



City Research Online

## City, University of London Institutional Repository

---

**Citation:** Chekakta, Z., Zenati, A., Aouf, N. & Dubois-Matra, O. (2022). Robust deep learning LiDAR-based pose estimation for autonomous space landers. *Acta Astronautica*, 201, pp. 59-74. doi: 10.1016/j.actaastro.2022.08.049

This is the accepted version of the paper.

This version of the publication may differ from the final published version.

---

**Permanent repository link:** <https://openaccess.city.ac.uk/id/eprint/32877/>

**Link to published version:** <https://doi.org/10.1016/j.actaastro.2022.08.049>

**Copyright:** City Research Online aims to make research outputs of City, University of London available to a wider audience. Copyright and Moral Rights remain with the author(s) and/or copyright holders. URLs from City Research Online may be freely distributed and linked to.

**Reuse:** Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge. Provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.

---

---

---

City Research Online:

<http://openaccess.city.ac.uk/>

[publications@city.ac.uk](mailto:publications@city.ac.uk)

---

# Robust Deep Learning LiDAR-based Pose Estimation for Autonomous Space Landers

Zakaria Chekakta<sup>a</sup>, Abdelhafid Zenati<sup>a</sup>, Nabil Aouf<sup>a</sup> and Olivier Dubois-Matra<sup>b</sup>

<sup>a</sup>City, University of London, London, EC1V 0HB, United Kingdom

<sup>b</sup>European Space Agency, 2200 AG, Noordwijk ZH, The Netherlands

---

## ARTICLE INFO

### Keywords:

Space Landing Operations  
Robotics  
Deep Neural Network  
Relative Pose Estimation  
LiDAR Navigation

## ABSTRACT

Accurate relative pose estimation of a spacecraft during space landing operation is critical to ensure a safe and successful landing. This paper presents a 3D Light Detection and Ranging (LiDAR) based AI relative navigation architecture solution for autonomous space landing. The proposed architecture is based on a hybrid Deep Recurrent Convolutional Neural Network (DR-CNN) combining a Convolutional Neural Network (CNN) with a Recurrent Neural Network (RNN) based on a Long-Short Term Memory (LSTM) network. The acquired 3D LiDAR data is converted into a multi-projected images and feed the DR-CNN with depth and other multi-projected imagery. The CNN module of the architecture allows an efficient representation of features, and the RNN module, as an LSTM, provides robust navigation motion estimates. A variety of landing scenarios are considered, simulated and experimented to evaluate the efficiency of the proposed architecture. A LiDAR based imagery data (Range, Slope, and Elevation) is initially created using PANGU (Planet and Asteroid Natural Scene Generation Utility) software and an evaluation of the proposed solution using this data is conducted. Tests using an instrumented Aerial Robot in Gazebo software to simulate landing scenarios on a synthetic but representative lunar terrain (3D digital elevation model) is proposed. Finally, real experiments using a real flying drone equipped with a Velodyne VLP16 3D LiDAR sensor to generate real 3D scene point clouds while landing on a designed down scaled lunar moon landing surface are conducted. All the test results achieved show that the suggested architecture is capable of delivering good 6 Degree of Freedom (DoF) pose precision with a good and reasonable computation.

---

## 1. Introduction

Space missions have demonstrated a crucial need for further research and development activities in autonomous guidance, navigation, and control systems (GNC). Space research roadmaps indicate that achieving safe and accurate landing operations is one of the top priorities that requires complex but efficient and robust GNC algorithms operating with advanced on-board sensors. Indeed, accurate relative navigation is one of the key modules for a safe space landing, as it allows the lander to approach the targeted landing site with high precision and adapts to inaccuracies related to winds, turbulence and other uncertainties available in the lander system. Therefore, relative pose estimation is considered an important topic and an active field of research to achieve autonomous space landing operations.


Relative pose estimation is required in the following space missions: *i*) orbital navigation and spacecraft rendezvous with a target for docking, refuelling or debris removal tasks; *ii*) planetary robots exploration and localisation; *iii*) autonomous landing and terrain navigation, where the space vehicle should be able to estimate its relative position and attitude using data acquired from on-board sensors. Present relative pose estimation techniques use visual 2D data from a monocular camera [1, 2] or stereo cameras [3, 4, 5, 6], thermal Infrared (IR) camera 2D data [7], or LiDAR 3D data [8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]. Small and lightweight onboard cameras are now the primary sensor adopted for autonomous rendezvous and close-proximity operations [20]. In [21], the authors review methods for spacecraft orbital pose estimation with great details.

