



Article

Crater Detection and Recognition Method for Pose Estimation

Zihao Chen ^{1,2} and Jie Jiang ^{1,2,*}

¹ School of Instrumentation and Optoelectronic Engineering, Beihang University, Beijing 100191, China; chenzihao@buaa.edu.cn

² Key Laboratory of Precision Opto-Mechatronics Technology, Ministry of Education, Beihang University, Beijing 100191, China

* Correspondence: jiangjie@buaa.edu.cn; Tel.: +86-10-8233-8497

Abstract: A crater detection and recognition algorithm is the key to pose estimation based on craters. Due to the changing viewing angle and varying height, the crater is imaged as an ellipse and the scale changes in the landing camera. In this paper, a robust and efficient crater detection and recognition algorithm for fusing the information of sequence images for pose estimation is designed, which can be used in both flying in orbit around and landing phases. Our method consists of two stages: stage 1 for crater detection and stage 2 for crater recognition. In stage 1, a single-stage network with dense anchor points (dense point crater detection network, DPCDN) is conducive to dealing with multi-scale craters, especially small and dense crater scenes. The fast feature-extraction layer (FEL) of the network improves detection speed and reduces network parameters without losing accuracy. We comprehensively evaluate this method and present state-of-art detection performance on a Mars crater dataset. In stage 2, taking the encoded features and intersection over union (IOU) of craters as weights, we solve the weighted bipartite graph matching problem, which is matching craters in the image with the previously identified craters and the pre-established craters database. The former is called “frame-frame match”, or FFM, and the latter is called “frame-database match”, or FDM. Combining the FFM with FDM, the recognition speed is enabled to achieve real-time on the CPU (25 FPS) and the average recognition precision is 98.5%. Finally, the recognition result is used to estimate the pose using the perspective-n-point (PnP) algorithm and results show that the root mean square error (RMSE) of trajectories is less than 10 m and the angle error is less than 1.5 degrees.



Citation: Chen, Z.; Jiang, J. Crater Detection and Recognition Method for Pose Estimation. *Remote Sens.* **2021**, *13*, 3467. <https://doi.org/10.3390/rs13173467>

Academic Editors: Stephan van Gasselt and Andrea Nass

Received: 4 August 2021

Accepted: 30 August 2021

Published: 1 September 2021

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Soft landing on the surface of a planet is one of the key technologies for space exploration missions and requires high positioning accuracy. The pose-estimation method based on inertial navigation has accumulated errors, and the landing error is within a few kilometers [1], while high-precision landing missions require landing errors of less than 100 m or even 10 m. Terrain relative navigation (TRN) [2] uses an optical camera to capture images and calculates information about the pose of the spacecraft. The optical flow method or feature-tracking method is a kind of TRN but cannot give the absolute pose [3]. Another method of TRN matches the image obtained by the image sensor with the image of the vicinity of the flight area and the stored terrain feature database to calculate the absolute pose. Craters are widely present on the surface of planets and are an ideal topographic feature. The NEAR mission successfully proved that high-precision pose information can be obtained using crater features [4]. Thus, crater-based TRN is possibly one of the most suitable solutions for pose estimation in soft landing missions.

Crater-based TRN has three parts: processing sensor input, i.e., crater detection; matching sensor data with a database containing topographical features, a process called recognition; and, finally, using correlated data to calculate pose. In this paper, crater image processing methods for pose estimation are discussed, including crater detection

and recognition, both of which are critical to final positioning accuracy. After recognition, perspective-n-point (PnP) [5] is used to solve the pose. During landing to the ground, the craters imaged in the landing camera are always ellipse and the scale of the craters changes. The detection and recognition method must be adaptable to these changes.

The accuracy and robustness of crater detection are sensitive to lighting conditions, viewing angle, and degree of noise. Large changes in crater scale, overlap, and nesting of different craters have challenged detection algorithms. At present, neural networks have been successfully applied in many computer-vision tasks; they can deal with complex scenes and thus comprise a possible solution. In the literature, one type of research uses a convolutional neural network to segment an image, separate the ellipse from the background, cluster the ellipse through a clustering algorithm, extract the edge of the crater, and finally fit the ellipse equation. This solution can obtain the complete information of the crater, which is beneficial to the recognition algorithm, but the detection algorithm is complicated. Klear [6] uses the UNet [7] network to predict the probability that each point may belong to the edge of the crater and detects the crater through the edge-fitting method. Silburt and Christopher [8] also use the UNet network, but the image used in their work is a digital elevation map (DEM), and there is no need to consider changes in lighting conditions. The network structure of Lena [9] is the same as that proposed in Ref. [8]. Lena expanded the data and used DEM images and optical images to analyze the effects of lighting factors in detail, making the network more robust to lighting. Another solution is using an object-detection neural network to detect the crater target and obtain the crater circumscribed rectangle. When the flying height is much larger than the radius of the crater, the rectangular center can be approximated as the crater center [10]. The algorithm complexity of this method is lower and using bounding boxes can improve the matching accuracy.

Recognition is the process of associating the crater in an image with the crater in the pre-established crater database, which can be divided into recognition with prior information and “lost in space”. The recognition algorithm should have strong robustness and cannot occupy a large amount of time for the entire navigation task.

If the prior pose existing error is known, the craters in the database can be projected to the current camera field of view, using an error function between the projected image and the real image to calculate the pose through optimization. Leroy et al. [11] used the affine transformation relationship of two crater sets on small celestial bodies for matching, which requires accurate prior knowledge. Clerc et al. [12] combined this type of recognition method with the random sample consensus (RANSAC) method to further improve its accuracy. Cheng et al. [13] used the affine invariant feature of the ellipse to select three possible matching pairs of craters and calculate their attitudes and then projected the craters from the database onto the image to verify the guessed matching pairs. This method does not require prior information but will take a significant amount of time to verify the position. In Refs. [14,15], it was assumed that the camera shoots vertically to the ground, which can realize absolute pose estimation without prior information. In addition to matching craters in the image with the pre-established database, it can also match images between sequences, which can provide more information and accelerate recognition. The matching methods between sequence images include pattern matching and image correlation-based matching [16], among others.

In a crater-based terrain navigation system, the recognition algorithm is related to the output of the detection algorithm, and many researchers study the detection and recognition algorithms separately. With CraterIdNet [17], a fully convolutional network was proposed for end-to-end crater detection and recognition. In the present article, the region proposal net [18] is used to detect the crater object, assuming that the crater is circular, and then a grid pattern is generated for each crater input, which is then input into the classification network. For generating the grid pattern, CraterIdNet assumes that the camera shoots vertically to the ground and must input the height information, which cannot be used to calculate posture and limits the scope of application. By fusing the

information of sequence images, projecting the pre-established database by the previous pose, or matching the craters in the before and after, our method does not need to consider whether the camera is shooting vertically.

Overall, the contribution of this research is to comprehensively study the crater detection and recognition algorithm, demonstrate a robust and efficient crater detection and recognition system, and, at the same time, fuse the information between the sequence images into the system, consisting of two stages: stage 1 for detection and stage 2 for recognition. In stage 1, adopting the idea of an anchor-free object detection neural network, a dense point crater detection net (DPCDN) with multiplying anchor points is designed. The dense anchor point of the DPCDN is beneficial to the detection of small and dense craters. DPCDN outputs the circumscribed rectangle of the crater as the input of the recognition algorithm. In stage 2, taking the encoded features and intersection over union (IOU) of craters as weights, a bipartite graph matching algorithm is used to match crater output by the DPCDN in the image with the previously identified craters and the pre-established craters database, which can achieve real-time speed on CPU and high precision.

The rest of this paper is organized as follows. In Section 2, the method proposed in this paper is introduced, i.e., the dense point crater detection network in Section 2.1 and crater recognition algorithm in Section 2.2. In Section 3, the experimental results on crater detection and recognition as well as pose estimation are described. In Section 4, the results are discussed, and conclusions and future research plans are presented in Section 5.

2. Materials and Methods

2.1. Methodology

The workflow of the detection and recognition system is shown in Figure 1, and it consists of two stages: stage 1 for crater detection and stage 2 for recognition. In stage 1, to effectively detect small-scale craters and dense crater scenes, a crater detection network with multiplier anchor points is proposed, i.e., the “dense point crater detection network,” or DPCDN. In stage 2, using DPCDN output, the Kuhn-Munkres (KM) algorithm [19,20] is used to quickly match the craters between the frames, which is beneficial to make full use of the information of the sequence images and the craters between the image frames and the pre-established database. The image taken at time k is input into DPCDN, extracted by the fast extraction layer (FEL) and the feature is enhanced by the feature pyramid network (FPN [21]) structure. Then the detection head with dense point is used for detecting craters. The FEL and the dense point detection head we designed will be described in detail. Then the crater bounding boxes are matched with the crater in the pre-established database to complete the recognition, which is called “frame-database matching”, or FDM. Another matching route is matching frame k ’s craters with frame $k - 1$ ’s craters, and if craters in $k - 1$ have been identified, frame k ’s craters can be also identified. This process is called “frame-frame matching,” or FFM. The latter is faster than the former.

