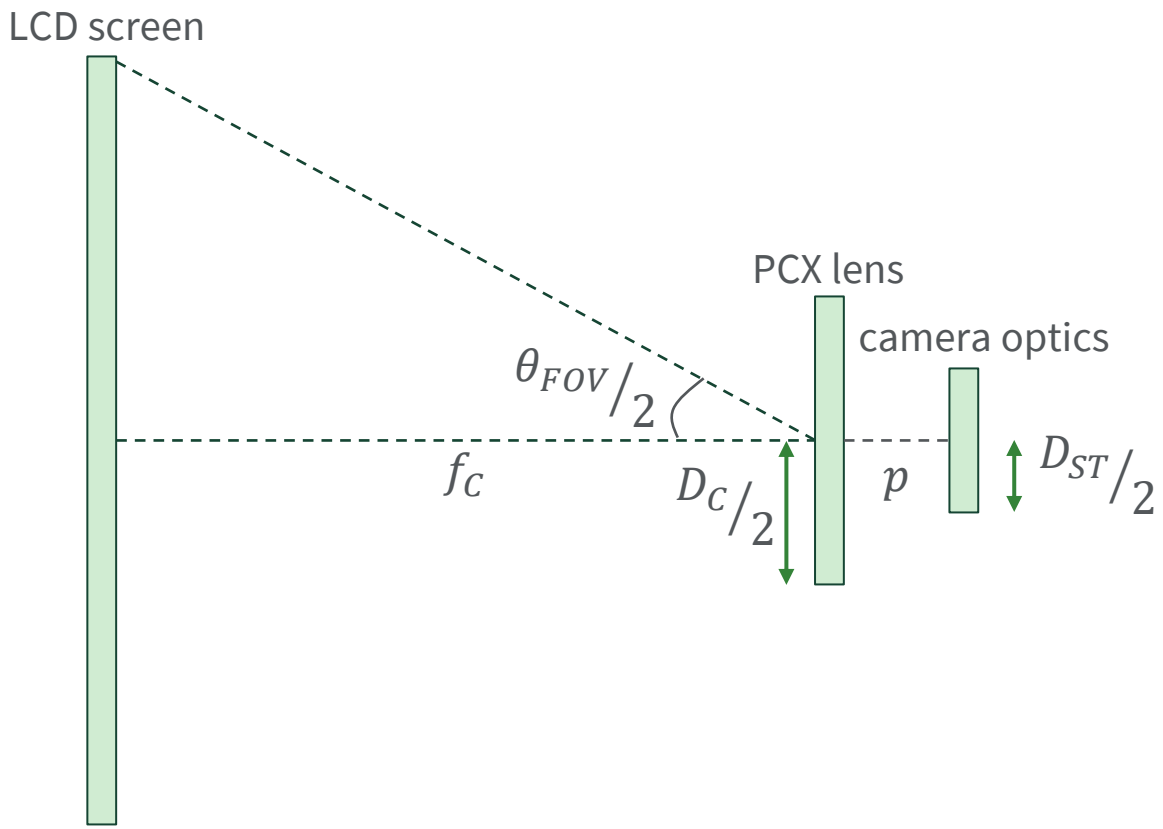
Hardware Design

System Sizing

Hardware Design

- A plano-convex lens is used to parallelize the light from the screen to overcome the effects of parallax
- The lens is sized based on the diameter of the screen and the field of view of the camera

Focal Length [mm]	Screen Diagonal [mm/in]
100	153.4/6.04
115.86 (ideal)	177.8/7.00
120	184.2/7.25



System Setup

Hardware Design

- 3D printed supports for the screen, camera, and lens
- Components are aligned by their centers but can be manually adjusted in the boresight direction