While precise autonomous space landing and navigation is always a critical challenge, safety is a crucial factor when landing a space vehicle on another planet. Future space landers are expected to travel to faraway and scientifically

---

\* note text

\*Corresponding author

 zakaria.chekakta (Z. Chekakta)

ORCID(s):

<sup>1</sup>bhry

interesting planets and will have to land on unknown and rough locations on dangerous terrains and under light changing conditions. To deal with this challenge, Terrain Relative Navigation (TRN) was developed, as a visual relative navigation technique, to place the Perseverance Rover and the Ingenuity helicopter safely during landing descent. TRN takes images, compares them with an on-board orbital map, and then pinpoints safely the lander during descent. Even though this technique was successful for the Mars 2020 mission, it is dependent on lighting conditions at the time of acquiring the images required to match images from a passively lit scene [22]. Due to this difficulty, landing safely and precisely with existing vision-based landing systems has proven to be hard since it requires detailed orbital reconnaissance, a prior hazard maps, and knowledge of time-of-day restrictions on landing to ensure similar lighting conditions in orbit and descent. To overcome these challenges, a qualified space 3D LiDAR sensor, which is robust against change of lighting conditions, can be adopted instead of/or in addition to a typical optical landing sensor suite.

On the other hand, machine learning (ML), specifically deep learning (DL), is becoming increasingly prevalent in space applications, echoing its unprecedented success on the terrestrial and computer vision fronts. Based on a Digital terrain Map (DTM) of the lunar surface, the CNN structure proposed in [23] can be trained and deployed to return the source's location relative to the DTM. In 2020, the work in [24] investigated the possibility of using the deep network LunaNet to address the TRN problem and presented a robust strategy for relative pose estimation. LunaNet has therefore been used to identify craters across a lander trajectory and match them with known features of the lunar. Compared to a classical image based crater detection approach [25], LunaNet and Extended Kalman Filter (EKF) improve TRN accuracy and reliability under different lighting conditions. In [26, 27], a comparison study of different learning-based approaches for terrain classification and recognition is provided. One of those approaches is a CNN-LSTM based algorithm [28]. In [29] a Deep Neural Network architecture dealing with autonomous lunar landing scenario and based on raw greyscale images obtained from an on-board camera, is proposed to predict fuel-optimal control actions. In spite of these new developments, space landing missions generally still adopt more traditional approaches [30, 31]. The next two sub-sections present related works on AI- based and LiDAR-based navigation techniques for orbital scenarios and for the Entry, Descent and Landing (EDL) phase scenarios, respectively.

### 1.1. Orbital navigation

Space navigation based on deep learning has been the subject of several previous studies. In [32], the authors demonstrate how ML can be applied to space data, and demonstrated the use of deep learning on-device to improve spacecraft navigation. Furthermore, Deep Neural Networks based algorithms showed robustness when it comes to setting the initial space lander attitude regardless of the dynamic lighting [33, 34]. As high-performance space-graded computation devices are being developed, CNN-based approaches are set to be deployed for pose determination problem without resource to complex dynamics models [23].

The work in [35] suggests a deep CNN to deal with the problem of relative pose classification. Using transfer learning, the AlexNet model [36], pre-trained on ImageNet data, [37] is adapted to Tango space images from the Prisma mission [38]. These results are further enhanced by establishing the Spacecraft Pose Network (SPN) in [39]. In [35], the authors investigate both aspects of the SPN: estimation of the target pose and quantification of the estimation uncertainty. Further experiments are performed to compare the SPN technique to their earlier CNN based work [35], [40]. In [41], an estimator, which directly relates spacecraft pose to an AlexNet-based CNN architecture, is presented. A synthesised image of a 3D model was used to create data set, using Gazebo simulation software [42], for the training stage of a CNN network. Using hardware simulators, real images are captured to test the trained CNN. Based on these produced data, the work in [43] proposes to estimate spacecraft attitude quaternion using a CNN-based regression scheme. Using Gazebo to generate synthetic images labelled with their poses, an architecture similar to that described in [35], is used in [44] to determine relative poses for space assembly tasks. The work in [45] proposes a pre-trained GoogLeNet on Unreal Rendered Spacecraft On-Orbit (URSO) dataset for spacecraft pose estimation. Experiments have shown that the proposed weighted Euclidean based pose estimation model has a reasonably good estimation accuracy, whereas an exponential model has low accuracy in estimating the orientation of the target's pose. Oestreich et al. [33] use AlexNet-based transfer learning for post-classification and attitude refinement technique, focusing on resolving issues about DL's applicability to this domain. Cosmas and Kenichi [46] assess the inference capability of space on-board devices to check if a CNN-based spacecraft pose estimation algorithm is feasible. Considering power and cost-effectiveness, their analysis reveals that a U-Net-based detection network outperforms a ResNet-50-based detection network, but falls short of the established ResNet34-U-Net model. Cassini's et al. [47] assesses the accuracy, during pose initialization, of the existing monocular CNN based pose estimation algorithms. In ref [48], authors present a deep learning pipeline, which estimates the relative pose of a spacecraft by incorporating temporal information in a

rendezvous sequence. A comprehensive survey of Deep Learning-based spacecraft relative navigation methods can be found in [49].

