On-board Processing of Vector Code Correlation Algorithm for Image-based Navigation System

Genki Ohira[1*], Shuya Kashioka[1], Yuki Takao[2], Miyuki Kadokura[3], Taketoshi Iyoda[3], Yuichi Tsuda[4]
[1]SOKENDAI, Japan; [2]The University of Tokyo, Japan; [3]SOKA University, Japan;
[4]ISAS/JAXA, Japan

-Abstract-

Image-based navigation is using to estimate the relative position of a spacecraft with a target body and control it for explorations to small solar system bodies. In the case of Hayabusa2, the asteroid-relative position is estimated by ground operators comparing a captured image and the point cloud model of the asteroid with landmark information. On the other hand, in the case of explorations to small bodies farther than Main-belt, the communication delay is unacceptably large for the asteroid-relative feedback guidance. This situation becomes worse for larger asteroids because the time constant of the dynamics becomes faster in such cases. Therefore, importance of on-board image-based navigation is highlighted for explorations to far distant small bodies. To accomplish on-board image-based navigation that can be applied to various bodies, the Vector Code Correlation (VCC) algorithm is focused on. The VCC algorithm is deemed suitable for FPGA. In this study, the VCC-based on-board position estimation method was developed and implemented on an FPGA board. The estimation accuracy and computation time were evaluated by comparing the proposed method with other position estimation methods applied to the actual missions such as Hayabusa2 .

画像航法用ベクトル符号相関アルゴリズムの オンボード処理化に関する研究

-摘要-

はやぶさ２などの太陽系小天体探査において，探査機の目標天体相対位置を高精度で推定・制御するために画像航法が使用されている．はやぶさ２の場合は地球との通信を使用する．地上で，特徴点データを持つ点群モデルと撮影画像を比較することで位置を推定している．しかしながら，メインベルト以遠の超遠方天体探査の場合，フィードバック制御上許容できないほどに地球との通信遅延が大きくなる．また，目標天体が大きくなるほど運動の時定数が速くなるため，この時間遅れはより大きくなる．したがって，超遠方天体を対象として高精度で探査機の位置を制御するには，画像航法のオンボード処理化が必要不可欠である．様々な天体へ適用可能なオンボード画像航法を実現するため，私たちは FPGA による処理に適したベクトル符号相関（VCC）アルゴリズムに注目した．本研究では,VCC アルゴリズムを用いた位置推定手法を開発し，FPGA 評価ボードに実装した．また，はやぶさ 2 などの実際のミッションで使用されている位置推定手法と提案手法を比較することで，推定精度と推定時間を評価した．

Nomenclature

f	:	luminance of a reference image (8bit)
g	:	luminance of a template image (8bit)
sd	:	discretized slope of luminance (4bit)
sx	:	discretized slope of luminance in the x direction (2bit)
sy	:	discretized slope of luminance in the y direction (2bit)
th	:	threshold level for discretization
V	:	velocity
X	:	asteroid-relative position in Home Position (HP) coordinates
x	:	position in camera coordinates, pixel
Y	:	asteroid-relative position in HP coordinates
y	:	position in camera coordinates, pixel
Z	:	asteroid-relative position in HP coordinates

Subscripts

r	:	reference image
t	:	template image

1. Introduction

In recent years, a lot of explorations to small solar system bodies have received much attention around the world. In these missions, image-based navigation is used to accurately estimate and control the asteroid-relative position of a spacecraft with a target body. In the case of Hayabusa2, an exploration mission to the asteroid Ryugu, the landing site is limited because of a lot of boulders[1]. Therefore, to realize the high precision control, GCP-NAV (Ground Control Point Navigation) is used as image-based navigation for the high precision control (Fig. 1). The position is estimated by ground operators comparing a captured image and the point cloud model of the asteroid with landmark information (e.g. rocks, edge)[3]. Ryugu is 2.4 AU away from Earth in landing phase. Therefore, it takes about 1 hour to estimate and control the position because of communication delay with Earth[1]. However, in this case, Hayabusa2 is controllable taking advantage of the microgravity of Ryugu. On the other hand, in the case of explorations to small bodies farther than Main-belt, the communication delay is unacceptably large for the asteroid-relative feedback guidance. This situation becomes worse for larger asteroids since the time constant of the dynamics becomes fast in such cases because of extreme gravitational forces. Therefore, importance of on-board image-based navigation is highlighted for explorations to far distant small bodies.

