Development of On-board Image Processing Algorithm to Detect Lunar Impact Flashes for DELPHINUS*

Masahiro FUJIWARA,1)† Satoshi IKARI,1) Hirotaka KONDO,1) Ryota FUSE,2) Yosuke MASUDA,2) Shinetsu ABE,2) Masahisa YANAGISAWA,3) Kenji YAMAMOTO,3) Hajime YANO,4) and Ryu FUNASE1)

1)Department of Aeronautics and Astronautics, The University of Tokyo, Tokyo 113–8656, Japan
2)Department of Aerospace Engineering, Nihon University, Funabashi, Chiba 274–8501, Japan
3)The University of Electro-Communications, Chofu, Tokyo 182–8585, Japan
4)Institute of Space and Astronautical Science, JAXA, Sagamihara, Kanagawa 252–5210, Japan

DELPHINUS is a camera system mounted on EQUULEUS, which is planned to be launched using NASA’s Space Launch System EM-1 in 2021. DELPHINUS aims to investigate size distribution, influx ratio, and daily variation of meteoroids in the cislunar space through observations of lunar impact flashes (LIFs) from the far side of the moon. DELPHINUS will observe the moon’s surface with the 60-fps camera modules to capture the flashes that are short duration phenomena. All image data cannot be downlinked due to constraints in memory size and communication capability. Therefore, an on-board image processing algorithm was developed to reduce downlinked data size by extracting only necessary pixel data including LIFs. Three experiments using three simulators were demonstrated to verify the real-time processing performance and detection capability. This paper reports the details of the proposed algorithm and the verification results.

Key Words: CubeSat, Science Payload, Lunar Impact Flashes, On-board Image Processing

1. Introduction

When meteoroids strike the moon surface at several tens of km/s, faint flashes having durations of approximately 0.01 to 0.1 s are generated by part of the kinetic energy that is converted to light energy.1) These phenomena are referred to as lunar impact flashes (LIFs). Figure 1 shows an example of LIFs observed on 24 Nov. 2017 by a telescope with a 28-cm aperture at The Univ. of Electro-Comm. (Yanagisawa et al., unpublished data). Observations of LIFs are important for investigating size distribution, influx ratio, and daily variation of meteoroids in the cislunar space. These parameters contribute to clarifying the origin of life and water on the earth and the risk assessment of the moon surface, which is planned as a new base for space development. So far, about 430 flashes have been detected over a period of 10 yr through ground-based observations conducted by NASA.2) Figure 2 shows the locations of the LIFs. The locations are biased because ground-based observations are conducted against the moon. In particular, space-based observations from the second Lagrangian point of the earth and the moon (EML2) have further advantages, such as no earthshine and the closeness of the distance to the moon. In addition, the results of space-based observations from EML2 complements those of ground-based observations because LIFs have never been observed from the far side of the moon. These days, much attention is being paid to space-based observations of LIFs, and Milan University3) has proposed the LUMIO mission to observe LIFs from EML2.

Figure 3 shows EQUULEUS, which is a 6U CubeSat planned to be launched using NASA’s Space Launch System Exploration Mission-1 (EM-1) rocket in 2021, and demonstrates several technological and science missions during and after the flight to EML2.4) DELPHINUS is a scientific instrument of EQUULEUS that will be used to observe LIFs from a halo orbit around EML2. DELPHINUS aims to reveal the unknown size distribution, influx ratio, and daily variation of meteoroids having diameters of several centimeters or meters through space-based LIF observations.5) DELPHINUS has two camera modules that will capture images at a high frame rate in order to detect LIFs, and therefore all image data cannot be stored in its memory or transmitted to the ground stations due to constraints in memory size and communication capability. Therefore, DELPHINUS has to detect LIFs on-board and store only images deemed necessary, including the LIFs detected to save memory consumption and reduce downlinked data size. However, it is difficult for the LIF observation to distinguish between true flashes and camera system noises in a large amount of image data. If a threshold to detect LIFs is too low, noises can be misrecognized as true LIFs, causing increases in the stored and downlinked data. To overcome this technical issue, an on-board LIF detection algorithm was developed. In the algorithm, firstly, an average image is calculated in real-time to subtract background. After that, from the subtracted image, true flashes are detected by comparing several targeted pixels with thresholds simultaneously based on the point spread function.
of true flashes. To verify the performance of the proposed algorithm, we demonstrated three types of experiments using the following simulators: MATLAB-based software simulator, FPGA-based hardware simulator, and hardware simulator. MATLAB-based software simulations were carried out to estimate flash detection rates and appropriate thresholds. In the FPGA-based hardware simulation, pixel data like LIFs were generated inside the FPGA of the system. The emulated LIFs were input into the on-board algorithm while changing the LIF appearance frequency and brightness. The hardware simulator can generate signals simulating those of a LIF having a first to fifth magnitude by changing the flashing time of the laser. The end-to-end experiments were achieved using this hardware simulator.