As a result of the proven advantages of 3D LiDAR sensor and its demonstrated robustness in space applications, the research community proposes several strategies utilizing LiDAR data for space relative navigation. Galante et al. [11] introduce the Argon relative navigation system, which consists of a flash LiDAR in addition to a stereo optical camera configuration that can be used for relative navigation. Argon estimates the pose using extracted edges from the collected data in combination with an Iterative Closest Point (ICP) algorithm. The use of an Oriented Unique Repeatable Clustered Viewpoint Feature Histogram (OUR-CVFH) or Spin Images [50] in combination with ICP, as well as a Multiplicative Extended Kalman Filter (MEKF) to track the target platform [10, 16, 51] are examples of other LiDAR based orbital navigation solutions that are proposed in the literature. An adaptive H inf filter is used to iteratively filter the HoD-S 3D local feature matches in Ref. [15]. Reference [52] elaborates on that work by evaluating several 3D features and recursive filtering algorithms on real and simulated data. Rather than estimating poses at single frames, Kechagias-Stamatis et al. [53] suggest a Deep Recurrent CNN (DRCNN) mechanism to predict a spacecraft's relative pose between two consecutive frames. Synthetic (Elite platform) and real data (Envisat mockup model) are used in the algorithm's test phases. The study demonstrates that the DRCNN outperforms current techniques like Iterative Closest Point (ICP) [54]. Reference [53] provides a list of the current 3D LiDAR-based space pose estimation methods.

## 1.2. Entry, Descent and Landing (EDL) navigation

The estimation of the spacecraft's status during EDL is linked to the several on-board sensor modalities. A star tracker and gyroscopes are used to determine the lander orientation. The procedure of measuring altitude using a radar or laser rangefinder is known as altimetry. The horizontal position is determined in visual TRN (also called Map Relative Localization (MRL)), and the velocity is obtained either by the use of specialized instruments such as Doppler radar or LiDAR or by using velocimetry (frame-to-frame synchronization of images). The work in [6] states that a flash LiDAR can be used for altitude and velocity measurements between 20 km and 100 m. A Doppler LiDAR is used for altitude and velocity measurements between 2.5 km and 10 m and a laser altimeter is used for TRN measurements between 15 km and 5 km. Figure 1 depicts the different LiDAR based sensors in operation during a lunar landing scenario.

Researchers involved in the Autonomous Landing and Hazard Avoidance Technology project (ALHAT) provide an overview of both passive and active sensor options for lunar landing [56]. Both frame-to-frame and frame-to-map DEM matching are included in the study. In [57], the authors discuss field testing using a laser altimeter and a flash LiDAR for LiDAR-based TRN. LiDAR point clouds were matched to similar resolution topographic maps.

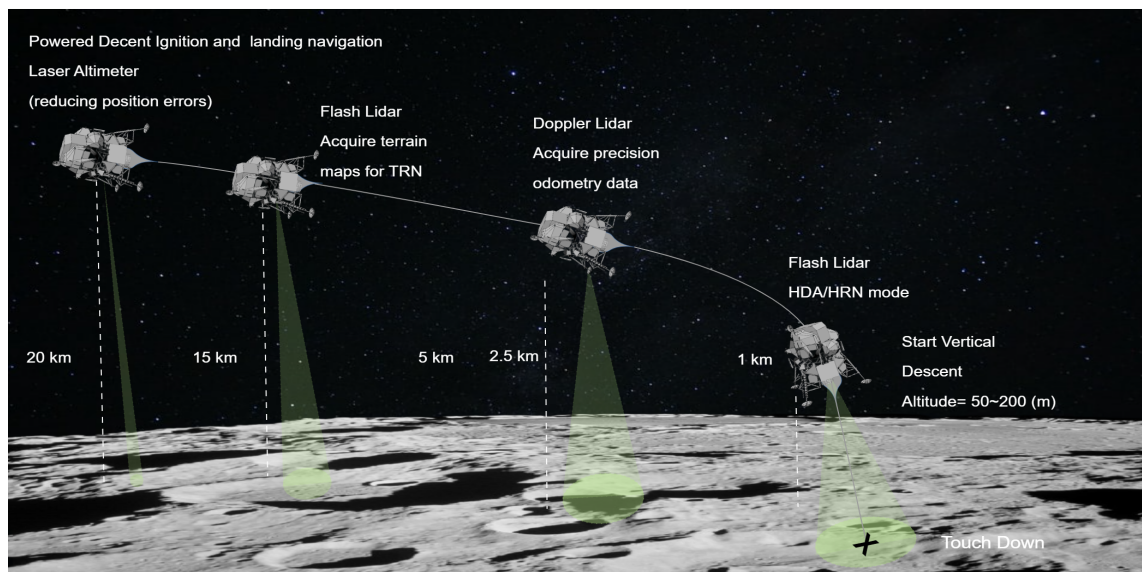


Figure 1: LiDAR-based Landing navigation [55]