The typical examples of on-board position estimation methods are shown below.

(1) Asteroid Image Tracking (AIT)[2]

This method estimates the asteroid-relative position of the spacecraft from centroid and area of the asteroid binary image. This method is used as an on-board position estimation method in Hayabusa2 (Fig .2). Although its calculation cost is obviously low, it is vulnerable to shape change of the target body in images due to the rotation of it. Thus, this method cannot be applied to bodies with unique shapes such as the asteroid Itokawa. In addition, this method also cannot be applied at low altitudes, where the outlines of the asteroid cannot be seen in the images.

(2) Crater matching[4]

This method is used in Smart Lander for Investigating Moon (SLIM), which aims for a pin-point lunar landing. Though this method can estimate the position of the spacecraft with high accuracy, it cannot be applied to bodies without craters.

(3) Natural Feature Tracking (NFT)[5]

In the case of the Origins, Spectral Interpretation, Resource Identification, Security-Regolith Explorer (OSIRIS-REx), NFT is used as an onboard image-based navigation method that compares captured images to a set of asteroid terrain models. In this method, a template matching with the Normalized Cross Correlation (NCC) algorithm is used for similarity calculation of them.

To accomplish on-board image-based navigation that can be applied to various bodies, the Vector Code Correlation (VCC) algorithm suitable for on-board processing is focused on.

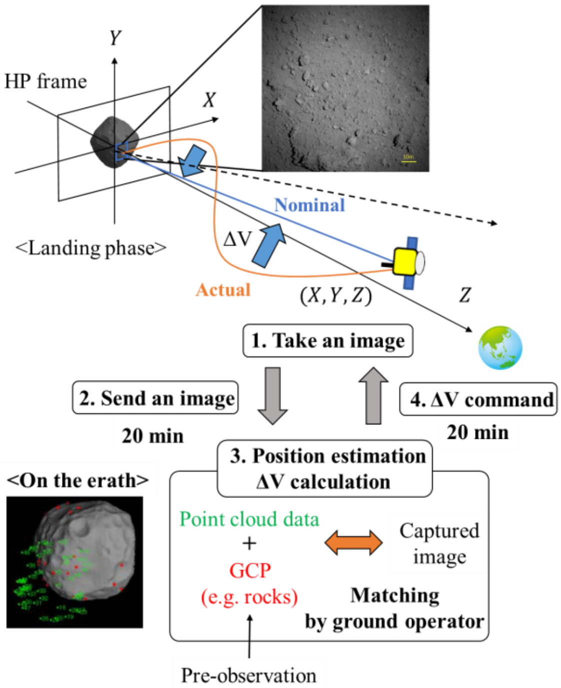

Fig. 1. Image-based navigation in Hayabusa2[2].

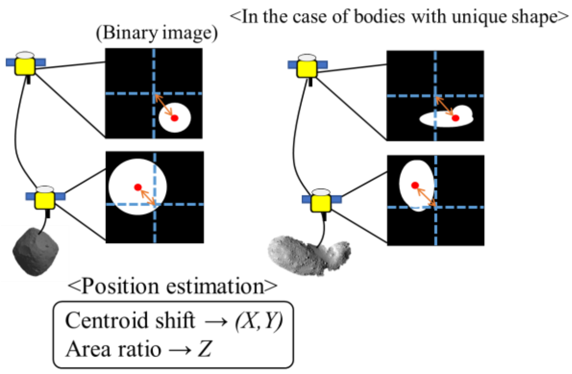

Fig. 2. Asteroid Image Tracking (AIT)[2].

2. The position estimation method using the VCC algorithm

2.1. The VCC algorithm

The VCC algorithm is a type of the template matching that finds the most similar part of 2 images by comparing them. Generally, the template matching uses luminance of images for similarity calculation such as Normalized Cross Correlation (NCC). The NCC of a template image ($N*N$ pixels) in a reference image ($M*M$ pixels, $N < M$) is defined as;

$$NCC(x_t, y_t) = \frac{RT(x_t, y_t)}{R(x_t, y_t)T}, \quad (1)$$

where

$$RT(x_t, y_t) = \sum_{x_r=0}^{N-1} \sum_{y_r=0}^{N-1} [f(x_t + x_r, y_t + y_r) g(x_r, y_r)], \quad (2)$$

and

$$T = \sqrt{\sum_{x_r=0}^{N-1} \sum_{y_r=0}^{N-1} [g(x_r, y_r)]^2}, \qquad (3)$$

and

$$R(x_t, y_t) = \sqrt{\sum_{x_r=0}^{N-1} \sum_{y_r=0}^{N-1} [f(x_t + x_r, y_t + y_r)]^2}. \qquad (4)$$

