Development of a Low-Cost Multi-Camera Star Tracker for Small Satellites

Hongrui Zhao, Michael Lembeck
University of Illinois at Urbana Champaign
104 S. Wright St., Urbana, Illinois 6180; 217-300-8009
hongrui5@illinois.edu

ABSTRACT

This paper presents a novel small satellite star tracker that uses multiple low-cost cameras to achieve viable attitude determination performance. The theoretical analysis of the star detectability improvement by stacking images from multiple cameras is presented. An image processing algorithm is developed to combine images from multiple cameras with various focal lengths, principal point offsets, distortions, and misalignments. The star tracker also implements other algorithms including the region growing algorithm, the intensity weighted centroid algorithm, the geometric voting algorithm for star identification, and the singular value decomposition algorithm for attitude determination. A star tracker software simulator is used to test the algorithms by generating star images with sensor noises, lens defocusing, and lens distortion. A hardware prototype is assembled, and preliminary night sky testing was conducted to verify the feasibility of the selected hardware. The flight hardware for the star tracker is being developed in the Laboratory for Advanced Space Systems at Illinois (LASSI) at the University of Illinois at Urbana Champaign for future CubeSat missions.

INTRODUCTION

Missions requiring precise attitude determination are starting to take advantage of CubeSat platforms for their low cost and design simplicity.[1,2] One example is the AeroCube-OCSD CubeSat developed by the Aerospace Corporation.[1] AeroCube utilizes a star tracker to provide three-axis determination accuracy better than 0.15° in support of laser communication downlinks. Historically, magnetometers and sun sensors have been used as attitude sensors for CubeSat missions. However, these sensors cannot provide attitude information better than 0.2°, and sun sensors do not work in eclipse, hindering their utility on high performance missions.[3] Star trackers, instead, provide attitude determination with accuracy better than 0.1° throughout the orbit, which is sufficient for the applications such as Earth observation and optical communication.[1,2]

While there are several commercial star trackers that are compatible with the CubeSat formats (such as Berlin Space Technologies ST-200, Sinclair Interplanetary ST-16, and Adcole Maryland Aerospace MAI-SS),[4,5,6] universities and other research institutes often choose to build their own star trackers for their CubeSat missions.[3,7,8] Commercial star trackers with high accuracy and robustness tend to be costly, and lower cost star trackers, mostly developed by universities, lack the desired levels of performance. For example, the star tracker developed by York University has failed its night sky test due to the high noise level in the star images.[7]

This paper focuses on the design of a low-cost star tracker compatible with CubeSat formats by combining images from multiple low-cost commercial cameras instead of using a single expensive low-noise camera.

IMAGE STACKING AND STAR DETECTION

Combing multiple images to remove noise and increase sensitivity of image sensing systems is implemented across various research areas. For fluorescence detection, Joshua et al. captured hundreds of low sensitivity images with a cheap Webcam and combined them using an image stacking algorithm, which results in a sensitivity similar to a high-cost, high-sensitive CCD camera.[9] Roberto and Pieter developed the Dragonfly telephoto array consisting of eight commercial CCD cameras, and the Signal-to-Noise Ratio (SNR) can be improved by increasing the array size.[10] Yuanman et al. have proposed an attitude-correlated frames adding approach to combine images taken by the star tracker at the different time, and the SNR of the star image is between $\sqrt{n}$ and $2\sqrt{n}$ of a single frame where n is the number of frames added.[11] As shown in the above-mentioned examples, the image stacking technique allows one to build a sensitive image sensing system without using high-cost scientific grade camera. Thus, combining images from multiple low-cost cameras is proposed in this paper to maintain a sensitivity similar to a single high-sensitivity camera without the attendant cost.

The sensitivity of a star tracker determines a star's detectability and sets a threshold for the dimmest stars the star tracker can detect. Improving the star detectability of a star tracker improves the attitude estimation accuracy and the robustness of the system. The improvement of the star detectability through the addition of multiple cameras can be quantified to determine the optimal camera array size. The detectability and sensitivity analysis start with the analysis of noise components. There are two types of noise that influence the pixel values of an image sensor: temporal noise (or time variant noise), and spatial noise (or time invariant noise).[7] The effects of spatial noise can be largely removed through calibrations. Thus, temporal

noise dominates the sensitivity of a star tracker. Shot noise, dark current, and read noise are the principal temporal noise sources.[12,13,14] Shot, or photon noise, results from the discrete and random nature of photons. Shot noise follows a Poisson distribution, but can be modeled as zero mean Gaussian noise. Dark current represents the accumulation of electrons in the absence of light. Dark current can be modeled as Gaussian noise with zero mean and a standard deviation varying with integration time and temperature. Read noise combines noise sources from the charge transfer and the read-out amplifiers. Read noise is modeled using a zero mean normal distribution with a constant standard deviation. Considering these temporal noise sources, the expression for SNR is given as

$$SNR = \frac{S}{\sqrt{S + {N_{sensor}}^2}} \tag{1}$$

where S is the electrons generated in the sensor by photons from a target star; N_{sensor} is the standard deviation of the temporal noise measured in electrons. N_{sensor} is given by