The results demonstrate that by utilising internal metrics on terrain relief and data correlation, LiDAR-based TRN can produce position estimations of less than 90 *m*. Sensitivity analyses indicate that the method maintains its matching performance even when the initial location is imprecise by up to 1.6 *km*. In [58], authors estimate velocity using a scanning LiDAR without feature matching. The study in [59] investigated the performance of LiDAR-based Map Relative Localization (MRL) for Europa Lander. Using realistic trajectory, LiDAR features, and future available maps, simulation results show promising performance of LiDAR-based MRL and prove to be competitive with passive-optical MRL when topography is sufficiently rich. In contrast to passive-optical MRL, LiDAR-MRL is not affected by illumination conditions. Recently, researchers developed a new LiDAR-inertial navigation and mapping algorithm for precision landing, where the landing spacecraft's overall trajectory can be estimated using a smoothing framework [22].

### 1.3. Paper contributions

Space-qualified LiDARs are developing and the current advanced 3D imaging LiDAR systems have the potential to operate from altitudes up to 5 *km* [22]. A landing system based on a LiDAR sensor can perform accurate altimetry, map-relative localisation, odometry, and map refinement in an illumination-insensitive manner, over unknown or partially known terrains. Indeed, more demanding planetary exploration requirements implies a technology development program that calls for more precise guidance systems capable of delivering rovers and/or landers with higher and higher degree of precision. A deep Recurrent Neural Network (RNN) architecture has been designed, tested and validated for predicting the fuel-optimal thrust from sequence of states during a powered planetary descent [60]. To our knowledge, this work presents the first research demonstrating Deep Learning LiDAR-based pose estimation approach applied to planetary landing operation. Similar approach has been explored by Ghilardi et. al [61], where authors present an image-based powered descent guidance via hybrid architecture to control the command acceleration along the three axes during the descent phase. Their research highlights one of the most significant obstacles in supervised learning: the need to create a training data set that adequately and efficiently covers the state space to be learnt. We introduce a hybrid CNN-RNN (LSTM) architecture for pose estimation presenting the following advantages:

- Effective feature representation due to the adoption of the CNN module. The latter allows for features to be generalised and used on untrained terrains;
- The RNN (LSTM) module provides a robust and efficient modelling of the navigation kinematics;
- The hybrid network is flexible to train and test on different data modalities from simulation and real LiDAR data. This capability allows the potential extension of the network to deliver accurate pose estimation on further data modalities;
- Combine both of CNN and LSTM networks after the training stage and perform the estimation in one step.

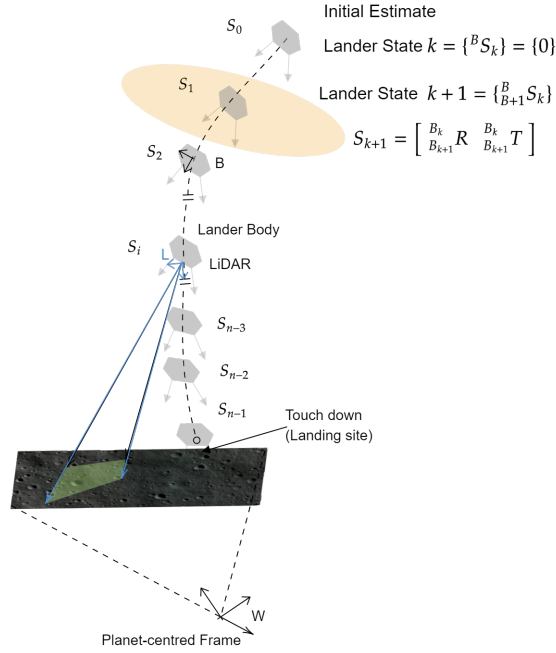
The paper is organized as follows: Section 2 is devoted to introduce the proposed hybrid CNN-RNN(LSTM) pose estimation architecture. Section 3 details the different techniques and simulation tools used for landing scenarios data collection, and presents some preliminary results. Section 4 describes the experiment that has been conducted to obtain realistic and real LiDAR data from real equivalent landing scenarios, evaluation results of the proposed deep pose estimation architecture are presented and discussed. Section 5 outlines our conclusions and future work directions.

## 2. CNN-LSTM pose estimation approach

### 2.1. LiDAR-based pose estimation

The Lidar based pose estimation we propose is set to determine a rigid body transformation from two consecutive LiDAR scene data scans,  $S_k$  and  $S_{k+1}$  collected from a 3D LiDAR sensor mounted on a space lander. Figure 2 illustrates the navigation states and frames adopted in this work. The planet-centered-inertial reference frame is referred to as  $W$ , for the “world” frame, which is non-rotating. From the lander’s CAD design, frame  $B$  is chosen as a reference frame. Body frame  $B$  is attached to the LiDAR frame  $L$ , which represents the sensing frame of the LiDAR.