On the other hand, in the case of the VCC algorithm, discretized slope of luminance is used for similarity calculation. The slope of luminance of a template image is calculated as;

$$sx_t(x_t, y_t) = \frac{1}{6}\sum xg(x_t + x, y_t + y),$$

and

$$sy_t(x_t, y_t) = \frac{1}{6}\sum yg(x_t + x, y_t + y),$$

where

$$-1 \leq x \leq 1, -1 \leq y \leq 1. \qquad (5)$$

The slope of luminance of a reference image is calculated in the same way as above. By using the threshold level defined in advance, the slope of luminance is discretized into the 3 patterns (Fig. 3, Table. 1).

Table 1. The discretization of the slope of luminance. The symbol "b" represents a binary number.

Conditions	Parameters
$sx, sy > th$	$sx, sy = 10b$
$th > sx, sy > (-th)$	$sx, sy = 00b$
$(-th) > sx, sy$	$sx, sy = 01b$

The discretized slope of luminance of an image in the x direction and the y direction are integrated into sd by shift operation. The VCC of a template image (N*N pixels) in a reference image (M*M pixels, $N < M$) is defined as;

$$VCC(x_t, y_t) = \sum_{x_r=0}^{N-1} \sum_{y_r=0}^{N-1} [sd_r(x_t + x_r, y_t + y_r) \oplus sd_t(x_r, y_r)]. \qquad (7)$$

This algorithm can reduce the data size of pixel from 8 to 4 bit without losing feature such as edges. Furthermore, the similarity of 2 images can be calculated via XOR operation. On the other hand, the NCC uses 8 bit data for calculation. For these reasons, the VCC algorithm is deemed more suitable for FPGA.

2.2. The processing flow of the position estimation method using the VCC algorithm

Figure 3 shows the processing flow of the position estimation method using the VCC algorithm. It consists of the 3 steps below.

(1) Rendering

The 3D shape model of the target body obtained from pre-observation, nominal position and attitude of the spacecraft, ephemeris of the Earth and the target body are necessary for input. A reference image is generated from rendering with input. It represents the appearance of the asteroid from nominal trajectory. In this study, in consideration of future implementation to FPGA, the reference images are generated without texture mapping and shading to reduce the computation cost.

(2) VCC algorithm

The pixel shift between the reference image and the captured image are calculated by the VCC algorithm. These results suggest the difference between nominal and actual appearance in screen coordinates. In this study, images are compared while scaling up and down for estimation of the altitude (Z).

(3) Transformation to HP coordinates

The difference between the nominal position and actual position is estimated by using transformation matrix from screen coordinates to HP coordinates. This matrix is calculated from camera parameters such as FOV (Field of View).

The advantage of the proposed method is that it can be applied to bodies with relatively flat surface because it uses slope of luminance for matching. It's also strong against bodies with unique shapes because it uses images generated from the 3D shape model of the target body.

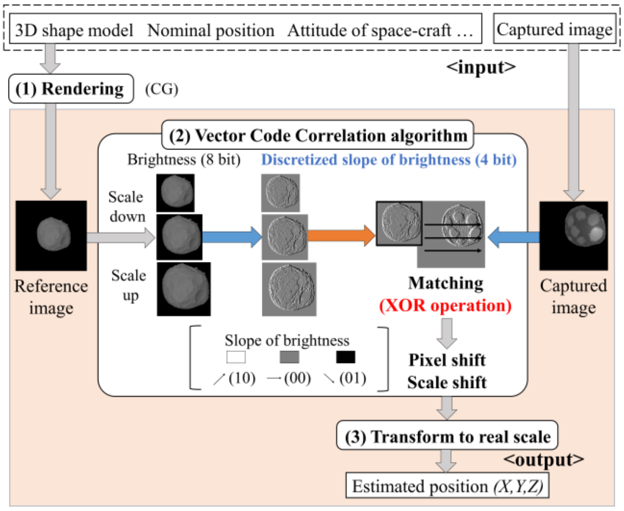

Fig. 3. The position estimation method using the VCC algorithm.

