

Smart Camera System On-board a CubeSat for Space-based Object Reentry and Tracking

Ravi Teja Nallapu
Aerospace & Mechanical
Engineering Department
University of Arizona
Tucson, AZ 85721
rnallapu@email.arizona.edu

Aaditya Ravindran,
School of Electrical,
Computer, and Energy
Engineering,
Arizona State University
Tempe, AZ 85287
aarivin11@asu.edu

Himangshu Kalita
Aerospace & Mechanical
Engineering Department
University of Arizona
Tucson, AZ 85721
hkalita@email.arizona.edu

Vishnu Reddy
Lunar and Planetary
Laboratory
University of Arizona
Tucson, AZ 85721
reddy@lpl.arizona.edu

Roberto Furfaro
Systems and Industrial
Engineering Department
University of Arizona
Tucson, AZ 85721
robertof@email.arizona.edu

Erik Asphaug
Lunar and Planetary
Laboratory
University of Arizona
Tucson, AZ 85721
easphaug@lpl.arizona.edu

Jekan Thangavelautham
Aerospace & Mechanical
Engineering Department
University of Arizona
Tucson, AZ 85721
jekan@email.arizona.edu

Abstract— The availability of low-cost, high-performance electronics, sensors, actuators, and communication systems for CubeSats and small-satellites open new avenues in Space Situational Awareness (SSA). Scores of low-cost CubeSats and small-satellites may be rapidly assembled into a constellation to constantly observe natural and man-made objects entering the Earth's atmosphere. Such a low-cost architecture can be constantly upgraded with rapidly advancing sensor technology and is robust to single point failures. SWIMSat (Space Weather and Meteor Impact Monitoring Satellite) is a Low Earth orbit (LEO) based 3U CubeSat demonstrator concept intended to observe objects entering the Earth's atmosphere. The proposed spacecraft uses a smart camera system to observe the Earth during the night time. The camera is placed on the long axis of the spacecraft so that the earth observation is achieved with Nadir pointing. Using this demonstrator, we hope to develop the technology to demonstrate a CubeSat constellation to observe object entering the Earth's atmosphere.

SWIMSat's smart camera system uses an onboard image processing algorithm to detect objects entering the Earth's atmosphere. The object detection algorithm uses a multilayered approach to autonomously detect and track entering objects using a single camera. The object detection approach at its base relies on simple vision-feature detection methods to filter and identify events of interest. This is followed by obtaining dynamic information of the objects including velocity, acceleration, trajectory. Using the hyperspectral and thermal imagers it may be possible to obtain first-order estimates of the composition of the reentering object. Using these estimated statistics, our approach develops a physical simulation model of the observed system and predicts its entry trajectory. In this work, the performance of our object entry detection algorithm coupled with a spacecraft guidance, navigation, and control system is demonstrated by simulating both the physical world and the orbiting observer. These experiments are

conducted in the laboratory using a mock spacecraft mounted on a robotic arm to facilitate 2-axis rotations. These two systems are connected to a computer equivalent to a CubeSat computer. When the experiment starts, the camera detects incoming meteors simulated on a high resolution TV-screen. The detection algorithm coupled with spacecraft control system then track and predict the object reentry trajectory. A thorough description of the detection algorithm, along with the tracking controller is presented in this work. The results of laboratory hardware-in-the-loop experiments are presented. Our work suggests both a critical need and the promise of such a tracking algorithm for implementation of an autonomous, low-cost constellation for performing Space Situational Awareness (SSA).

Keywords— *Space Situational Awareness; Autonomous Object detection; Meteor tracking;*

I. INTRODUCTION

Meteor impacts have been known to have caused catastrophic extinction events on Earth. Very little is known of the risks of these big impacts. For instance, the airburst of the Chelyabinsk meteorite (shown in Fig. 1) in February 2013 culminated in explosion mid-air releasing 500 kilotons of energy [2]. These entry events are not too rare and are expected once every 50 years. Between 1988-2018, US government sensors recorded at least 604 meteor events of various energies ranging from 0.1 Kiloton to the 500 kilotons all around the world [3]. Similar risks also exist from man-made objects such as long-range missiles and reentering space debris [4].

A problem which is common to most of the entry events is that the data on the events is obtained after their impact, which is often too late. This mandates the need for having a space-based monitoring network which can provide timely warning when a dangerous meteor entry event occurs.



Fig. 1. The entry of the Chelyabinsk meteor observed from the ground just before its airburst.