$$N_{sensor} = \sqrt{N_{dark} \cdot T + {N_{read}}^2} \tag{2}$$

where $\sqrt{N_{dark} \cdot T}$ is the standard deviation of the dark current in electrons. T is the exposure time in seconds. N_{read} is the standard deviation of read noise in electrons. By stacking n images or calculating the mean of n images, SNR will increase by $\sqrt{n}$ times

$$SNR = \frac{S \cdot n}{\sqrt{(S + {N_{sensor}}^2) \cdot n}} \tag{3}$$

To determine highest star visual magnitude a star tracker can detect based on SNR, it is necessary to calculate how many photons are generated by a target star. The photons collected on the focal plane from a magnitude m star is given as[15]

$$N_{ph} = A_l \cdot T_l \cdot \Delta\mathrm{B} \cdot \Phi_m \cdot T \tag{4}$$

where A_l is the area of light collecting surface in cm^2, $A_l = \frac{\pi}{4} d^2$ and d is the diameter of the camera's aperture; T_l is optical transmittance, usually 0.6~0.8. T_l represents the fraction of light of a specific wavelength which passes through the lens; ΔB is the bandwidth of the lens, usually 3000-6000 Å (Angstrom); T is the exposure time in seconds; Φ_m is the luminous flux of m visual magnitude (unit $\frac{photons}{cm^2 \cdot Å \cdot s}$) , given by[15]

$$\Phi_m = 10^{(15-2m)/5} \tag{5}$$

Using Eq(4), the signal S (electrons per pixel) received from a magnitude m star can be expressed as[15]

$$S_m = N_{ph} \cdot QE \cdot K_{fill} \cdot K \tag{6}$$

where QE is the general quantum efficiency, usually ranged from 0.3 to 0.7; K_{fill} is the fill factor, the ratio of a pixel's light sensitive area to its total area, ranged from 0.3 to 0.7; K is the energy concentrative degree. The value of K is about 0.25~0.4 when the starlight is spread over a 3x3 pixel grid.

An SNR threshold V_{th} is defined to determine if an SNR value is sufficient for detecting stars. According to Rose criterion, SNR > 5 can guarantee the certain detection of a signal.[16] Based on Eq(3), SNR should fulfill the expression given below to detect a star with magnitude m when N images are stacked

$$SNR = \frac{S_m \cdot n}{\sqrt{({S_m} + {N_{sensor}}^2) \cdot n}} \geq V_{th} \tag{7}$$

Eq(7) can be rewritten as

$$S_m \geq \frac{{V_{th}}^2 + \sqrt{{V_{th}}^4 + 4{V_{th}}^2 \cdot {N_{sensor}}^2 \cdot n}}{2n} \tag{8}$$

Substituting Eqs(2)(6) into Eq(8) gives the highest star visual magnitude a star tracker can detect

$$m_x \leq 7.5 - 2.5 log_{10}\left(\frac{{V_{th}}^2 + \sqrt{{V_{th}}^4 + 4{V_{th}}^2 \cdot (N_{dark} \cdot T + {N_{read}}^2) \cdot n}}{2n \cdot A_l \cdot T_l \cdot \Delta\mathrm{B} \cdot T \cdot QE \cdot K_{fill} \cdot K}\right) \tag{9}$$

Eq(9) can be used to plot the star detectability as a function of the number of cameras added, as shown in Figure 1. The parameters are set to represent MT9V022 image sensor and LS-12020 lens used for the hardware prototype, where $N_{dark} = 25\ e^-$, $N_{read} = 2.6\ e^-$, $A_l = \frac{\pi}{16}\ cm^2$, $T_l = 0.88$, $\Delta\mathrm{B} = 3000$ Å, $T = 0.1\ s$, $QE = 0.4$, $K_{fill} = 0.3$, $K = 0.25$. The circular field of view (FOV) of LS-12020 lens paired with MT9V022 image sensor is 13.69 °. The star visual magnitude higher than 5 is required to achieve sky coverage above 90% with

this circular FOV.[3] Thus, there shall be more than three cameras in star tracker system to achieve adequate sky coverage. However, this analysis doesn't consider the impacts of errors in image processing algorithms, which would hinder the improvement of star detectability. That impact is examined in the next section.

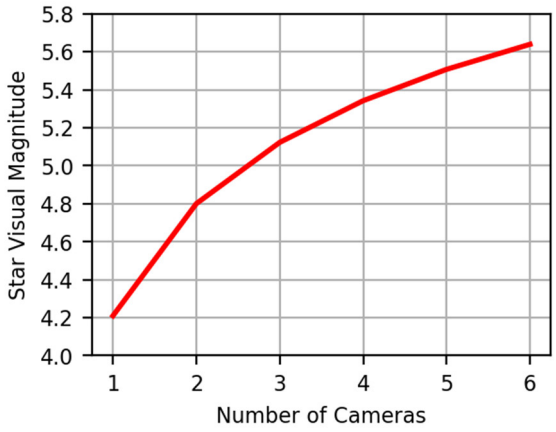

Figure 1: Star Detectability as a Function of the Number of Cameras Added

ALGORITHMS

The software system for our multi-camera star tracker consists of image processing algorithm, centroiding algorithm, star identification algorithm, and attitude determination algorithm. The image processing algorithm corrects the lens distortions and combines images from multiple cameras by applying projective transformation. The centroid algorithm applies the region growing algorithm to find connected regions representing detected stars, and implements the intensity weighted centroid algorithm to find the coordinates of the stars on the focal plane. The star identification algorithm constructs vectors using camera parameters and stars' coordinates, and identifies them in a star catalog using a geometric voting algorithm. Finally, by using the identified stars, the singular value decomposition algorithm is implemented to obtain the attitude quaternion of the star tracker