3. Software simulations

3.1. Evaluation of the proposed method

Firstly, the position estimation method using the VCC algorithm was developed and simulated on software processing. The simulation parameters are indicated in table 2. Figure 4 shows the evaluation of the proposed method. The training data including the actual position in Hayabusa2 operation was used for evaluation. The estimation errors are calculated by comparing the estimated position with the actual position. Furthermore, the estimation results of the proposed method are compared with GCP-NAV and AIT.

Table 2. The simulation conditions using the training data

Image	・Number of images = 125 ・Resolution = 512×512 *altitude ≈ 20 ~ 1 km *It takes approximately 20 hours to descend approximately 20 km to 1 km.
Asteroid 3D shape model	・Number of polygons (3D resolution) = 5,450,420 ・Diameter ≈ 1.0 km ・Rotation period ≈ 7.316 hours
Searching method for optimal scale of 2 images	・Sequential search *Range of scaling = 0.95 ~ 1.05 *Step of scaling = 1 pixel size
Scaling method	・Nearest neighbor method
The VCC algorithm	・Threshold level = 15 (fixed)

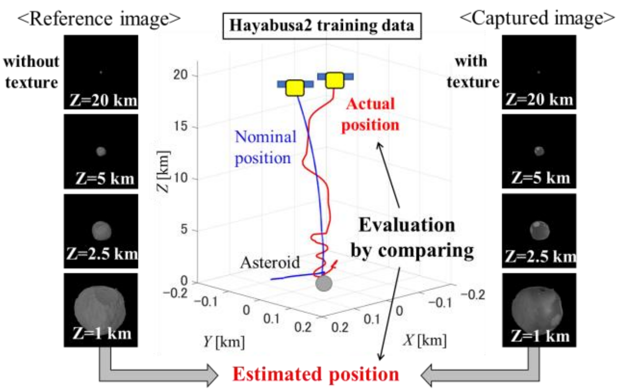

Fig. 4. The evaluation of the proposed method using the training data

3.2. Results

Figures 5 and 6 show the estimation results of the proposed method compared with GCP-NAV. In these simulation conditions, the lower the altitude is, the smaller the estimation errors are. The reason is that the pixel size in real space at low altitude is smaller than that at high altitude. Additionally, because the estimation errors are approximately 1 pixel size in real space, it is clear the proposed method can estimate the position with high accuracy of approximately 1 pixel size. Moreover, by comparing both results, the estimation errors of the proposed method are almost the same as that of GCP-NAV overall in the *X* direction and *Y* direction. However, in the *Z* direction, estimation errors of the proposed method are larger since GCP-NAV uses the altitude data from the light detection and ranging instrument (LIDAR). If the LIDAR is used for the proposed method, the estimation errors will be shorter. Not only that, the computation time will be faster because the searching for optimal scale is not necessary.

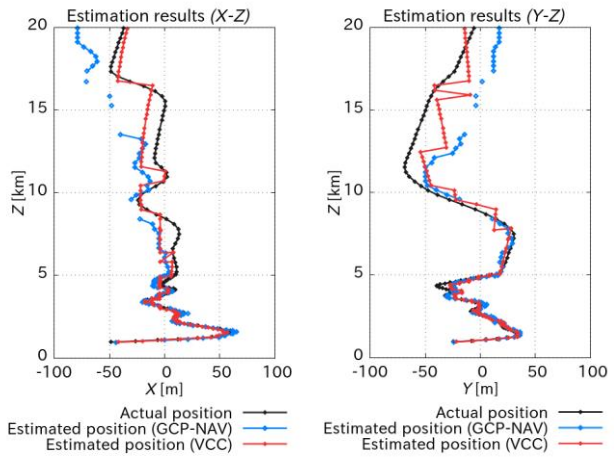

Fig. 5. The position estimation results (The VCC algorithm and GCP-NAV). The results of GCP-NAV are incomplete because of missing image data, etc.