Fortunately, the availability of low-cost, high-performance electronics, sensors, actuators, and communication systems for small spacecraft such as CubeSats open new avenues in Space Situational Awareness (SSA). These small spacecrafts can be rapidly deployed as a constellation to constantly observe natural and man-made objects entering the Earth's atmosphere. Deploying a constellation of spacecraft also makes the observation campaign robust to single point failures. The SWIMSat mission is a first step toward realizing such a space-based surveillance network.

SWIMSat (Space Weather and meteor Impact Monitoring Satellite) is a Low Earth orbit (LEO) based 3U CubeSat demonstrator concept intended to observe objects entering the Earth's atmosphere [1]. The proposed spacecraft uses a smart camera system to observe the Earth during the night time. The camera is placed on the long axis of the spacecraft so that the earth observation is achieved with nadir pointing.

SWIMSat uses a smart camera system to detect objects entering the Earth's atmosphere. The object detection

algorithm uses a multilayered approach to autonomously detect and track the entering objects using a single camera. The object detection approach at its base relies on simple vision-feature detection methods to filter and identify events of interest. This is followed by obtaining dynamic information (in coordination with ground assets) of the objects including velocity, acceleration, trajectory. Using the hyperspectral and thermal imagers it may be possible to obtain first-order estimates of the composition of the reentering object. Using these estimated statistics, our approach develops a physical simulation model of the observed system and predicts its entry trajectory. To this end, a hardware implementation of the SWIMSat's smart camera is carried out.

This paper presents the hardware demonstration of the SWIMSat's smart camera. The work proceeds in the following manner. The operation of the SWIMSat spacecraft is introduced in Section II. The operation and workings of the smart camera system is described in Section III. Section IV describes the hardware testbed built to demonstrate the performance of the smart camera system. The results of the demonstrations are presented in Section V. Finally, conclusions are presented in Section VI.

II. THE SWIMSAT MISSION

The objective of SWIMSat is to observe, detect, and track atmospheric entry events. The concept of operation for the SWIMSat mission is presented in Fig. 2. As seen here, once the spacecraft is commissioned and calibrated, SWIMSat begins its meteor observing mode, where the spacecraft uses its body fixed camera to point towards the Earth's center. Once an entry event occurs in the camera's field of view, the event is detected by isolating the fiery trail being lit. This event is then tracked till the meteor and its trail is no longer visible. Once no event is detected the spacecraft reenters observation mode.

