

# Position and Displacement Estimation Using Crater-Based Line Segments for Pinpoint Lunar Landing

By Kazuki KARIYA,<sup>1)</sup> Takayuki ISHIDA,<sup>2)</sup> Shujiro SAWAI<sup>2),1)</sup> and Seisuke FUKUDA<sup>2),1)</sup>

<sup>1)</sup>*Department of Space and Astronautical Science, SOKENDAI, Sagamihara, Japan*

<sup>2)</sup>*Institute of Space and Astronautical Science, JAXA, Sagamihara, Japan*

(Received June 21st, 2017)

Next-generation landing missions require autonomous precision landing capability, with for example an accuracy of typically 100 m for a Moon landing to investigate hazardous but scientifically interesting terrain. However, current navigation systems using inertial measurement units (IMUs) do not have the navigational precision to meet this requirement. The purpose of this paper is to estimate the lander position and displacement in order to augment a navigation system. For a gravitational planetary lander, high precision and high-speed position estimation is required even with low calculation resources, e.g., a space-grade field-programmable gate array (FPGA). In this method, the lunar lander position or displacement is estimated by matching crater point patterns with database point patterns. This is accomplished by finding topological correspondences using crater-based linear features as a low-complexity method. In addition, the amount of resources and the processing time are evaluated when this algorithm is implemented on an FPGA using high-level synthesis.

**Key Words:** Pattern Matching, Position Estimation, Pinpoint Landing, TRN, FPGA

## Nomenclature

$p$	:	model points
$q$	:	world points
$S$	:	line segment composed of $p$
$S'$	:	line segment composed of $q$
$s$	:	scaling factor
$l$	:	length of line segment
$t$	:	number of corresponding points

## 1. Introduction

Recent planetary orbiters, such as those to Moon and Mars, acquired versatile data such as digital elevation models (DEM) and gravity maps. This narrows down interesting exploration areas. Next-generation landing missions require autonomous precision landing capability with, for example, an accuracy of typically 100 m for a Moon landing, to investigate hazardous but scientifically interesting terrain. However, current navigation systems using inertial measurement units (IMUs) do not have the navigational precision to meet this requirement. Thus, the inertial error should be reduced in some way.

To meet this challenge, some research studies have proposed to augment navigation systems by providing the spacecraft position.<sup>1)</sup> The position information in the guidance path where the lander performs a powered descent is important in correcting the drift error. In particular, Terrain Relative Navigation (TRN), which estimates the lander position by comparing the preliminary maintained terrain information as a database with the observed terrain, is actively researched as an effective method.

A TRN that inputs image information is divided into two types: extracting characteristic points from topographic information contained in the image and matching the extracted characteristic points. The former is a problem of extracting feature

points that can be equally distinguished even in images with different time and sunshine/posture conditions, such as an image obtained by the orbiter and by the lander. The latter is a problem of matching their feature points (where missing, insertion and misalignment exists) in real-time with robustness and high accuracy. This study aims to realize a highly accurate position estimation method using crater positions obtained by the lunar lander in real-time, with a space-grade field-programmable gate array (FPGA) as a target of the latter problem.

In a matching problem in image processing such as computer vision, the main approach, such as SIFT<sup>2)</sup> and SURF,<sup>3)</sup> used involves matching by extracting feature descriptors that provide robust information about changes in the image. Although it has high robustness against changes in photographing conditions, the information extraction process becomes complicated and the calculation cost increases. This is unsuitable for real-time processing.

On the other hand, there is a method of matching only feature points extracted from the image as a set of position information. In this method, the feature point information of the matching source comprises two point sets given in two-dimensional Euclidean space. The goal is to superimpose them by performing geometric transformation. This matching method is called point pattern matching (PPM). It plays an important role in the field of fingerprint collation and object recognition.

Since PPM is a combination problem, a key discussion point involves controlling the calculation cost with high computational complexity, thus allowing image rotation and scale changes. However, in practical use, it is not always necessary to allow rotation or scale changes, and it is possible to perform high-speed matching by relaxing these restrictions. This paper proposes a method to estimate the position of the acquisition image in the database from the crater position of the lunar surface and the line segment information constituted by crater coordinates. This approach uses a two-dimensional point pattern

matching method from the viewpoint of calculation resources and time to evaluate the possibility of mounting.

This paper first introduces the scope of the position estimation method for the lunar lander. Next, a matching method using PPM is proposed. Then, the effectiveness of the proposed method is verified by conducting a simulation experiment and evaluating the feasibility using an FPGA. Finally, this paper concludes in the final section.

## 2. Scope of This Method

In this method, we focus on position estimation for the lunar lander during its powered descent phase and vertical descent phase as shown in Fig. 1. The power descent phase fluctuates at an altitude of 10 km to 30 km. In this case, a database prepared in advance is used as a map. In the vertical descent phase, global matching is unnecessary, but the resolution of the pre-acquired data is insufficient. Therefore, the movement amount of the current frame is obtained using the image of the previous frame as a database.

As shown in Fig. 2, this method consists of two parts: a preprocessing part that creates a database, and a real processing part that is mounted on the lander and actually performs the position estimation. In the preprocessing part, a computer graphic (CG) image of the moon surface is generated by assigning the lander attitude and sunshine condition to a DEM and a reflectance map. Next, the crater coordinates are extracted from the CG image as feature points that are robust to changes in sunshine and time. Finally, a list of line segment information is created, and this is combined to form a database after maintenance such as deleting unnecessary points. In the real processing part, the lander position on the lunar surface is estimated by matching the position of the database and the crater extracted from the acquisition image using rotation, altitude and position information obtained by other sensors.