Analysis and Results

System Integration

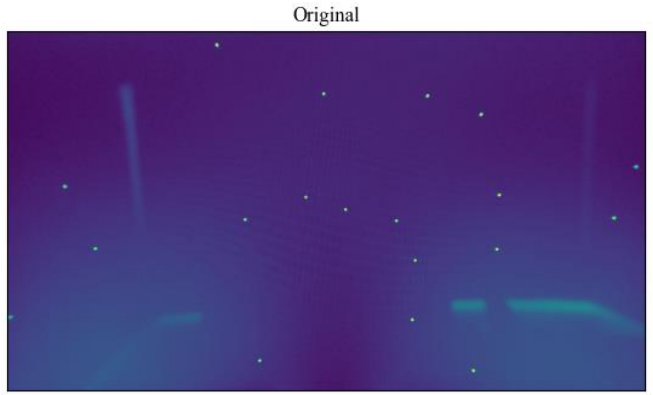
Analysis and Results

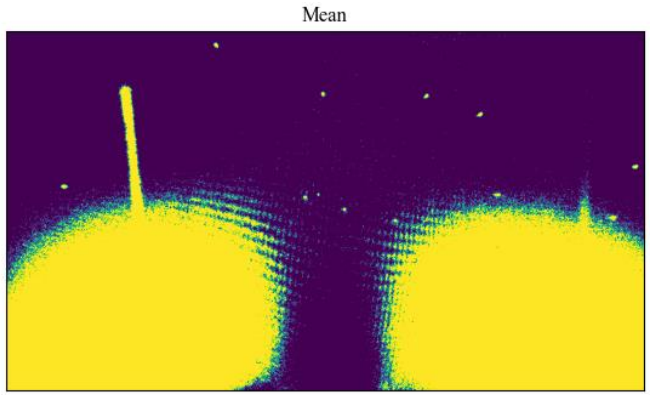
- System placed under a box to operate in semi darkness
- Star field is displayed on the screen and imaged by the camera

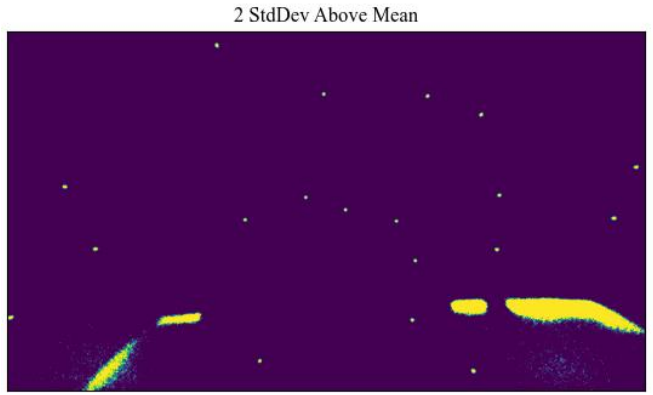
Software Modifications

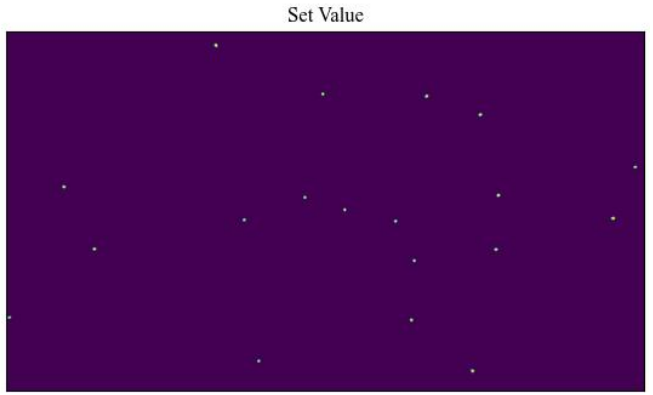
Analysis and Results

- Threshold value changed
- Focal length computed manually
  - Choose two stars a known distance apart and numerically compute the focal length necessary to produce the correct angular distance in the image
  - Alignment errors led to the focal length not being accurate for every star pair









System Validation

Analysis and Results

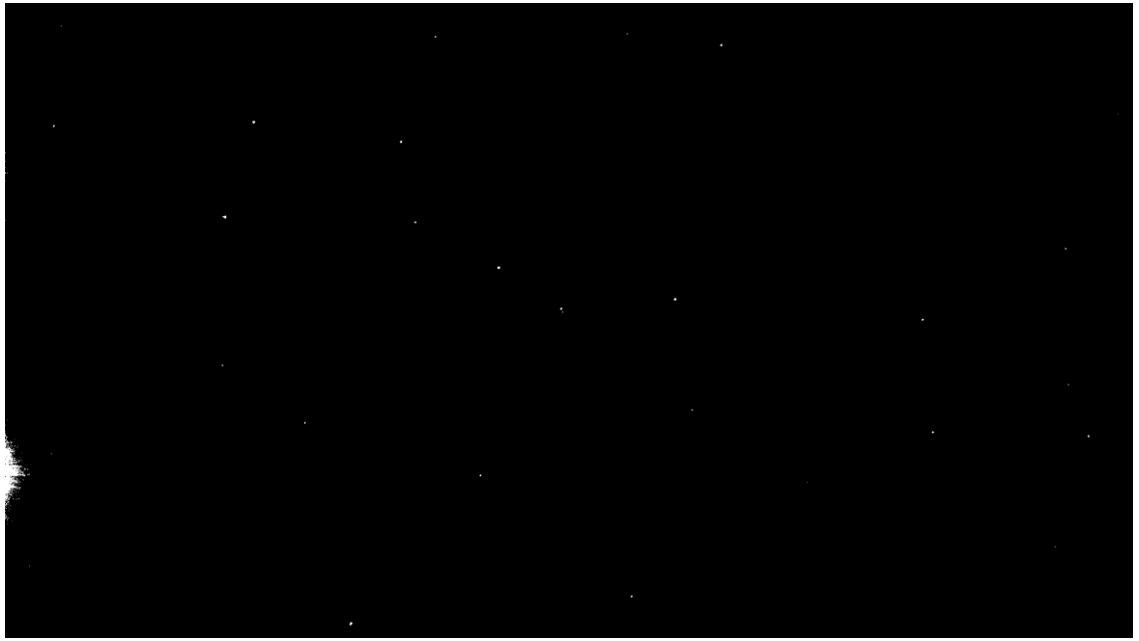
- Large error bounds necessary to account for errors in focal length
- Ran two sets of 100 random trials