This smart camera systems consists of a CMOS sensor

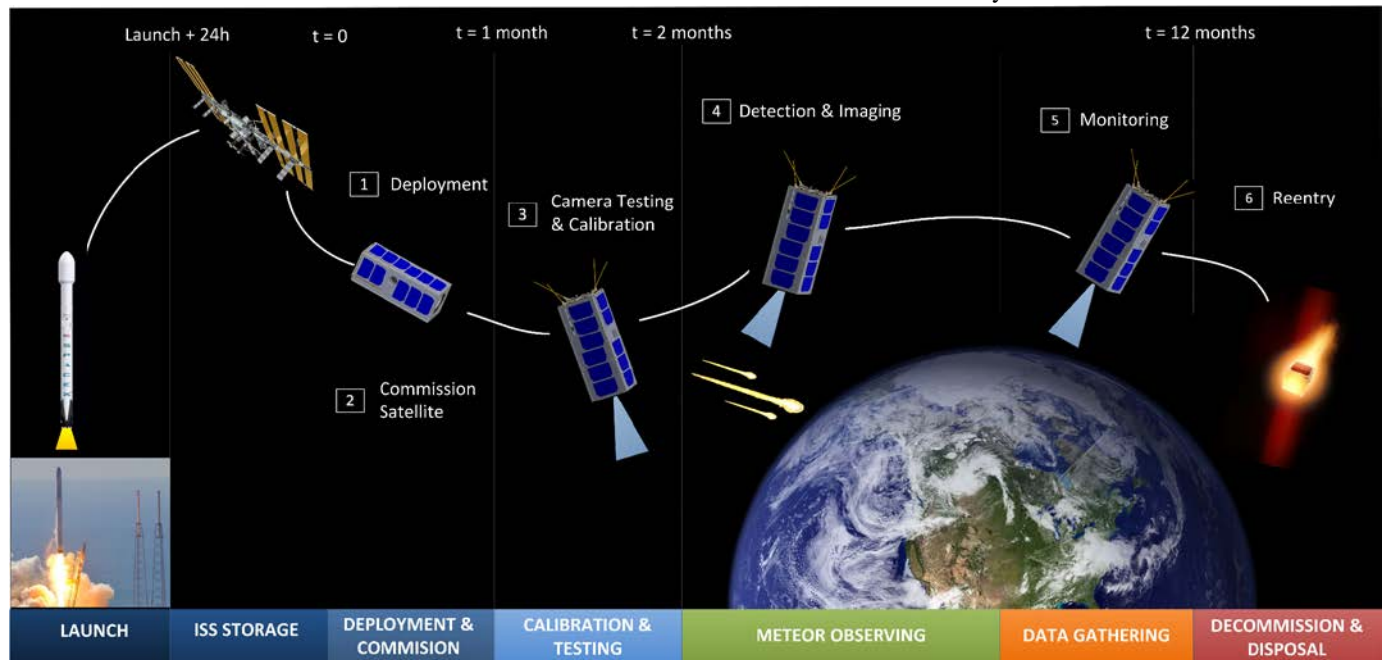


Fig. 2. Concept of operation for the SWIMSat mission.

called PHY367C coupled with a wide-angle field of view lens, Tech Spec C, which provides the spacecraft with wide a field of view of 143°. The algorithm runs onboard the FPGA fabric within the command and data handling (CD&H) subsystem. SWIMSat has selected the CSP C&DH board by Space Micro [5].

When an entry event is observed, the onboard algorithm will detect the event through a color-based thresholding and centroiding algorithms, which computes an attitude error: the difference between the center of the observed event, and center of the camera. The error is then minimized by using the onboard attitude control system. SWIMSat uses MAI-400 by Maryland Aerospace Inc, for its attitude determination and control system (ADCS). The ADCS has 3-axis reaction wheels, and 3-axis magnetorquers, along with an attitude determination suite to maneuver the spacecraft towards the desired attitude. The process is repeated until the entry event is no longer visible. After the data is gathered by the spacecraft, SWIMSat will downlink the data using a UHF radio. The radio communication system consists of the AstroDev Lithium-1 transceiver and a Nanocom ANT 430 antenna [5]. A sectional view of the SWIMSat spacecraft along with its different subsystems are shown in Fig. 3.

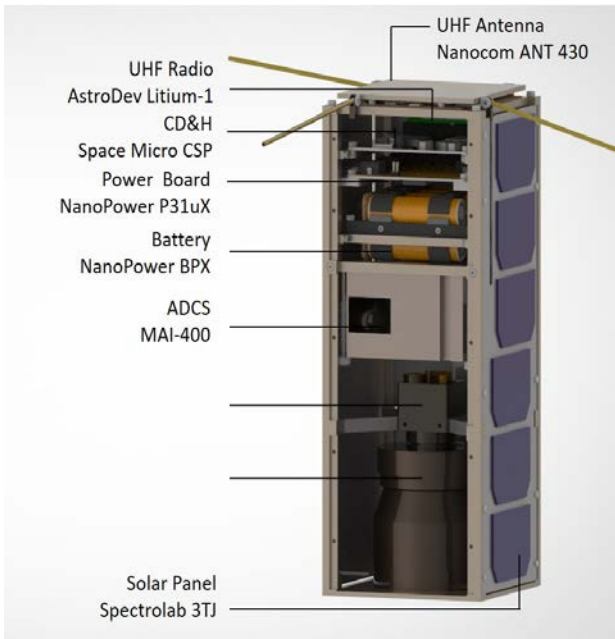


Fig. 3. Sectional view of the SWIMSat spacecraft to present various subsystems

III. SYSTEM SETUP

The SWIMSat smart camera system needs to perform three critical tasks described as follows: (A) Detect an observed meteor/object entry event, (B) Use optical measurements to predict meteor object trajectory, and (C) Generate real-time tracking error statistics. These three tasks are carried out by the on-board command and data handling system. The tracking error is sent to the attitude determination and control system, which then reacts to minimize this error by aligning the center

of the camera with the center of the event. The whole process occurs iteratively until the meteor event is no longer visible or couldn't be tracked any further. The operations of the smart camera are shown in Fig. 4. The three critical tasks are described in further detail below.

A. Entry detection

The smart camera system is constantly on the look-out for entry events when the spacecraft is in eclipse. The detection algorithm first applies a mask to zones note of interest or could be indeterminate such as the Earth's surface. Detection occurs when the algorithm isolates a group of pixels which are unique signatures of a meteor entry event. Once isolated, a box and centroid are drawn to keep track of the event.