In this method, we estimate the lander position by aligning the database (world) point pattern and the acquisition crater (model) point pattern using point pattern matching, and assume that the crater position, as shown in Fig. 3, can be obtained by another detection method.<sup>4)</sup>

### 2.1. Overview of Point Pattern Matching

First, model point pattern  $M$  is a set of certain points given in two-dimensional Euclidean space, and a set of points given as a positioning target is defined as a world point pattern  $W$ . When the number of elements is  $m = |M|$ ,  $n = |W|$  ( $m < n$ ), these sets are expressed as follows:

$$M = \{p_1, p_2, \dots, p_m\} \quad (1)$$

$$W = \{q_1, q_2, \dots, q_n\} \quad (2)$$

where  $p_i, q_j \in \mathbb{R}^2$ . The purpose of PPM is to grasp the position of  $M$  in  $W$  by deriving the best overlapping geometric transformation from finding the topological correspondence. A conceptual diagram of PPM is shown in Fig. 4

Considering a superposition based on an affine transformation, the transformation matrix  $T$  that superimposes the point of

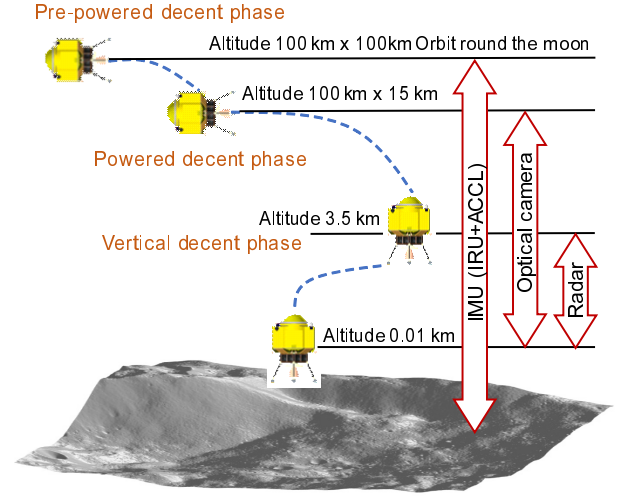


Fig. 1. Image of landing sequence.

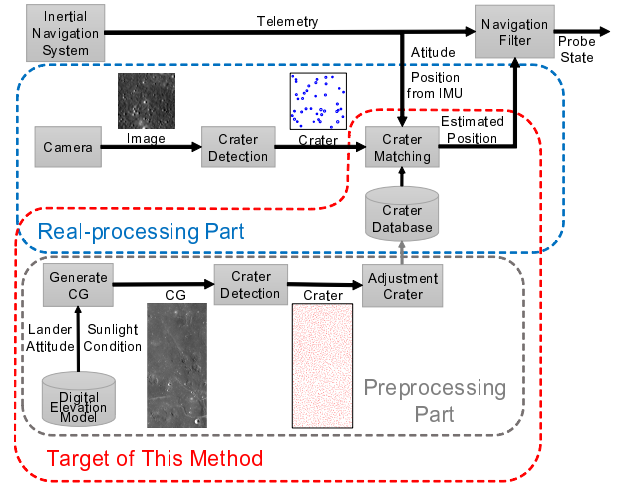


Fig. 2. Flow of position estimation of proposed method.

$M$  on the point on  $W$  is obtained as follows:

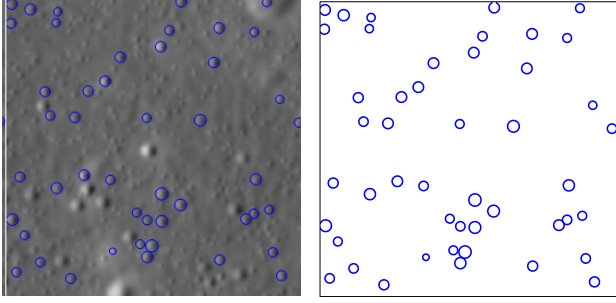
$$T \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} t_x \\ t_y \end{pmatrix} + s \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad (3)$$

where  $(x, y) \in \mathbb{R}^2$ ,  $t_x$  and  $t_y$  are movements in each axis direction,  $s$  is the scaling factor, and  $\theta$  is the rotation angle. These transformation parameters can be determined by selecting two points from each set. Therefore, it is possible to find the optimal solution by superimposing on all combinations of two points and evaluating the overlapping condition of the other points at that time. However, as the value of  $m, n$  increases, the number of combinations of points increases significantly. Thus, it takes a lot of time to process and is not a realistic approach.

To solve this problem, Ogawa<sup>5)</sup> proposed a method of dividing a point pattern into multiple triangles by the Delaunay triangulation method, and determining transformation parameters from the largest pair of triangles. Chang et al.<sup>6)</sup> focused on  $s$  and  $\theta$  among the parameters in the above Eq. (3), and tried to speed up the algorithm by using a hierarchical approach (2D cluster method). In addition, various approaches such as a method<sup>7,8)</sup> using relaxation have been attempted.