The details of the design of DELPHINUS are described in Section 2, and the details of the on-board algorithm to detect LIFs are described in Section 3. The results of the three types of simulations are described in Section 4. Finally, Section 5 concludes the paper with a summary.

2. System Overview of DELPHINUS

The requirements for DELPHINUS to achieve the LIF observation are described as follows:

- The capability to observe LIFs comparable to the ninth magnitude from the ground, corresponding to the fourth magnitude from the halo orbit, in order to detect more than 100 LIFs during the whole observation period which is one month in the best orbit.

- The capability to detect faint flash phenomena for approximately 0.01 to 0.1 s on-board in order to efficiently downlink the candidates of true LIFs.

- The lens system to prevent the stray light from the sun, the earth, and the moon from entering the cameras.

Table 1 shows the specifications of DELPHINUS. A camera module, WATEC T065, is utilized because the frame rate is high enough to detect flashes with a duration of 0.01 to 0.1 s. As shown in Fig. 4, DELPHINUS has two camera modules to determine whether or not detected LIFs are true by comparing two images captured by each camera on the ground. Additionally, DELPHINUS has a hood designed to shield the stray light and incident light at an angle of...
45 deg or more. The overall size is about 0.5U, that is; 10 cm × 5 cm × 10 cm.

The electrical system of DELPHINUS has an FPGA chip implemented for high-speed image processing. This FPGA has a soft-core processor, and it is possible to design both hardware and software by programming in Verilog-HDL and C language, respectively. Three types of memories are implemented on the electric board: a NAND flash, an SDRAM, and an SRAM. The NAND flash is the only non-volatile memory in this system, and this is used to store the extracted LIF image data. The SDRAM is mainly used to store the raw image data, and the SRAM is a working space for the soft-core processor (CPU). Figure 5 shows the data flow of DELPHINUS. The captured image data is transferred into the SDRAM by FPGA hardware logic and stored temporarily. When LIFs are detected or the CPU receives a command to store image data on the NAND flash, the CPU transfers the image data to the NAND flash for downlink.

The data size of a raw image is

\[ \text{data size of a raw image} = 659 \times 494 \times 10 \div 8 = 406932.5 \text{ bytes} \]

\[ \approx 397 \text{ Kib} \]  

(1)

The data rate of the captured images by two cameras at 60 fps is 46.6 MiB/s. The NAND flash, the storage of which is 512 MiB, is filled in only 11 s at this data rate. It is also impossible to downlink all image data to the ground station because the downlink data rate of EQUULEUS is up to 32 Kbps. This is why the system needs an on-board image processing algorithm to extract and store only the necessary image data.

3. On-board Image Processing Algorithm to Detect LIFs

The objective of an on-board image processing algorithm is to detect LIFs on-board and extract only the pixels including a LIF in real-time to reduce the amount of image data stored and downlinked. As shown in Fig. 6, the algorithm can be divided into two main parts in terms of those processing cycles. The first one is for LIF detection and the second one is for LIF extraction. The detection part is composed of three subtasks: calculation of an average image, subtracting the average image from the raw image to remove background, and comparing the values of pixels with the thresholds to detect LIFs. These subtasks run simultaneously on the hardware inside the HW logic with the operation clock of the camera (24.54 MHz). Although DELPHINUS has two camera modules, the images from only one camera are input into the detection algorithm due to the computational limitation. In the extraction part, the following subtasks run sequentially on the CPU. Firstly, the extraction part checks the list of detected LIFs. When the list is not empty, the CPU buffers the LIF data, and finally stores the data on the NAND flash. The details of each part are described in Section 3.1 and 3.2.

To detect the fourth-magnitude LIFs, the algorithm utilizes the point spread function of the LIFs. Figure 7 shows a point spread function obtained from the actual image of a fourth-magnitude fixed star observed by the Proto-Flight Model (PFM) of DELPHINUS. The refractive index of air is 1.0003, which is slightly different from that in a vacuum (1.0000). Therefore, one of the camera lenses of the PFM dedicated to air conditions for the pre-flight calibration. The PSF indicated approximately 5 pixels with the full-width half-maximum of 2.62 pixels, which satisfies the requirements for the camera design.
The point spread function of the LIFs is designed to spread over several pixels of the cameras. On the other hand, the noises rarely appear with the same spread as the LIFs because the noises in the image are random variations of each pixel’s brightness. This property is used to distinguish between true LIFs and the noises in the algorithm. According to the analysis based on the specifications of the cameras and the lens system, the probability that the value of noises exceeds the threshold for the fourth magnitude is at least 4.75%, as shown in Fig. 8.