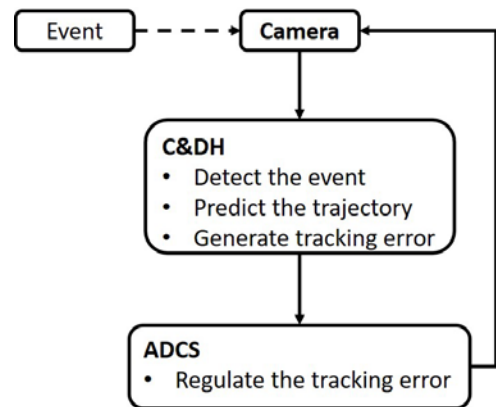


Fig. 4. Operation of the smart camera

The first task of isolating the pixels is achieved by a color-based thresholding method, i.e., observe only events that fall under a certain color range. The thresholding is done with the cylindrical color coordinates which consist of hue, saturation, and value (h, s, v) [6]. Therefore, the entry event detection is done by filtering pixels that lie within certain (h, s, v) bounds, which can be expressed as:

$$(h_{min}, s_{min}, v_{min}) \leq (h, s, v) \leq (h_{max}, s_{max}, v_{max}) \quad (1)$$

The minimum and maximum bounds shown in Eq. 1 are determined during the process of calibration. During the calibration phase, SWIMSat captures images of a known meteor entry event. These obtained images are downlinked to the ground station where they are processed to determine the (h, s, v) bounds. These values are then uplinked to SWIMSat, where it is now ready to begin its observing phase.

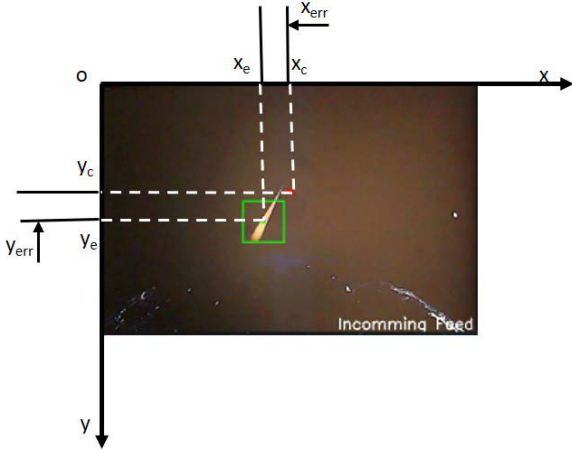


Fig. 4. Simulated camera view of an entry event to show the event center, camera center, and tracking errors

The algorithm proceeds to apply the HSV filter and obtains a thresholded view of the scene. Then a bounding rectangle is fit around the largest group of pixels. In the 2-dimensional image space, the reference centroid of the entry event (x_e, y_e) can, therefore, be defined as the center of the bounding rectangle as shown in Fig. 4. The location of the event in the image space is therefore recorded in terms of corresponding pixel values of (x_e, y_e) in the image frame [7].

B. Trajectory prediction

The trajectory of the entry event is obtained by noting the time history of (x_e, y_e) in the image frame. When an event persists in the field of view, the algorithm takes the centroid coordinates of the last n_f frames and then fits a polynomial curve through the observed trend. This trend is predicted forward in time for the next n_f frames. This process of observing, curve fitting, and predicting, occurs iteratively until the event vanishes from the field of view of the camera.

C. Tracking error

Once the meteor or object has been detected and confirmed, it is continually tracked until it is no longer visible to SWIMSat. A tracking error is defined as the relative distance between the center of the detected event and the center of the camera. The Cartesian components of the tracking error are expressed as

$$(x_{err}, y_{err}) = (x_e - x_c, y_e - y_c) \quad (3)$$

Finally, we summarize the smart camera system algorithm for the readers benefit in Fig. 5.

IV. HARDWARE DEMONSTRATION

A hardware testbed was developed to demonstrate the performance of SWIMSat's smart camera. The flight hardware of the SWIMSat was emulated by low cost hardware.