### 2.2. Requirement for Crater Matching

When considering PPM for the position estimation problem of a planetary lander, the preconditions differ in some respects



(a) Lunar surface image and crater position. (b) Acquisition crater point pattern.

Fig. 3. Example of crater point pattern.

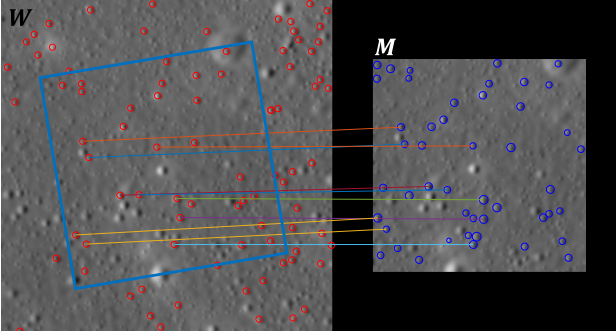


Fig. 4. Conceptual diagram of point pattern matching.

as compared with the general method.

- **Extensive Search Range** : There is a difference of nearly 100 times between the number of elements and the range of  $M$  and  $W$ .
- **Existence of False Detected/Undetected Points** : It is impossible to perfectly superimpose because the addition, deletion, and misalignment of points owing to the difference in observation conditions between the terrain image as the detection source of  $W$  and the image obtained by actual observation.
- **Computational Cost** : A calculation cost that enables real-time estimation is required, even for a spacecraft computer that has a clock frequency of about several tens of megahertz.

A particular problem is the extensive range to be searched. Application of iterative methods such as relaxation and sequential evaluation of local matching relations while allowing for scale and rotation changes is difficult from the viewpoint of calculation cost. Moreover, from the existence of false detection and undetected, it is difficult to create an equivalent graphical polygon in a method that narrows the range at once using polygon correspondence.

One of the algorithms for matching patterns of a few points with such a global map is a star identification algorithm used for Star Tracker. In the star identification algorithm, some mainstream methods are based on finding the angle of separation between the star images from the acquisition image and associating it with the star map.<sup>9)</sup> Such a method is adopted that narrows down the candidate positions on the map by repeating the correspondence using the separation angle and combining. In addition, in recent years, a method of rapidly matching the correspondence using the separation angle by using the star magnitude information has also been developed.<sup>10)</sup> These star

identification algorithms have features such as weakness against false detection and undetection of points and using brightness information unique to a fixed star. Therefore, it is difficult to apply directly as a position estimation method of the planetary lander.

From these viewpoints, we propose a method to estimate the lander position by evaluating matching locally after global matching using line segments which is the minimum element necessary for the overlapping of point patterns.

### 3. Proposal of Crater-Based Matching Method

In this section, the position estimation method based on point pattern matching, which is the real processing part, will be explained.

#### 3.1. Create Line Segments and Narrow Down Databases

First, Crater coordinates are rotated from the lander attitude given as input. At this time, the rotation is assumed to occur in the image plane, and correction is performed using a two-dimensional rotation matrix. Let  $M$  be the crater point pattern after this correction.

Next, two arbitrary points are selected from the point pattern  $M$ , and a line segment is created. Since the transformation matrix  $T$  is derived by the correspondence between two points, this line segment is a directed segment.

When the created line segment is  $S$ , the set  $L_M$  of the line segment consisting of all combinations of points included in  $M$  is as follows:

$$L_M = \{S_1, S_2, S_3, \dots, S_u\} \quad (4)$$

Here,  $u = |L_M|$ . Likewise, the set  $L_W$  of the line segment  $S'$  in  $W$  is expressed as  $v = |L_W|$  as follows:

$$L_W = \{S'_1, S'_2, S'_3, \dots, S'_v\} \quad (5)$$

When an arbitrary  $S \in L_M$  and  $S' \in L_W$  are selected, the transformation matrix  $T$  to superimpose these is determined, but all of these can correspond unless some restriction is given. For a solution, these restrictions are given according to an assumed error range.

For the assumed altitude, let  $s_{max}$ ,  $s_{min}$  be the maximum and minimum values of scale changes that can occur at the time of image acquisition, and  $\pm\theta_{max}$  ( $|\theta_{max}| < \pi/2$ ) the maximum value of the rotation. When choosing a certain  $S$ ,  $S'$  that can correspond to this is determined by the following formula:

$$s_{min}l < l' < s_{max}l \quad (6)$$

$$\cos(\theta_{max}) < \cos(\phi) \quad (7)$$

where  $l$ ,  $l'$  are the length of  $S$  and  $S'$ , and  $\phi$  is the angle between  $S$  and  $S'$ .

For a given  $S_i \in L_M$ , sets of all  $S' \in L_W$  that satisfy the above Eq. (6) and (7) is  $L_W^i$ . In this method, matching is sequentially evaluated for these sets of line segments.

The relationship between  $L_M$  and  $L_W^i$  is shown in Fig. 5. From this figure, the matching confirmation number  $N$  is as follows:

$$N = \sum_{i=1}^m |L_W^i| \quad (8)$$

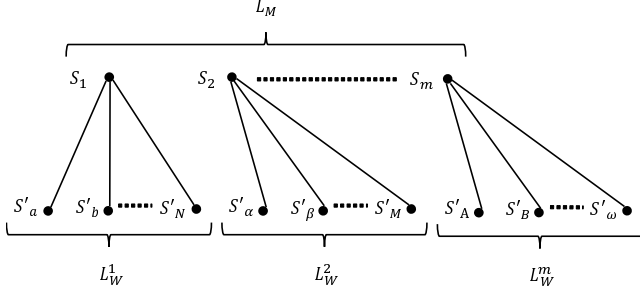


Fig. 5. Correspondence of line segment list.