Let us consider  $R$  to be a rotation matrix [62], which adheres to the  $SO(3)$  requirements throughout the training and testing stages. This representation of translation and rotational motion of a rigid-body is suited for the regression we perform over our network’s last layer. In terms of the restrictions applied on the rotation matrix elements by the  $SO(3)$ , the training strategy takes these constraints into account by learning them, at the training stage, from the known,



**Figure 2:** Coordinate System Frames and Navigation States

free-constraints, ground truth  $R$ . This results in generating free of  $SO(3)$  constraints rotation matrices, at the testing stage.

$$S_{k+1} = RS_k + T \quad (1)$$

$$\mathcal{R}^* = \begin{bmatrix} R & | & T \\ 0 & | & 1 \end{bmatrix}, \quad R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, \quad T = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} \quad (2)$$

Where  $R$  denotes the rotation matrix, and  $T$  represents the translation component. These components remap  $S_k$  to  $S_{k+1}$ .

## 2.2. CNN-RNN(LSTM) approach

Hereinafter, we dive into the proposed approach and explore its essential parts and modules including the network configuration and its selected parameters.

### 2.2.1. Transforming frames to feature vectors

CNN is used to extract features by putting two consecutive frames into the network in order to determine the activation and thus transforming the input images into sequences of feature vectors. The feature vectors represent the outputs of the activation function at the fully connected layer of the CNN.

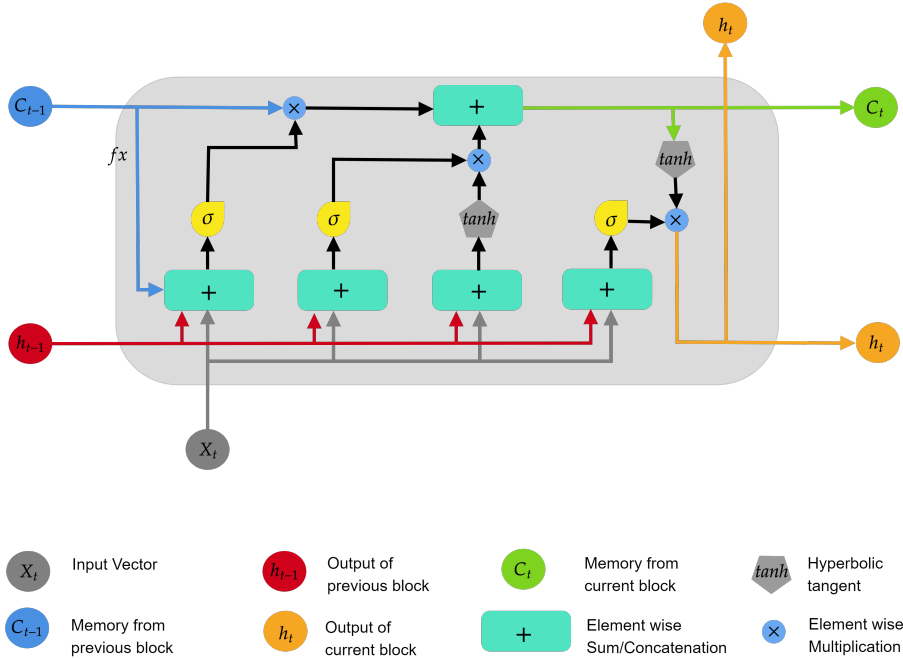
The CNN's parameters are given in Table 1.

### 2.2.2. Estimate pose from feature vector

The RNN (specifically LSTM) module intends to automatically model the spacecraft landing navigation and the relationships between the  $I_{k,k+1}$  features that were extracted by the CNN network. The LSTM network is adaptable when it comes to learning the model of motion. LSTM layers implemented in the proposed architecture, are capable

Layer N	Layer Type	Parameters
1	Input Conv <sub>1</sub>	depth image
2	Conv <sub>1</sub>	Filter size 5 × 5, padding 3, stride 2, channels 64
3	ReLU <sub>1</sub>	-
4	Conv <sub>2</sub>	Filter size 5 × 5, padding 2, stride 2, channels 128
5	ReLU <sub>2</sub>	-
6	Conv <sub>3</sub>	Filter size 5 × 5, padding 2, stride 2, channels 256
7	ReLU <sub>3</sub>	-
8	Fully Connected <sub>1</sub>	1024 × 1 matrix
9	Regression	1024 × 1 matrix

**Table 1**  
Configuration of the CNN Network



**Figure 3:** Block diagram of a LSTM recurrent memory unit

of learning long-term relationships between two sequential depth image projections  $I_k$  and  $I_{k+1}$ . This is beneficial to the problem of spacecraft pose estimation by learning how the current frame is being linked to the motion estimates of previous frames.