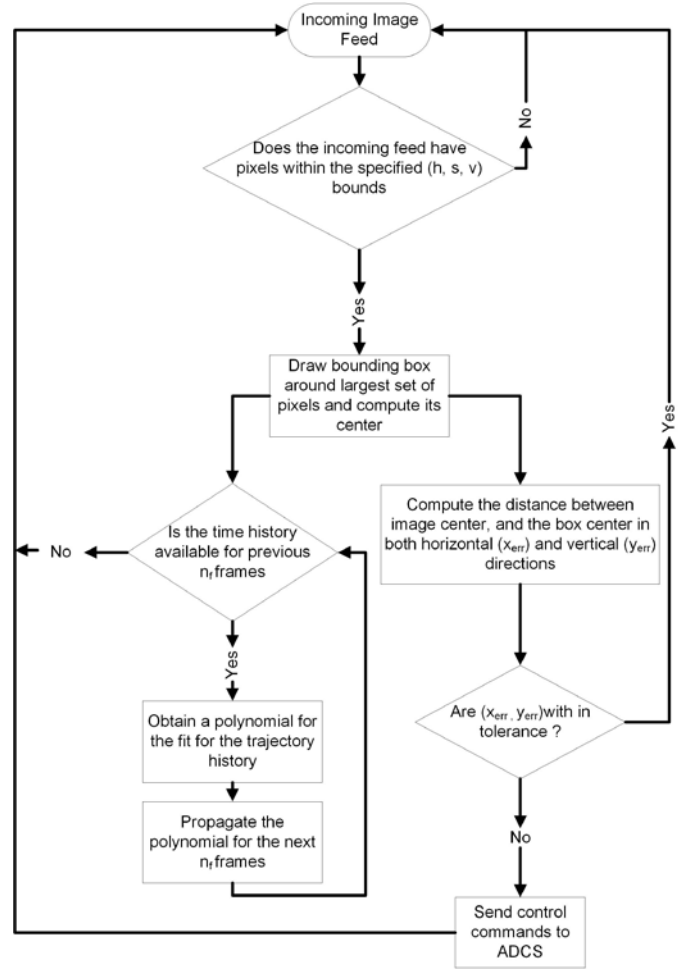


Fig. 5. Operation algorithm of the smart camera system of SWIMSat

The setup consists of the main computer, to mock the C&DH of the spacecraft, a robotic arm to mock the ADCS, and a webcam to mock the spacecrafts imaging payload. A computer simulation of the entering meteor is placed in the field of the camera. The different systems of the hardware demonstration are presented in Fig. 6.

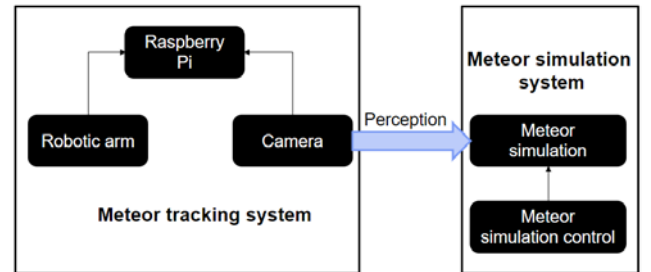


Fig. 6. Systems diagram of the smart camera demonstration.

The individual components of these subsystems are explained as follows:

A. Meteor Tracking System

The main computer for the meteor tracking system is the Raspberry Pi 3 B+ which is an ARM v8 computer running Ubuntu Mate. This closely resembles the cubesat computer, SpaceMicro CSP or SpaceCube Mini, both of which are ARM-based command and data handling (CDH) systems. This Raspberry Pi commands the PhantomX Pincher Robotic arm, which mimics an essential 2-axis Attitude Determination and Control System (ADCS). A camera is mounted on the robotic arm, which mocks the camera used in SWIMSat.

B. Meteor Simulation System

The Meteor Simulation is a physics-based model of a meteor entering the Earth's atmosphere, which is displayed on a Monitor. This simulation is generated by a computer running the Meteor Simulation Control. This controls the brightness, size, color of the meteors displayed on a Nighttime view of the Earth in the background.

C. Robotic arm control

The robotic arm controller is responsible for commanding the angles of the servos, which in turn determines the position of the arm in 3d space. The controller is a python library built using the pypose [8] class as a driver. The controller is always running, waiting to receive the move command from stage 3 (Part C).

D. Meteor Detection

The meteor detection algorithm performs HSV thresholding of the image from the camera and draws a green box around the biggest contour in the image. The trajectory of the object is then tracked. Trajectory prediction of the tracked object is then performed. The pixel difference between the center of the tracked object and the camera center is calculated and is sent to the next stage.

Apart from this, at this stage, trajectory detection is performed by using linear regression. Using this trajectory, a prediction is made for the estimated trajectory of the meteor. Due to the continuous movement of the robotic arm, a smoothing on the images must be performed to stabilize the detection in real-time. This is performed on the detected meteor using the Lucas-Kanade optical flow algorithm.

E. Arm control and meteor detection interfacing

The pixel difference from the motion detection stage is received and is converted into angle difference using a PID controller so that this can be fed into the robotic arm controller. This way, there are two controllers in the system, one to maintain the angle of the servos and one which performs tracking, combining perception data from the camera and the angle data from the robotic arm.

F. Meteor Event Simulation

The meteor event simulation is based on first-order principles of objects moving through the Earth's atmosphere. Various parameters like velocity of the meteor, size of the meteor affect its trajectory through the Earth's atmosphere. A GUI was made using Matlab and VRML based on these principles.

Fig. 7 presents the physical setup of the demonstration testbed.