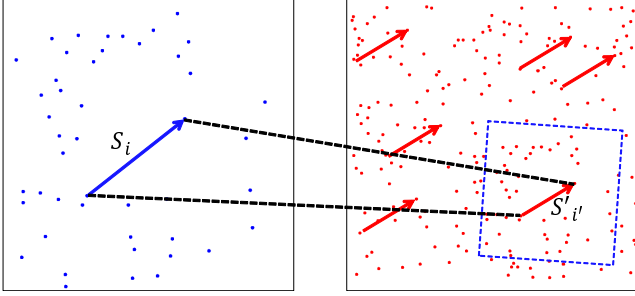


Fig. 6. Conceptual diagram of superimposition.

In order to speed up processing, consider to terminating the process when sufficient matching is confirmed for position estimation. It is desirable to evaluate the superimposition of line segments in certain  $S_i$  and  $L_W^i$  in order of likelihood. Therefore, after sequentially rearranging the line segment  $S'$  in  $L_W^i$  from the one closest to the current position  $q_{IMU} \in \mathbb{R}^2$  predicted by the IMU, matching is confirmed consecutively. In addition, when the IMU propagation error can be defined, the calculation speed can be improved by reducing the combination of line segments accordingly.

### 3.2. Determination of Search Range

Next, confirm the matching between  $M$  and  $W$  transferred by the overlapping of line segments. Using a certain  $S_i = \{p_a, p_b\}$  and  $S'_i = \{q_{a'}, q_{b'}\}$  in  $L_W^i$ , superposition by matrix  $T$  is determined from correspondence between directed segments, as shown in Fig. 6. Evaluate matching by searching  $q$  corresponding to  $p$  after doing this superposition.

On the other hand, since  $M$  contains positional deviations as an influence upon extraction, an error occurs in the matrix  $T$  used for overlay. Therefore, in order to investigate  $q_{c'}$  corresponding to a certain  $p_c$ , it is necessary to set a search range.

It is assumed that an error exists at points  $p_a$  and  $p_b$  in  $M$  by a distance  $E$  from true points  $p'_a$  and  $p'_b$ . As shown in Fig. 7, by matrix  $T$  using  $S_i$  constituted by  $p_a, p_b, p_c \in M$  is transformed to point  $T(p_c)$  away from the corresponding  $q_{c'}$  on  $W$ . Since the true points  $p'_a$  and  $p'_b$  exist within the range of  $p_a$  and  $p_b$  to  $E$  obtained, the corresponding point  $q_{c'}$  can be obtained as follows using Lagrange's undefined multiplier method.

The binding condition  $g$  is represented as follows:

$$g(p'_a, p'_b) \equiv d(p_a, p'_a) + d(p_b, p'_b) - 2E = 0 \quad (9)$$

Here,  $d(A, B)$  is the Euclidean distance between  $A$  and  $B$ . Using this, the Lagrange function  $\mathcal{L}$  becomes as follows:

$$\mathcal{L}(p'_a, p'_b, \lambda) = T'(p_c) + \lambda g(p'_a, p'_b) \quad (10)$$

$T'$  is a transformation matrix obtained by a pair of certain  $p'_a, p'_b$  and  $q_{a'}, q_{b'}$ , and  $\lambda$  is an undetermined Lagrangian multiplier.

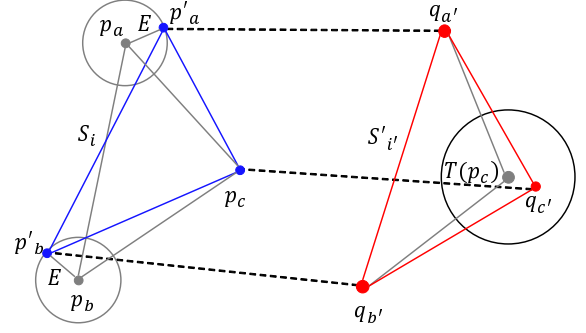


Fig. 7. Error range.

By obtaining matrix  $T'$  using  $p'_a, p'_b$  calculated from these equations, it is possible to calculate the maximum and minimum values of the range of  $T'(p_c)$  in which  $q_{c'}$  can exist. Actually, since the point  $p_c$  also includes an error by  $E$ , the search range at the time of matching is expanded by  $E$ .

### 3.3. Confirm Matching

Finally, after superimposing the line segments, search points are searched for corresponding points on  $W$  by setting the above search range for all points on  $M$ . At this time, recalculate  $T$  every time a corresponding point is found in order to reduce its error. Since there is a positional shift at each point, it is impossible to superimpose completely using affine transformation in correspondence of three or more points. Therefore, a transformation that minimizes the square of the Euclidean distance between corresponding points is obtained by the least squares method. By repeating the recalculation, it is possible to reduce the error of the transformation  $T$  caused by the positional shift without bias.