DPCDN is a one-stage, anchor-free target detection network, and its output is a bounding box circumscribed by the crater. Compared with the nearest neighbor matching using the center point, using bounding boxes can improve the matching accuracy. In the case of dense craters, the distance between the center points is more likely to be mismatched. Figure 2 shows this result intuitively.

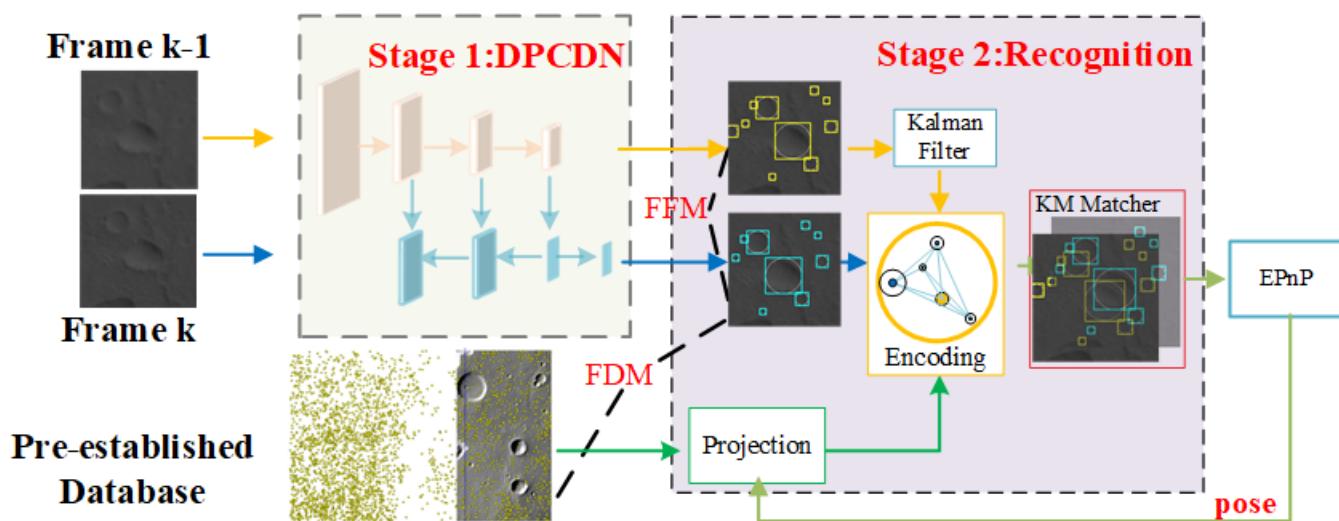


Figure 1. Crater detection and recognition system workflow. The whole workflow consists of two stages. In stage 1, a dense point crater detection network obtains craters in the frame k and frame $k - 1$. Then, in stage-2, we use the KM algorithm to match k 's craters with $k - 1$'s craters or the pre-established database.

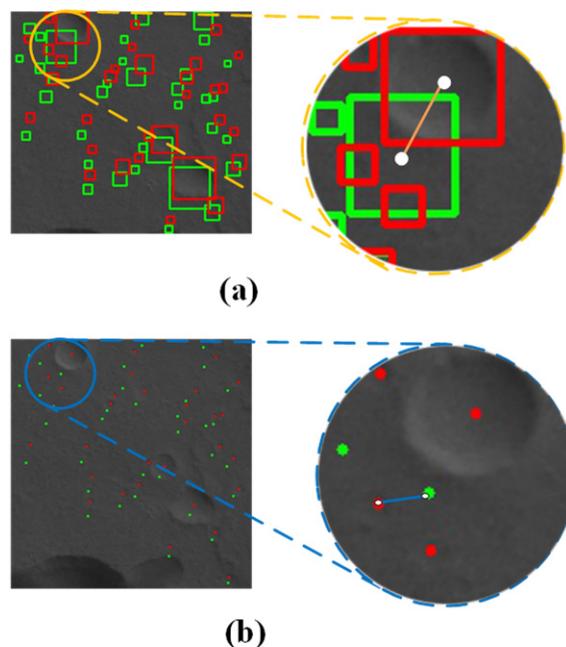


Figure 2. Matching (a) with bounding box and (b) with the center of craters.

2.2. Stage 1: Crater Detection

Inspired by the idea of using a convolutional neural network to detect craters [18], a dense point crater detection network (DPCDN) was designed. The architecture of the proposed network is shown in Figure 3. The crater detection algorithm used for navigation should be efficient and robust. A fast feature-extraction layer (FEL) was designed as the backbone. In the anchor point design, each position of the low-level feature map (P_2) is associated with multiple anchor points to detect dense small-scale craters.

FEL is used to quickly extract the features of the input image, as shown in the upper part of Figure 3a, and has four convolutional layers. The stride between the previous convolution layer and the next is 2. Both conv1 and conv2 are followed by a CReLU [22] activation function:

$$\text{CReLU}(x) = \text{Concat}[\text{ReLU}(x), \text{ReLU}(-x)] \quad (1)$$

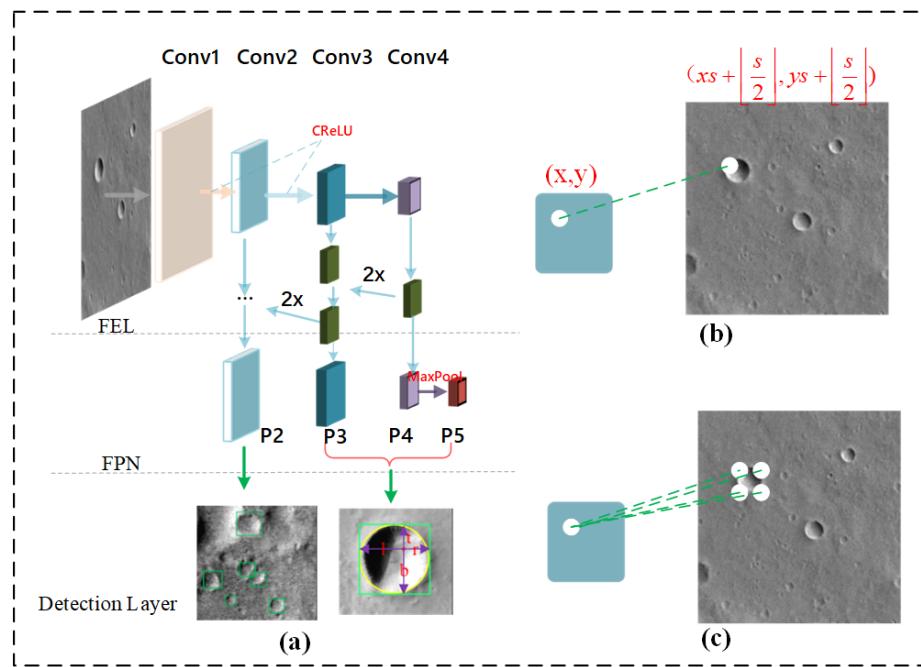


Figure 3. (a) Architecture of DPCDN; (b) one point in feature map mapping to an anchor point in original image applied in P3, P4, and P5; (c) one point in feature map mapping to multiple anchor points in the original image, applied in P2.

The detail in the FEL is shown in Figure 4a. Shang et al. [22] proved that the weights learned by the low-level network are approximately symmetrical; that is, for the weights of 2C channels, those of the first C channels and the last C channels are approximately in the opposite relationship. Using this, the activation function can double the number of output channels by simply concatenating negated outputs before applying ReLU.

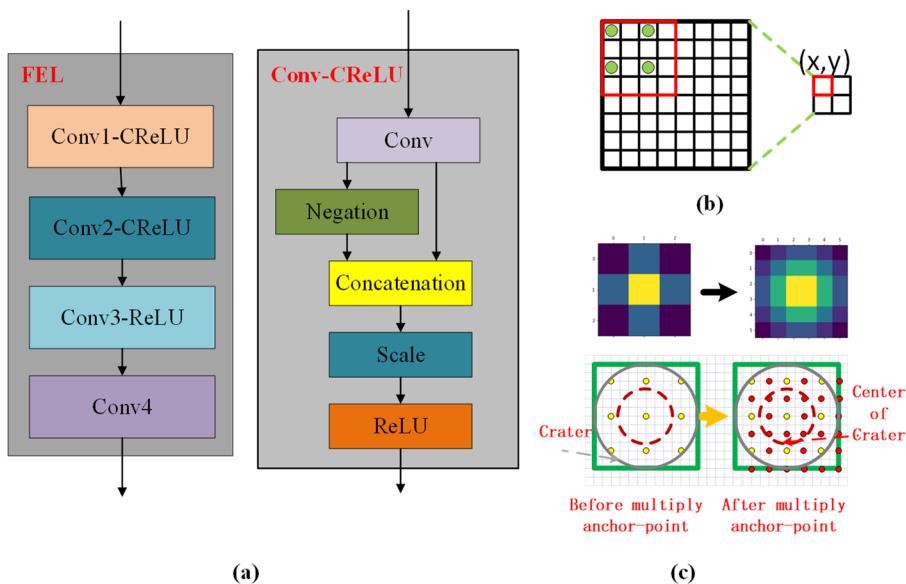


Figure 4. (a) Structure of FEL and Conv-CReLU. (b) Relationship between point (x, y) of feature map and point of original map. (c) Changes in centrality before and after dense anchor points.

