Low-cost Raspberry Pi star sensor for small satellites

Bharat Chandra P,a,b,* Mayuresh Sarpotdar,c
Binukumar G. Nair,a Richa Rai,a Rekhes Mohan,a
Joice Mathew,d Margarita Safonova,a and Jayant Murthy,a

*aIndian Institute of Astrophysics, Bangalore, Karnataka, India
bUniversity of Calcutta, Kolkata, West Bengal, India
cNanyang Technological University, Satellite Research Centre, Singapore
dAustralian National University, Advanced Instrumentation and Technology Centre, RSAA,
Mount Stromlo Observatory, Canberra, Australia

Abstract. We present a low-cost Raspberry Pi-based star sensor StarberrySense using commercial-off-the-shelf components, developed and built for applications in small satellites and CubeSat-based missions. A star sensor is one of the essential instruments on board a satellite for attitude determination. However, most commercially available star sensors are expensive and too bulky for use in small satellite missions. StarberrySense is a configurable system—it can operate as an imaging camera, a centroiding camera, or a star sensor. We describe the algorithms implemented in the sensor, its assembly, and its calibration. This payload was selected by a recent announcement of an opportunity call for payloads to fly on the PS4-Orbital Platform by the Indian Space Research Organization. © 2022 Society of Photo-Optical Instrumentation Engineers (SPIE) [DOI: 10.1117/1.JATIS.8.3.036002]

Keywords: star sensor; attitude sensor; small satellites; Raspberry Pi; CubeSat.

Paper 22020G received Feb. 9, 2022; accepted for publication Jun. 30, 2022; published online Jul. 20, 2022.

1 Introduction

A star sensor on board a satellite determines the satellite’s attitude in space by identifying the stars in its field of view (FOV). The advantage of a star sensor over other sensors, such as magnetometers and sun sensors, is that it provides more precise orientation information of the satellite, on the order of a few arcseconds to microarcseconds depending on the star sensor model. In recent years, small satellites have been marking a new era of space exploration. Among these, a class of small satellites is called the CubeSats. They are usually used in low-earth orbits (LEO) for communication, remote sensing, scientific missions, and more. There might be a need for pointing information in these kinds of missions to perform orientation corrections to perform a particular observation. Most of the commercial radiation-hardened star sensors are made for long-duration space missions. However, using these expensive star sensors is not economical for the low-budget short-duration CubeSat or any SmallSat mission in LEO, where radiation levels are low. Also, in some cases, commercial star sensors are bulky and consume a lot of power, making them impractical for use in CubeSats. To address this problem, our group started the development of a low-cost star sensor using either readily available components or building some of them in house. The first prototype, StarSense, was built using a Star 1000 CMOS detector, an MIL-grade FPGA board, and a custom-fabricated lens system at the cost of around 10,000 USD. Sarpotdar et al. also developed a software package to evaluate the performance of star sensor algorithms, which included parameters, such as attitude accuracy, sky coverage, and catalog size, for different values of focal length, FOV, distortion effects, and other attributes. We have continued the development of low-cost star sensors for short duration missions using readily available off-the-shelf components with a fast development cycle, modular design, and easy source code portability to different hardware. One concept of such a low-cost
star sensor development, called lab open star tracker, based on Raspberry Pi (RPi) and using existing open-source astronomy software, was recently described by Gutiérrez et al.6 They employed the Source Extractor to extract the bright sources from the captured image and Match for matching the image of the sky with the projected segments of the sky stored on board, followed by the attitude determination. From the real-sky test, they estimated the sensor accuracy to be about 30 arcseconds to 60 arcseconds with an update rate of 0.05 Hz.

In StarberrySense, we used RPi Zero7 as the main controller, costing around 15 USD, along with an RPi camera as the detector, off-the-shelf lens system for imaging, a custom-made power supply, and a custom-made housing to mount all of the components, bringing the total price to about 1600 USD. RPi is a small single-board computer developed by the RPi foundation.8 It provides a compact, low-power, flexible platform to which different devices can be interfaced for applications ranging from home automation to aeronautical communication. RPi hardware design and computational capability allow it to be used in various small satellite missions, and this was the main reason for choosing the RPi. Using off-the-shelf components makes the development of a star sensor fast, cheap, and easy without relying on complicated hardware and optics design. StarberrySense has an accuracy of 30 arcseconds with an update rate of 0.2 Hz and an average power consumption of under 1.25 W (see Sec. 2 and Table 3). Table 1 shows the comparison between StarberrySense and other commercially available star sensors.9–11

The software for StarberrySense is written in standard C/C++ for faster processing on a free, open-source (FOSS) Linux platform and is easily portable to any other system, such as Embedded Linux or real-time operating system, allowing for faster development and deployment. The modular code architecture allows for easy customization, which is vital in the design of such subsystems in CubeSat or small satellite missions. The sensor operates in lost-in-space (LIS) mode,12 where centroids of possible stars in the image are identified by a region-growing algorithm. The geometric voting algorithm12 is used to match the centroids with stars stored in the onboard catalog. The final attitude determination is performed using the quaternion-estimator (QUEST)13 algorithm. A minimum of three stars is required for attitude determination. The stored onboard catalog is the Hipparchus bright star catalog,14 which lists stars with magnitudes up to 6.5.

2 Instrument

A star sensor is essentially an imaging camera that images the stars, processes the images, identifies the bright stars, and matches their positions with the stored star catalog for determining the look direction. As such, it consists of several subsystems that include optics, the image sensor, electronics, and the housing (Fig. 1). The main controller in our star sensor is the RPi Zero (Table 2).

2.1 Design Requirements

The design requirement was to build a low-cost star sensor for small satellites and CubeSat class missions. The pointing accuracy requirement was based on the subarcminute pointing...
requirement for small satellite astronomy missions. The total mass was constrained to under 500 g, and the system’s power consumption to below 2 W as per the mass and power budget of CubeSats and small satellites. Because we had a stringent need for a shorter development cycle, we chose to design the star sensor from readily available off-the-shelf electronics and ruggedized optical components.

