VISUAL-BASED INTEGRATED NAVIGATION SYSTEM APPLIED TO A SIMULATION OF LUNAR MODULE LANDING

C.T. Kuo¹, Y.T. Tien¹, K.W. Chiang¹

¹ Dept. of Geomatics, National Cheng Kung University, No.1, Daxue Road, East District, Tainan, Taiwan – alex57133@gmail.com

KEYWORDS: Visual-based Navigation System, Inertial Navigation System, Lunar Landing Module, PANGU, Direct Sparse Odometry

ABSTRACT:

With the development of space technology, more and more lunar researches are performed by different countries. For the lunar landing mission success, the lunar landing module should equip with advanced Positioning and Orientation System (POS) for the navigation requirements. For the pinpoint landing mission formulated by NASA, a good POS with error less than 100 meters is needed in order to make the lunar module land safely at the exact destination on lunar surface. However, the existing technologies for lunar navigation, such as satellite positioning and star tracker, have poor performance for the navigation requirements. The visual-based positioning technology is an alternative way to make sure a lunar landing module reaches the destination. There are two types of visual-based positioning technology, absolute and relative navigation. The relative navigation system can provide the solution at a higher rate, but the error would accumulate over time. On the contrary, the absolute navigation could provide an initial position or updates of position and attitude for relative navigation. Thus, the integrated navigation system from those two methods can take advantage of both stand-alone systems. On the other hand, the Inertial Navigation System (INS) can help it overcome the disadvantage that the images much closer to the lunar surface are not available. This study shows an integrated navigation system that integrates a visual-based navigation system and an INS, which is implemented in a simulated lunar surface.

1. INTRODUCTION

The moon is the closest celestial body to the Earth. Before starting the exploration, how to land on lunar surface safely is an important issue. Advanced positioning and navigation technology are required to land on target precisely. Traditional inertial navigation integrates the angular increment and velocity increment of the IMU to obtain the position and the attitude. However, the error would accumulate due to the initial error, measurement deviation and noise, even can reach kilometers. Since satellite positioning system is not yet complete on the lunar environment, visual-based navigation technology, which has high autonomy and accuracy for positioning, is an alternative way for navigation.

The extraction, matching, and tracking effects determine the quality of visual-based navigation, which means that the feature extraction and matching during lunar landing are more important. However, these feature detections, such as Scale-Invariant Feature Transform (SIFT) algorithms (Lowe, D. 1999), and tracking algorithms are computationally intensive and require long processing time. Therefore, there are still difficulties in real-time application in landing.

In order to reduce computation time and avoid the effects of the environment, that is, to have too few features, a new Visual Odometry (VO)method called Direct Sparse Odometry (DSO) is proposed (Engel, J., Koltun, V., & Cremers, D. 2017). DSO has strong robustness and speeds up the operation compared to the past. The visual-based navigation method continuously provides stable position and attitude estimations with the error not exceeding a certain range. However, the navigation algorithm has the disadvantages of large amount of data calculation and low data update rate, which sometimes cause temporary loss of tracking targets.

The purpose of this research is to establish an integrated visualbased navigation system to realize the combination of inertial navigation and visual-based navigation and improve the navigation accuracy of the lunar landing process. The integrated navigation system overview is shown in the Figure 1.



Figure 1. Integrated Navigation System Overview

2. VISUAL-BASED NAVIGATION ALGORITHMS

The necessary input for the visual-based navigation system is the image. First, using the Planet and Asteroid Natural scene Generation Utility (PANGU) software to simulate the lunar surface image. Once the image is simulated, it can be used in a visual-based navigation system. The visual- based navigation system is divided into two parts; one is the relative navigation algorithm and the other is the absolute navigation algorithm.

2.1 Visual-based Relative Navigation

In this research, the algorithm of visual-based relative navigation is Direct Sparse Odometry (DSO) (Engel, J., Koltun, V., & Cremers, D. 2017), which is a kind of VO algorithm using the direct- method. It has strong robustness and speeds up the operation compared with the past. Direct-method tracks the points in the image and computes the camera displacement and attitude changes by minimizing the photometric error. The direct method eliminates the feature extraction and matching process, thereby reducing computation time. Therefore, the direct method is used in this research under the assumption that the system operates in real time.