Evaluate the matching after completing the corresponding point search of all points in  $M$ . Evaluation is performed by the number of corresponding points  $t$ , and it is judged whether it exceeds the threshold  $R$  necessary for sufficient matching. This method is not aimed finding a match with the largest corresponding point; rather, it mainly focuses on finding a sufficiently match as quickly as possible, and finishing matching if  $t \geq R$ . If  $t < R$ , use the next line segment on the list of  $S'$  to repeat the matching until  $t \geq R$  is reached.

After matching is completed, recalculate  $T$  from all corresponding points, and estimate the position by shifting the image center coordinates to the database coordinates.

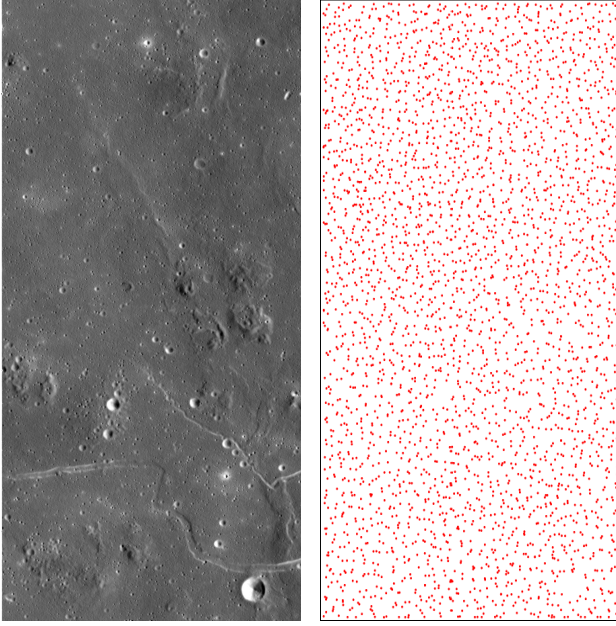
## 4. Outline of the Crater Matching Algorithm and Simulation Experiment

Based on the understanding obtained in section 3, we implement an algorithm to estimate the lander position.

### 4.1. Create Database

First, a database is created as preprocessing. The database assumed this time was the image obtained by the terrain camera equipped with SELENE in the range of about  $35 \text{ km} \times 73 \text{ km}$  around the vertical hole<sup>11)</sup> of Marius hill. This moon surface image is shown in Fig. 8 (a). The resolution of the original image is  $10 \text{ m / pix}$ . Center coordinates of craters included in this topographic image were extracted, and neighboring points in them were deleted so that the density of craters was constant





(a) Original image for database. (b) Database point pattern.

Fig. 8. Database range.

Table 1. Database description information.

Number of point patterns( $n$ )	3356
Total line segment number( $p$ )	6105
Used line length	75 - 125 pix
Described information	length
	Start point coordinate value
	End point coordinate value
	Mid point coordinate value
	Horizontal component
	Vertical component

in the database range. The crater coordinates maintained by this were used as database point pattern. This point patterns are shown in Fig. 8 (b).

Next, a set of line segments  $L_W$  is created. At that time, considering the reduction of the number of combinations, thus restricting the length of the line segments to be used. In addition, it tried to reduce the calculation time for Eq. (6) and (7) and the ordering by preliminarily calculating and listing parameters, shown in Table 1, in each line segment.

#### 4.2. Procedure of the Crater Matching

The position estimation procedure is as follows:

- Step 0 Acquisition of lunar image and detection of crater position ( $M$ ).
- Step 1 Create a line segment  $S$  from model point pattern  $M$ .
- Step 2 Create a line segment list on the database  $L_W^i$  that has the same length and inclination angle as a line segment  $S_i$ .
- Step 3 Search for  $q$  corresponding to any  $p \in M$  by superimposing  $M$  on  $W$  from correspondence between line segments.
- Step 4 The position is calculated when the number of corresponding points  $t$  is larger than the threshold value  $R$ .

In the vertical descending phase, the position is estimated first. After that, the crater pattern acquired in the previous frame is used as a database and displacement is estimated using same logic. The conceptual diagram for each step is shown in Fig. 9.