For the star sensor to work, three or more stars are required to be visible in its FOV. Therefore, the FOV (which depends on the focal length of the optics and the size of the image sensor) and the magnitude detection limit must be chosen carefully to meet the requirement. For our design, the RPi V2 camera module was used as the image sensor and the Schneider Xenoplan 23-mm lens system as the imaging optics. This setup provided the system with a FOV of 9.31 deg × 7 deg, and the magnitude of the star detection limit was set to 6.5, ensuring a minimum of three stars in the FOV.

| Chip         | Broadcom BCM2835 |
|--------------|------------------|
| CPU          | ARM1176JZF-S Single Core |
| Operating system | Raspbian |
| GPU          | Broadcom VideoCore IV |
| CPU clock    | 1 GHz |
| Memory       | 512 MB DDR2 |
| Interfaces   | 1× Micro USB |
|              | 1× UART |
| I/O          | 40 GPIO Pins |
| Onboard storage | SD, MMC, SDIO card slot |
| Weight       | 45 g |
| Power rating | 250 mA (1.25 W) |
| Dimensions   | 66.0 mm × 30.5 mm × 5.0 mm |

For the star sensor to work, three or more stars are required to be visible in its FOV. Therefore, the FOV (which depends on the focal length of the optics and the size of the image sensor) and the magnitude detection limit must be chosen carefully to meet the requirement. For our design, the RPi V2 camera module was used as the image sensor and the Schneider Xenoplan 23-mm lens system as the imaging optics. This setup provided the system with a FOV of 9.31 deg × 7 deg, and the magnitude of the star detection limit was set to 6.5, ensuring a minimum of three stars in the FOV.
2.2 Image Sensor and Lens System

The star sensor uses the Xenoplan Schneider C-mount lens system designed to work in the visible and near-IR range (400 to 1000 nm). The RPi V2 No-IR camera used in the star sensor has a Sony IMX219 CMOS image sensor with no IR filter that is sensitive from the visible band to the near-IR band. With the 23-mm lens system and the 1.12-μm pixel sensor, the star sensor has a pixel scale of 20.5″ per pixel (with 2 × 2 binning). The dark noise characterization for the RPi V2 camera was done by Pagnutti et al. To ensure that stars with a magnitude of 6.5 are imaged with a good SNR, we chose the exposure to be 500 ms. In this setup, the mean value of the background was 1.87 counts/pixel (σ = 0.89), and 6.5 mag stars were detected with SNR = 30.

2.3 Electronics

The RPi Zero with the ARM1176JZF-S Single Core processor is the main controller of our star sensor; it performs the image capture, image processing, star identification, and attitude determination. The RPi is interfaced with the camera module with the MIPI camera serial interface (CSI), and the Broadcom VideoCore IV, onboard RPi handles the image readout from the Sony IMX219 CMOS sensor. The universal asynchronous receiver–transmitter (UART) port of the RPi is configured for transmitting the output data (quaternions and centroids). The RPi Zero and the camera module consume less than 1.25 W, which is within our power limit requirement. The power supply for the star sensor was designed based on a switching mode regulator LMR33630 from Texas Instruments, which provides a 5-V regulated voltage over a wide input range from 5 to 30 V. The LMR33630 has a wide operating temperature range from −40°C to +125°C, making it suitable for our application (the summary of the technical specifications is presented in Table 3). Figure 2 shows the overall electronics block diagram of the star sensor.

2.4 Mechanical Structure and Assembly

The mechanical enclosure for the star sensor was custom-designed and precision-manufactured for holding the camera, C-mount lens, power supply, and RPi Zero. The enclosure was made with 3-mm-thick-aluminum 6061. Figure 3 shows the exploded view of the star sensor. The star

| Table 3 StarberrySense technical specifications. |
|-----------------------------------------------|
| Weight (without baffle) | 315 g |
| Weight (with baffle)    | 470 g |
| Board                  | RPi Zero |
| Dimensions (without baffle) (L × W × H) | 98 × 75 × 69 mm |
| Dimensions (with baffle) (L × W × H)    | 98 × 75 × 160 mm |
| Power                   | 1.25 W |
| Image sensor            | Sony IMX219 |
| FOV                     | 9.31 deg×7 deg |
| Wavelength range        | 450 to 750 nm |
| Limiting magnitude      | 6.5 (V) |
| Mode of operation       | LIS |
| Pointing accuracy (3σ)  | 27.18° |
| Roll accuracy (3σ)      | 38.76° |
The sensor has a mass of about 315 g, excluding the baffle. The entire housing is black anodized for better optical performance and to protect the surface from corrosion.

To enhance the Sun and Moon avoidance angle, the star sensor requires a baffle. Otherwise, if the Sun appears near the FOV, it can saturate the detector, and the star sensor will not work. To prevent this, we designed a custom baffle following Asadnezhad et al.\textsuperscript{16} to achieve the Sun avoidance angle of 30 deg and manufactured it in house from aluminum 6061. The baffle was also black anodized to avoid scattering (Fig. 6).

The assembly of the star sensor was initiated by bonding the lens mount for the C-mount lens system with the mechanical enclosure of the star sensor using epoxy Scotch-Weld 2216\textsuperscript{TM}. Once the glue on the mount was cured, the C-mount lens assembly was threaded onto the mount. The RPi camera and RPi Zero were mounted on an intermediate aluminum plate along with the voltage regulator board as shown in Figs. 4 and 5. The intermediate aluminum plate was
mounted inside the star sensor enclosure by aligning it with respect to the optomechanical axis of the lens barrel and the RPi camera using a collimated light source. The alignment precision of the system was bound by the surface finish of the mechanical components of the star sensor, which was in the range of 50 to 100 μm. Then, the lens system was moved on the C-mount thread to bring the focal plane onto the image sensor plane, and the lens system was locked in position using the lock screw. Once the RPi camera was aligned, all of the required electrical connections were made, and the enclosure was sealed using the enclosure lid. Once the main body of the entire star sensor unit was assembled, the baffle was fitted using the mounting holes provided on the mounting plate of the enclosure case (Fig. 6).