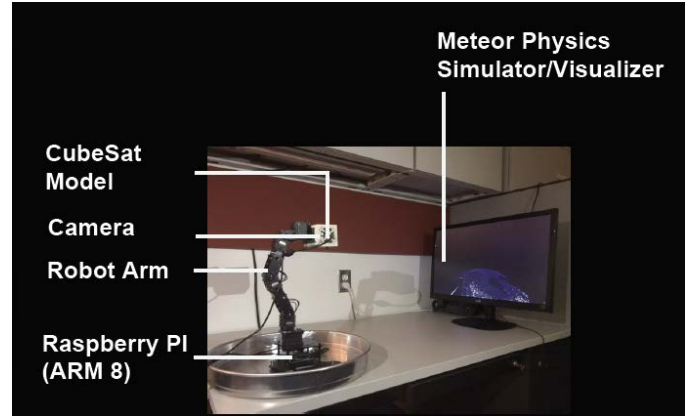


Fig. 7. Smart camera demonstration testbed.

V. RESULTS

A. Calibration

As described in the previous section, the camera system was calibrated to establish the bounds on the minimum and maximum (h, s, v) values. This was done by holding a fixed visual of the meteor on the display system. An uncalibrated camera system would detect all the colors in its field of view. This can be seen in Fig. 8, where the monochromatic thresholded view on the right has is colored white for all the colors in its field of view. Then the (h, s, v) values are tuned on a graphical user interface, until only the pixels of entry event are isolated in the thresholded view as shown in Fig. 8. Calibrating the camera used on the demonstration system yielded the hue-saturation-value bounds which are presented in Table I.

TABLE I. RESULTS OF CALIBRATION

Parameter	Value
$(h_{min}, s_{min}, v_{min})$	(0, 0, 254)
$(h_{max}, s_{max}, v_{max})$	(172, 14, 255)

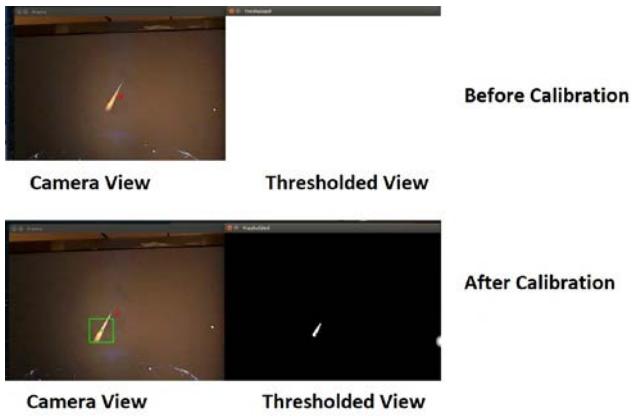


Fig. 8. Calibration of the smart camera system

B. Trajectory estimation

Through testing, the camera system predicted trajectories better when the regression occurred on the previous 6 frames, ie, ($n_f = 6$). It was also observed that the trajectories were acceptably tracked when a linear polynomial was used to fit this data. Fig. 9 shown an image of the demonstration of the trajectory prediction. The green line shows the history of the trajectory from the camera's perspective, while the red line indicates the trajectory predicted for the next 6 frames.

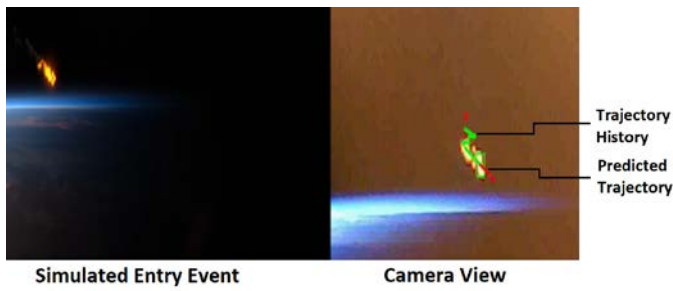


Fig. 9. Trajectory prediction by the smart camera

C. Stationary tracking demonstration

The results of the tracking demonstration are presented in Fig. 10 and 11 respectively, for the case of stationary tracking. In this demonstration, the meteor is held fixed on the screen, similar to the calibration procedure. However now, the robotic arm tries to align the center of the camera with the center of the detected image. Fig. 10 shows how the x and y coordinates (in pixels) of the meteor detected changes with time, as the algorithm is tracking it.

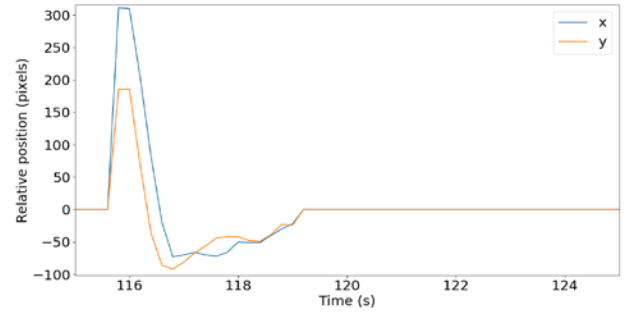


Fig. 10. Cartesian coordinates of the relative position of the event center with respect to the camera center in the image frame during stationary tracking.