After the features extracted by the FEL are enhanced by the feature pyramid network (FPN [21]), the feature maps $P_i \in R^{H \times W \times C}$, where i is in $\{2, 3, 4, 5\}$ for detection, are obtained. Each location (x, y) on the feature map P_i is mapped to the input image $(x_s + \lfloor \frac{s}{2} \rfloor, y_s + \lfloor \frac{s}{2} \rfloor)$ and s is the total stride until the layer is reached. The detection layer

outputs four distances (t, l, b, r) from the center to the four sides of the bounding box. To suppress low-quality detected bounding boxes, the “center-ness” of the location is used and the low center-ness output is ignored. The center-ness reflects how close the anchor point is to the target center, and its formula is

$$\text{centerness} = \sqrt{\frac{\min(l, r)}{\max(l, r)} \times \frac{\min(t, b)}{\max(t, b)}} \quad (2)$$

The center-ness is useful for large-scale craters because the low-quality bounding boxes can be filtered by non-maximum suppression (NMS). However, when the crater scale is small, there are very few locations associated with it. Only one location can be used to detect small-scale craters, as shown in Figure 4c. To avoid this situation, the anchor point is made dense. For each location (x, y) in feature maps, it is made dense to $\left(x + \frac{k_i}{s}, y + \frac{k_i}{s}\right)$, $i \in Z$ and $i < s$, so that each location can be associated with more than one location in the input image. After being made dense, more effective output bounding boxes can be used because of more anchor points in the center of the crater.

2.3. Stage 2: Crater Recognition

The purpose of crater recognition is to obtain information, such as the location of the crater in the pre-established database and its three-dimensional coordinates in the stellar coordinate system. Either the crater can be directly matched with the database, or the crater detected in the k th frame can be matched with the craters that have been successfully matched in the image of the previous frame. In this paper, the former is called “frame-database match”, or FDM, and the latter is called “frame-frame match”, or FFM. To achieve real-time speed, for both matching strategies the KM algorithm is employed for data association because it is effective and fast. In feature space, an encoding method to measure the distance between the craters to be matched is designed, and in physical space IOU is used as the distance. Two distance measures improve the robustness of the algorithm. Figure 5 shows the workflow of the recognition algorithm.

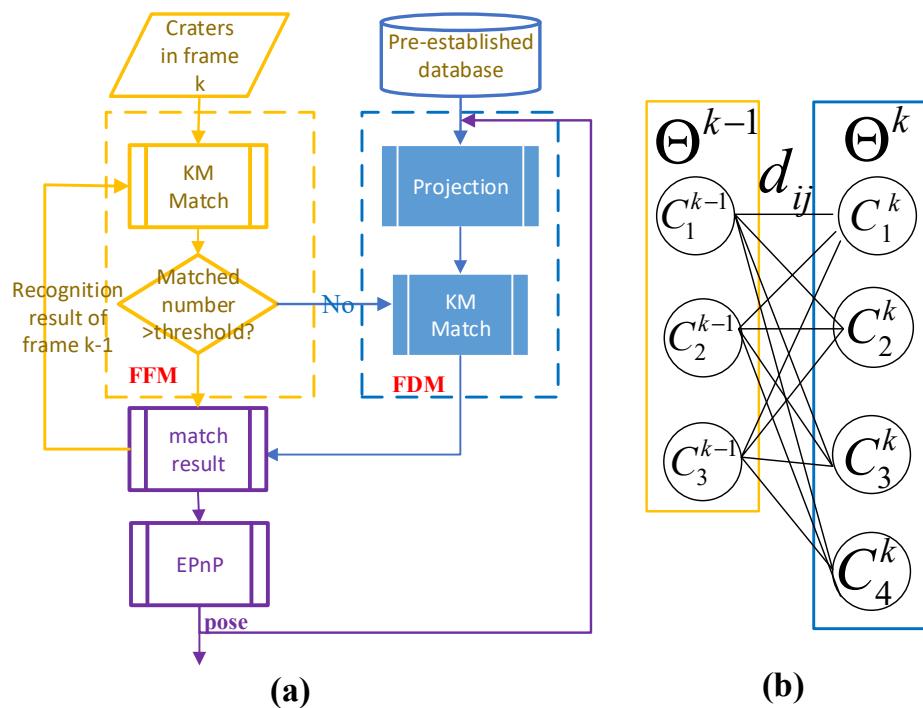


Figure 5. (a) Recognition workflow. (b) KM matching algorithm where the d_{ij} is the distance between craters, including the feature distance and IOU distance. Θ is for the craters and C_i^k is the i th crater in the k th frame.

The bounding boxes detected in the k th frame are denoted

$$\Theta^k = \left\{ C_i^k = (x_i^k, y_i^k, w_i^k, h_i^k) \mid i = \{0, 1, \dots, m\} \right\} \quad (3)$$

where k is the sequence number of this frame, C_i^k the i th crater of the k th frame, and x, y, w , and h the coordinates of the upper left-hand corner of the target, the length, the width of the target, respectively.

The FFM is the matching of the sets Θ^k and Θ^{k-1} . To match the frame to the database, the pre-established database must be projected to the current field of view according to the pose $\Omega^k = \Gamma(\Omega, \text{pose}^k)$, where Γ represents the camera projection equation, and the FDM is to match Ω^k and Θ^k . Both use the KM match algorithm, which is a weighted bipartite graph matching algorithm. The weight in KM algorithm is the distance between craters, including IOU and features:

$$d_{ij}^{iou}(C, C') = \frac{\text{area}(C) \cap \text{area}(C')}{\text{area}(C) \cup \text{area}(C')} \quad (4)$$

$$d_{ij}^{feature}(C, C') = \|f_i^C - f_j^{C'}\| \quad (5)$$

When there is a large overlap area, the IOU distance metric is used to match the crater, and when the IOU fails, the distance between the encoding features is used for matching because the features are more robust to noise.

The encoding method proceeds as follows and is shown in Figure 6.

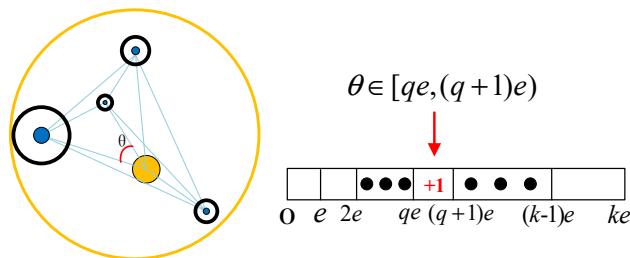


Figure 6. Feature-encoding method.

- (1) Initialize the feature vector $v = (0, 0, \dots, 0)^T \in \mathbb{R}^{n \times 1}$, where $n = \lfloor 2\pi/e \rfloor$, and e is the discrete factor.
- (2) Using the “constellation” composed of the crater to be matched and m surrounding craters, calculate the angle θ between the craters, according to $\theta \in [qe, (q+1)e]$, $v[q] = v[q] + 1$; this process is discretization, and discretization makes the feature more robust.

2.3.1. Frame-Frame Match

“FFM” matches the crater detected in the k th frame Θ^k with the craters $\Theta' = \Theta^{\{0, 1, \dots, k-1\}}$ that have been successfully matched in the image of the previous frame. There are three statuses of the crater in the Θ' , namely “new” crater S_1 , “mature” crater S_2 , and “old” crater S_3 . The status is updated by the matching result. If a crater is detected for the first time, i.e., the crater is not associated with any previous craters, then the crater will be added to the set Θ' , and the status is “new”. If a crater in the Θ' is associated with a crater for N consecutive frames, its status is updated to “mature.” If a crater lost matching for M consecutive frames, the status changes to “old” and will be deleted later. The state-transition process is shown in Figure 7a.