3 Software

For the onboard operating system, we used a minimalist version of Raspbian with in-house developed algorithms written in C++ running as a standalone application. For the image capture

Fig. 5 Photo of the RPi v2 camera mounted on the intermediate plate.

Fig. 6 StarberrySense after assembly.
and the camera control, a C++ API developed by the AVA research group was used. The output of the sensor is attitude quaternions, which can be stored on board and transmitted through the UART serial communication when needed.

3.1 Lost-in-Space Algorithm

The LIS algorithm is used to determine the orientation of the spacecraft in space without the need for any initial knowledge of the satellite orientation. The algorithm identifies stars in the FOV and determines the attitude of the satellite. The algorithm uses a search table for star identification, which is generated and stored on board prior to launch. During the star sensor in-flight operation, the distance value between star pairs in the image is compared with the search table for the voting. Once the voting is completed, and the actual stars are identified, the QUEST algorithm is used to determine the orientation in terms of quaternions of rotation. Figure 7 shows the flowchart of the algorithm implemented on board the star sensor.

![Flowchart of the algorithm on board the star sensor.](image-url)
3.1.1 Star identification process

Several methods have been developed for the identification of stars in the images. \cite{18} Initially, stars were identified based on their magnitudes, which required a flux-calibrated image sensor. The first-star sensor, based on the CCD, was developed by Salomon at JPL. \cite{19} The major flaw with this technique was that, over time, as the sensor degraded, the method was prone to errors. Then came the development of the pattern-matching based algorithms \cite{18} for star identification. Among them, one of the well-known methods was proposed by Liebe; \cite{1} it was based on finding the angular distance from a star to its two closest neighbors and the spherical angle between them to identify the star. This method paved the way for the development of algorithms that use star triangles for star identification. \cite{20,21,22} Another well-known method, proposed by Mortari et al., \cite{24} is the pyramid algorithm that uses four stars to perform the star identification. This technique is more robust and reliable.

In our star sensor, we used the geometric voting algorithm with a binary search algorithm for star identification. \cite{4,12} The geometric voting uses angular distances from a star to all stars in the FOV for voting to get the identity of the star. This is followed by the secondary voting to confirm these identities and reject false stars, making it robust and less sensitive to noises.

3.1.2 Search table generation

The search table is generated manually and stored on board before the flight. The search table was generated by calculating the angular distances between all possible star pairs in the Hipparchus catalog and saving only those star pairs with an angular separation that is less than the longest FOV $D_{fov}$ (Table 2). Once the distances of these star pairs are calculated, they are sorted based on the increasing order of the distance between them and stored on board the star sensor along with the unit vector of each star in the Earth-centered inertial (ECI) coordinate system. The unit vectors of stars are calculated using the equation:

$$
\begin{bmatrix}
\hat{x} \\
\hat{y} \\
\hat{z}
\end{bmatrix} = \begin{bmatrix}
\cos \alpha \cos \delta \\
\sin \alpha \cos \delta \\
\sin \delta
\end{bmatrix},
$$

(1)

where $\alpha$ is the right ascension and $\delta$ is the declination of a star. Thus, the search table stored on board contains the list of all possible star pairs along with the angular distances between them. Algorithm 1 was used to generate the search table, where $D_{fov}$ is the longest FOV and $d_{ij}$ is the minimum required angular separation between the stars, assumed to be $30^\circ$ to avoid blending.

Algorithm 1  Search table generation.

1: \textbf{for} $i = 1$ to $N$ \textbf{do}
2: \hspace{1em} \textbf{for} $j = i + 1$ to $N$ \textbf{do}
3: \hspace{2em} $d_{ij} =$ distance between $N_i$ and $N_j$
4: \hspace{2em} \textbf{if} $d_{ij} < D_{fov}$ and $d_{ij} > D_{min}$ \textbf{then}
5: \hspace{3em} Append the table row with elements $i$, $j$, $d_{ij}$,
6: \hspace{2em} \textbf{end if}
7: \hspace{1em} \textbf{end for}
8: \hspace{1em} \textbf{end for}
9: \hspace{1em} Sort the table based on $d_{ij}$. 
3.1.3 Region growing and centroiding

The first part of the algorithm on board the star sensor is finding the centroids of the stars. For this, an image of the sky is acquired and run through a region-growing algorithm to grow the brightest regions in the image above a threshold value. The threshold value is \( p_{th} = \overline{p} + 5\sigma_p \), where \( \overline{p} \) is the mean pixel value and \( \sigma_p \) is the standard deviation in the pixel values. The pixels above a threshold that lies in the neighborhood of 2 to 3 pixels are considered to be part of the same region. Here, single-pixel regions are ignored because they are mostly due to either cosmic ray events or random noise. After the regions are grown, the weighted centroids for all regions in the image are calculated as

\[
x = \frac{\sum_{i=0}^{N} x_i I_i}{\sum_{i=0}^{N} I_i}, \quad y = \frac{\sum_{i=0}^{N} y_i I_i}{\sum_{i=0}^{N} I_i}.
\]

where \((x, y)\) is the centroid, \((x_i, y_i)\) is the \(i\)th pixel position, and \(I_i\) is the intensity of the \(i\)th pixel. The weighted centroiding helps us achieve subpixel level accuracy. Once the centroids are calculated, the orientation unit vectors for each star in the camera reference coordinate system are calculated as

\[
\begin{bmatrix}
\hat{x} \\
\hat{y} \\
\hat{z}
\end{bmatrix} = \begin{bmatrix}
\frac{pp_x(x-x_0)}{Kx_{mm}} \\
\frac{pp_x(y-y_0)}{Ky_{mm}} \\
\frac{1}{K}
\end{bmatrix},
\]

where

\[
K = \sqrt{\left(\frac{pp_x \times (x-x_0)}{f_{mm}}\right)^2 + \left(\frac{pp_y \times (y-y_0)}{f_{mm}}\right)^2 + 1}.
\]

Here, \(f_{mm}\) is the focal length of the lens system, \((x_0, y_0)\) is the location of the central pixel of the detector, \(pp_x\) is pixel size along the \(x\) axis, and \(pp_y\) is the pixel size along the \(y\) axis. Once the unit vectors for all stars in the image are calculated, geometric voting begins.