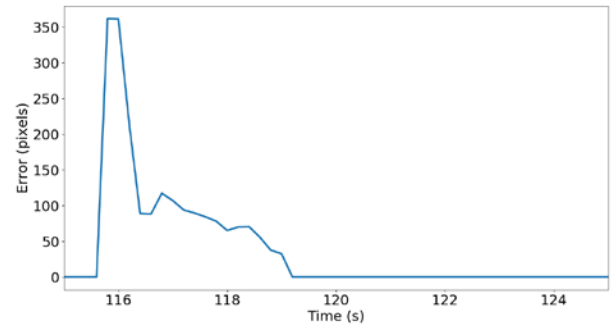


Fig. 11. Magnitude of the tracking error of event center with respect to the camera center in the image frame during stationary tracking.

D. Moving object Tracking demonstration

In this demonstration, the simulated meteor now moves on the screen. The robotic arm of the camera system then aligns the centers of the event and the camera. Fig. 12 and 13 show the tracking of a stationary meteor on the display. After the meteor is tracked, the meteor tracking is turned off and the robotic arm returns to its home position. This can be seen from the peak at around 116 seconds in the graphs. It takes about 3 seconds to track an object on an average.

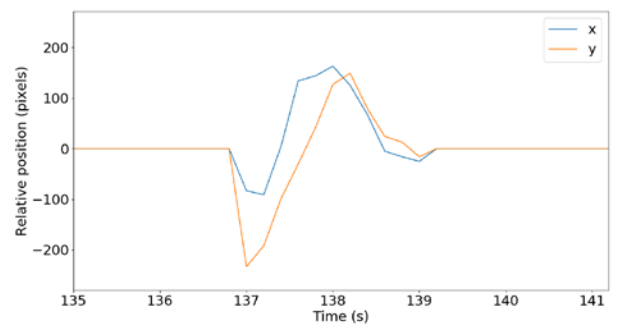


Fig. 12. Cartesian coordinates of the relative position of the event center with respect to the camera center in the image frame during moving meteor tracking.

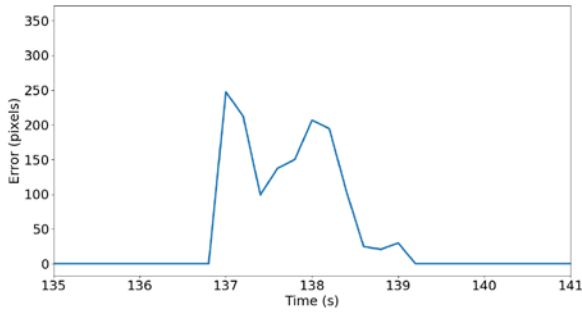


Fig. 13. Magnitude of the error in the case of tracking the moving meteor.

Fig. 12 and 13 show the tracking of a meteor moving on the display, generated randomly. A meteor is detected at about 137 seconds and is tracked till it moves out of the display, and no object is being detected by the algorithm. After some frames of non-detection, the robotic arm is reset to its home position. This takes about 4 seconds.

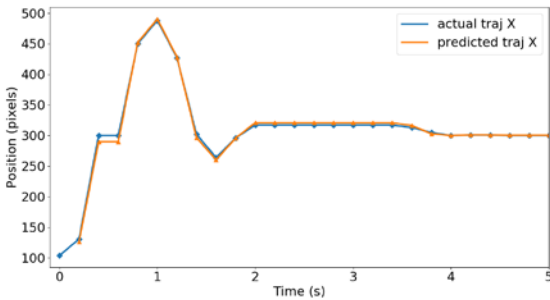


Fig. 14. x-components of the predicted trajectory and the recorded event position.

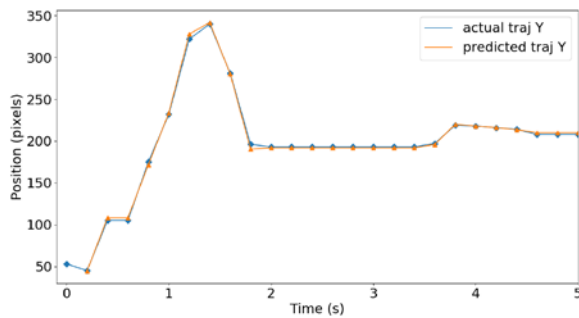


Fig. 15. y-component of the predicted trajectory of the predicted trajectory and the recorded event position.