The expected value of false detection per second is

\[
659 \times 494 \times 0.0475 \times 60 = 9.28 \times 10^5 \text{ counts/s.} \tag{2}
\]

The probability that all values of 3 x 3 pixels are larger than the thresholds due to noises is

\[
0.0475 \times 0.0708^4 \times 0.0968^4 = 9.96 \times 10^{-11}. \tag{3}
\]

This probability is much lower than that of the case where only one pixel is compared with the threshold. If 3 x 3 pixels are simultaneously compared with the thresholds, the expected value of false detection is

\[
659 \times 494 \times 9.96 \times 10^{-11} \times 60 = 1.95 \times 10^{-3} \text{ counts/s.} \tag{4}
\]

While the false detection rate is 7.02 counts/h, the number of LIFs that can be detected by the thresholds for the fourth magnitude is approximately 0.5 counts/h. We can distinguish whether the extracted image data is true LIFs or noises out of only eight images per hour.

### 3.1. LIF detection

Figure 9 shows the details of the detection part. The following processes are executed at the same time when new pixel data is transferred from the camera. Firstly, an average image is calculated from 2 to 64 frames and subtracted from a raw image in order to remove background such as offsets derived from CCDs and the stray light from the sunlit side of the moon. Secondly, the pixel values of three lines including the new pixel data are temporarily stored in registers on the FPGA. Thirdly, the value of each 3 x 3 pixels out of the image data of three lines is simultaneously compared with thresholds based on the point spread function of a LIF. If the values of one or more pixels are larger than the thresholds, those 3 x 3 pixels are determined as a candidate of a LIF. If LIFs are smaller than 3 x 3 pixels or if irregular pixels are generated by radiation, the number of pixels to determine LIFs can be changed in the range from one to nine pixels. When a LIF candidate is detected, the image frame number and the row and column numbers in the image are written in the FIFO.

According to the position of EQUULEUS in the halo orbit, the field of view of DELPHINUS changes as shown in Fig. 10. When DELPHINUS is far from the moon, for example, at 40,000 km and 60,000 km in Fig. 10, not only the moon surface but also other areas are within a frame. A rectangular mask to limit the detection areas in the frame is implemented since the pixel data outside of the moon surface need not be compared with the thresholds.

### 3.2. LIF extraction

Figure 11 shows the details of the extraction part. All of the raw images captured by two cameras at 60 fps are stored in the range from 0 to 119 MiB of the SDRAM. In short, the LIFs must be extracted within 1 s before a raw image is overwritten by a new image. The CPU accesses the FIFO at 60 times/s to check whether or not the FIFO has the address data of LIF. If LIFs has been detected, the CPU reads out the address where the central pixel value of a LIF candidate is stored. Then, the CPU transfers the values of 11 x 11 to 31 x 31 pixels including a LIF candidate into a buffer of the SRAM. At this moment, pixel data of the same position in the same frame captured by the other camera are stored in the buffer to compare with the other data on the ground. When the data of the other camera are extracted, its position...
of the extracted pixels can be compensated to remove the misalignment between the two cameras. In addition, according to a command, pixel data at the same position in multiple frames before and after the LIF frame detected are also stored to capture the flash phenomena from the start to the end. The buffer data are sent to the NAND flash when the size of the buffer exceeds the page size of the NAND flash (2,048 bytes) to minimize the access times to the NAND flash. If the buffer is not less than 2,048 bytes for 30 s, the LIF extraction algorithm is programmed to transfer the data in the buffer to the NAND flash. If the number of pixels and frames extracted is large, the quality of the LIF data extracted is high. However, large data cannot be extracted and stored within a second when the false detection rate is high. For example, it takes 0.12 s to transfer a LIF data size of 31 × 31 pixels and four frames into the NAND flash. We have to determine the proper thresholds and the number of pixels and frames extracted using simulations and on-orbit tests.