Many factors cause the measured centroids to shift from their actual positions. Centroiding accuracy is proportional to the square root of the signal from the stars, and if the point-spread function is < 0.5 pixels, the centroiding accuracy will be limited by the sampling theorem.26 Another factor is the radial distortion caused by the lens system. This results in the centroids being displaced radially. This distortion must be accounted for during the calibration process with a suitable distortion model and corrected during the processing12,26 (Sec. 4.2).

3.1.4 Geometric voting

In geometric voting, the distance between each star pair is calculated for the matching process. The distance can be calculated by finding the inverse cosine of the unit vector dot product of the star pairs. Once the distance is found, the star’s ID should be in the search table in the distance range from \(d_{ij} - \varepsilon\) to \(d_{ij} + \varepsilon\), where \(\varepsilon\) is the uncertainty in calculated distance,26 assumed to be 20° in our case. Using a binary search algorithm, the location on the search table is found where this distance range lies. Once the location of the distance range is known, the stars IDs, lying between the distance \(d_{ij} - \varepsilon\) and \(d_{ij} + \varepsilon\) are used for the voting. After the voting, the total votes are counted to identify the star in the image that corresponds to the star in our catalog. The second round of voting is used to remove falsely identified stars. In the end, the total number of votes from the primary and secondary rounds is used to identify the stars and to remove false detections. Once the stars in the image are identified, the star sensor proceeds to the attitude determination.

3.2 Attitude Determination

Algorithm 2 is used to identify stars and determine the attitude. Once stars in the image are identified, the final step is to calculate the quaternion of rotation between the ECI coordinate
Algorithm 2  Star sensor algorithm.

1: Calculate threshold for region growing and perform region growing
2: Calculate the centroids of the stars along with uncertainty $\varepsilon$ of the centroid
3: Correct centroids for distortion
4: Assign $n$ as the total number of centroids
5: Calculate $\tilde{p}$, the unit vector for the centroids in sensor body coordinate
6: \textbf{for} $i = 1$ to $n$ \textbf{do}
7: \hspace{1em} \textbf{for} $j = i + 1$ to $n - 1$ \textbf{do}
8: \hspace{2em} Compute distance $d_{ij} = \cos^{-1}(\tilde{p}_i \cdot \tilde{p}_j)$
9: \hspace{2em} \textbf{if} $d_{ij} < D_{fov}$ \textbf{then}
10: \hspace{3em} upper$_\text{Index} = \text{binary_search}(d_{ij} + \varepsilon_{ij})$
11: \hspace{3em} lower$_\text{Index} = \text{binary_search}(d_{ij} - \varepsilon_{ij})$
12: \hspace{3em} \textbf{for} $k = \text{lower}_\text{Index}$ to $\text{upper}_\text{Index}$ \textbf{do}
13: \hspace{4em} for all entries $T(k)$ \textbf{do}
14: \hspace{5em} Append voting list $V_i$ and $V_j$ of possible stars $S_i$ and $S_j$
15: \hspace{4em} \textbf{end for}
16: \hspace{3em} \textbf{end for}
17: \hspace{2em} \textbf{end if}
18: \hspace{1em} \textbf{end for}
19: \textbf{end for}
20: \textbf{for} $i = 1$ to $n$ \textbf{do}
21: \hspace{1em} Find the catalog star with maximum votes in voting list $V_i$
22: \hspace{1em} Assign $St_i$ to the catalog star that got the maximal votes
23: \textbf{end for}
24: \textbf{for} $i = 1$ to $n$ \textbf{do}
25: \hspace{1em} \textbf{for} $j = i + 1$ to $n$ \textbf{do}
26: \hspace{2em} Compute distance $d_{ij} = \cos^{-1}(\tilde{p}_i \cdot \tilde{p}_j)$
27: \hspace{2em} \textbf{if} distance between stars $St_i$ and $St_j \in d_{ij}$ \textbf{then}
28: \hspace{3em} Add a vote for the match of $(St_i, S_i)$ and $(St_j, S_j)$
29: \hspace{2em} \textbf{end if}
30: \hspace{2em} \textbf{end for}
31: \hspace{1em} \textbf{end for}
32: \textbf{end for}
33: Identify true stars based on primary and secondary votes.
34: Estimate attitude using QUEST algorithm.
The geometric voting algorithm provides the unit vector of the identified star in the ECI coordinate system, and the unit vector for that star in the sensor coordinate system is calculated using Eq. (2).

Two commonly used algorithms to determine the attitude from the unit vectors are triaxial attitude determination (TRIAD)\textsuperscript{27} and QUEST\textsuperscript{13}. TRIAD determines the direction cosine matrix that describes the relationship between two coordinate systems. The TRIAD algorithm only uses two of the unit vector pairs and discards the rest, thus not utilizing the complete information from all star unit vector pairs. Because each identified star has a pair of unit vectors, the total amount of pairs can be 20 or more (see Fig. 10 for an example of 18 registered stars). Because we want to use the complete information, the algorithms suited for handling such cases are Davenport’s q-method,\textsuperscript{28} QUEST, and SVD\textsuperscript{29,30}. Because QUEST bypasses the eigenvalue problem and is computationally less expensive, we implemented it in our attitude determination algorithm.