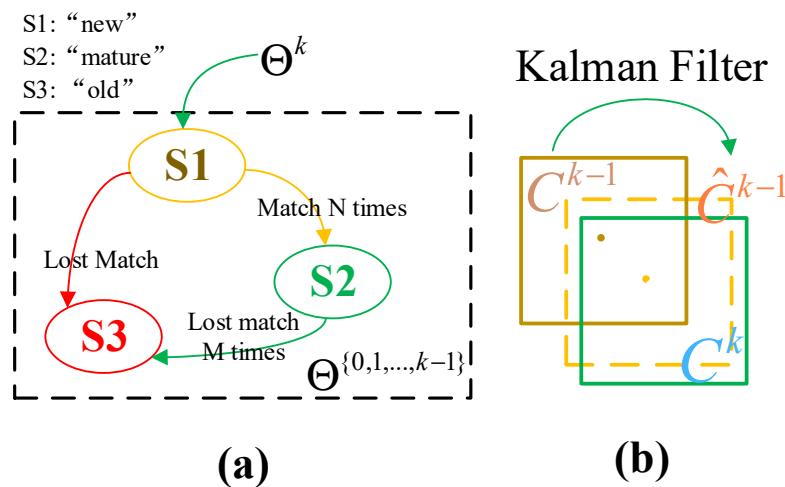


Figure 7. (a) State-transition diagram. (b) Frame $k - 1$ matching with frame k , using Kalman filter to predict crater’s position of frame $k - 1$. C^{k-1} is the craters’ state in the frame $k - 1$ and \hat{C}^{k-1} is the prediction of C^{k-1} in frame k by Kalman filter.

To calculate the IOU distance, it is necessary to predict the position of the crater in Θ' under the current field of view, which is shown in Figure 8. In this paper, the Kalman filter [23] is used to predict it. The state of a crater in Θ' is expressed as

$$x = [x, y, r, h, \dot{x}, \dot{y}, \dot{r}, \dot{h}]$$

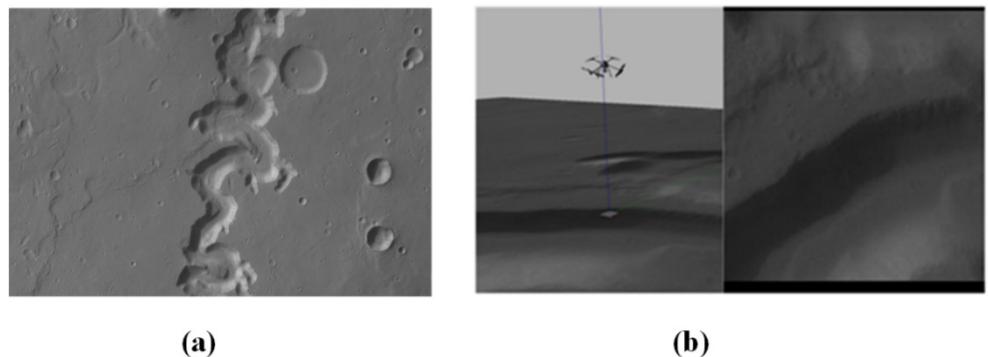


Figure 8. (a) Bandeira Mars Crater Database. (b) Left, gazebo simulation environment; right, image captured by camera.

Among them, x and y are the coordinates of the center of the target frame, r is the aspect ratio, h is the height, the modeling is a linear motion model, and the remaining variables are the corresponding speeds. The prediction equations are

$$x' = Fx^k \quad (6)$$

$$P' = FP'F + Q \quad (7)$$

where x' is the predicted value of the state quantity, F is the state-transition matrix, P is the predicted value of the covariance matrix, and Q is the system error.

The newly arrived detection frame at the k th moment and the matched craters in Θ' are matched through the KM algorithm, and the matching result is input as the observation value into the update equation of the Kalman filter:

$$y = z - Hx' \quad (8)$$

$$K = P'H^T (HP'H^T + R)^{-1} \quad (9)$$

$$x^{k+1} = x' + Ky \quad (10)$$

$$P = (I - KH)P' \quad (11)$$

where z is the mean vector of the bounding box, excluding the speed change, H is the measurement matrix, R is the noise matrix, and K is the gain of the Kalman filter.

Summarizing the above process, the FFM algorithm executes as follows.

1. The Kalman filter calculates the predicted value C^{k-1} of the crater \hat{C}^{k-1} . Calculate the IOU between \hat{C}^{k-1} and C^k .
2. Encode the feature of craters C^k and calculate the distance between features.
3. Input the distance into the KM algorithm, matching craters by IOU first, and match the remaining unmatched craters using the distance of the feature.
4. Use successfully matched craters to update the Kalman-filter parameters and update the state of craters in Θ' .

2.3.2. Frame Database Match

The pre-established database contains the three-dimensional coordinates $X = (x_i^w, y_i^w, z_i^w)$, where w represents the world coordinate system (the celestial Cartesian coordinate system). In the first frame, assuming that a rough pose P_{initial} is available and the camera internal parameter K is known, one must project the craters in the database to the pixel coordinates:

$$(u, v)^T = \Gamma(C) = K P X^T \quad (12)$$

It is necessary to calculate the circumscribed rectangle of the ellipse projected from the crater on the pixel plane. One takes m points on the circle and projects them under the pixel plane to obtain the ellipse equations and corresponding circumscribed rectangles of all craters in the database under the current field of view. This crater set is denoted Θ^{global} . The IOU distance and feature encoding distance of the target frame in Θ^{global} and Θ^k are calculated and input into the KM algorithm to complete the match. This process is similar to FFM.

During navigation, the recognition algorithm first uses FFM, and when the number of successful matches is less than the threshold, FDM is used. The input pose of FDM is the predicted value of the output pose of the previous pose through the Kalman filter. Finally, the matching result is input into the EPnP algorithm to obtain the pose of the current frame. On one hand, the pose is output directly; on the other hand, as FDM input, it rematches the crater of the current frame with the database and updates the state of craters in Θ' .

3. Results

3.1. Experimental Dataset

At present, many researchers have studied crater detection methods and established a crater database of the Moon, Mars, and other planets through artificial or automatic detection methods [8,24–27]; herein, the Bandeira Mars Crater Database is used for experimental data (Figure A1). The Bandeira database contains a length of $40 \text{ km} \times 59 \text{ km}$, covering an area of 2360 km^2 , which is located between Mars $-47.66^\circ\text{E}/-48.68^\circ\text{E}$ and North latitude $7.28^\circ\text{N}/7.95^\circ\text{N}$. This area contains a huge number of craters. There are a total of 3050 hand-marked craters in the database, with a radius between 20 and 5000 m. This dataset has been used in many studies to facilitate the comparison of method performance [17].

Under the Robotic Operating System (ROS), the abovementioned Mars area was input to the gazebo [28] physical simulation platform, and the PX4 [29] flight control system equipped with a camera was used to obtain the flight attitude and camera data, as shown in Figure 8.

Using gazebo, the ascent, descent, and level flight of the spacecraft were simulated. The sequence rate is 10 frames per second (fps). The experiment used four sequences, i.e., Seq1–4. The shooting area of different sequences is different, as is the average crater density. Figure 9 shows their ground-truth trajectories. The image resolution used in the

experiment was 1024×1024 . The flight altitude range of simulations Seq1–4 was between 600 and 1000 m.

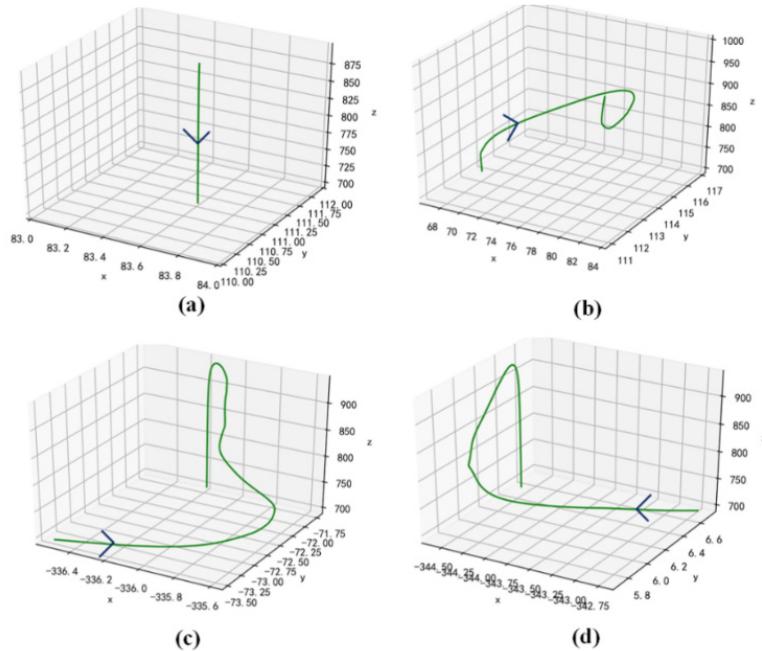


Figure 9. (a–d) Ground-truth trajectories of Seq1–4.

3.2. DPCDN Validation

3.2.1. Training Details

During training, the Bandeira database and images taken by the high-resolution stereo camera (HRSC) in the Express spacecraft were used to form a dataset. There are 9000 images in the training set and 3000 in the test set. The resolution of each image was 600×600 . The strategies of data augmentation include random rotation, random flips, random shift, and random Gaussian blur.

The strides of the feature-extraction layer are 4, 8, 16, and 32. The positive samples in the training set were allocated to different detection layers according to their sizes. P3–P6 were associated with the size ranges of the positive samples, i.e., $[-1, 12], [12, 32], [32, 64], [64, 99999]$ respectively. The stochastic-gradient-descent optimizer was used, and the parameters were randomly initialized by the Kaiming strategy [30]. The training epochs' number was 30 and the learning rate was 0.01.