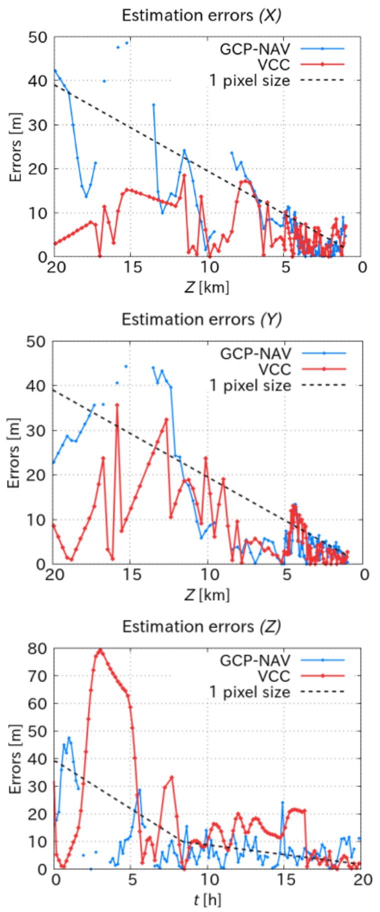

Fig. 6. The position estimation errors (The VCC algorithm and GCP-NAV). The black dotted lines show the 1 pixel size in real space.

Figures 7, 8 show the estimated results of the proposed method compared with AIT. The position estimation errors of AIT are extremely larger than that of the proposed method because of the influence of centroid errors due to the rotation.

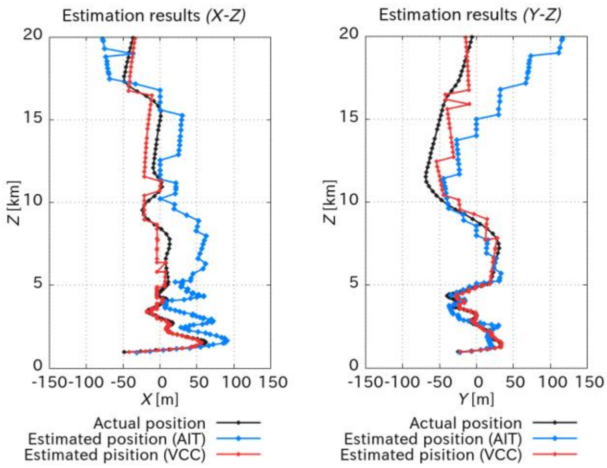

Fig. 7. The position estimation results (The VCC algorithm and AIT).

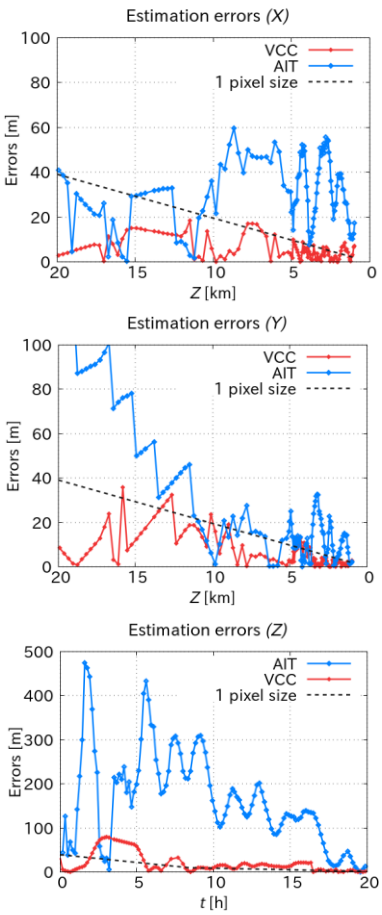

Fig. 8. The position estimation errors (The VCC algorithm and AIT). The black dotted lines show the 1 pixel size in real space.

4. Development and evaluation with FPGA

4.1. Development process with FPGA

In this study, it is assumed that reference images are generated on Earth and registered to the spacecraft in advance. Therefore, the VCC algorithm part is only implemented on FPGA. Figure 9 shows the development process with FPGA. An HLS (High Level Synthesis) tool that generates Verilog HDL code from C code written in consideration of hardware processing is used. After development in Verilog HDL, the Arrow SoCKit (terasic) including the Cyclone V FPGA (Altera), 2 GB SDRAM are used for verification experiments. The reason for choosing this product is that Cyclone V has the same performance as RTG4 (Microsemi). RTG4 FPGA has experience in space use.

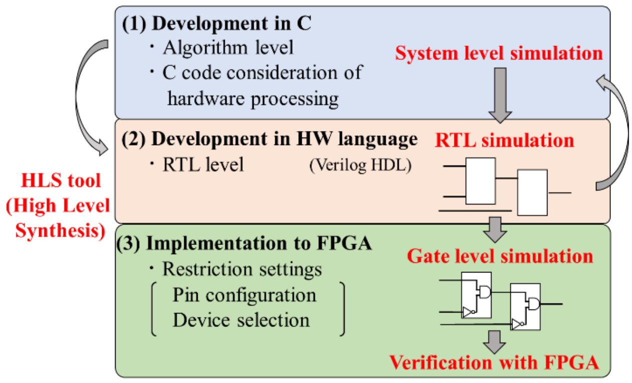

Figure. 9. Development process using HLS tool with FPGA.