LSTM's ability to learn long-term dependencies is owed to its gated design that determines which sectors of the previous hidden state should be kept or discarded in the current iteration. This is achieved not only in combination with the current input, processed by four different units, but also by a cell state which acts as an "information motorway" that bypasses the cells. The LSTM structure is illustrated in Figure 3. In order to get the pose estimation vectors, we employ an RNN network composed of few LSTM layers i.e taking as input the feature vectors extracted from the CNN network. The LSTM network's parameters are given in Table 2.

### 2.2.3. Building the CNN-RNN(LSTM) architecture

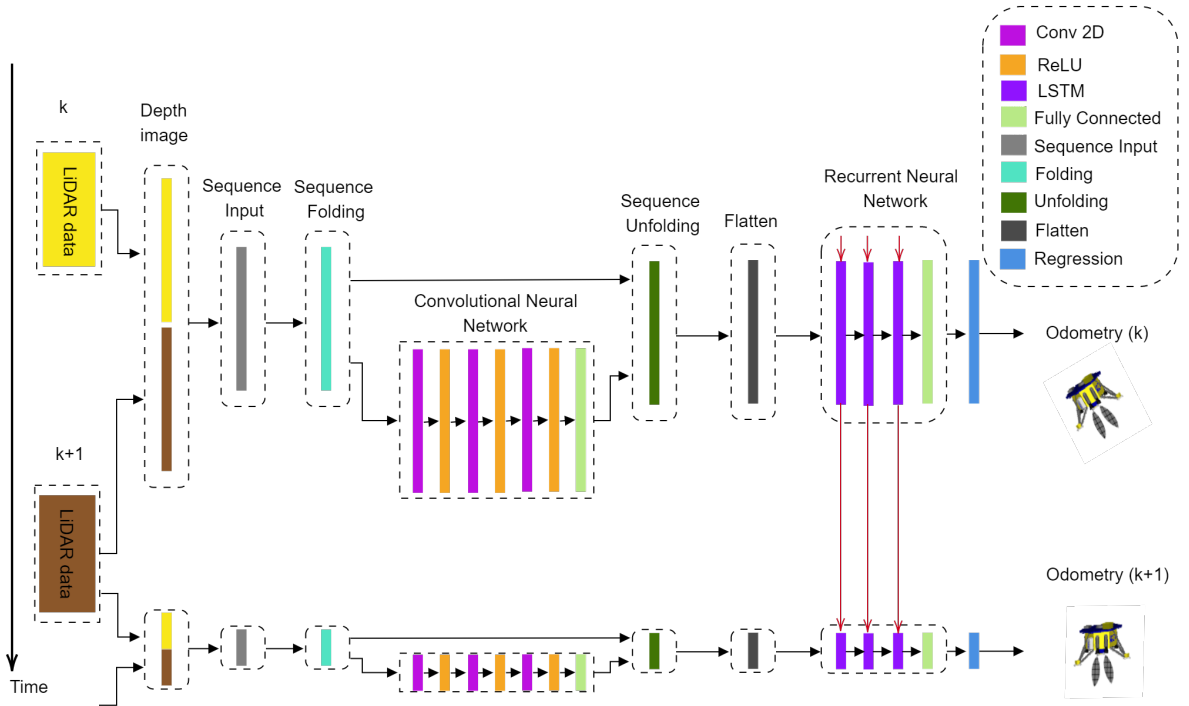
The CNN and RNN(LSTM) networks are trained and tested separately. Thus, to perform estimation, we need first to extract the features between two frames using the trained CNN network. Then, feeding the extracted feature vector to the trained RNN network (LTSTMs) as a sequence of inputs.



Layer N	Layer Type	Variables
1	Sequence input layer	$1024 \times 1$ matrix
2	LSTM <sub>1</sub>	hidden values 1000
3	LSTM <sub>2</sub>	hidden values 1000
4	LSTM <sub>3</sub>	hidden values 1000
5	Fully Connected <sub>2</sub>	$1024 \times 1$ matrix
6	Regression	$1024 \times 1$ matrix

**Table 2**  
Configuration of the LSTM Network

Following, we build the network that contains the trained layers from the CNN network in addition to the LSTM layers of the RNN network. Figure 4 shows the proposed architecture. The CNN module extracts features from  $I_{k,k+1}$ , and because both components of  $I_{k,k+1}$  are composed of 2D depth images, the corresponding features are geometric in nature. This is essential since it allows for the features to be generalised, hence boosting the overall robustness.



**Figure 4:** Hybrid CNN-LSTM architecture.

Initially, the 2D depth image  $I_{k,k+1}$  is given to the sequence input to layer. The size of  $I_{k,k+1}$  is empirically defined during training, and it is the greatest possible to maintain the details of point cloud frames.