Using the QUEST algorithm with multiple unit vector pairs as inputs, the star sensor calculates the quaternion of rotation between the two coordinate systems. The rotation can be described in many ways, such as rotation matrix, Euler angles, quaternions, etc. In our case, we chose quaternions as they are computationally less intensive. Now, if $v_b$ is the unit vector for the star in the sensor body coordinate system and $v_i$ is the unit vector for the ECI coordinate system (Fig. 8), we have

$$v_b = R v_i,$$

where $R$ is the rotation matrix that describes the rotation between the ECI and the sensor body coordinate systems. Once the stars in the image are identified, we know the unit vector of those stars in both coordinate systems. Now, the star sensor needs to find the rotation matrix $R$ from the unit vectors. The value of $R$ must be such that it minimizes the loss function

$$J = \frac{1}{2} \sum_{k=1}^{N} w_k |v_{kb} - R v_{ki}|^2,$$

where $J$ is the loss function, $w_k$ is the weight of each star unit vector pair, and $N$ is the number of correctly identified stars. Because the measurements are not ideal, we will always have a value of $J$ greater than zero. The star sensor needs to find a solution that minimizes the loss function $J$, and the method should not be computationally intensive. There are several existing methods to solve this classic Wahba’s problem.\textsuperscript{31} We restate the loss function in terms of quaternions in such a way that it becomes an eigenvalue problem, in which the largest eigenvalue is to be found. The QUEST algorithm was developed to bypass the expensive eigenvalue problem by approximating the process. It reduces to solving for parameter $p$, called the Rodriguez parameter, in the equation...
\[ [(\lambda_{\text{opt}} - \sigma)I - S]p = Z, \quad (6) \]

where

\[ Z = [B_{23} - B_{32}B_{31} - B_{13}B_{12} - B_{21}^T], \quad B = \sum_{k=1}^{N} w_k (v_{kl} e_k^T), \quad \lambda_{\text{opt}} = \sum w_k, \quad \sigma = \text{Tr}(B), \]

and

\[ S = B + B^T. \quad (7) \]

After finding the value of \( p \), the attitude quaternion is given as

\[ q_{\text{quest}} = \frac{1}{\sqrt{1 + p^T p}} \begin{bmatrix} p \\ 1 \end{bmatrix}, \quad (8) \]

where

\[ q_{\text{quest}} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}. \quad (9) \]

### 4 Calibration and Testing

#### 4.1 Performance and Precision Limit

We calculated the theoretical limit of accuracy for the noise equivalent angle (NEA) following calculations in Ref. 32. The NEA measures the ability of a star sensor to reproduce the same attitude information for the same part of the sky. Even though the NEA depends on different instrumental parameters such as dark noise, read noise, etc., it is possible to quickly estimate it. For that, we first find the average number of stars in the FOV \( N_{\text{stars}} \):

\[ N_{\text{stars}} = N_{\text{catalog}} \frac{1 - \cos \frac{D}{2}}{2} = 11, \quad (10) \]

where \( N_{\text{catalog}} = 8874 \) is the total number of stars from the Hipparcos catalog used to generate the search table (stars with magnitudes \( \leq 6.5 \)). The average StarberrySense FOV is \( D = 8.15 \text{ deg} \), and the average number of pixels is \( N_{\text{pixels}} = 1436 \).

The cross boresight NEA (\( E_{\text{CB}} \)), which gives the bound of error, or uncertainty, in the pointing direction, and the roll axis NEA (\( E_{\text{Roll}} \)), which gives the bound of error, or uncertainty, in the roll axis, are calculated as

\[ E_{\text{CB}} = \frac{DE_{\text{centroid}}}{N_{\text{pixels}}\sqrt{N_{\text{stars}}}} = 3.05'' , \quad (11) \]

and

\[ E_{\text{Roll}} = \tan^{-1} \left( \frac{E_{\text{centroid}}}{0.3825N_{\text{pixels}}} \right) \frac{1}{\sqrt{N_{\text{stars}}}} = 56.06'' , \quad (12) \]

where \( E_{\text{centroid}} \) is the centroiding accuracy of the star sensor, assumed to be 0.5 pixels as the worst-case scenario.
4.2 Real Sky Test and Distortion Correction

After the star sensor was assembled, a night-sky test was conducted at Vainu Bappu Observatory (VBO) facility of IIA, Kavalur, Tamil Nadu. The star sensor was mounted on a stable platform and pointed toward the night sky with no pointing knowledge. Two subsequent night-sky tests were conducted: the first to correct for the distortion of the lens system and the second to verify the pointing performance.

The main distortion that affects the angular measurement of the distance between stars is the radial distortion, which causes a significant error during the voting process leading to false star identification. The obtained image of the sky in jpeg format was fed to Astrometry.net33 for astrometric calibration and generation of distortion-correction polynomial coefficients. Thereafter, these polynomial coefficients were stored on board the star sensor for performing distortion correction using the SIP convention method.34 If \((u, v)\) is the centroid of the star in the pixel coordinate system and \((x, y)\) is the centroid of the star after the correction, we have the distortion correction equation

\[
x = u + \sum_{a,b} A_{a,b} u^a v^b, \quad y = v + \sum_{a,b} B_{a,b} u^a v^b,
\]

(13)

where \(A_{a,b}\) and \(B_{a,b}\) are the polynomial coefficients of the second order and \(a\) and \(b\) are integers with \(a + b \leq 2\). This correction was implemented in Algorithm 2 after the centroid calculation, and the corrected centroids were used for unit vector calculation. We found that second-order polynomial correction was sufficient as the distortion was <26\(^{\circ}\) even at the edges.