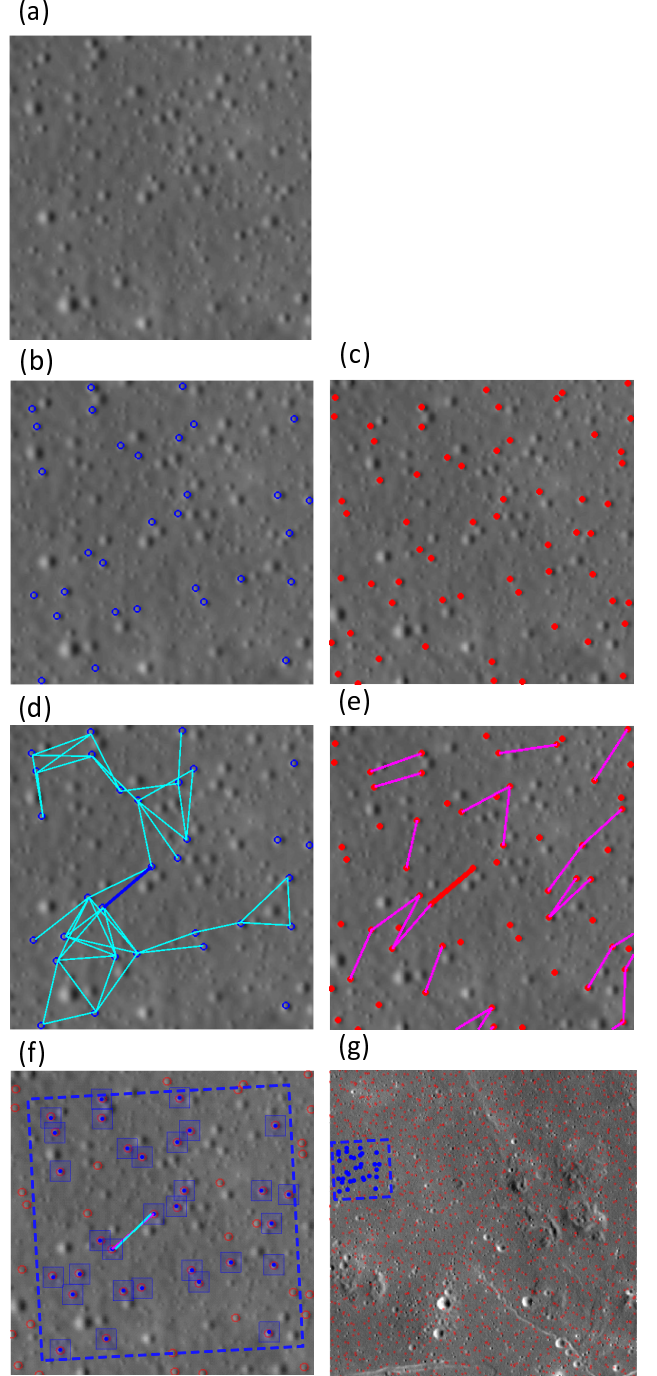


Fig. 9. Procedure of the position estimation. (a) Acquisition image. (b) Model point patterns  $M$ . (c) Part of world (database) point patterns  $W$ . (d) Set of model line segments  $L_M$  (cyan), and selected line segment  $S_i$  (blue). (e) Set of world line segments  $L_W^i$  (magenta), and selected line segment  $S'_i$  (red). (f) Overlap result and search range (blue hatch). (g) Position estimation result.

#### 4.3. Experimental Method and Results

In order to confirm the effectiveness of this method, an experiment was conducted that simulated the input database and crater point pattern. In this experiment, information obtained or limited in the guidance path for position estimation was assumed. The specifications are shown in Table 2.

An algorithm was implemented in the C language, and a simulation was performed on a general-purpose PC. The environ-

Table 2. Specification of information to be input.

Acquisition image size	512×512 pix
Database size	3520×7279 pix
Resolution of image for database	10 m/pix
IMU propagation error	±512 pix(in $3\sigma$ )
$s_{max}$	1.15
$s_{min}$	0.85
$\theta_{max}$	15 deg
$E$	2 pix
$t$	12
Maximum false detected rate	30 %
Maximum undetected rate	30 %

Table 3. Computer experiment environment.

OS	Windows 7 Professional 64bit
CPU	Intel Xeon CPU E5-2698 v3 @2.30 GHz x2
RAM	32.0 GB

ment of the computer used for the simulation is detailed in Table 3.

In this experiment, a global position estimation using the database in the power descent phase was performed. The input point pattern was simulated by randomly applying the error parameter shown in Table 2 to the database point pattern within the range. Such simulated point patterns were created in 10,000 cases with random locations within the database range, and an estimated position for each case. The position estimation results are shown in Fig. 10. These represent position estimation errors in the horizontal direction and the vertical direction. Looking at estimation results, the average value of the estimation error in the horizontal direction is 0.00 pix, and  $\sigma$  is 0.25 pix. The average value of the estimation error in the vertical direction is 0.01 pix, and  $\sigma$  is 0.26 pix. In other words, given the input as presupposed in Table 2, the proposed method can estimate the position with an accuracy of 0.75 pix in the horizontal direction and 0.78 pix in the vertical direction in  $3\sigma$ . In the database with the resolution of 10 m / pix, this is an estimation error of about 11 m in terms of linear distance. It can be said that it satisfies the requirement of a 100 m class high precision landing.

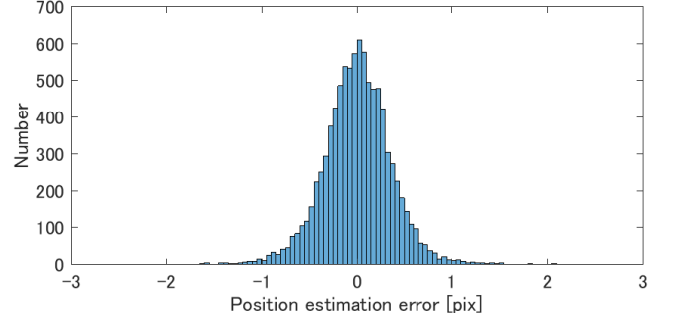
## 5. Evaluation of Mountability on FPGA

We evaluated the resource amount and processing speed consumed by the proposed algorithm for the spacecraft FPGA.