In the detection part, if a LIF that is larger than 3 × 3 pixels is generated, redundant detection occurs in the same LIF. For example, a bright LIF that spreads in 5 × 5 pixels causes detection nine times. In order to prevent this redundant detection, the CPU does not buffer the data when the difference between the address of a new LIF and that of a previous LIF is less than 2,048 bytes for 30 s, the LIF extraction algorithm is programmed to transfer the data in the buffer to the NAND flash. If the number of pixels and frames extracted is large, the quality of the LIF data extracted is high. However, large data cannot be extracted and stored within a second when the false detection rate is high. For example, it takes 0.12 s to transfer a LIF data size of 31 × 31 pixels and four frames into the NAND flash. We have to determine the proper thresholds and the number of pixels and frames extracted using simulations and on-orbit tests.

In the detection part, if a LIF that is larger than 3 × 3 pixels is generated, redundant detection occurs in the same LIF. For example, a bright LIF that spreads in 5 × 5 pixels causes detection nine times. In order to prevent this redundant detection, the CPU does not buffer the data when the difference between the address of a new LIF and that of a previous LIF is less than a specific value that can be changed by a command.

4. Verification of Algorithm

To verify the performance of the algorithm mentioned in the previous section, we demonstrated three types of simulations: MATLAB-based software simulation, FPGA-based hardware simulation, and hardware simulation.

4.1. MATLAB-based software simulation

Firstly, we developed a LIF software simulator using MATLAB for software experiments to consider and choose an effective LIF detection algorithm. The simulator generates full-frame images with background and random noises and LIFs. The brightness of these noises and LIFs are controlled by users. The number of LIFs in a frame and their positions are randomly decided for each frame. The simulator also emulates image signal output behavior of the DELPHINUS cameras. According to the camera specifications document, the images generated are divided to 347,776 pixels including invalid padding pixels, and the count value of each pixel is output to a buffer one-by-one with the pixel clock, the line flag, and the frame flag. This kind of camera output protocol emulation is important for real-time image processing when using low-performance computers because the computers usually cannot have enough memory or computational speed for full-size image processing. The algorithms to store and monitor several pixels or lines are required.

In the MATLAB software simulator, we demonstrated two types of LIF detection algorithms: a single-pixel detection algorithm and a multi-pixel detection algorithm. The algorithms are coded in MATLAB, thus they are not suitable for the FPGA. As mentioned above, the single-pixel algorithm is weak for random noises. Actually, in an experiment using the MATLAB simulator applying the same noise situation as the cameras, the single-pixel algorithm detected hundreds of false LIFs in a frame, even if the frame had only one LIF. However, the multi-pixel algorithm detected all true LIFs without false detection. Using the MATLAB simulator, we verified the performance of the multi-pixel detection algorithm, and started to implement the algorithm onto the FPGA hardware logic.

4.2. FPGA-based hardware simulation

We developed FPGA logic that can generate test signals like LIFs and add them to raw images captured by the cameras. We can control the size of pixels, frequency of appearance, and brightness of the emulated LIFs. A real-time experiment to detect and extract fourth-magnitude LIFs was demonstrated using the FPGA-based hardware simulator. The brightness of the emulated LIFs was set based on the results of the observations of the fourth-magnitude fixed star image taken from the ground. In this simulation, one emulated LIF with a size of 3 × 3 pixels was generated per second. This frequency is high enough to verify that the test signals can be detected and extracted in real-time since the number of LIFs that can be detected by the thresholds for the fourth-magnitude LIF is 0.5 counts/h.5) As mentioned above, Figure 12 shows images of the extracted LIF during this simulation and the fourth-magnitude fixed star captured
from the ground. The algorithm could detect the emulated LIF having nearly the same point spread function as the fixed star. As a result of carrying out the experiments for 20 seconds more than 10 times, all of the LIF emulations were able to be detected without false detection.

We also demonstrated this experiment in a temperature cycling test of the integrated flight model of EQUULEUS. Noises of the cameras were measured at −20 and 50°C in the integrated system. Although noises of the integrated system are larger than that of DELPHINUS alone, it was confirmed that the false detection rate was sufficiently low in the detection of LIFs comparable to the fourth-magnitude fixed star.

4.3. Hardware simulation

A hardware simulator was developed to demonstrate an experiment that is close to the actual situation. The hardware simulator is composed of a pulse generator, a filter, and a collimator, as shown in Fig. 13. A laser diode is used as a light source and its flash duration can be changed according to the pulse width generated by the pulse generator. The flash is generated as a LIF ranging from the first to fifth magnitudes through the collimator and filter.7) Figure 14 shows the configuration of the hardware simulation. In this experiment, a splitter was used to input the fourth-magnitude flash generated by the hardware simulator to both cameras. The extracted LIF images are shown in Fig. 15. Although the brightness of the emulated LIF generated by the hardware simulator was lower than that of the fourth-magnitude fixed star, the algorithm could detect it. A total of eight frames, including the detected frame, were extracted since the frames captured at the same time by the other camera were also extracted, as described in Section 3.2. The extracted frames were shifted by one frame between cameras because of the clock difference of each camera. It can also be seen that a diminishing flash was captured in the frame immediately after the detected frame. As a result, it was confirmed that LIFs could be detected and extracted in this end-to-end experiment using the hardware simulator.