Image Processing

Due to the cameras' misalignments and different intrinsic parameters, images are processed and transformed before stacking. One camera is selected as the reference camera and its star images are defined as the reference images. The images from non-reference cameras are then mapped to the reference images. The reference images and non-reference images are processed differently.

The applicable coordinate frames are shown in the Figure 2, where $(u,\ v)$ is the star image centroid coordinate under the focal plane frame and $(u_0,\ v_0)$ is the coordinate of the principal point under the focal plane frame (the intersection point of the boresight and the focal plane). The star tracker body frame XYZ has its origin located at the vertex of the optical system (the pinhole camera model is used) and its z-axis aligned with the star tracker's boresight. The focal plane frame UV is a 2D coordinate frame, originating at the corner of the focal plane.

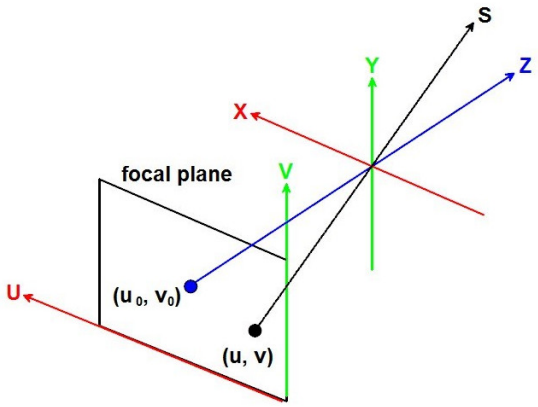

Figure 2: Coordinate Frames

For the reference images, the distortion correction is applied. We need a mapping from the undistorted coordinates to the distorted position to obtain the image intensity at the corrected position, which follows:[17]

(1) A raw distorted image of the reference camera is obtained and saved into a matrix A.

(2) An empty matrix B is created to represent the undistorted image.

(3) The coordinate (u_c, v_c) of a pixel from j row k column of the matrix B relative to the principal point is calculated.

(4) The UV frame coordinate (u_d, v_d) of the corresponding pixel on the matrix A is calculated using the tangential and radial distortions model.

(5) The value for the j row k column pixel on the matrix B is computed by interpolation of pixels on the matrix A.

The coordinate of a pixel from j row k column of the matrix B relative to the principal point is calculated by

$$(u_c, v_c) = (k + 0.5, j + 0{,}5) \cdot p - \ (u_0, v_0) \qquad (10)$$

where p is the pixel width/length of the image sensor, (u_0, v_0) is the principal point of the reference camera. (u_c, v_c) is the coordinate of an undistorted pixel, and we can calculate the UV frame coordinate of the corresponding distorted pixel by applying radial and tangential distortions.[18]

$$(u_d, v_d) = (u_c, v_c) + (\delta u^r, \delta v^r) + (\delta u^t, \delta v^t) + (u_0, v_0) \quad (11)$$

The radial distortion term is given by

$$(\delta u^r, \delta v^r) = (u_c, v_c) \cdot (k_1 \cdot r^2 + k_2 \cdot r^4) \quad (12)$$

where k_1 and k_2 are the coefficients for radial distortion, and $r = \sqrt{u_c^2 + v_c^2}$. The tangential distortion term is given by

$$\delta u^t = 2 \cdot p_1 \cdot u_c \cdot v_c + p_2 \cdot (r^2 + 2 \cdot u_c^2) \quad (13)$$

$$\delta v^t = 2 \cdot p_2 \cdot u_c \cdot v_c + p_1 \cdot (r^2 + 2 \cdot v_c^2) \quad (14)$$

where p_1 and p_2 are the coefficients for tangential distortion

The distortion computation usually results in a non-integral address result lying between known pixel values of the distorted image (the matrix A).[17, 19] Thus, it is necessary to estimate the unknown pixel amplitude from its neighbors using image interpolation.[17, 19] The nearest neighbor interpolation is the most basic interpolation algorithm, where each pixel is given the value of the sample closet to it.[20] The nearest neighbor interpolation requires the least processing time and preserves some high frequency signals, but cannot preserve subpixel image realtions.[20,21] Bilinear interpolation considers the closest 2x2 neighborhood of known pixel values surrounding the unknown pixel, and then takes a weighted average of these 4 pixels to arrive at its final interpolated value. [22] This results in much smoother looking images than nearest neighbor method, but also results in strong attenuation of high frequencies [21,22] Nobach et al. has implemented the Gaussian interpolation, which can achieve sub-pixel precision, to process the images from particle image velocimetry and particle tracking velocimetry.[23] Star signals usually spread over a small pixel area, and have much higher pixel values in comparison with the background noises. Thus, they are considered high frequency signals because the pixel values are rapidly changing in space. Also, having a fast processing speed is crucial to a star tracker's robustness against high rotation rates. As a result, the nearest neighbor interpolation is selected for implementation here. After all pixels in the matrix B are assigned with values through interpolation, the image processing of the reference image is complete, and the matrix B is used to stack with other processed images.