The next layer is the sequence folding layer which is used to convert a batch of image sequences to a batch of images. A sequence folding layer allows performing convolution operations on time steps of image sequences independently. The layers after that are convolutional layers, each one, accompanied by a Rectified Linear Unit (ReLU) activation layer. To connect the CNN and RNN (LSTMs) modules, the CNN module has a Fully Connected layer at the output. Whereas the sequence unfolding layer restores the sequence structure of the input data after sequence folding, the flatten layer fits the spatial dimensions of the input into the channel dimension.

The RNN module is comprised of three LSTM layers, which work to improve the RNN's ability to learn a high-level representation and complex navigation modelling, respectively. Following the three LSTM layers, a Fully Connected

Layer N	Layer Type	Variables
1	Sequence Input	Depth image
2	Sequence Folding	
3	Conv <sub>1</sub>	Filter size 5 × 5, padding 3, stride 2, channels 64
4	ReLU <sub>1</sub>	-
5	Conv <sub>2</sub>	Filter size 5 × 5, padding 2, stride 2, channels 128
6	ReLU <sub>2</sub>	-
7	Conv <sub>3</sub>	Filter size 5 × 5, padding 2, stride 2, channels 256
8	ReLU <sub>3</sub>	-
9	Fully Connected <sub>1</sub>	1024 × 1 matrix
10	Sequence Unfolding	
11	Flatten layer	
12	LSTM <sub>1</sub>	hidden values 1000
13	LSTM <sub>2</sub>	hidden values 1000
14	LSTM <sub>3</sub>	hidden values 1000
15	Fully Connected <sub>2</sub>	1024 × 1 matrix
16	Regression	1024 × 1 matrix use 12 first variables to predict the pose

**Table 3**  
Assembled Hybrid CNN-LSTM Network configuration

layer is added. The final layer of the RNN (LSTM) module is a Regression layer, the output of which is transformed into  $\mathcal{R}^*$  by extracting its resulting elements 1 – 12, as following:

$$regression_{layer} = [r_{11} r_{21} r_{31} r_{12} r_{22} r_{32} r_{13} r_{23} r_{33} t_1 t_2 t_3] \quad (3)$$

Elements 13 – 1024 are omitted. Table 3 shows the CNN-RNN(LSTM) network configuration with the selected parameters for each layer.

### 3. Landing scenario simulation

It is not trivial for a spacecraft to autonomously determine its pose relative to the planetary bodies terrains. The following sub-sections describe the methods and software used for this work to simulate the landing scenario and render the sensor data from the LiDAR.

#### 3.1. PANGU simulation

PANGU (Planet and Asteroid Natural Scene Simulation Utility) [63] software is utilised to simulate the landing scenario and generate data producing a sequence of LIDAR frames. PANGU enables users to create models of both extraterrestrial objects or surfaces and generate images/point clouds of them similar to those acquired by real on-board imaging/LIDAR sensors. This helps to test and verify the approaches of landing and autonomous spacecraft missions.

The descent and landing scenario is provided in Figure 5a. It highlights the scenario phases and the landing trajectory leading to safe landing. [64]. Our landing approach is based on LiDAR sensor and the landing path would be a projectile trajectory starting at 1200 (m) in altitude, as shown in Figure 5b. Data collection is performed using the LiDAR interface function provided by PANGU. The function provides several outputs and we choose to model the following ones:

- Range: is the distance from the LiDAR emitter to the target point;
- Slope angle: is the angle formed by the surface normal and the LiDAR pulse direction;
- Elevation angle: is identical to the slope except that the elevation angle is of the LiDAR beam inside the emitter;

Figure 5b illustrates the data generated through the synthetic LiDAR sensor. At each specified position and orientation, we trigger the LiDAR and store those sensor outputs for each frame.