Figure 2 shows the basic concept of DSO. The camera displacement $T_{k,k-1}$ and attitude changes between k and k – 1 frame are estimated by minimizing the photometric error of the pixels corresponding to the two images.



Figure 2. Basic concept of DSO

When a new image inputs to DSO algorithm, it will go through the flow chart shown in Figure 3, performing steps such as initialization, attitude estimation, calculating residuals, optimizing, and removing outliers. First, use the initial few images to complete the initialization, calculate the initial position and attitude, and then use the images to update the depth of the immature point, which is called "trace new coarse" in the process. The optimization is to process the key frame, add new residual items, remove the wrong residuals, extract immature points, and update the points with new images.



Figure 3. DSO system flow chart (J. Engel et al., 2017)

2.2 Visual-based Absolute Navigation

In order to complete the visual-based absolute navigation model, two main processes are needed. One is to use the Speeded Up Robust Features (SURF) to find the image feature points, and the other is to use the image from the resection technology to get the camera position and attitude.

2.2.1 Speeded Up Robust Features (SURF)

The SURF algorithm consists of the following three steps: feature point detection, feature proximity description, and descriptor matching. For feature point detection, SURF uses the integral map and square filter as the approximation of the Gaussian filter. Furthermore, SURF maintains the original image on a spatial scale, uses the 9×9 square filter results as the initial scale group to build the corresponding layer under each scale by changing the square filter size, and then constructs the layer into a pyramid-like concept. After each layer is set according to the scale space, each pixel is compared with the adjacent 26 points. If the pixel is of maximum value and greater than the threshold value, it is then a feature point.

When the feature points are found, their descriptors are created. The method of descriptor construction is to describe the changes between the feature point and its adjacent phase points. In order to ensure rotation invariance, the main direction is first assigned to the feature points. Focusing on the feature points and with 6σ as the radius of the circle, the Harr wavelet of all pixels is counted and multiplied by the Gaussian weight of the corresponding position.

In order to obtain the main direction, a sector-shaped sliding window with an opening angle of 60 degrees is used to calculate the sum of the horizontal and vertical responses of the Harr wavelet in the region, and the direction corresponding to the largest response area obtained by sliding the sector window is the main direction of the feature point. Along the main direction, a rectangular area of 20×20 is taken as its neighborhood and divided it into 16 sub-areas. Then, each sub-area calculates the sum of Harr wavelet responses of the pixel points in the area. The response of each pixel point is multiplied by the corresponding Gaussian weight of position. In this way, each sub-area vector has a total of four magnitudes, and a total of 64-dimensional data descriptors is generated. Finally, matching pairs can be found by comparing descriptors obtained from different images.

2.2.2 Photogrammetry Space Resection (PSR)

Photogrammetry Space Resection (PSR) technology uses the camera to observe at least three noncollinear known feature points and the exterior orientation parameters (EOPs), which means that the position and attitude of the photos are calculated by these known points. The core theory of the space resection is collinearity. The local coordinates (X_A, Y_A, Z_A) , the image coordinates (x_a, y_a) of point A and camera perspective center (X_C, Y_C, Z_C) are collinear. The mathematical expressions are as shown in equations (1) and (2).

$$\begin{aligned} \mathbf{x}_{a} &= \mathbf{x}_{0} - f \, \frac{m_{11}(\mathbf{X}_{A} - \mathbf{X}_{C}) + m_{12}(\mathbf{Y}_{A} - \mathbf{Y}_{C}) + m_{13}(\mathbf{Z}_{A} - \mathbf{Z}_{C})}{m_{31}(\mathbf{X}_{A} - \mathbf{X}_{C}) + m_{32}(\mathbf{Y}_{A} - \mathbf{Y}_{C}) + m_{33}(\mathbf{Z}_{A} - \mathbf{Z}_{C})} \\ \mathbf{y}_{a} &= \mathbf{y}_{0} - f \, \frac{m_{21}(\mathbf{X}_{A} - \mathbf{X}_{C}) + m_{22}(\mathbf{Y}_{A} - \mathbf{Y}_{C}) + m_{23}(\mathbf{Z}_{A} - \mathbf{Z}_{C})}{m_{31}(\mathbf{X}_{A} - \mathbf{X}_{C}) + m_{32}(\mathbf{Y}_{A} - \mathbf{Y}_{C}) + m_{33}(\mathbf{Z}_{A} - \mathbf{Z}_{C})} \end{aligned}$$
(1)