There are misalignments between the reference camera and the non-reference cameras. As a result, in addition to the distortion correction, the projective transformation is applied for the non-reference images, as follows:

(1) A raw distorted and misaligned image of a non-reference camera *i* is obtained and saved into a matrix C_i.

(2) An empty matrix D_i is created to represent the aligned and undistorted image.

(3) The UV frame coordinate (u_1, v_1) of a pixel from j row k column of the matrix D_i is calculated.

(4) The coordinate (u_2, v_2) of the corresponding misaligned but undistorted pixel relative to the principal point is calculated using the projective transformation.

(5) The UV frame coordinate (u_d, v_d) of the corresponding misaligned and distorted pixel on the matrix C_i is calculated using the tangential and radial distortions model.

(6) The value for the *j* row *k* column pixel on the matrix C_i is computed by interpolation of pixels on the matrix D_i

The UV frame coordinate (u_1, v_1) of a pixel from j row k column of the matrix D_i is calculated by

$$(u_1, v_1) = (k + 0.5, j + 0{,}5) \cdot p \quad (15)$$

Since all stars can be considered infinitely far away from the star tracker, translations between reference camera and non-reference cameras can be ignored. When only the rotations between the cameras are considered, point correspondences between the images are related by an explicit one-to-one projective transformation given by[24]

$$u_2 = -f_2 \frac{r_{11}(u_0 - u_1) + r_{12}(v_0 - v_1) + r_{13} \cdot f_1}{r_{31}(u_0 - u_1) + r_{32}(v_0 - v_1) + r_{33} \cdot f_1} \quad (16)$$

$$v_2 = -f_2 \frac{r_{21}(u_0 - u_1) + r_{22}(v_0 - v_1) + r_{23} \cdot f_1}{r_{31}(u_0 - u_1) + r_{32}(v_0 - v_1) + r_{33} \cdot f_1} \quad (17)$$

Where f_2 is the focal length of the non-reference camera, f_1 is the focal length of the reference camera, $(u_0,\ v_0)$ is the principal point of the reference camera. The (u_2, v_2) is a coordinate relative to the principal point of the non-

reference camera. The rotation matrix from the reference camera to the non-reference camera is given by

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (18)$$

After the projective transformation, (u_d, v_d) can be calculated by replacing (u_c, v_c) in Eq(11)~(14) with (u_2, v_2). When calculating Eq(11) for a non-reference camera, the principal point of the non-reference should be used instead of $(u_0,\ v_0)$. After all pixels in the matrix D_i are assigned with values through interpolation, the image processing of the non-reference image is complete. The image stacking is simply averaging the pixel values from matrix B and D_i (i = 1, 2, 3, …).

An example of the image processing results is shown in Figure 3. From the top to the bottom, the first image is the raw image taken by the reference camera, the second image is the raw image taken by the non-reference camera, and the third image is the final stacked image. The X and Y axes are the pixel coordinate, and the grey scale bars represent the pixel values. All images are generated by the star image simulator. The locations of the stars on the second image are different from that of the first image due to the rotation between two cameras. After being processed, the stars on the second image are mapped to the first image, as can be seen in the third image. Also, the effects of distortion correction can be seen on the third image.

Other Algorithms

Existing algorithms found in the literature are implemented here for further processing. After the stacked image is obtained, the region growing algorithm is applied.[25] The region growing algorithm scans over the entire stacked image searching for pixels brighter than a pre-determined value. Once a pixel is found, the algorithm determines the area connected to this given pixel by checking its eight neighboring pixels for values higher than the threshold. The same region growing process is also applied to the neighboring pixels. After a region is identified, the number of contiguous lit pixels in this region, and the integrated intensity of all contiguous lit pixels in this region are calculated.[5] Both of the values need to be higher than corresponding thresholds for the region to be considered as a region generated by a star. After a star region is identified, the centroid of the region is calculated by the intensity weighted method.[12] In the geometric voting algorithm, a pair of stars in the image gets a vote from a pair of stars in the catalogue if the angular distance between the stars of both pairs is similar. [26] This algorithm is selected for its robustness and simplicity. After the star identification, the singular value decomposition (SVD) algorithm is implemented to obtain the attitude quaternion.[27] The SVD method is slower than QUEST, FOAM, ESOQ, and ESOQ2 algorithms, but is numerically more robust.[28]