In the second test, the star sensor was pointed at different parts of the sky, and the captured images were stored on board along with the calculated quaternions. Figure 9 shows one of the captured images, pointed at a random part of the sky, and the calculated centroids of the stars after running the region-growing algorithm are shown in Fig. 10. Quaternions, obtained from the star sensor, were used to find the pointing,35 which in the case of Fig. 9 was \(a = 91.0176\) deg and \(\delta = 14.1498\) deg, whereas the actual pointing from Astrometry.net was \(a = 91.019\) deg and \(\delta = 14.149\) deg. The ground-based observations error was found to be 6\(^{\circ}\) when pointing from the star sensor was compared with the pointing obtained from Astrometry.net.

To determine the accuracy of StarberrySense in pointing and roll direction, the star sensor was mounted on a stable platform and pointed toward the sky. The star sensor was set to calculate the quaternions and store them on board along with the sky images for a long duration. The actual pointing obtained from the Astrometry.Net using the stored images, and the pointings obtained from the star sensor was used to determine the accuracy in the pointing and in the roll axis. Figures 11 and 12 show the histogram of error in the pointing and roll axis, respectively.
Fig. 10 Same image as Fig. 9 with marked centroids of the stars above the threshold (Sec. 3.1.3) after running the region-growing algorithm. $x$ and $y$ axes are pixel numbers.

Fig. 11 Histogram of error for StarberrySense in the pointing axis.

Fig. 12 Histogram of error for StarberrySense in the roll axis.
The measured accuracy in the pointing axis was $3\sigma = 27.18$ arcsec and in the roll axis was $3\sigma = 38.76$ arcmin. The success rate of the star sensor was found to be 94.69% from the real-sky test. Also, the sky condition during the test was close to 5 on the Bortle scale, which is a measure of the night sky brightness.\textsuperscript{36} We also determined the slew rate limit by simulating the star trailing on the images captured by the star sensor. We found that, after the simulated motion rate at 0.2 deg/s, the algorithm failed to determine the quaternions.

4.3 Power Consumption and Processing Time

Table 4 shows the average power consumption during different processes: when the star sensor is idle, capturing the image, processing the image, or during UART transmission. The star sensor draws an average current of 140 mA when idle, and the current can peak to 220 mA during the image capture.

Table 5 shows the processing time taken by each of the subroutines in the star sensor algorithm. The image capture and readout take the longest amount of time, and the star sensor can provide 14 quaternions per minute.

5 Flight Qualification

To qualify for the space flight, payloads have to undergo the standard environmental tests: vibration and thermal-vacuum tests. This is to ensure that the payload will be able to withstand all launch loads and operate in the space environment. The most stringent vibration requirements for launch vehicle platforms are the following: the natural frequency of the payload must be above 100 Hz, and the instrument should be able to withstand acceleration loads of up to 10 g.\textsuperscript{37,38} The thermal-vacuum and vibration tests for the sensor were performed in the M. G. K. Menon Lab, CREST Campus, IIA, Bangalore, as per the Polar Satellite Launch Vehicle (PSLV) Stage 4 flight requirements.

5.1 Thermal-Vacuum Test

A thermal-vacuum chamber was used to simulate the thermal environment that the sensor will be subjected to in space. Four temperature probes were mounted on the StarberrySense body as
shown in Figs. 13 and 14 using aluminum tape; then the pressure inside the chamber was pumped down to $10^{-6}$ mbars with the help of a roughing pump and a turbo-molecular pump. The temperature in the chamber was varied through seven cold and hot cycles to simulate the orbital environment in the LEO. Figure 15 shows the temperature measurement from the four sensors mounted on the star sensor overplotted on the programmed chamber profile.39

After each hot and cold cycle, the functionality of the payload was checked by turning on the system and capturing dark frames, which were stored on board the system memory. The camera module and RPi functioned without any errors, and the dark frames were consistent without any abnormalities. Thus, the payload remained functional and was able to handle the temperature variations.

5.2 Vibration Test

The vibration test was conducted as per the requirement of the launch provider (ISRO-PSLV). The launch load profiles are shown in Tables 6 and 7. The payload was subjected to sine and random vibrations according to the launch requirements. Accelerometers were mounted on the StarberrySense body as shown in Fig. 16. StarberrySense successfully passed the vibration test without any structural damage, with all of the connecting bolts remaining intact. The first natural
frequency was found to be 385 Hz (well above the frequency level requirement of PSLV stage 4), and the next five successive frequencies were 585, 911, 1327, 1376, and 1655 Hz, respectively.

After the vibration test, the star sensor was tested in the open sky. We verified that the star sensor operates normally as expected and distortion parameters were unchanged.

### 6 Conclusions and Future Work

In this paper, we have presented the design and development of StarberrySense, a low-cost star sensor based on RPi Zero. The star sensor was assembled, aligned, calibrated, and tested in house. A computationally efficient algorithm was implemented and tested against the real sky conditions, and the results were favorably compared with obtained values from Astrometry.net. The thermal-vacuum test for the star sensor, along with the vibration test, was completed successfully, as per PSLV launch requirements. StarberrySense was selected by the recent

![Average temperature values recorded from the four sensors mounted on the star sensor body during the thermal-vacuum test (blue continuous) overplotted on the programmed temperature profile (red dotted).](image-url)

**Table 6** Random vibration specifications for StarberrySense.

| Frequency (Hz) | PSD ($g^2$/Hz) | Level     | Duration | Axis               |
|----------------|----------------|-----------|----------|--------------------|
| 20             | 0.001          | $9g_{rms}$| 1 min    | All three axes     |
| 68             | 0.001          |           |          |                    |
| 250            | 0.062          |           |          |                    |
| 1000           | 0.062          |           |          |                    |
| 2000           | 0.015          |           |          |                    |

**Table 7** Sinusoidal specifications for StarberrySense.

| Frequency       | Longitudinal axis | Lateral axis |
|-----------------|-------------------|-------------|
|                 | Level             | Sweep rate  | Level       | Sweep rate  |
| 10 to 16 Hz     | 20 mm DA          | 2 oct/min   | 15 mm DA    | 2 oct/min   |
| 16 to 100 Hz    | 10 g              | 2 oct/min   | 6 g         | 2 oct/min   |

frequency was found to be 385 Hz (well above the frequency level requirement of PSLV stage 4), and the next five successive frequencies were 585, 911, 1327, 1376, and 1655 Hz, respectively. After the vibration test, the star sensor was tested in the open sky. We verified that the star sensor operates normally as expected and distortion parameters were unchanged.