3.2.2. DPCDN Results

In this subsection, the detection effects of DPCDN are compared with different networks (Urbach [31], Ding [32], CraterIdNet, and Bandeira). The results are shown in Table 1, which reports the average F1 scores of the different regions, including the West, Central, and East regions. The smallest crater instance detected by DPCDN is 6 pixels, which is smaller than DPCDN because of the dense anchor point. The DPCDN performance was better than that of the other networks. To facilitate the comparison with other methods, only craters larger than 16 pixels are compared in the present work because the previous method cannot detect small craters.

The FEL in DPCDN improved the detection speed. Then, the performance of DPCDN was compared under different backbone networks; in addition, the parameters, operating speed, precision rate, recall rate, and F1 score were compared. Operating time refers to the time from image input to network forward propagation and output NMS operation. Table 2 shows the results. The backbone network includes Vgg16 [33], resnet18 [34], and the FEL proposed in this paper. The F1 scores of each backbone all exceed 97%, and the number of

parameters using the FEL is much lower than that of Vgg16 and Resnet18, indicating that the FEL has the function of compressing parameters without losing performance.

Table 1. F1 score of different regions and detection networks.

Network	Region		
	West	Central	East
Urbach	67.89%	69.62%	79.77%
Bandeira	85.33%	79.35%	86.09%
Ding	83.89%	83.02%	89.51%
CraterIDNet	90.86%	90.02%	93.31%
DPCDN	95.40%	96.30%	96.40%

Table 2. Performance of different backbones.

	FEL	Vgg16	Resnet18
Parameters (Mbits)	9.7	40.3	73.3
Speed (fps)	9.43	3.23	3.4
Recall (%)	96.9	97	97.0
Precision (%)	97.4	97.4	97.3
Average F1 score (%)	97.2	97.2	97.0

The design of dense anchor points effectively improves the detection of dense, small craters, which is why DPCDN is better than other methods, and it also improves detection precision. The above experiment uses F1 score when $\text{IOU} = 0.5$ for evaluation. However, $\text{IOU} = 0.5$ has the problem of insufficient accuracy in relative terrain navigation based on the crater. IOU indicates how close the detection bounding box is to the ground truth. Average Precision (AP) value is used to evaluate the performance of using the dense anchor points. The AP value is the area enclosed by the precision-recall curve (PR curve) and the horizontal axis of the coordinate axis under a certain threshold. Figure 10 shows the PR curve with and without dense anchor points when $\text{IOU} = 0.8$. The interval between the two curves in the blue box represents the difference between the two methods. It can be seen from the figure that, without dense anchor points when the recall rate is 0.87, the accuracy is close to 0, while the accuracy of the network with dense anchor points is still 0.9 at the same recall value. Using the 0.5–0.95 threshold proposed by MSCOCO [35], the mean AP (mAP) value is used as the evaluation method. The mAP of the network without dense anchor points is 0.767, and that of the network with dense anchor points is 0.855. The performance improved by 11.47%.

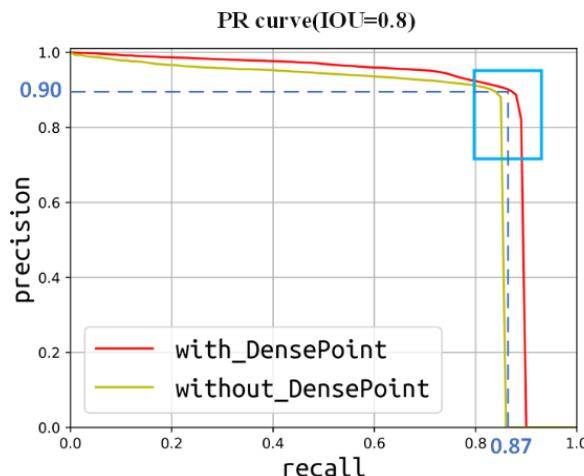


Figure 10. PR curve with or without dense anchor point.

Using the same structure, DPCDN was tested on the lunar database after retraining to verify the robustness of the network to different scenarios. Figure 11 shows the experimental results. The image is from the 120-m-resolution scientific data of the Chang'e-1 CCD stereo camera (Figure A2) [36]. The data comprise an entire month using simple cylindrical projection. The longitude and latitude range were chosen as $[-53^\circ, -23^\circ]$ and $[-15^\circ, -8^\circ]$, respectively. This area exhibits large changes in illumination, complex crater scenes, large-scale changes, and large differences compared to Mars scenes. The crater data used are the lunar data released by Robbins [25] in 2018. The data cover craters with a radius of more than 1 km. The F1 score is 95.6%. Figure 11 shows the results for the Moon and Mars.

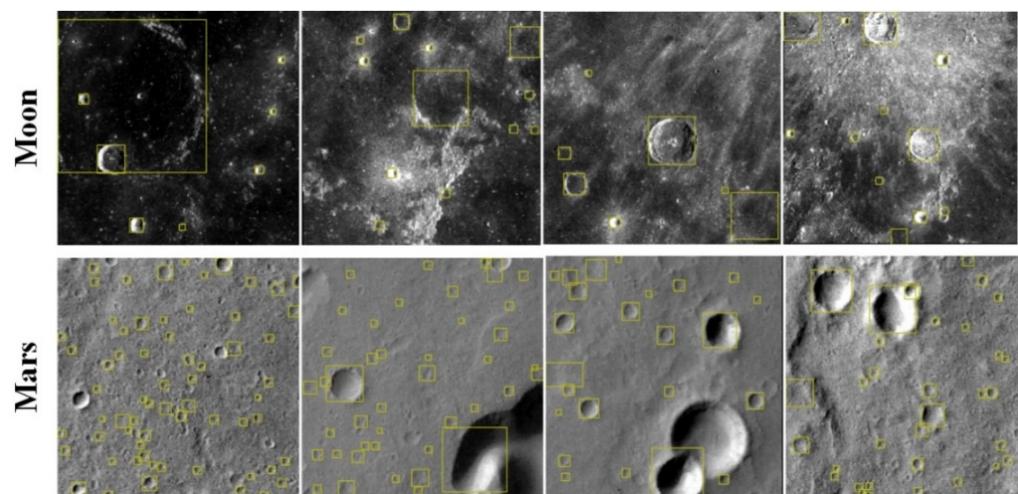


Figure 11. Craters on the Moon and Mars detected by DPCDN.

Experiments were conducted on detection precision because the identification of craters requires the use of crater position. One can simply use the center of the bounding box as the crater position. Using root-mean-square error (RMSE) to evaluate the precision, the RMSE of a total of 24,629 craters (with repeated craters and different noises) of 600 randomly generated images is 0.46 pixels, which means that DPCDN achieves pixel-level accuracy.

3.3. Recognition Validation

Random position noise and length, as well as width noise, were added to the ground-truth bounding boxes to simulate the output of the detection algorithm. First, the performance of FDM and FMM were evaluated separately, and, finally, the position estimation performance of the entire system was checked.

3.3.1. Validation of FDM Performance

Using bounding boxes instead of points as matching objects improves the success rate of matching and is more robust to noise. Figure 12 compares the matching accuracy of the FDM algorithm under different disturbance levels. The perturbation angle ranges from 0° to 4° , the step length is 0.4° , and the position perturbation ranges from 0% to 25%. The relationship between the matching rate and noise level is shown in Figure 12. In Ref. [3], images are used to show the corresponding results, in which the angle perturbation is 4° and the matching rate is already less than 20%, but the FDM matching rate is 58%. The matching method used in this paper is more robust than that in the literature [3] because using the IOU of bounding boxes and features information improves the matching rate.

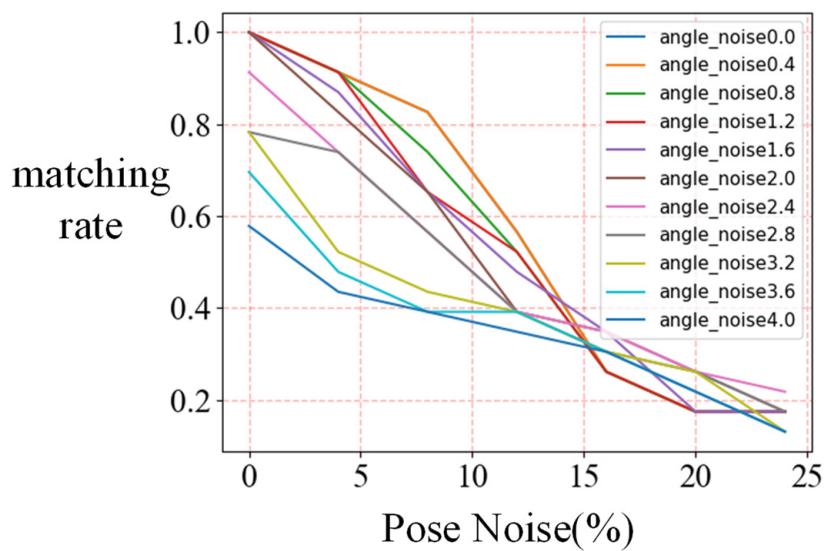


Figure 12. Matching rate vs. pose noise and angle noise.