Where (x_0, y_0, f) is the internal orientation parameter which means the principal point and focal length of the camera, and $(m_{11}, m_{12}, ..., m_{33})$ are the rotation matrix elements of the camera rotating from the local coordinate system to the camera

coordinate system. The concept is shown in Figure 4. Let the EOPs of the photo be unknown and the image coordinate of the target be the observation, then input to the least squares model. After calculation, the EOPs of the photo are obtained.



Figure 4. Space resection concept

2.2.3 Visual-Based Absolute Navigation Model

The visual-based absolute navigation model is a combination of SURF and PSR algorithm. The first step is to use the SURF algorithm to extract the feature points of each simulated image, find out their object coordinate, and then store them into the database. The flow chart of database construction is shown in Figure 5. The images in the database are composed of a few groups of images, and every image has its own information, such as the EOPs and SURF parameters for its feature points.



Figure 5. Database construction flowchart

The second step is to load target images, process SURF algorithm on them, and use the current position, which means the estimated EOPs from visual- based relative navigation, as the center point to make a circle within a 10 kilometers radius. Then, match target images with database images one by one with the set SURF parameters to find the closest image with the highest success rate in the database.

The third step is to find out the object coordinates of the feature points, which are extracted from the closest image found in the previous step in the database. Finally, figure out the object coordinates corresponding to the image coordinates extracted from the target image. Thus, the collinear equations are obtained in order to process the PSR algorithm. The flow chart and illustration of the whole process are shown in Figure 6.



Figure 6. Visual-based absolute navigation model flow chart

Figure 7 is the illustration of visual-based absolute navigation model, it expresses the first and second step that is mentioned above, including how the moon image looks after SURF processing and the concept of searching within a radius. The red points in the lunar surface images indicates the point number of the extracted feature point.



Figure 7. Illustration of visual-based absolute navigation model

3. VISUAL-BASED INTEGRATED NAVIGATION SYSTEM

The visual-based integrated navigation system proposed in this research is integrated by visual-based relative navigation, visualbased absolute navigation and inertial navigation system. From the system architecture shown in the Figure 1, it is known that the system will first fuse the visual-based relative navigation with visual-based absolute navigation, and then integrate the fused visual-based navigation result with inertial navigation system. This chapter will first describe the comparison of different fusion methods, and then present the methods in which the proposed system is used.

3.1 Loose and Tight Coupling

Visual information and IMU data fusion can be divided into two kinds of data interaction, loose coupling and tight coupling. Loose coupling adopts an independent inertial positioning model and a positioning navigation model. The update frequency of the two models are inconsistent, and there is a certain information exchange between the two models. In loose coupling, the inertial data is taken as the core, and the visual measurement data corrects the cumulative error in the inertial measurement data.

Tight coupling uses IMU to perform motion estimations in VO. The error of IMU integration between image frames is relatively small, and IMU data is used to predict frame-to-frame motion, accelerates the point matching process, and completes VO position and attitude estimation. In loose coupling, GPS calculates the result and then input to KF in order to integrate with INS. In tight coupling, GPS raw measurement is input to KF to integrate with INS. Comparison between characteristics of loose coupling and tight coupling is shown in Table 1.

Comparison	Loose Coupling	Tight Coupling
	System	System
Basic	Each processor has	Processors have
	its own memory	shared memory
	module.	modules.
Memory	Generally do not	Generally
	encounter memory	experiences more
	conflict.	memory conflicts.
Data rate	Low	High
Price	Less expensive	More expensive

Table 1. Comparison between Loose Coupling and Tight Coupling (Tech Differences, 2017)

There are several tight coupling schemes (Mourikis, Trawny, Roumeliotis, Johnson, & Matthies, 2007; Trawny, Mourikis, Roumeliotis, Johnson, & Montgomery, 2006) and loose coupling schemes (Pham, 2010) in the literature that combine absolute and relative state estimates. One disadvantage of tight coupling is that algorithms are difficult to disassemble. In contrast, loose coupling is more modular and simpler. The loosely coupled system can easily choose different absolute navigation system or relative navigation system to replace the original algorithm. In this research, therefore, the visual-based integrated navigation system uses loose coupling to fuse the two kinds of sensors in order to achieve its flexibility.