6 Conclusions and Future Work

In this paper, we have presented the design and development of StarberrySense, a low-cost star sensor based on RPi Zero. The star sensor was assembled, aligned, calibrated, and tested in house. A computationally efficient algorithm was implemented and tested against the real sky conditions, and the results were favorably compared with obtained values from Astrometry.net. The thermal-vacuum test for the star sensor, along with the vibration test, was completed successfully, as per PSLV launch requirements. StarberrySense was selected by the recent
announcement of opportunity call for payloads to fly on the PS4-Orbital Platform by ISRO. PS4-Orbital Platform is an innovative use of the spent PS4 stage (fourth stage of the PSLV) as it has standard interfaces/packages for power and telemetry and can be stabilized. Thus, it can carry the scientific payloads to perform scientific experiments for up to 6 months in LEO. We are preparing the payload for flight before the end of 2022 on board the ISRO PSLV stage-4 platform.

Further improvements in the next version of StarberrySense include an upgrade of the hardware processing capability with the newer Rpi Zero 2 as well as a reduction in the form factor and weight of the payload. In addition, an improvement in the optics is envisaged to enhance the light collection area and thereby reduce the exposure time.

**Acknowledgment**

We wish to thank Mr. S. Sriram of the Indian Institute of Astrophysics (IIA) and Mr. Ajin Prakash from Arksa Research Labs, Bangalore, India, for their valuable suggestions and help. MS acknowledges the financial support by the Department of Science and Technology (DST), Government of India, under the Women Scientist Scheme (PH), project reference number SR/WOS-A/PM-17/2019. We also thank the staff at the M. G. K. Menon Laboratory (CREST) for helping us with the assembly and storage of payload components in the cleanroom environment.

**References**

1. C. Liebe, “Star trackers for attitude determination,” *IEEE Aerosp. Electron. Syst. Mag.* **10**, 10–16 (1995).
2. T. Reid, “Orbital diversity for global navigation satellite systems,” PhD thesis (2017).
3. S. Lee, R. Saleem, and S.-S. Lee, “Micro star tracker with a curved vane for a short baffle length and sharp stray light attenuation,” *Appl. Opt.* **59**, 4131–4142 (2020).
4. M. Sarpotdar et al., “A software package for evaluating the performance of a star sensor operation,” *Exp. Astron.* **43**(1), 99–117 (2016).
5. M. Sarpotdar et al., “Design and development of a star sensor cum asteroid tracker,” poster presented at SPIE AS-16, 9904-195 (2016).
6. S. Gutiérrez, C. Fuentes, and M. Diaz, “Introducing SOST: an ultra-low-cost star tracker concept based on a Raspberry Pi and open-source astronomy software,” *IEEE Access* **8**, 166320–166334 (2020).
7. Raspberry Pi Zero, “Raspberry Pi Foundation,” [http://raspberrypi.org](http://raspberrypi.org) (2000).
8. Raspberry Pi Foundation, “Raspberry Pi,” http://www.raspberrypi.org (2021).
9. T. Dzamba et al., “Success by 1000 improvements: flight qualification of the st-16 star tracker,” in Small Satellite Conf. (2014).
10. B. Seng, “Space sensor commercialization: a small company approach,” https://www.osti.gov/servlets/purl/1249037 (2015).
11. R. Mozzillo, “Highly accurate, flexible, robust and scalable multicamera system for spacecraft autonomous attitude determination through low cost cameras,” https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5de83b5&applied=PPGMS.
12. M. Kolomenkin et al., “Geometric voting algorithm for star trackers,” IEEE Trans. Aerosp. Electron. Syst. 44, 441–456 (2008).
13. G. Wahba, “A least squares estimate of satellite attitude,” Soc. Ind. Appl. Math. Probl. 8(3), 65 (1966).
14. M. A. C. Perryman et al., “The hipparcos catalogue,” Astron. Astrophys. 500, 501–504 (1997).
15. M. Pagnutti et al., “Laying the foundation to use Raspberry Pi 3 v2 camera module imagery for scientific and engineering purposes,” J. Electron. Imaging 26, 013014 (2017).
16. M. Asadnezhad, A. Eslamimajd, and H. Hajghassem, “Optical system design of star sensor and stray light analysis,” J. Eur. Opt. Soc. 14, 9 (2018).
17. R. M. Salinas, “RaspiCam: C++ API for using Raspberry camera with/without OpenCv,” http://www.uco.es/investiga/grupos/ava/node/40 (2020).
18. B. Spratling and D. Mortari, “A survey on star identification algorithms,” Algorithms 2(1), 93–107 (2009).
19. P. Salomon and W. Goss, “A microprocessor-controlled CCD star tracker,” in 14th Aerosp. Sci. Meeting (1976).
20. B. Quine and H. F. Durrant-Whyte, “Rapid star-pattern identification,” Proc. SPIE 2739, 351–360 (1996).
21. E. Heide et al., “Development and validation of a fast and reliable star sensor algorithm with reduced data base,” in Int. Astronaut. Congr. (1998).
22. A. Nabi, Z. Ahmed-Foitih, and M. E.-A. Cheriet, “Improved triangular-based star pattern recognition algorithm for low-cost star trackers,” J. King Saud Univ. – Comput. Inf. Sci. 33(3), 258–267 (2021).
23. G. Lamy Au Rousseau, J. Bostel, and B. Mazari, “Star recognition algorithm for APS star tracker: oriented triangles,” IEEE Aerosp. Electron. Syst. Mag. 20(2), 27–31 (2005).
24. D. Mortari et al., “The pyramid star identification technique,” Navigation 51(3), 171–183 (2004).
25. Q. Fan and X. Zhong, “A triangle voting algorithm based on double feature constraints for star sensors,” Adv. Space Res. 8(4), 1132–1142 (2018).
26. C. Liebe, “Accuracy performance of star trackers: a tutorial,” IEEE Trans. Aerosp. Electron. Syst. 38(2), 587–599 (2002).
27. H. D. Black, “A passive system for determining the attitude of a satellite,” AIAA J. 2(7), 1350–1351 (1964).
28. J. E. Keat, “Analysis of least-squares attitude determination routine DOAOP,” Computer Sciences Corporation (1977).
29. F. Markley, “Attitude determination using vector observations and the singular value decomposition,” J. Astronaut. Sci. 36(3), 245–258 (1988).
30. A. O. Erlank, “Development of cubestar a cubesat-compatible star tracker,” PhD thesis (2013).
31. G. Wahba, “A least squares estimate of satellite attitude,” SIAM Rev. 7(3), 409–409 (1965).
32. C. C. Liebe, “Accuracy performance of star trackers: a tutorial,” IEEE Trans. Aerosp. Electron. Syst. 38(2), 587–599 (2002).
33. D. Lang et al., “Astrometry net: blind astrometric calibration of arbitrary astronomical images,” Astron. J. 139(5), 1782 (2010).
34. D. Shupe et al., “The sip convention for representing distortion in fits image headers,” ASP Conf. Ser. 347, 491 (2005).
35. F. Markley and J. Crassidis, *Fundamentals of Spacecraft Attitude Determination and Control*, Space Technology Library, Springer New York (2014).
36. J. E. Bortle, “Gauging light pollution: the Bortle dark-sky scale,” *Sky & Telescope* Retrieved 2020-05-29 (2001).
37. J. Mathew et al., “Prospect for UV observations from the Moon. II. Instrumental design of an ultraviolet imager LUCI,” *Astrophys. Space Sci.* 362, 1–11 (2017).
38. A. Kumar et al., “Ultra violet imaging telescope (UVIT) on ASTROSAT,” *Proc. SPIE* 8443, 84431N (2012).
39. A. Kandala et al., “Design and development of a PS4-OP payload for solar spectral irradiance measurements and technology demonstration of small-satellite subsystems,” in *Small Satellite Conf.* (2021).