3.3.2. Validation of FFM Performance

Matching accuracy and error between images are used herein to evaluate FFM performance. These two indicators can be used to evaluate the matching effect of the entire time series.

$$\text{accuracy} = 1 - \frac{C_{\text{miss}} + C_{\text{switch}} + C_{\text{fp}}}{C_{\text{all}}} \quad (13)$$

where C_{all} represents all consecutive crater sequences, C_{miss} represents missed craters, C_{fp} represents wrongly matched craters, and C_{switch} represents the number of orbital switches.

$$\text{error} = \frac{\sum_{i=0}^{i=N} \text{dist}(\hat{\text{pos}} - \text{pos})}{N} \quad (14)$$

where N represents the total number of craters to be matched, and the formula calculates the average distance between the predicted value and the ground truth.

Table 3 lists the accuracies and errors of FFM in different sequences. With different crater densities, the matching accuracy is between 96% and 97%. The average Gaussian noise used in the experiments is 0 and the variance is 0.5 pixels. Densities have an impact on detection speed because the time complexity of the KM algorithm used in the matching algorithm is related to the number of craters.

Table 3. Average crater density, matching accuracy, error, and speed under different sequences.

Sequence	Average Crater Density	Accuracy (%)	Error	Speed (fps)
Seq1	123.1	97.0	0.196	21.65
Seq2	89.0	96.8	0.176	17.26
Seq3	38.4	96.8	0.184	79.21
Seq4	116.2	96.5	0.185	18.80
Average	91.7	97.2	0.185	34.23

Figure 13 visualizes the matching results between frames, where the images of (1-a)–(2-a) are the experimental sequences discussed in this paper, and the algorithm is applied to the landing sequence of Chang'e 3. The yellow rectangles are the DPCDN detection result, and the rectangles with the same color in different images indicate the same crater.

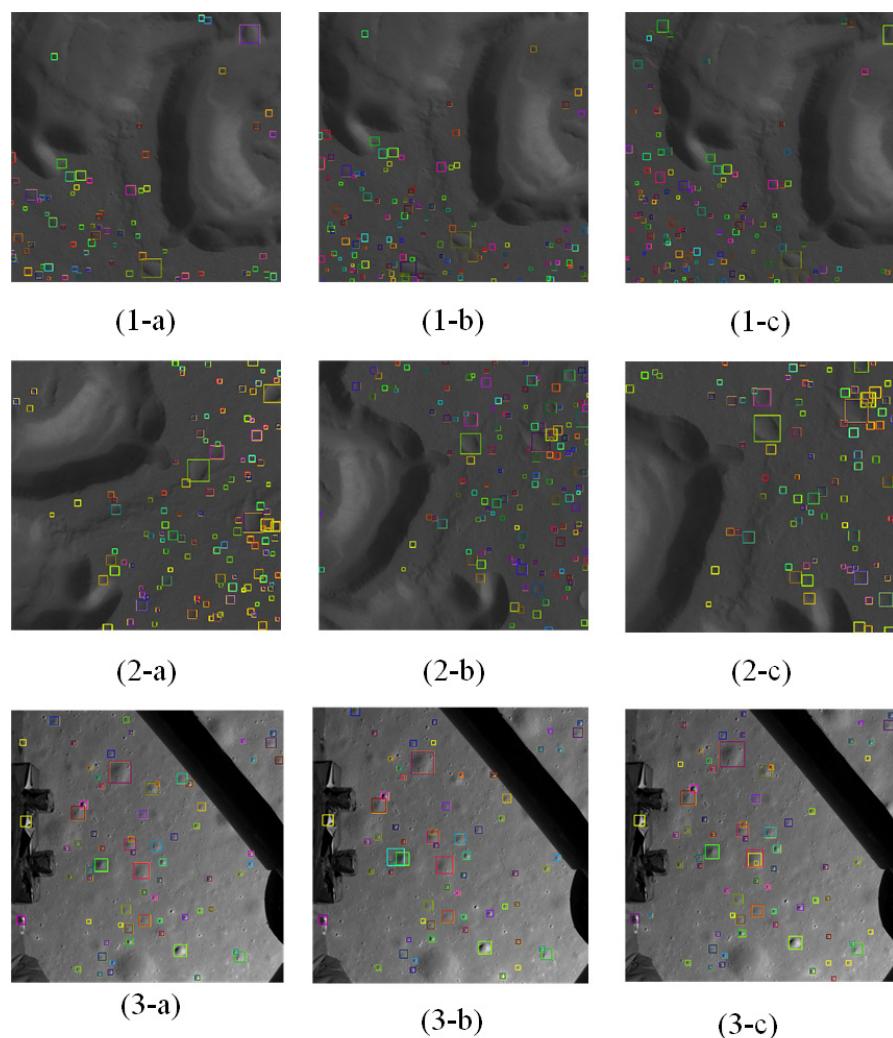


Figure 13. Visualization of FFM results. The images of (1-a)–(2-c) are the experimental landing sequences simulated by Gazebo and (3-a)–(3-c) are the landing sequence of Chang'e 3. The yellow rectangles are the DPCDN detection result, and the rectangles with the same color in different images indicate the same crater.

3.3.3. Validation of Recognition

The recognition algorithm combines the matching result of FFM and FDM. When the number of correctly matched craters is greater than four, the recognition is considered successful [17]. Because the correct recognition of more than four craters can be input into the PnP algorithm to calculate the pose, the input of the wrongly identified craters into the PnP will reduce the accuracy of position estimation and even obtain an incorrect result. In related papers, the mismatched input is called an “outlier point” and the proportion is called the outlier rate. The recognition algorithm is evaluated by accuracy in the present paper, where accuracy = $1 - \text{outlier rate}$. Table 4 shows the average accuracy of different sequences, and the accuracy rate is 98.5%. Seq2 and Seq4 have more frequent motion acceleration mutations in flight, so the recognition accuracy is lower than that of Seq1 and Seq3.

Table 4. Recognition accuracy and speed in different sequences.

Sequence	Seq1	Seq2	Seq3	Seq4
Accuracy (%)	99.6	96.5	99.5	98.5
Speed (Only FDM, fps)	7.41	10.84	26.61	8.11
Speed (FDM + FFM, fps)	17.34	20.07	46.93	16.56

In the existing research, the Kalman filter is used to estimate the pose and the pose is used for crater recognition. The FDM in this paper is similar to this process. The matching between sequence images is integrated. In the FFM, the Kalman filter is used to estimate the parameters of each crater and its motion state at the same time, which accelerates the recognition speed because FDM requires additional projection operations and fitting of the ellipse equation for each crater in this view. Table 4 shows the recognition speed of FDM only and the combination of FDM and FFM. Results show that the latter's average speed is 25.22 fps, which is 1.99 times faster than the former. An Intel core-i7 processor and the pytorch libraries by python language were used in the test, and the matching threshold used in the recognition algorithm was 10.

3.4. Pose-Estimation Experiment Results

The above-described experiments verify the performance of the detection and recognition algorithms. The error between the ground-truth pose and the pose estimated by the proposed system were tested next. The crater detection and recognition system proposed requires only the input of the initial pose of the first frame; no additional input is required.

The pose estimated by the crater using PnP is projected onto the image sequence, and the result is shown in Figure 14. The projection result is consistent with the real crater ellipse, indicating that the pose estimation is effective.

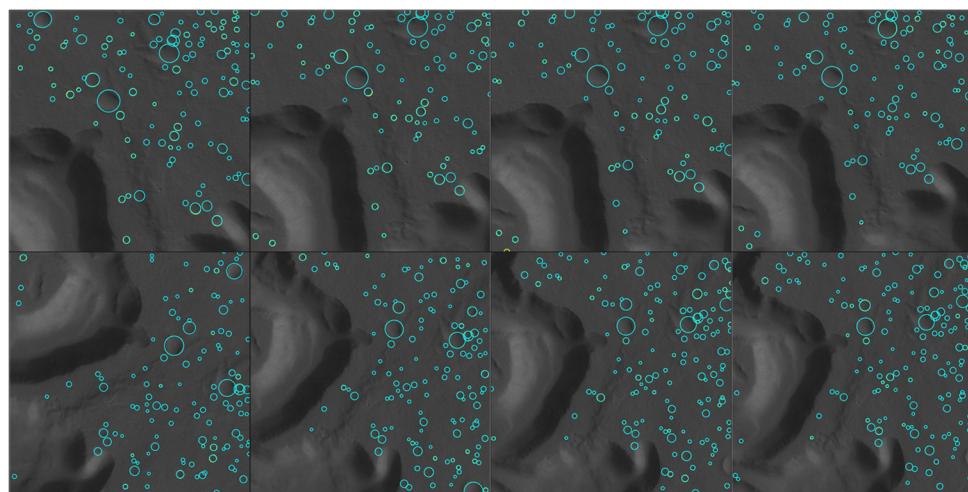


Figure 14. Result of projecting craters onto images by estimated pose.