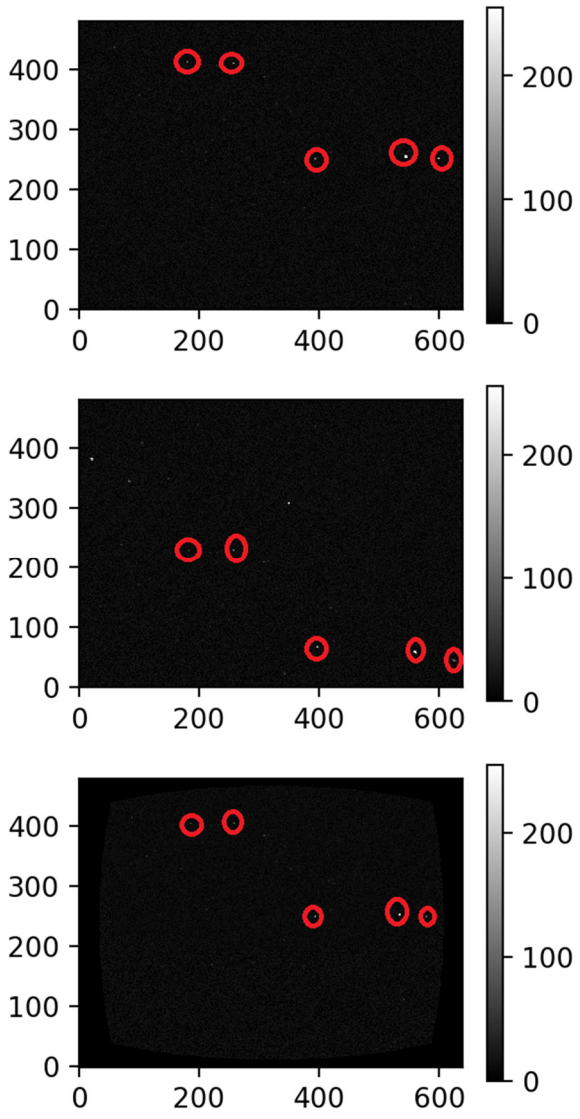

Figure 3: An Example of Image Processing Results

SOFTTWARE SIMULATION

A star tracker software simulator is designed to test the algorithms by generating star images with sensor noises, lens defocusing, and lens distortion. The Hipparcos catalog from the European Space Agency (ESA) is used to obtain stars coordinates and magnitude.[29] A star's image is defocused by the optics and spreads over several pixels. A point spread function (PSF) is selected to model the defocusing of the star light. PSF of an ideal lens can be described by an Airy Disk.[31] However, Gaussian distribution can also be used as a good approximation of the actual PSF.[30] There are two typical

types of PSF models : IPSF and SPSF.[31] The SPSF is usually preferred for its simplified computation as compared with integral IPSF. The SPSF used here is given by[32]

$$g(i,j) = \frac{1}{2\pi\sigma^2} exp\left[-\frac{(i-x_c)^2+(j-y_c)^2}{2\sigma^2}\right] \quad (19)$$

where (i,j) is the coordinate of an arbitrary pixel, $g(i,j)$ is the probability a photon will fall into pixel (i,j), (x_c, y_c) is the coordinate of the center of a star image, σ is the Gaussian radius describing the size of the defocused image. The other photons and electrons calculations follow Eqs(1)~(9). The quantization is applied to obtain 0~255 (8 bits analog-to-digital converter) digital counts from the electrons.

In the software simulator, the lens distortion is applied to an undistorted image. Therefore, the inverse distortion mapping, which maps a distorted pixel to an undistorted pixel, is needed. If both radial and tangential distortion components are considered, we know that there is no analytic solution to the inverse mapping.[18] Pierre and Julien proposed an iterative approach to approximate the inverse mapping.[33] Janne and Olli used a polynomial approximation method to solve the back-projection problem.[18] The residual errors of both methods are calculated and shown in Figure 4 and 5. The color bars represent the residual errors in pixels. The iterative method is a more accurate approximation. However, the polynomial method has much faster processing speed, and its maximum residual error is less than 0.1 pixel. As a result, the polynomial method is selected here to simulate lens distortion.