**Bharat Chandra P** is a PhD student at the Indian Institute of Astrophysics (IIA). He received his M Tech degree in astronomical instrumentation from Calcutta University in 2019. His research interests are UV instrumentation, high altitude ballooning, and optoelectronic instrument design.

**Mayuresh Sarpotdar** received his PhD in astronomical instrumentation from the Indian Institute of Astrophysics, Bangalore, India. He then worked as a research associate at the Satellite Research Centre, Nanyang Technological University. Currently, he is working as a principal engineer at Dhruva Space. His research interests are astronomical instrumentation and electronics system design.

**Binukumar G. Nair** is a visiting scientist at the Indian Institute of Astrophysics (IIA). He was awarded the Marie-Curie fellowship for pursuing a PhD. He received his PhD from The Open University, Milton Keynes, United Kingdom, in 2015. He has worked extensively with UV beamlines in NSRRC, Taiwan, SOLEIL, France and ISA, ASTRID, Denmark. His research interests are UV instrumentation, high altitude ballooning, UV beamlines, and experimental molecular astrophysics.

**Richa Rai** is a postdoctoral fellow at the Indian Institute of Astrophysics. She received her PhD from the Department of Physics, Banaras Hindu University and later joined Osaka University as a foreign young scientist. Her research centers around the optimization of the quantum detection efficiency of alkali halides photocathode-based microchannel plate detectors and gaseous honeycomb-structured RPC detectors.

**Rekhes Mohan** is a scientist (D) at the Indian Institute of Astrophysics, Bangalore. He received his PhD in physics from RRI, Bangalore. His doctoral thesis was based on radio observations using the giant meter wave radio telescope (GMRT, operated by NCRA-TIFR). He joined IIA as a postdoctoral fellow in 2005 and later obtained a permanent position at the institute. His expertise is in software design, data visualization tools, and computational astrophysics. He was a software lead in the TAUVEX project. He is also a member of the payload operations team of the UVIT on board Astrosat.

**Joice Mathew** received his PhD in astronomical space instrumentation from the Indian Institute of Astrophysics, Bangalore, India. He then worked as a visiting instrument scientist at the Max Planck Institute for Solar System Research, Germany, on the ESA/NASA Solar Orbiter mission. He joined the Advanced Instrumentation and Technology Centre as an instrumentation scientist at the Australian National University. He works on space instrumentation for astronomical and Earth observation optical payloads and space qualification. His research interests include astronomical instrumentation, instrument modeling, electro-optical payload development, and space instrumentation.

**Margarita Safonova** is a DST woman scientist at the Indian Institute of Astrophysics (IIA). She received her PhD in astrophysics from the Department of Physics and Astrophysics, University of Delhi in 2002. She has worked in Cambridge, United Kingdom and Tehran, Iran. Since 2006, she has worked at the Indian Institute of Astrophysics (IIA), Bangalore. She is also visiting faculty at MP Birla Institute of Fundamental Research, Bangalore. Her research interests are...
the application of gravitational lensing in astrophysics and cosmology, UV astronomy from space and near-space, exoplanets, and habitability.

Jayant Murthy is a senior professor at the Indian Institute of Astrophysics (IIA). Prior to joining IIA, he received his PhD in physics from the Department of Physics and Astronomy, Johns Hopkins University, Baltimore, Maryland, United States, and later joined as a research scientist there. His research revolves around space missions, starting with a payload on the space shuttle (TAUVEK and UVIT missions), interstellar dust and diffuse radiation field studies, SETI (Search for ExtraTerrestrial Intelligence) project, and many more.