5. Conclusion

One of the science missions of EQUULEUS is the observation of lunar impact flashes (LIFs) from a halo orbit around the second Lagrangian point of the earth and the moon (EML2) in order to investigate size distribution, influx ratio and daily variation of meteoroids in the cislunar space. An on-board image processing algorithm was developed to detect the faint and short-duration phenomena. The proposed algorithm utilizes the difference between a point spread function of a LIF and noise distribution on the detector. The algorithm successfully detected LIFs of the fourth magnitude efficiently by simultaneously comparing 3 × 3 pixels out of the image with the thresholds. Several functions are also implemented such as a detection mask, misalignment correction, and multi-frame LIF extraction to achieve the mission more reliably.

The performance of the algorithm was verified through three types of experiments. In MATLAB-based simulation, false detection rates and appropriate thresholds were estimated. In FPGA-based hardware simulation, test signals like LIFs generated inside an FPGA hardware logic were added to a raw image to confirm that the algorithm can detect and store LIFs in real-time. In the hardware simulation, the first to fifth-magnitude flashes were generated by a hardware simulator directly input to DELPHINUS. LIFs of the fourth magnitude can be detected under conditions that are more realistic in this end-to-end experiment. In addition to these three experiments, we have already conducted observations of fixed stars and a study of the temperature effect on the point spread function in a vacuum environment using the DELPHINUS flight model. We confirmed that the algorithm could detect the emulated LIF based on the observations of the fourth-magnitude fixed stars and the temperature has little effect on the point spread function in the flight model. Even if the point spread function of the actual LIF is different from our assumption, we can tune the thresholds using some stars visible from the halo orbit. Thus, the algorithm could be verified and the thresholds can be tuned during flight.
Acknowledgments

The authors appreciate COSINA Co., Ltd., WATEC Co., Ltd., IMAGETECH Co., Ltd., and all of the members of the EQUULEUS project for developing of DELPHINUS.

References

1) Yanagisawa, M. and Kisaichi, N.: Lightcurves of 1999 Leonid Impact Flashes on the Moon, *Icarus*, **159** (2002), pp. 31–38.
2) Moser, D. E.: Lunar Impacts NASA, https://www.nasa.gov/centers/marshall/news/lunar/lunar._impacts.html (accessed April 17, 2019).
3) Cipriano, A., Dei Tos, D. A., and Topputo, F.: Orbit Design for LUMIO: The Lunar Meteoroid Impacts Observer, *Front. Astron. Space Sci.*, **5** (2018), pp. 1–23.
4) Funase, R., Ikari, S., Kawabata, Y., Nakajima, S., Nomura, S., Kakihara, K., Takahashi, R., Yanagida, K., Matsushita, S., Ishikawa, A., Funabiki, N., Murata, Y., Suzumoto, R., Shibukawa, T., Mori, D., Fujiwara, M., Tomita, K., Koizumi, H., Asakawa, J., Nishi, K., Yoshikawa, I., Yoshioka, K., Hirai, T., Abe, S., Fuse, R., Yanagisawa, M., Miyoshi, K., Kobayashi, Y., Tomiki, A., Torii, W., Ito, T., Kuwabara, M., Yano, H., Ozaki, N., Ikenaga, T., and Hashimoto, T.: Flight Model Design and Development Status of the Earth-Moon Lagrange Point Exploration Cubesat EQUULEUS Onboard SLS EM-1, 32nd Annual AIAA/USU Conference on Small Satellites, Logan, USA, SSC18-VII-05, 2018.
5) Abe, S., Yanagisawa, M., Yano, H., and Funase, R.: Mining Earth’s Mini-moons near Cislunar Space by 6U Spacecraft, Proceedings of the 60th Space Sciences and Technology Conference, Japan, Hakodate, 2016, pp. 1–3.
6) Fuse, R., Abe, S., Yanagisawa, M., Funase, R., and Yano, H.: Space-based Observation of Lunar Impact Flashes, *Trans. JSASS Aerospace Technology Japan*, **17** (2019), pp. 315–320.
7) Yamamoto, K.: Development of a Hardware Simulator for Lunar Impact Flashes, Master Thesis, The University of Electro-Communications, 2017 (in Japanese), http://id.nii.ac.jp/1438/00008596/