	Equal Error Bounds	One-Sided Error Bounds
Failed Identifications	7	28
False Positive Identifications	51	7
Successful Identifications	42	65
Successful Average Error	1.0079 deg	1.2781 deg

Night Sky Test

Analysis and Results

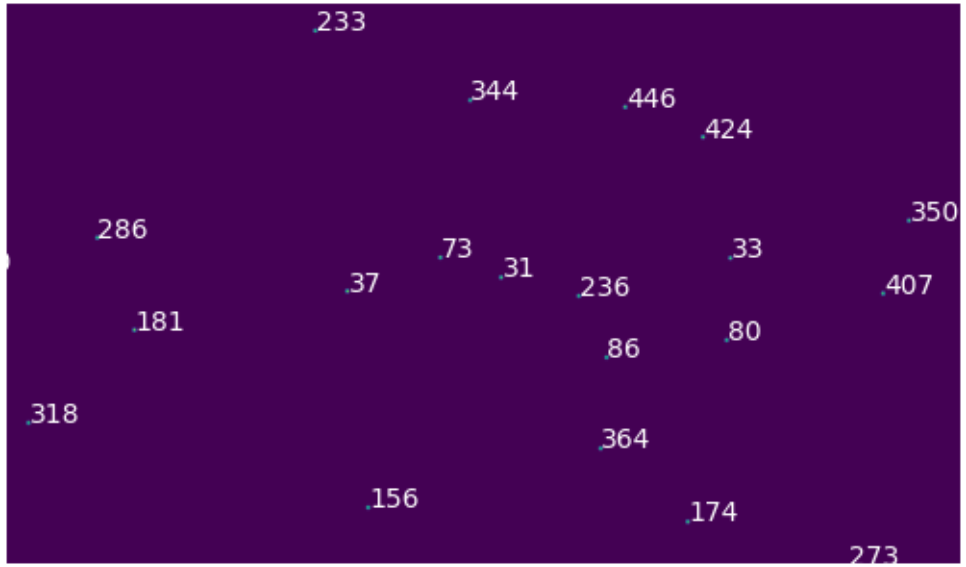
Captured night sky image



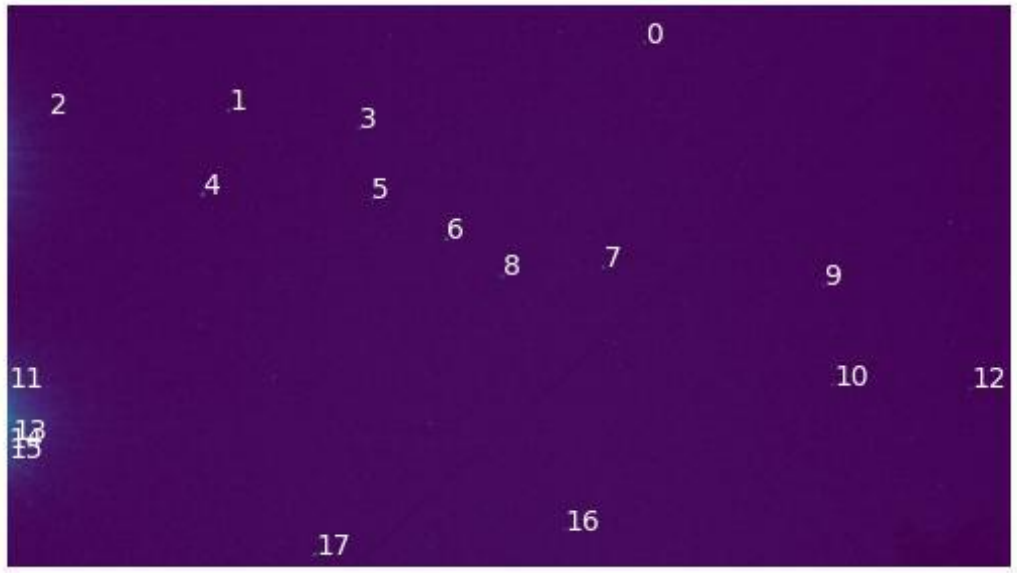
Thresholded night sky image

Night Sky Verification

Analysis and Results



Global labels from image generation star catalog



Local labels of night sky image

Global Label	Local Label
73	8
31	6
236	5
37	7

Matches returned from star identification

Conclusions

Recommendations for Future Work

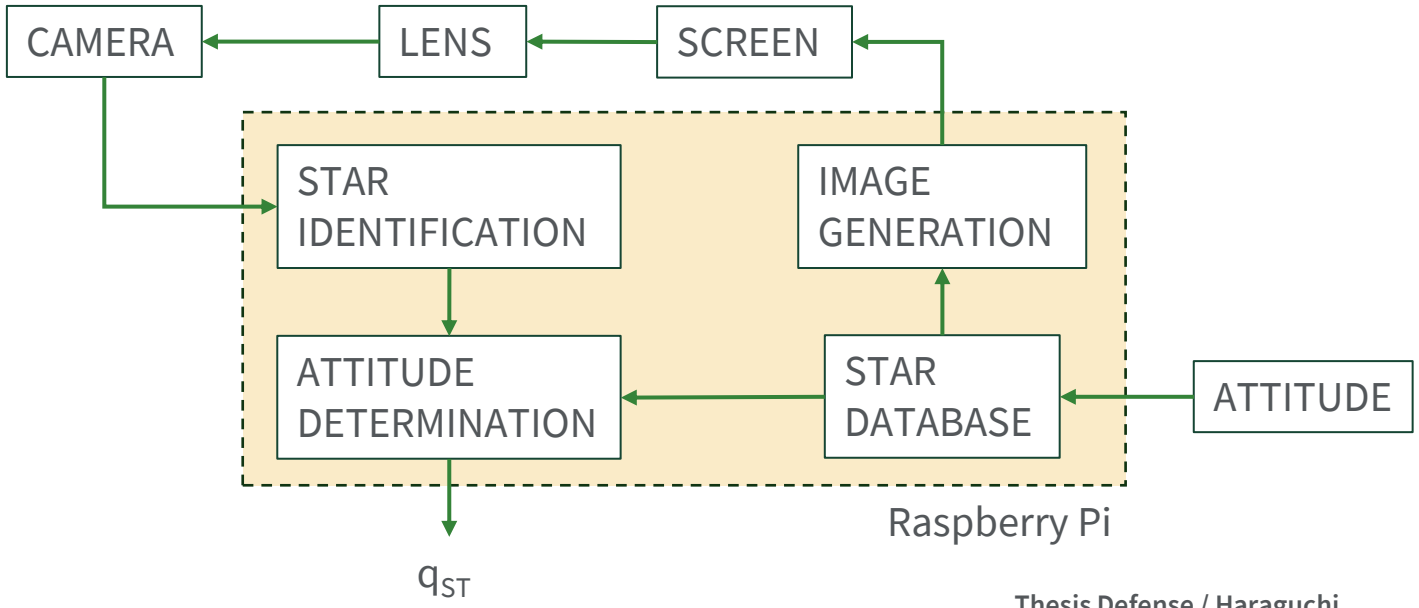
Conclusions

- Improve system operation
  - Optics table for precise alignment of components
  - OLED screen for more accurate black and white representation
  - Camera with better mounting
- Increase system fidelity
  - Add additional noise sources in image generation, such as false stars and dropped stars
  - More accurately represent star magnitude
- Improve system application
  - Integrate with SADS
  - Allow for continuously updating star field images
  - Use a star tracking algorithm (as opposed to lost-in-space)
- Apply to educational purposes
  - Use additional star identification or attitude determination algorithms

Conclusion

Conclusions

- Full software written and tested
- System integrated and shown to accurately determine attitude
- Star tracker algorithms verified with a night sky image



Questions?

References

[1] Tomoyuki Kato. “Modification of the Cal Poly Spacecraft Simulator System forRobust Control Law Verification”. Master’s Thesis. San Luis Obispo, CA: Cal-ifornia Polytechnic State University, San Luis Obispo, June 2014. 109 pp. url:https://digitalcommons.calpoly.edu/theses/1201/ (visited on 10/13/2023).