3.2 Fusion of Visual-Based Navigation

Figure 8 shows the fusion process of visual-based navigation. The output of both visual-based navigation algorithms is camera state, that is, the position and attitude of the camera, but one is relative camera state, and the other is absolute camera state. The position of this moment plus the relative movement distance can get the position of the next moment. When the absolute navigation has a solution, the position of the next moment is replaced by this solution, and then the position continues to accumulate the subsequent relative movement distance until the next absolute navigation solution input. When the absolute navigation has no solution, the relative navigation is not updated and continues to accumulate the subsequent relative movement distance.



Figure 8. Fusion of Visual-Based Navigation

3.3 Fusion of Visual-Based Navigation and INS

The visual-based integrated navigation system overview is shown in Figure 9. The output of visual-based navigation algorithms is low frequency absolute camera state. At the same time, the inertial navigation requires the simulated IMU data to generate high frequency relative position and attitude of the camera as its output. These output states will eventually become the input of KF, and the final absolute camera state result is obtained.



Figure 9. Fusion of Visual-Based Navigation and INS

The Kalman filter (KF) (Welch G., & Bishop G.1995) is a highly efficient recursive filter that estimates the state of a dynamic system from a series of incomplete and noise- containing measurements and has numerous applications in technology. One common application is guidance, navigation, and the control of vehicles, aircrafts and space crafts. It is also widely used in time series analysis, such as signal processing and econometrics.

The main concept of KF is as shown in Figure 10. The algorithm is a two- step procedure. First is the estimation step, also called time update, which is shown in equations (3) and (4). The KF produces an estimate of the current state, which also includes uncertainty. Next is the update step, also called measurement update, which is shown in equations (5) to (7). If the next measurement is observed, the estimated value will be updated by a weighted average called the Kalman gain. The higher the certainty, the higher the weighted weight of the measurement. This algorithm is iterative and can be executed in a real-time control system. (4)



Figure 10. The main concept of KF (Welch G., & Bishop G.1995)

$$\hat{x}_{t|t-1} = F_t \times \hat{x}_{t-1} + B_t \times u_t \tag{3}$$

$$P_{t|t-1} = F_t \times P_{t-1} \times F_t^T + O_t \tag{4}$$

$$K_t = \frac{P_{t|t-1} \times H_t^T}{\tau}$$
(5)

$$H_t \times P_{t|t-1} \times H_t + R_t$$

$$\begin{aligned} x_{t|t} &= x_{t|t-1} + K_t (2_t - H_t x_{t|t-1}) \\ P_{t|t} &= P_{t|t-1} - K_t \times H_t \times P_{t|t-1} \end{aligned} \tag{6}$$

where x_t = state vector at time t

- u_t = vector that controls the parameters
- F_t = the state transition matrix
- B_t = control parameter matrix
- $K_t =$ Kalman gain
- Q_t = process noise covariance matrix
- R_t = measurement noise covariance matrix
- P_t = estimation error covariance matrix
- z_t = measurement vector
- H_t = rotation matrix of state vector to measurement

At each discrete time increment, KF use a linear dynamical system and IMU data to predict the new state, and to get the less noise result of position by the update from other measurement.

4. EXPERIMENT SETTING

4.1 Flight Path Region

The landing site is set at the lunar longitude of 59.921°E and the latitude of 87.185 °S, and the simulated reference trajectory is about 80 km long and 10 km height from the landing site. In Figure 11, the entire reference trajectory is displayed in the local level frame that uses the landing point as the origin. The whole landing process descends from 10km-0km high from the lunar surface. Figure 12 shows the simulated reference attitude, which means the roll, pitch, and yaw angle of the lunar module. The reference attitude shows that after the first 60 seconds of the large yaw angle changing, the lunar module will move smoothly. The simulations of all images are based on an approximate vertical image, so the variations of roll and pitch angle are close to zero.