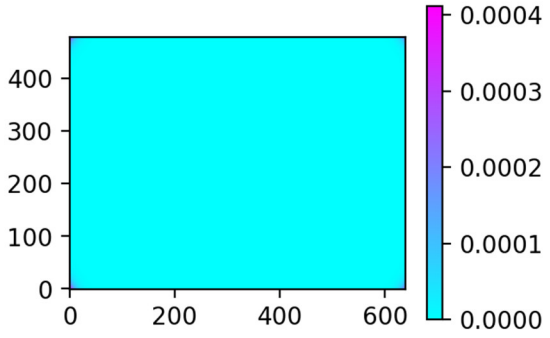

Figure 4: Residual Error of the Iterative Method

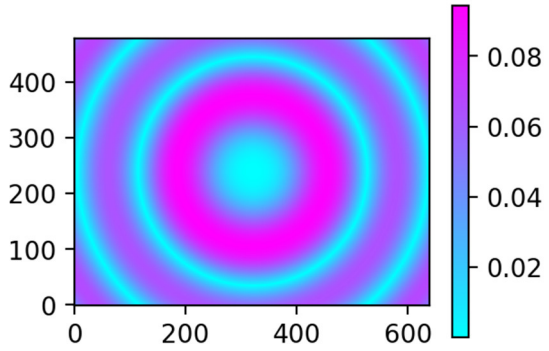

Figure 5: Residual Error of the Polynomial Method

Simulations at thirty different attitude orientations are run for one-camera, two-camera, three-camera, and four-camera star tracker configurations. The number of stars detected and the attitude estimation errors are recorded, as shown in Figure 6 and Figure 7. The average attitude estimation errors are 0.0204° for one camera, 0.0198° for three cameras, and 0.0188° for four cameras. Since there is a false attitude estimation for the two-camera configuration, its attitude estimation error isn't presented. The average numbers of stars detected are 9.8 for one camera, 7.0 for two cameras, 7.1 for three cameras, and 6.7 for four cameras. Interestingly, the number of stars detected in the configurations with more than one camera are slightly lower than the number of stars detected in one camera configuration, which conflicts with the theoretical analysis presented in the previous section. One possible explanation is that some stars in the reference image would not fall into other non-reference images due to misalignments and different intrinsic parameters. As a result, although most of the star signals are enhanced through stacking, the stars that only show in the reference image could become dimmer and fall below the detection threshold. Nevertheless, the attitude estimation accuracy is improved when there are three or more cameras, as the noise level is reduced by the image stacking. Two camera configuration has the worst performance because the errors brought by the image processing cancel out the improvement in noise reduction. These simulation results show the negative impacts of the image processing on the performance of the star tracker, but also show how the estimation accuracy is improved by the image stacking.

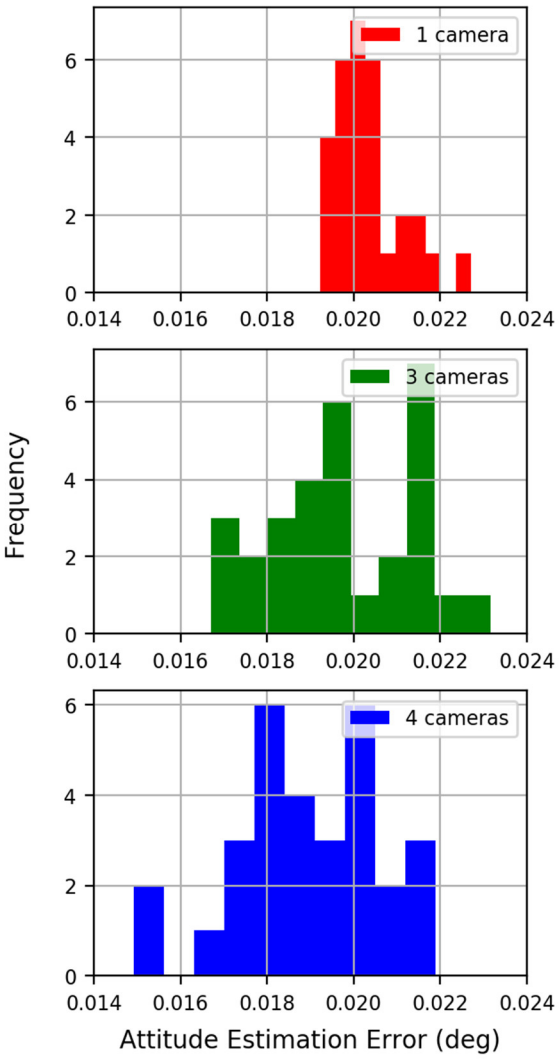

Figure 6: Attitude Estimation Error Results for Different Camera Configurations

HARDWARE VERIFICATION

A hardware prototype has been assembled to verify the performance of the multi-camera star tracker in a more realistic setting, as shown in Figure 8. Two MT9V022 monochrome image sensors are used to prove the concept of stacking, two more sensors will be added for testing in the future. The MT9V022 image sensor was selected for the star tracker of the AeroCube CubeSat mission.[1] Its performance has been verified on orbit, and it only costs 50 USD. The LS-12020 lens is also used, which has no IR filter. A Raspberry Pi 3 is selected as the processor to obtain images from the cameras and run the star tracker algorithms. The Pi 3 is also remotely controlled through a laptop. A DXL360S two-axis digital inclinometer is implemented to point the star tracker towards zenith.