The sequences are input into the detection and recognition system, and the estimated pose is the output. Figure 15 shows the estimated and ground-truth poses. The relative pose error (RPE) and absolute trajectory error (ATE) are then computed, and the results are shown in Table 5. The RPE measures the local accuracy of the trajectory over a fixed time interval and ATE compares the absolute distances between the estimated and ground-truth trajectories [37]. Seq3 tests the situation in which the spacecraft is flying downwards approximately vertically, which is simpler, and the pose-estimation accuracy is thus higher. Seq4 has larger motion changes and complex conditions, so it has lower accuracy than other sequences. Results show that the RMSE of trajectories is less than 10 m and the angle error is less than 1.5°. This accuracy meets the requirements of soft landing.

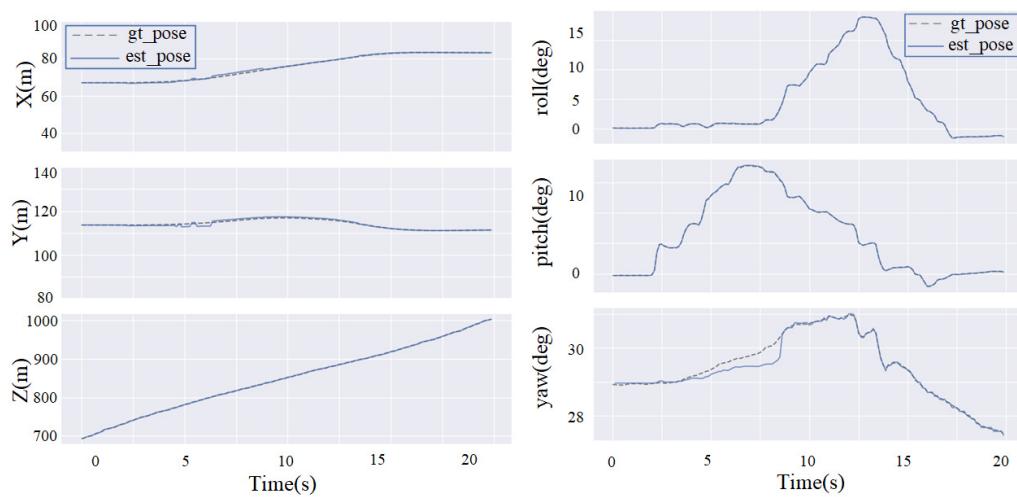


Figure 15. Ground-truth vs. estimate trajectories. The subfigures from the upper left to the lower right represent the estimated and ground truth trajectories of x , y , z direction, and rolling, pitch, and yaw, respectively. Dashed lines represent ground truth trajectories, and solid lines represent estimated trajectories.

Table 5. APEs and RPEs of trajectories.

	Sequence			
	Seq1	Seq2	Seq3	Seq4
APE (m)	3.93	5.89	0.11	4.74
RPE (m)	3.26	6.02	0.15	3.84

4. Discussion

In deep-space exploration missions, crater-based optical autonomous navigation algorithms must be able to detect and identify craters quickly and robustly. The recognition algorithm is related to the output of the detection network, and one must consider the detection and recognition algorithm as a whole. Using the object detection network instead of the segmentation network for crater detection can save the time spent on crater edge detection and fitting. According to the bounding box of the object detection network, the KM algorithm and Kalman filter are used to associate the current frame with pre-matched crater data, which can quickly identify the crater.

In terms of algorithm speed, Tables 2 and 4 suggest that on a 1024×1024 image the detection speed can reach 9.43 fps and the recognition speed can reach 25.22 fps. In terms of algorithm robustness, Figures 2 and 12 suggest that using the bounding box instead of the center point for direct matching is more robust to noise. When the position noise reaches 5% and the angle noise reaches 0.8° , the matching recognition rate exceeds 90%. Figure 10 suggests that the single-stage target detection network with dense anchor points can deal with the situation of small and dense craters more effectively. Compared with no dense anchor point, the mAP of DPCDN increased by 11.47%.

Moreover, the pose-estimation method in this paper uses the PnP algorithm directly on the results of the crater recognition. How to fuse other sensors, such as inertial measurement units, to achieve higher accuracy and design a long-term stable and real-time pose-estimation algorithm based on craters deserves further attention.

5. Conclusions

Detection and recognition methods used in existing crater-based pose estimation systems were analyzed in this paper, and a crater detection and recognition method consisting of two stages was designed. In stage 1, the single-stage crater object detection network with dense anchor points can deal with small and dense scenes and achieved

state-of-art crater detection performance. In stage 2, the recognition algorithm matches associated craters in the image with previously identified craters and a pre-established craters database using the KM algorithm. The pattern composed of the target crater and the surrounding craters is encoded. The distance of encoding features and the IOU is used as the weight of the KM algorithm. Experimental results show that the F1 score of the detection network is better than those of other detection methods. The performance with dense points improved by 11.47% compared to without dense points. Experiments on images of Mars and the Moon show that the proposed DPCDN can handle a variety of scenarios. Matching based on the bounding boxes is more robust to noise than the direct matching method using the center of the crater. Combining the FFM with FDM, the recognition enables the achievement of real-time speed on the CPU (25FPS). Relying on a high-accuracy network and the sequence image information, we provide an efficient crater detection and recognition method for pose estimation.

Author Contributions: Z.C. is responsible for the research ideas, overall work, the experiments, and the writing of this paper. J.J. provided guidance and modified the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Natural Science Foundation of China (No.61725501).

Data Availability Statement: Data are contained within the article or Appendix A.

Acknowledgments: This work was supported by the Key Laboratory of Precision Opto-mechatronics Technology, Ministry of Education, Beihang University, China.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

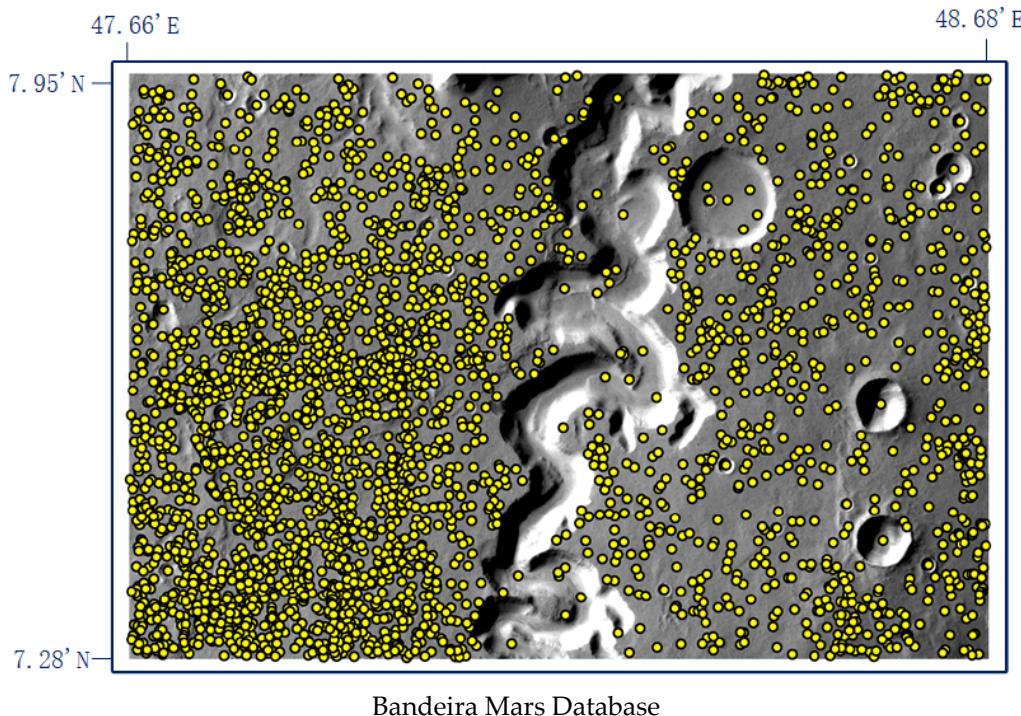


Figure A1. Bandeira Mars Crater Database. Yellow points in image are centers of craters.