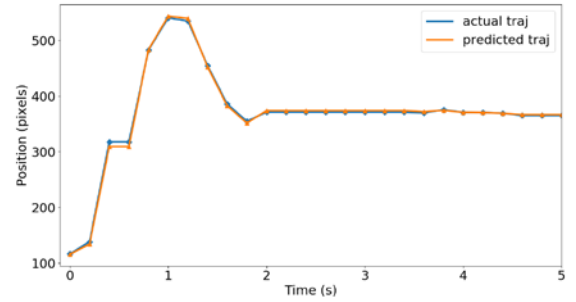


Fig. 16. Magnitude of positions of the predicted trajectory the recorded event.

The results of the trajectories predicted in the case of moving object simulations are presented here. Fig. 14 and 15 show the X and Y coordinates of the trajectory, while Fig. 16 shows the Euclidean position of the predicted and actual trajectories of the meteor over time. These plots show a close correlation between the predicted trajectory and actual trajectory.

The meteor simulation is a physics model of meteor entry events. Care has been taken to emulate a real-world Nighttime view of the Earth, including lights. It is evident from the detection results that it is robust, in terms of eliminating noise from lights on Earth. Tracking of the detected meteor events, both for stationary objects and moving objects has been shown. Tracking is considered successful when the error in the relative position of the detected meteor event and the camera center is reduced to a few pixels. Along with detection and tracking, the simulation system demonstrates trajectory prediction of meteor events. The trajectory prediction results show that the predicted trajectory converges with the actual trajectory, with an error of five pixels. These results can be taken as a preliminary proof, demonstrating the capability of SWIMSat's science algorithm.

VI. CONCLUSION AND FUTURE WORK

This paper presents the meteor entry and impact detection system onboard the SWIMSat mission. A CubeSat in Low Earth Orbit (LEO) or higher altitude has an excellent vantage point to monitor meteor entry events and impacts compared to ground observation. A ground demonstration system of meteor detection and tracking has been developed and is set up to mimic conditions expected on a CubeSat in LEO. Meteor detection and tracking are performed with trajectory prediction in order to maximize camera view of the event. The algorithm is being demonstrated on a low-power ARM 8 based processor like Raspberry Pi which makes porting onto a CubeSat computer system like the Space Micro CSP with an ARM 9 readily feasible.

The current implementation is limited by the speed of the meteor entering the Earth's atmosphere. Efforts are being made to improve the tracking accuracy. The capability of the SWIMSat algorithm can be extended to a thermal camera or a hyperspectral imager to better detect meteor entry events. The images from multiple cameras maybe fused to obtain additional

information about the falling meteor. Moreover, improvements could be made to the prediction algorithm to use lesser system resources.

REFERENCES

- [1] Hernandez, V., Gankidi, P., Chandra, A., Miller, A., Scowen, P., Barnaby, H., Adamson, E., Asphaug, E., Thangavelautham, J., 8220; SWIMSat: Space Weather and Meteor Impact Monitoring using a Low-Cost 6U CubeSat," Proceedings of the 30th Annual AIAA/USU Conference on Small Satellites, AIAA, Logan, Utah, 2016.
- [2] Popova, O. P., et al. "Chelyabinsk Airburst, Damage Assessment, Meteorite Recovery, and Characterization." *Science*, 342 (6162), 2013. Post-Print Version, vol. 342, no. 6162, 2013, pp. 1069-1073.
- [3] "Fireball and Bolide Data." NASA, NASA, neos.jpl.nasa.gov/fireballs/.
- [4] Hankey, Wilbur L., and Inc ebrary. *Re-Entry Aerodynamics*. American Institute of Aeronautics and Astronautics, Washington, D.C, 1988, doi:10.2514/4.862342.
- [5] Hernandez, Victor and Ravindran, Aaditya. "On-Orbit Demonstration of the Space Weather and Meteor Impact Monitoring Network." (2017).
- [6] Zetterlind III, Virgil E., and Stephen M. Matechik. Hue-Saturation-Value Feature Analysis for Robust Ground Moving Target Tracking in Color Aerial Video, vol. 6235, SPIE, 2006, doi:10.1117/12.666728.
- [7] Suzuki, Satoshi, and Keiichi Abe. "Topological Structural Analysis of Digitized Binary Images by Border Following." *Computer Vision, Graphics and Image Processing*, vol. 30, no. 1, 1985, pp. 32-46.
- [8] Vanadium Labs. "GitHub – vanadiumlabs/pypose: PyPose and NUKE",