Figure 11. The reference trajectory in local level frame



Figure 12. The reference attitude local level frame

4.2 Image Data Setting

Figure 13 shows the height change over time. The blue line is the height of the whole reference trajectory, and the orange point means the location of the image database which is prepared for visual-based absolute navigation. An important factor that affects the feasibility of visual-based navigation algorithms is the image resolution. If the image is too blurry, the two visual- based navigation algorithms would not work or would have very poor accuracy. However, the lunar DEM source that covers the region of reference trajectory is downloaded from the NASA website and has the highest spatial resolution of only 10m/pixel. As shown in Figure 14, if the gradually decreasing ground distance corresponding to one pixel is smaller than the spatial resolution, which means the closer it is to the ground, the image may be too blurred to be used in visual-based navigation.







Figure 14. Relationship between a pixel and the ground during descending

Therefore, according to the limitation of the image resolution, which is, the dependence on the space resolution of the lunar DEM that inputs to the PANGU software, this study determines the most appropriate height for the visual-based system algorithm through multiple tests. The input image for DSO algorithm is limited to 10km-2km high, and the input image for PSR algorithm is limited to 10km-6km high. If the position is beyond the limited height, it is assumed that the image input is interrupted and that the algorithm would stop executing. Therefore, the simulation of the image database is also from 10km-2km high.

For visual-based absolute navigation, it is assumed that the execution time interval is 5 to 10 seconds. Therefore, a set of images around the current reference position is created every 7.5 seconds and is stored in the image database. For example, a set of images is stored at a reference height of 7.5 seconds, and another set of images is stored at a reference height of 15 seconds. In 7.5 seconds between executions, the relative positioning algorithms will continue to run so that the estimation of the trajectory will not be interrupted. The number of every set of images is shown in Table 2. The increase in the number of images at 22.5 seconds is since the surface is too flat at this location, resulting in too few feature points for extraction and matching. Therefore, more images are needed to improve the extraction and matching success rate of the feature points.

Flight time	Height	Number of
(centisecond)	(km)	images
0	10	15
7.5	10.39314	15
15.0	10.14727	15
22.5	9.41365	18
30.0	8.36972	15
37.5	7.20211	15
45.0	6.09739	15

Table 2. Number of images for Database

The camera parameters used in research are set as an ideal pinhole camera. The parameters are shown in Table 3. If there is an actual camera, the camera parameters will be changed to the corrected parameters. The experiment settings for visual-based navigation algorithms and the inertial navigation are shown in Table 4. The three methods used in the system have raw data with different efficiencies. The full flight time of the trajectory is 228.08 seconds.

Parameters	Content	Setting
fx	Camera focal length /	0.78125000
	X direction resolution	
fy	Camera focal length /	0.97656252
	Y direction resolution	
сх	Image center point	0.5
	X position offset	
су	Image center point	0.5
	Y position offset	
distortion	Camera geometry	0
	correction parameters	
Input width	Input image wide	1280
-	resolution (pixels)	
Input height	Input image high	1024
- 0	resolution (pixels)	

Table 3. Camera parameters setting in DSO algorithm

FOV	60
image size (pixel)	1280*1024
DSO efficiency	10Hz
PSR efficiency	0.2 Hz
IMU data efficiency	100 Hz
Number of DSO images	1485
Number of PSR images	108
Full flight time	228.08 sec

Table 4. Settings for the full system

5. RESULTS AND ANALYSIS

5.1 Results and Analysis of Position

Currently, the PSR algorithm can only be used down to 6 km in height, and the DSO algorithm can only be used down to 2 km in height. Thus, this research assumes that when the time is less than 60 seconds, the PSR, DSO, and INS in the system are working at the same time; when the time is between 60 and 150 seconds, the PSR cannot operate, leaving the DSO to assist the INS algorithm; when the time is more than 150 seconds, the visual navigation algorithms is invalid, and only the INS operates independently. The execution time period of each algorithm is shown in Figure 15, with the vertical axis as the height of algorithm execution. It can be observed that the time is about 60 seconds when the PSR algorithm stops at about 6 km high, and the time is about 150 seconds when the DSO algorithm stops at about 2 km high.