4.2. Evaluation using verification board

Figure 10 shows the verification experiments method taking advantage of the Cyclone V including the ARM CPU. Firstly, the input is sent to the hardware (HW) part from the software (SW) part in the Cyclone V. After that, the position is estimated in the HW part. At the same time, the position is also estimated in the SW part. Finally, the output including estimation errors and computation time are returned to the SW part from the HW part and compared with results of the SW part.

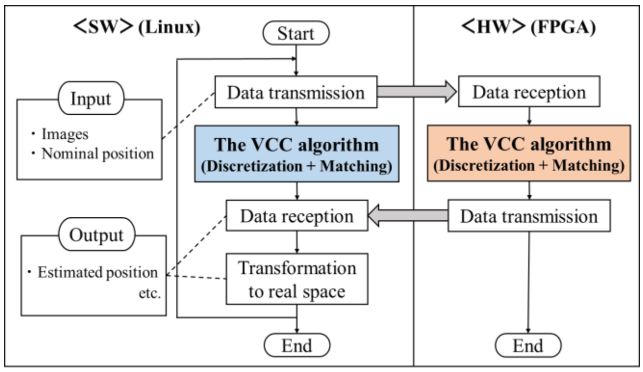

Figure. 10. The verification experiments method

4.3. Results

Table 3 shows the difference in estimation errors and computation time of each method including the proposed method. These results indicate that the proposed method can estimate the position with high accuracy such as GCP-NAV applied to the actual mission. Additionally, it was achieved that the computation time of the proposed method on HW processing is one fourth of that on SW processing. If the high speed SRAM is used instead of SDRAM on the SoCKit, the computation time will be shorter.

Table. 3. The average estimation errors and the average computation time of each method. The GCP-NAV, AIT, VCC (SW) were simulated on the PC (CPU : Intel Core i7-7700 3.6GHz, RAM : 8GB).

	Average errors (Z >10 [km])			Average errors (Z<10 [km])			Average computation time [s] (/1 position)
	X [m]	Y [m]	Z [km]	X [m]	Y [m]	Z [km]	
GCP-NAV	23.9	27.6	18.1	4.7	3.5	5.6	2400 (Including communication delay)
AIT	21.4	64.4	200.0	34.3	11.3	146.8	0.01
VCC (SW)	8.3	13.5	44.9	4.4	4.6	27.4	300
VCC (HW)							75

5. Conclusion

The position estimation method using the VCC algorithm on software processing was developed and evaluated by comparing with other position estimation methods. As a result, it was found that the proposed method can estimate the position with high accuracy of approximately 1 pixel size such as GCP-NAV applied to the actual mission. After that, by implementation of the proposed method to the FPGA board, the computation time has been reduced to 1/4 compared to that of the proposed method on software processing.

References

1) Tsuda, Y., Yoshikawa, M., Abe, M., Minamino, H., and Nakazawa, S.: System design of the Hayabusa 2—Asteroid sample return mission to1999 JU3, *Acta Astronautica*, **91** (2013), pp. 356–362.

2) Terui, F., Tsuda, Y., Ogawa, J., and Mimasu, Y.: Autonomy for Guidance, Navigation and Control of Hayabusa 2, *Transactions of the Japanese Society for Artificial Intelligence*, **29-4** (2014) pp.327-334 (in Japanese).

3) Morita, H., Shirakawa, K., Uo, M., Hashimoto, T., Kubota, T., and Kawaguchi, J.: Hayabusa Descent Navigation based on Accurate Landmark Tracking Scheme, *The Journal of Space Technology and Science*, **22-1** (2006) pp.21-31.

4) Kariya, K., Ishida, T., Sawai S., and Fukuda, S.: Position and Displacement Estimation Using Crater-Based Line Segments for Pinpoint Lunar Landing, *Trans. Jpn. Soc. Aeronaut. Space Sci.*, **14** (2016), pp. 7–14.

5) David, A. L., Ryan, O., Alexander M., et al.: Lessons learned from OSIRIS-REx autonomous navigation using natural feature tracking, 2017 IEEE Aerospace Conference, Big Sky, MT, USA, 2017.