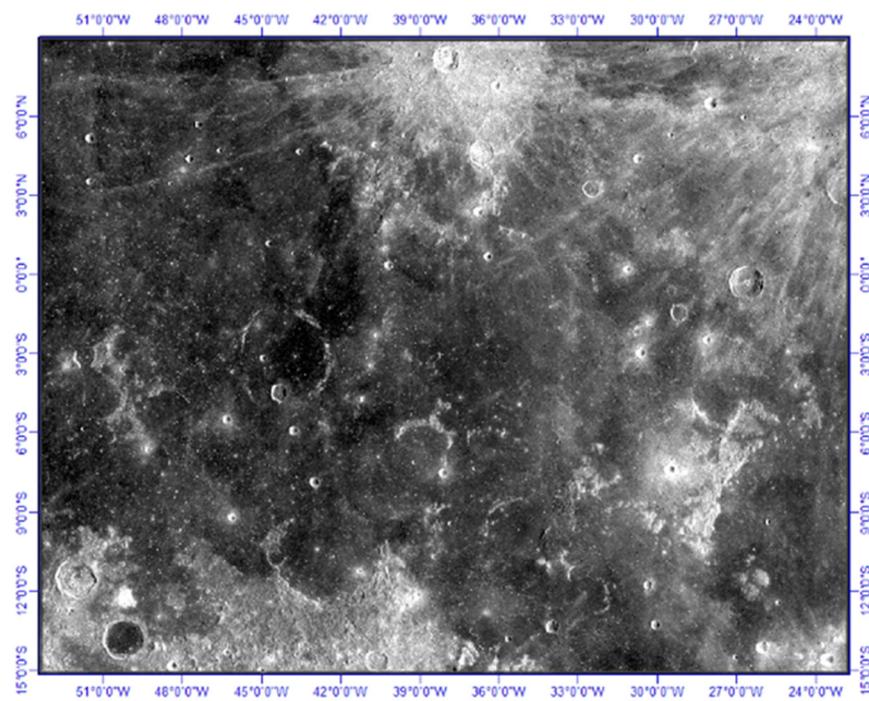


Figure A2. Lunar image from Chang'E-1 CCD stereo camera.

References

1. Downes, L.; Steiner, T.; How, J. Lunar Terrain Relative Navigation Using a Convolutional Neural Network for Visual Crater Detection. In Proceedings of the 2020 American Control Conference (ACC), Online, 1–3 July 2020; pp. 4448–4453. [[CrossRef](#)]
2. Johnson, A.E.; Montgomery, J.F. Overview of Terrain Relative Navigation Approaches for Precise Lunar Landing. In Proceedings of the Aerospace Conference, Big Sky, MT, USA, 1–8 March 2008.
3. Maass, B.; Woicke, S.; Oliveira, W.M.; Razgus, B.; Krüger, H. Crater Navigation System for Autonomous Precision Landing on the Moon. *J. Guid. Control Dyn.* **2020**, *43*, 1414–1431. [[CrossRef](#)]
4. James, K. Introduction Autonomous Landmark Based Spacecraft Navigation System. In Proceedings of the 13th Annual AAS/AIAA Space Flight Mechanics Meeting, Ponce, Puerto Rico, 9 February 2003.
5. Lepetit, V.; Moreno-Noguer, F.; Fua, P. EPnP: An Accurate O(n) Solution to the PnP Problem. *Int. J. Comput. Vis.* **2009**, *81*, 155–166. [[CrossRef](#)]
6. Klear, M.R. PyCDA: An Open-Source Library for Autonmated Crater Detection. In Proceedings of the 9th Planetary Crater Consortium, Boulder, CO, USA, 8–10 August 2018.
7. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015. [[CrossRef](#)]
8. Lee, C. Automated crater detection on Mars using deep learning. *Planet. Space Sci.* **2019**, *170*, 16–28. [[CrossRef](#)]
9. Downes, L.; Steiner, T.J.; How, J.P. Deep Learning Crater Detection for Lunar Terrain Relative Navigation. In Proceedings of the AIAA SciTech Forum, Orlando, FL, USA, 6–10 January 2020.
10. Tian, Y.; Yu, M.; Yao, M.; Huang, X. Crater Edge-based Flexible Autonomous Navigation for Planetary Landing. *J. Navig.* **2018**, *72*, 649–668. [[CrossRef](#)]
11. Leroy, B.; Medioni, G.; Johnson, E.; Matthies, L. Crater detection for autonomous landing on asteroids. *Image Vis. Comput.* **2011**, *19*, 787–792. [[CrossRef](#)]
12. Clerc, S.; Spigai, M.; Simard-Bilodeau, V. A crater detection and identification algorithm for autonomous lunar landing. *IFAC Proc. Vol.* **2010**, *43*, 527–532. [[CrossRef](#)]
13. Olson, C.F. Optical Landmark Detection for Spacecraft Navigation. In Proceedings of the 13th Annual AAS/AIAA Space Flight Mechanics Meeting, Ponce, Puerto Rico, 9 February 2003.
14. Singh, L.; Lim, S. On Lunar On-Orbit Vision-Based Navigation: Terrain Mapping, Feature Tracking Driven EKF. In Proceedings of the Aiaa Guidance, Navigation & Control Conference & Exhibit, Boston, MA, USA, 19–22 August 2013.
15. Hanak, C.; Ii, T.P.C.; Bishop, R.H. Crater Identification Algorithm for the Lost in Low Lunar Orbit Scenario. In Proceedings of the AAS Guidance and Control Conference, Breckenridge, CO, USA, 5–10 February 2010.

16. Wang, J.; Wu, W.; Li, J.; Di, K.; Wan, W.; Xie, J.; Peng, M.; Wang, B.; Liu, B.; Jia, M. Vision based Chang'E-4 landing point localization. *Sci. Sin. Technol.* **2020**, *50*, 41–53. [[CrossRef](#)]
17. Wang, H.; Jiang, J.; Zhang, G. CraterIDNet: An End-to-End Fully Convolutional Neural Network for Crater Detection and Identification in Remotely Sensed Planetary Images. *Remote Sens.* **2018**, *10*, 1067. [[CrossRef](#)]
18. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [[CrossRef](#)] [[PubMed](#)]
19. Kuhn, H.W. The Hungarian method for the assignment problem. *Nav. Res. Logist.* **2010**, *52*, 7–21. [[CrossRef](#)]
20. Munkres, J. Algorithms for the assignment and transportation problems. *SIAM J.* **1962**, *10*, 196–210. [[CrossRef](#)]
21. Lin, T.Y.; Dollar, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
22. Shang, W.; Sohn, K.; Almeida, D.; Lee, H. Understanding and Improving Convolutional Neural Networks via Concatenated Rectified Linear Units. In Proceedings of the International conference on machine learning 2016, New York, NY, USA, 19–24 June 2016; Volume 48, pp. 2217–2225.
23. Kalman, R.E. A New Approach to Linear Filtering and Prediction Problems. *J. Basic Eng.* **1960**, *82*, 35–45. [[CrossRef](#)]
24. Bandeira, L.; Saraiva, J.; Pina, P. Impact Crater Recognition on Mars Based on a Probability Volume Created by Template Matching. *Geoscience and Remote Sensing. IEEE Trans.* **2007**, *45*, 4008–4015. [[CrossRef](#)]
25. Robbins, S.J. A New Global Database of Lunar Impact Craters >1–2 km: 1. Crater Location and Sizes, Comparisons with Published Databases and Global Analysis. *J. Geophys. Res. Planets* **2018**, *124*, 871–892.
26. Stuart, J.R.; Hynek, B.M. A new global databases of Mars impact craters>1Km:1.Database creation, properties and parameters. *J. Geophys. Res.* **2011**, *117*. [[CrossRef](#)]
27. Bandeira, L.; Ding, W.; Stepinski, T.F. Automatic Detection of Sub-km Craters Using Shape and Texture Information. In Proceedings of the Lunar & Planetary Science Conference, Woodlands, TX, USA, 23–27 March 2010.
28. Koenig, N.; Howard, A. Design and Use Paradigms for Gazebo, an Open-Source Multi-Robot Simulator. In Proceedings of the Intelligent Robots and Systems, Chicago, IL, USA, 14–18 September 2004.
29. Meier, L.; Honegger, D.; Pollefeys, M. PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms. *IEEE Int. Conf. Robot. Autom.* **2015**, *2015*, 6235–6240.
30. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In Proceedings of the 2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, 7–13 December 2015.
31. Urbach, E.R.; Stepinski, T.F. Automatic detection of sub-km craters in high resolution planetary images. *Planet. Space Sci.* **2009**, *57*, 880–887. [[CrossRef](#)]
32. Ding, W.; Stepinski, T.F.; Mu, Y.; Bandeira, L.; Ricardo, R.; Wu, Y.; Lu, Z.; Cao, T.; Wu, X. Subkilometer crater discovery with boosting and transfer learning. *Acm Trans. Intell. Syst. Technol.* **2011**, *2*, 1–22. [[CrossRef](#)]
33. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
34. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 July 2016; pp. 770–778.
35. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In Proceedings of the ECCV 2014—European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014.
36. Wei, Z.; Li, C.; Zhang, Z. Scientific data and their release of Chang'E-1 and Chang'E-2. *Chin. J. Geochem.* **2014**, *33*. [[CrossRef](#)]
37. Sturm, J.; Engelhard, N.; Endres, F.; Burgard, W.; Cremers, D. A benchmark for the evaluation of RGB-D SLAM systems. In Proceedings of the Intelligent Robots and Systems (IROS), Vilamoura, Algarve, Portugal, 7–12 October 2012.