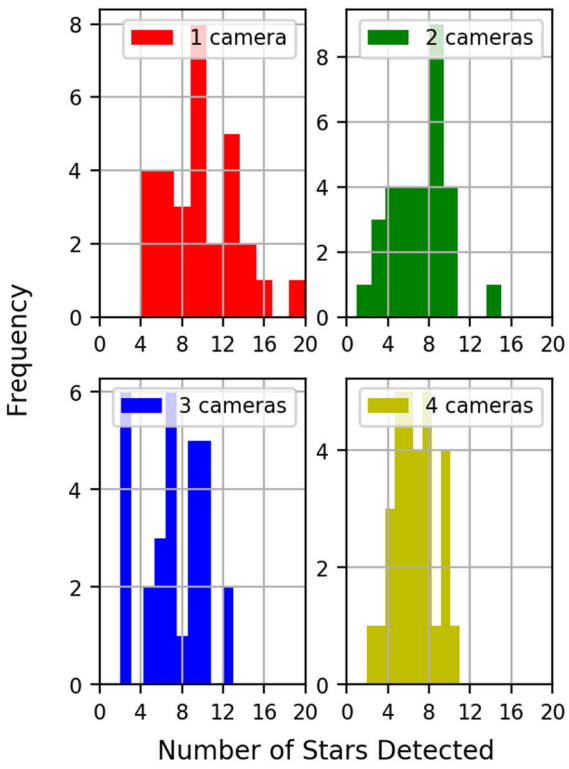

Figure 7: Number of Stars Detected Results for Different Camera Configurations

Star tracker performance verification can be challenging on the ground. The arcsecond accuracy level test equipment is needed for any indoor testing, which is costly and requires clean room environments for successful implementation. To control verification costs, night sky testing provides the most cost-effective environment for estimating the relative accuracy of the star tracker. While subject to weather constraints, another challenge of any night sky test is to verify the attitude output obtained during the test. Malak et al. used the estimated attitude from a star tracker to estimate the local coordinates of the camera on the Earth, which can be used to compare with latitude and longitude from a GPS to qualitatively verify the performance of the star tracker.[34] The attitude output of the star tracker can also be transformed into celestial coordinates: declination, roll angle, and right ascension. It is possible to estimate the relative accuracy of a star tracker as the statistical fluctuations of the declination and roll angle or the right ascension when the sidereal rate has been subtracted.[12] The night sky testing to date only verified the sensitivity of the hardware prototype as a proof-of-concept demonstration. As shown in Figure 9, Orion is clearly visible in the image. The centroid algorithm was applied to the image, and most of the stars can be detected as shown in Figure 10. This night sky test shows the image

sensor selected for the prototype has adequate sensitivity. More tests are planned to further verify the accuracy performance of the star tracker.

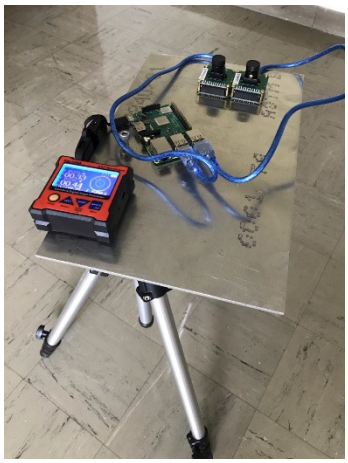

Figure 8: Hardware Protype

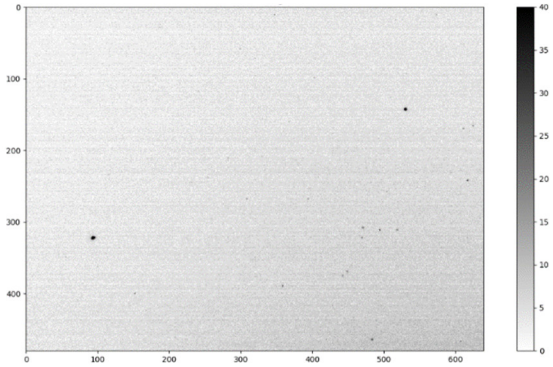

Figure 9: An Image Taken During the Night Sky Test

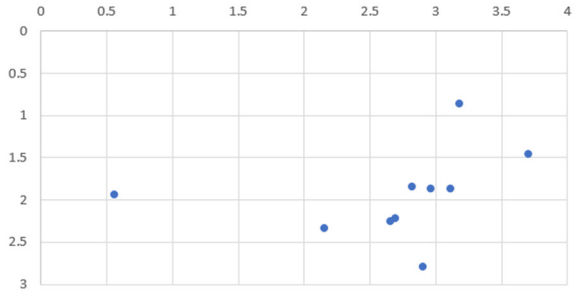

Figure 10: Centroid Results for the Night Sky Test Image

CONCLUSION

The concept of a multi-camera star tracker is proposed to improve the sensitivity while reducing integrated system assembly and test costs. The feasibility of the concept has been verified through theoretical analysis and numerical simulation. A hardware prototype has been assembled, and preliminary night sky testing was conducted to verify the feasibility of the selected hardware. For the future work, additional analysis and testing is planned to complete the verification of the low -cost multi-camera star tracker concept. A flight qualified system will be assembled and flown on a future CubeSat mission to complete the verification program. If successful, CubeSats will be able to take advantage of a low-cost sensor for achieving higher levels of pointing accuracy in future payload applications.