[2] Ball Aerospace. Star Trackers. url: https://www.ball.com/aerospace/capabilities/technologies-components/star-trackers (visited on 11/24/2023).

[3] Josh Grace et al. “A Low Cost Star Tracker for CubeSat Missions”. In: AIAASCITECH 2022 Forum. AIAA, Dec. 2021, p. 10. doi: https://doi.org/10.2514/6.2022-0520. url: https://arc.aiaa.org/doi/10.2514/6.2022-0520 (visited on10/18/2023).

[4] Christopher Ryan McBryde and E. Glenn Lightsey. “A Star Tracker Design forCubeSats”. In: IEEE Aerospace Conference. Big Sky, Montanna: IEEE, Mar.2012. isbn: 978-1-4577-0557-1. doi: https : / / doi . org / 10 . 1109 / AERO . 2012 .6187242. (Visited on 11/19/2023).

[5] Samuel T. Gutierrez, Cesar I. Fuentes, and Marcos A. Dıaz. “Introducing SOST:An Ultra-Low-Cost Star Tracker Concept Based on a Raspberry Pi and Open-Source Astronomy Software”. In: IEEE Access 8 (Aug. 28, 2020), pp. 166320–166334. issn: 2169-3536. doi: https://doi.org/10.1109/ACCESS.2020.3020048.url: https://ieeexplore.ieee.org/document/9179736 (visited on 11/01/2023).

References

[6] Mehdi Nasiri Sarvi et al. “Design and Implementation of a Star-Tracker forLEO Satellite”. In: Optik 208 (Apr. 2020). doi: https://doi-org.calpoly.idm.oclc . org / 10 . 1016 / j . ijleo . 2020 . 164343. url: https : / / www - sciencedirect -com.calpoly.idm.oclc.org/science/article/pii/S0030402620301777 (visited on11/19/2023).

[7] W. Grunwald and E. D. Swenson, “Design of a Programmable Star Tracker-Based Reference System for a Simulated Spacecraft,” Kissimmee, Florida, Jan. 2015, p. 21. doi: https://doi.org/10.2514/6.2015-0402. [Online]. Available: https://arc. aiaa.org/doi/10.2514/6.2015-0402 (visited on 10/19/2023).

[8] N. Filipe, “Miniaturized Star Tracker Stimulator for Closed-Loop Testing of CubeSats,” Journal of Guidance, Control, and Dynamics, vol. 40, no. 12, Dec. 2017, issn: 0731-5090. doi: https://doi.org/10.2514/1.G002794. [Online]. Available: https://arc.aiaa.org/doi/full/10.2514/1.G002794 (visited on 10/23/2023).

[9] G. Rufino, D. Accardo, M. Grassi, G. Fasano, A. Renga, and U. Tancredi, “RealTime Hardware-in-the-Loop Tests of Star Tracker Algorithms,” International Journal 27 of Aerospace Engineering, vol. 2013, p. 14, Sep. 2013. doi: https://doi.org/10. 1155 / 2013 / 505720. [Online]. Available: https : / / www . hindawi . com / journals / ijae/2013/505720/ (visited on 10/23/2023).

References

[10] F. Wessling and M. V. Does, “Star Field Simulator for the Spacelab Instrument Pointing System Fixed-Head Star Trackers,” in SPIE, Orlando, FL, Jul. 1994. doi: 10 . 1117/12.178926. (visited on 11/14/2023).

[11] V. Nardino, D. Guzzi, M. Burresi, and M. Cecchi, “MINISTAR: A Miniaturized Device for the Test of Star Trackers,” vol. 11180, Chania, Greece: SPIE, Oct. 2018. (visited on 11/23/2023).

[12] G. Zhang, Star Identification, 1st ed. Springer Berlin, Heidelberg, Jan. 2017. [Online]. Available: https : / / link . springer . com / book / 10 . 1007 / 978 - 3 - 662 - 53783 - 1 (visited on 10/19/2023).

[13] Daniele Mortari. “Search-Less Algorithm for Star Pattern Recognition”. In: TheJournal of the Astronautical Sciences 45.2 (June 1997), pp. 179–194. issn: 0021-9142, 2195-0571. doi: 10.1007/BF03546375. url: https://link.springer.com/10.1007/BF03546375 (visited on 04/09/2024).

[14] Daniele Mortari et al. “The Pyramid Star Identification Technique”. In: Nav-igation 51.3 (Sept. 2004), pp. 171–183. issn: 00281522. doi: 10.1002/j.2161-4296.2004.tb00349.x. url: https://onlinelibrary.wiley.com/doi/10.1002/j.2161-4296.2004.tb00349.x (visited on 04/08/2024).