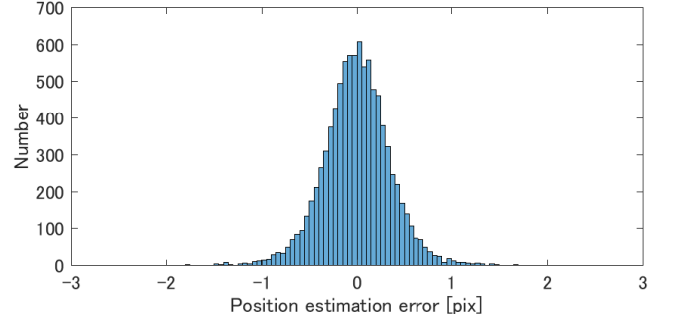
In hardware design and verification of complicated algorithms, it is not easy to design changes by optimization or bug correction in Register Transfer Level (RTL) design using the Hardware Description Language (HDL). Thus, the evaluation cycle will be prolonged. In order to avoid this problem, an evaluation method using a high-level synthesis that converts high-level languages into HDL descriptions has attracted attention. In this section, we design and evaluate the use of such a high level synthesis system, shortening the evaluation cycle and carrying out highly reliable evaluation.

Fig. 11 shows the evaluation method of the algorithm by the high-level synthesis system.

As shown, after converting a C language based algorithm designed as a floating point type into a 32-bit integer data type, it is divided into a software part responsible for data input



(a) Horizontal direction position estimation error.



(b) Vertical position estimation error.

Fig. 10. Position estimation result.

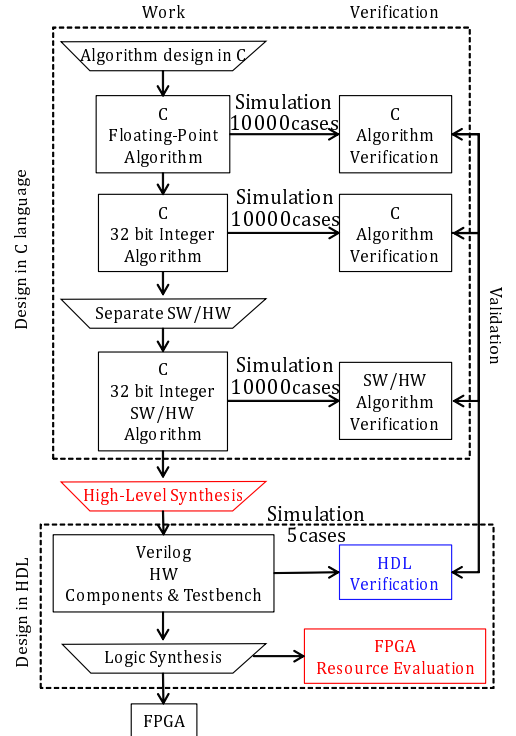


Fig. 11. Evaluation scheme by high-level synthesis system.

and a hardware part for performing actual calculations. Next, the hardware part is converted into an HDL description using high-level synthesis. In each of these stages, it is verified that there is no difference in the original C description algorithm by confirming the result obtained by giving the same input respectively. By using such an evaluation method, it is possible to correct differences in descriptions caused by the conversion at each stage, and to perform evaluation with minimal change in description when performing optimization. In addition, it is

Table 7. Processing time.

Case number	1	2	3	4	5
True position	(2191,2236)	(2191,2236)	(191,2630)	(191,2630)	none
Expected position ( $q_{IMU}$ )	(2191,2236)	(191,236)	(191,2630)	(2191,4630)	(0,0)
Rotation	8.12 deg	8.12 deg	5.27 deg	5.27 deg	none
Scale	+3.80%	+3.80%	+1.28%	+1.28%	none
Undetected rate	25%	25%	23%	23%	100%
False detection rate	7.8%	7.8%	28%	28%	100%
Estimated position	(2191,2236)	(2191,2236)	(191,2630)	(191,2630)	No estimation
Estimated scale	+3.70%	+3.70%	+1.11%	+1.11%	No estimation
Processing time@50MHz	75 ms	88 ms	110 ms	123 ms	283 ms

Table 4. Applicable FPGA.

Provider	Microsemi
Family	RTG4
Device	RT4G150_ES-CG1657M
Package	1657 CG
Speed	STD

Table 5. Evaluation tool.

C design	Microsoft Visual Studio 2012
HLS	Y Explorations eXCite 5.3n
Logic synthesis	Microsemi Libero SoC 11.7 SP3
HDL verification	Aldec Active-HDL 10.1

possible to obtain a highly reliable evaluation result by advancing the evaluation cycle.

Table 4 shows the space-grade anti-radiation FPGA to be evaluated, and the tool used for this evaluation is shown in Table 5. The results of verifying the proposed algorithm are shown in Tables 6 and 7.

Looking at Table 6 for the consumption resources of this algorithm, resource usage other than RAM of  $64 \times 18$  is around 20 % of the total logic, and the value of only the location estimation part (excluding the detection part) is sufficient. Meanwhile, 59 % of the internal memory is consumed. The cause appears to be that it places great pressure on resources by mounting the array used for processing inside the FPGA. This can be solved by storing the intermediate data in the memory external to the FPGA in addition to sharing the number of working arrays and finding the necessary number. However, since it is expected that the processing speed will be slowed by accessing the external memory, care should be taken for optimization.

Next, to evaluate the processing time, the evaluation flow, shown in Fig. 11, was applied to each of the five case patterns acquired in the experimental environment of Chapter 4. In order to check the difference in processing time owing to the difference between the IMU estimated position, these five cases are given different inputs respectively. In case 1 and 3, the IMU estimated position is equivalent to the true position. Case 2 or 4 has the same true position as case 1 or 3, but the IMU estimated position deviates from the true value by 2,000 pix. Case 5 is the worst case that is a point pattern that does not exist on the database.