Figure 15. Execution time period of each algorithm

The complete result trajectory of the algorithm and simulated reference trajectory are shown in Figure 16. In the figure, the blue line is the reference trajectory, the red line is the result trajectory of the inertial navigation, the yellow line is the result trajectory of the visual-based navigation system, the purple star symbolizes the absolute positions calculated by the PSR algorithm, and the green line is the result trajectory of the integrated navigation system. Figure 17 to Figure 19 show the trajectories on different planes. Figure 20 shows the position errors of the INS, and Figure 21 shows the position errors of the integrated navigation system.



Figure 16. Trajectory result of the integrated system

The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLIII-B1-2020, 2020 XXIV ISPRS Congress (2020 edition)



Figure 17. Trajectory result on X-Y plane



Figure 18. Trajectory result on Y-Z plane



Figure 19. Trajectory result on X-Z plane



Figure 20. Position error of INS on the three axes



Figure 21. Position error of integrated system on the three axes

In the above position error, the amount of error reduction provided by the PSR in the first 60 seconds exists but is less obvious, because the simulated INS is assumed to be just turned on, and the error accumulation is not fast. If a higher spatial resolution DEM can be obtained in the future to make the PSR execution time more durable, the effect of the PSR on reducing the errors should be more obvious.

In Table 5, the comparison between INS and Integrated navigation system is shown, including the position error at the end point of the trajectory and error Root Mean Square (RMS). The error of the integrated navigation system decreased obviously, especially in the horizontal direction. As shown in Table 6, the trend of the error in the horizontal directions is affected by the length of the trajectory. Unlike the horizontal trajectory that moves almost in a straight line, the cumulative error in the vertical direction is less affected by the length of the trajectory which is possibly because that the vertical trajectory has undulations and the error is eliminated.

Error Analysis		X (m)	Y (m)	Z (m)
End Point	INS	52.5267	99.1625	4.7218
Error (m)	Integrated	17.0813	21.1907	1.6610
Error RMS	INS	25.5665	49.2207	3.9800
(m)	Integrated	9.8208	13.3538	2.2628

Table 5. Analysis for position errors

Trajectory Length (km)		33.3772	77.5748	11.9894
End Point Error	INS	1.5737	1.2783	0.3938
Trajectory Length)	Integrated	0.5118	0.2732	0.1385
Accuracy Improvement (%)		67.48	78.63	64.82

Table 6. Analysis for position accuracy

Table 7 shows the position error in the end point of different algorithms and that in the horizontal direction, the error in INS-only is larger than that of visual-based-only; but in the vertical direction, the error in INS-only is smaller than that of visual-based-only. The possible reason is that the length of the simulated reference trajectory in the horizontal direction is larger than the length in the vertical direction, resulting in a larger error of the INS-only has a larger error in the vertical direction may be that the later in the trajectory, the closer to the lunar surface the camera is, and the more blurred the image.

(meters)	Horizontal	Vertical	Total Error	
	Error	Error		
INS only	112.2152	4.7218	112.3145	
DSO only	111.6353	601.3034	611.5784	
Visual-based only	44.4339	291.6924	295.0573	
Integrated	27.2179	1.6610	27.2685	

Table 7. Position error in the end point of different algorithm

5.2 Results and Analysis of Attitude

The execution time period of each algorithm on attitude is shown in Figure 22, with the vertical axis as the heading result, which is the yaw angle, of the algorithm execution. The PSR algorithm stops at about 60 seconds and the DSO algorithm stops at about 150 seconds.



Figure 22. Time period of each algorithm execution on attitude

Figure 23 and Figure 24 show the attitude errors of INS and of the integrated system. The errors are reduced in all three angles. In Figure 24, the pitch angle error has a decreasing trend at 50 to 150 seconds. At this time, it should be the period in which DSO and INS are executed. Therefore, it can be observed that DSO provides a good contribution to the reduction of pitch angle error. After 150 seconds, the period in which only the INS is being executed, it can be observed that the error of the pitch angle starts to accumulate again.