References

1. Janson, S. W., and Welle, R. P., "The NASA Optical Communication and Sensor Demonstration Program: An Update," Proceedings of the 28th Annual AIAA/USU Conference on Small Satellites, Next on the Pad, SSC14-VI, vol. 1, 2014.
2. Ofodile et al., "ESTCube-2 Attitude Determination and Control: Step Towards Interplanetary CubeSats," 2019 IEEE Aerospace Conference, Big Sky, MT, USA, 2019, pp. 1-12.
3. Erlank, A.O., "Development of CubeStar, A CubeSat-Compatible Star Tracker," Master's thesis, Stellenbosch University, 2013.
4. Segert et al., "Development of the Pico Star Tracker ST-200-Design Challenges and Road Ahead," Proceedings of the AIAA/USU Conference on Small Satellites, 2011.
5. Dzamba et al., "Success by 1000 Improvements: Flight Qualification of the ST-16 Star Tracker," Proceedings from 28th Annual AIAA/USU Conference on Small Satellites, Logan, Utah, 2014.
6. Adcole Maryland Aerospace Corporation, "MAI-SS Specification," 2017.
7. Lohmann, A.J., "Star Imager for Nanosatellite Applications," Master's thesis, York University, 2018.
8. McBryde, C.R., "A Star Tracker Design for CubeSats," Master's thesis, the University of Texas at Austin, 2012.
9. Balsam et al., "Image Stacking Approach to Increase Sensitivity of Fluorescence Detection using a Low-Cost Complementary Metal-Oxide-Semiconductor (CMOS) Webcam," Sensors and Actuators B Chemical, vol. 171-172, August–September 2012
10. Abraham, R., and Dokkum, R., "Ultra-Low Surface Brightness Imaging with the Dragonfly Telephoto Array," Publications of the Astronomical Society of the Pacific, 2014.

11. Yuanman et al., “Attitude-Correlated Frames Adding Approach to Improve Signal-to-Noise Ratio of Star Image for Star Tracker,” Optics Express, vol. 27, May 2019.
12. Liebe, C.C., “Accuracy Performance of Star Trackers - a Tutorial,” in IEEE Transactions on Aerospace and Electronic Systems, vol. 38, No. 2, pp. 587-599, April 2002.
13. Markley, L., and Crassidis, J., “Fundamentals of Spacecraft Attitude Determination and Control,” Springer-Verlag New York, 2014.
14. Tjorven et al., “An Accurate and Efficient Gaussian Fit Centroiding Algorithm for Star Trackers,” AAS/AIAA Space Flight Mechanics Meeting, Kauai, HI, 2013.
15. Li et al., “Study on the Detection Sensitivity of APS Sar Tacker,” Proceedings of SPIE - The International Society for Optical Engineering, 2007.
16. Christoph et al., “Field Guide to Astronomical Instrumentation”, SPIE Press Spi edition, 2015.
17. Junhee et al., “Lens Distortion Correction Using Ideal Image Coordinates,” in IEEE Transactions on Consumer Electronics, vol. 55, No. 3, pp. 987-991, August 2009.
18. Heikkila, J., and Silven, O., “A Four-Step Camera Calibration Procedure with Implicit Image Correction,” Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Juan, Puerto Rico, USA, 1997.
19. Pratt, W.K., “Introduction to Digital Image Processing,” CRC Press, Boca Raton, 2013.
20. Anthony et al., “Comparison of Interpolating Methods for Image Resampling,” in IEEE Transactions on Medical Imaging, vol. 2, No. 1, pp. 31-39, March 1983.
21. Philippe et al., “Image Interpolation and Resampling,” Handbook of Medical Image Processing and Analysis, 2000.
22. Cambridge in Colour, “Digital image interpolation” https://www.cambridgeincolour.com/tutorials/image-interpolation.htm
23. Nobach, H., Damaschke, N., and Tropea, C., “High-Precision Sub-Pixel Interpolation in Particle Image Velocimetry Image Processing,” Experiments in Fluids, vol. 39, No.2, pp.299-304, 2005.
24. Aghajan, H., and Cavallaro, A., “Multi-Camera Networks Principles and Applications,” Academic Press, 2009.
25. Mayuresh et al., “A Software Package for Evaluating the Performance of a Star Sensor Operation,” Experimental Astronomy, vol. 43, pp. 99-117, 2017.
26. Michael et al., “Geometric Voting Algorithm for Star Trackers,” in IEEE Transactions on Aerospace and Electronic Systems, vol. 44, No. 2, pp. 441-456, April 2008.
27. Markley, L., “Attitude Determination Using Vector Observations and the Singular Value Decomposition,” Journal of the Astronautical Sciences, vol. 38, 1987.
28. Yang, C., and Malcolm, D. S., “Improvement to the Implementation of the QUEST Algorithm,” Journal of Guidance, Control, and Dynamics, vol. 37, No. 1, pp. 301-305, 2014.
29. European Space Agency, “The Hipparcos and Tycho Catalogues,” ESA SP-1200, 1997.
30. Delabie, T., Schutter, J.D., and Vandenbussche, B., “An Accurate and Efficient Gaussian Fit Centroiding Algorithm for Star Trackers,” Journal of the Astronautical Sciences, vol. 61, pp. 60–84, 2014.
31. Wei et al., “A Systematic Error Compensation Method Based on an Optimized Extreme Learning Machine for Star Sensor Image Centroid Estimation,” Applied Sciences, vol. 9, 2019.
32. Haibo et al., “Focal Adjustment for Star Tracker”, Defence Science Journal, vol. 60, No. 6, 2010.
33. Drap, P., and Lefèvre, J., “An Exact Formula for Calculating Inverse Radial Lens Distortions,” Sensors, vol. 16, No. 6, pp. 807, 2016.
34. Samaan, M.A., Mortari, D., and Junkins, J.L., “Compass Star Tracker for GPS-ike Applications,” in IEEE Transactions on Aerospace and Electronic Systems, vol. 44, no. 4, pp. 1629-1634, Oct. 2008.