Looking at Table 7, for all cases including the worst case, processing can be performed at a speed of 1 s or less when operating at 50 MHz. There is no significant difference in processing time in all cases, and it can be said that sufficient real-time processing is possible even in the computer environment of the

Table 6. Consumption resources.

Logic	Used	Available	Utilization
LUT	38032	151824	25.05 %
SLE	26949	151824	17.75 %
RAM64x18	103	210	49.05 %
MACC	55	462	11.90 %
Operating frequency	50.4 MHz		

spacecraft.

These are the results of optimizing the circuit area and calculation time considering the balance. If the circuit area is sacrificed, the calculation time becomes faster for processing such as parallelization, and the circuit area can be reduced by sacrificing calculation time. In future work, it is necessary to consider more balanced optimization, including verification of the actual device.

## 6. Conclusion

In this paper, we aim to realize a position estimation method for lunar high precision landing in real time even in the computer environment of a spacecraft. We proposed a method to estimate the position with high accuracy using the crater position extracted from the lunar surface image and the line segment constituted by crater coordinates. In that case, in the PPM method which is a combination problem with high computation complexity, the attitude of the spacecraft and the inertial navigation position are given as inputs. In addition, fast matching was performed by reducing the number of combinations using correspondences between line segments consisting of two points in a crater point pattern and database line segments.

In this method, it is assumed that errors such as rotation and scale changes, undetected points and false detected points in the acquisition image are within a certain range. We simulated 10,000 cases of such a point pattern and confirmed its effectiveness by applying this method. As a result, we showed that position estimation is possible with an accuracy of 0.75 and 0.78 pix with  $3\sigma$  in the horizontal and vertical directions, respectively. This value is an error that does not apply a bias of about 12 m in linear distance in the original image for database with a resolution of 10 m / pix. It is possible to estimate the position with accuracy that can achieve the 100 m class precision landing indicated.

Subsequently, we evaluated the resource amount and processing speed consumed by the proposed algorithm for the space-grade FPGA.

By using a high level synthesis system as an evaluation

method, the evaluation cycle was shortened and a highly reliable evaluation was carried out. As a result, the calculation time was less than 1 s at 50 MHz operation in a case where the position can be estimated. This has a sufficient real-time property even when the landing aircraft is descending. In addition, we confirmed that the resource amount is 25 % or so excluding the memory. This is satisfactory for mountability to the spacecraft.

## References

- 1) Yoshikawa, S., Kunugi, M., Yasumitsu, R., Sawai, S., Fukuda, S., Mizuno, T., Nakaya, K., Fujii, Y. and Takatsuka, N.: Conceptual Study on the Guidance, Navigation and Control System of the Smart Landing for Investigating Moon (SLIM), Proceedings of Global Lunar Conference 2010, (2010), paper ID 5644.
- 2) David, G. L.: Object Recognition from Local Scaleinvariant Features, Proc. of IEEE International Conference on Computer Vision, **2**(1999), pp.1150-1157.
- 3) Herbert, B., Andreas, E., Tinne, T. and Luc, V. G.: Supeeded-Up Robust Features (SURF), Computer Vision and Image Understanding, **110**(2008), pp.346359.
- 4) Takino, T., Nomura, I., Moribe, M., Kamata, H., Takadana, K., Fukuda, S., Sawai, S. and Sakai, S.: Crater Detection Method using Principal Component Analysis and its Evaluation, Transactions of the Japan Society for Aeronautical and Space Sciences, Aerospace Technology Japan, **14** ists 30(2016), pp.7-14.
- 5) Ogawa, H.: Labeled Point Pattern Matching by Delaunay Triangulation and Maximal Cliques, Pattern Recognition, **19.1**(1986), pp.35-40.
- 6) Chang, S-H., Cheng, F-H., Hsu, Q-H. and Wu, G-Z.: Fast Algorithm for Point Pattern Matching: Invariant to Translations, Rotations and Scale Changes, Pattern recognition, **30.2**(1997), pp.311-320.
- 7) Ranade, S. and Rosenfeld, A.: Point Pattern Matching by Relaxation, Pattern recognition, **12.4**(1980), pp.269-275.
- 8) Ogawa, H.: Labeled Point Pattern Matching by Fuzzy Relaxation, Pattern Recognition, **17.5**(1984), pp.569573.
- 9) Junkins, J. L., White, C. and Turner, J.: Star Pattern Recognition for Real-Time Attitude Determination, Journal of the Astronautical Sciences, **25**(1977), pp.251270.
- 10) Tabur, V.: Fast Algorithms for Matching CCD Images to a Stellar Catalogue, Publications of the Astronomical Society of Australia, **24.4**(2007), pp.189-198.
- 11) Haruyama, J., Hioki, K., Shirao, M., Morota, T., Hiesinger, H., Van der Bogert, C. H., Miyamoto, H., Iwasaki, A., Yokota, Y., Ohtake, M., Matsunaga, T., Hara, S., Nakanotani, S., and Pieters, C. M.: Possible Lunar Lava Tube Skylight Observed by SELENE Cameras, Geophysical Research Letters, **36.21**(2009), L21206.