Figure 23. Attitude error in INS



Figure 24. Attitude error in the integrated system

Error Analysis		Roll	Pitch	Yaw
End Point	INS	0.2543	0.3554	0.4258
Error (deg)	Integrated	0.1507	0.1609	0.3662
Error RMS	INS	0.1511	0.2066	0.3033
(deg)	Integrated	0.0936	0.0764	0.1888
Accuracy Improvement		40.74 %	54.73 %	14.00 %
Table 8. Analysis for attitude errors				

The exact statistics are shown in Table 8, displaying the attitude errors of the INS and integrated navigation system. It shows that errors are significantly reduced in all three angles at the end point error and the error RMS. This indicate that the visual-based navigation results have a good effect in reducing attitude errors. The attitude error of less than 0.5 degrees is enough for the pinpoint landing problem of the moon landing process.

6. CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

This paper presents a visual-based integrated navigation system. Relative positioning and absolute positioning can be achieved simultaneously by using different technologies. First, the relative but drifting positioning is achieved by the DSO algorithm and the INS. At the same time, through feature point matching and PSR algorithm, the absolute but time-consuming positioning is realized. When visual-based navigation does not work, the INS continues providing navigation solutions; and when visual-based absolute navigation does not work, the visual-based relative navigation continues providing navigation solutions to correct the INS in a short time. The proposed method effectively reduces the position and attitude error and only needs a monocular camera and a tactical level IMU.

6.2 Future Works

The proposed method effectively reduces the position and attitude error in this research. However, the algorithm still has some shortcomings. First, the IMU data of the algorithm uses simulated data, and the simulation process may be overly idealized, which is not enough to match the real environment. The camera parameters also use an ideal monocular camera, which requires additional consideration for camera errors such as lens distortion when using an actual camera. The scale variability between image coordinate system and navigation coordinate system must also be considered more rigorously in the design of the filter. Secondly, the lunar surface image simulated by the software is affected by the DEM resolution, and the accuracy of the actual image may increase when it is close to the surface. The difference clarity between the images captured by the actual camera and the reference images of the database may also affect the navigation results. Finally, the simulated image is dominated by vertical surface images, and the image may have a large tilt angle during actual shooting. The simulation of images with large tilt angles is also the direction of future testing.

In future research, efforts will be made to bring the simulation situation closer to the real lunar environment and to improve the visual-based navigation system. If a higher spatial resolution DEM open resource is available, it is expected that the execution time of the visual-based system will be longer and the improvement in error will be more significant. Testing the algorithm with actual camera and hardware devices in the earth's environment is also a plan in the future. Finally, an important plan is that the parameters used in the fusion method and the filter are also more fully analysed after various tests.

REFERENCES

Cheng, Y. & Ansar, A. (2005). Landmark Based Position Estimation for Pinpoint Landing on Mars. Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA), Barcelona, Spain, pp. 4470-4475.

Engel, J., Koltun, V., & Cremers, D. (2017). Direct Sparse Odometry. IEEE Transactions on Pattern Analysis and Machine Intelligence.

Lowe, D. (1999). Object recognition from local scale-invariant features. In ICCV.

Mourikis, A. I., Trawny, N., Roumeliotis, S. I., Johnson, A. E., & Matthies, L. (2007). Vision-aided inertial navigation for precise planetary landing: Analysis and experiments. Proceedings of Robotics: Scientific Systems, Atlanta, GA

Pham, B. V. (2010). Vision-based absolute navigation for interplanetary spacecraft descent and landing. Système de navigation absolue pour l'atterissage d'une sonde interplanétaire. Diss. ISAE.

Tech Differences, (2017). Difference Between Loosely Coupled and Tightly Coupled Multiprocessor System (with Comaprison Chart) - Tech Differences. https://techdifferences.com/ difference-between-loosely-coupled-and-tightly-coupledmultiprocessor-system.html

Trawny, N., Mourikis, A. I., Roumeliotis, S. I., Johnson, A. E., & Montgomery, J. F. (2007). Vision-aided inertial navigation for pinpoint landing using observations for mapped landmarks. Journal of Fields Robotics, 24(5), pp. 357-378.

Welch G., & Bishop G. (1995). An introduction to the kalman filter. Technical report, UNC-CH Computer Science Technical Report 95041.

Xie, F., Liu, J., Li, R., Jiang, B., & Qiao, L. (2014). Performance analysis of a federated ultra-tight global positioning system/inertial navigation system integration algorithm in high dynamic environments. Journal of Aerospace Engineering, 229, pp.56-71.