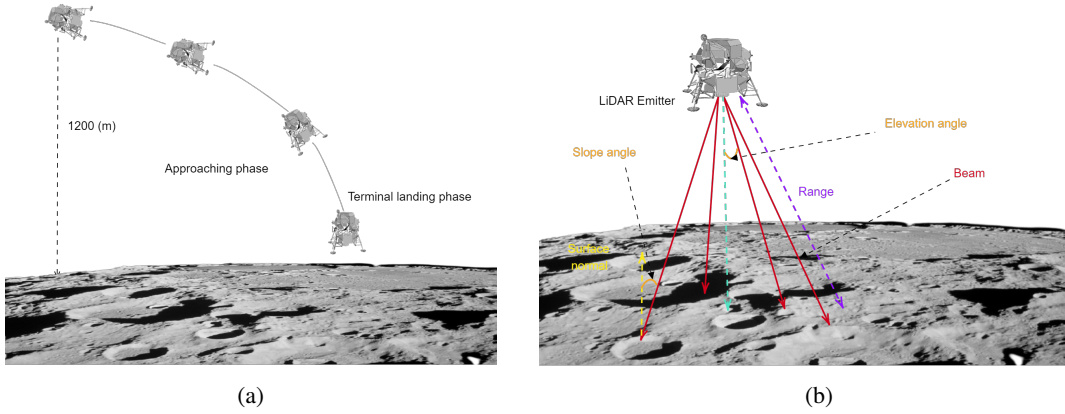


Figure 5: (a) Landing scenario, (b) LiDAR function outputs

*Data Processing:* The data generated from the LiDAR function is encoded as images as presented in Figure 6. The images are rainbow colour-coded with red representing small ranges/slopes/elevation and blue/violet representing large ranges/slopes/elevation. Following different predefined landing trajectories, the collected images from each frame are concatenated and resized with a conversion operation to grayscale to produce an image that includes the three data.

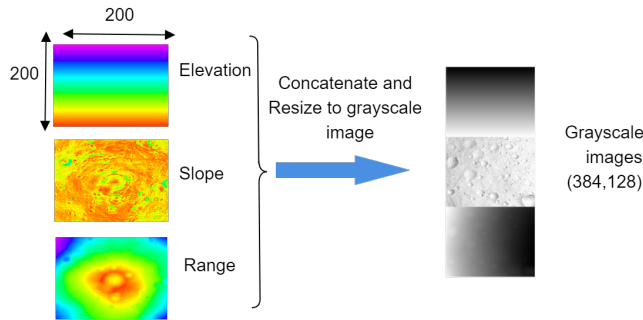


Figure 6: LiDAR data processing

### 3.2. Tests on PANGU LiDAR data

In this trials, we consider several landing trajectories on the Lunar Apollo 16 surface. We use those trajectories during the CNN and RNN (LSTMs) network’s training stages. The  $768 \times 64$  pixel depth image  $I_{k,k+1}$  is the input.  $I_{k,k+1}$  is chosen to be as large as possible during training in order to maintain image information while yet providing a satisfying pose estimation. Several examples of landing scenarios along with their simulated ground truth trajectories are presented in Figure 7(a-b). The ground truth pose of the spacecraft is used to perform a comparison with the estimated pose delivered by the proposed deep network architecture.

Figure 8 shows the lander position estimation, starting from 1.2 km altitude, compared to the ground truth. We observe that the hybrid CNN-RNN (LSTMs) estimation achieves a high accuracy estimation of the relative position. Figure 9 displays the attitude estimation along with the 3D position estimation curve, both compared to the ground truth. The orientation is noticed as less accurate than the result obtained for the the position. This is due to the training data collected from PANGU does not have enough variation in terms of the spacecraft attitude to achieve a better training and feature detection from the CNN module to improve the attitude estimation. The performance metrics of the deep learning algorithm estimation on PANGU data are given in Table 4. It is noticed that the network is providing a solution at a promising computational time.

## Robust Deep learning LiDAR-based pose estimation

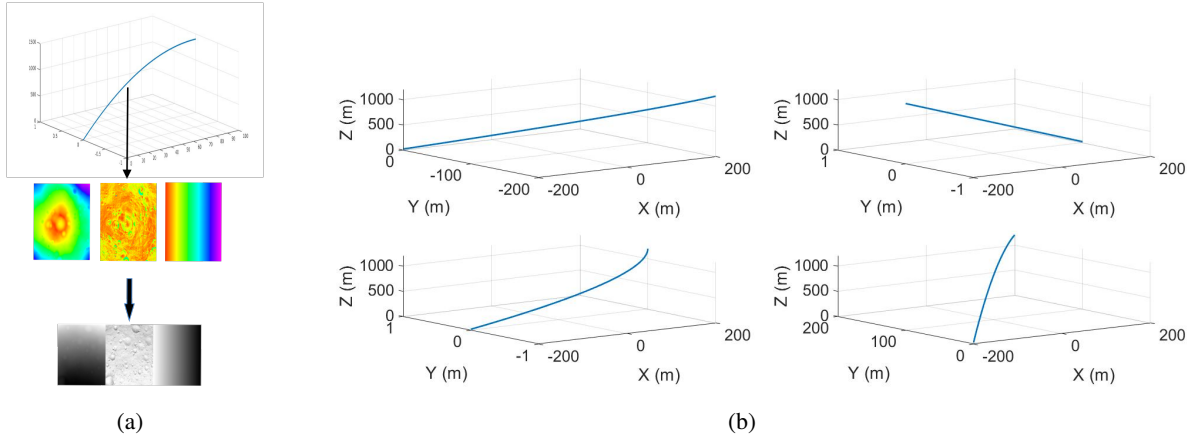


Figure 7: PANGU Landing scenario samples

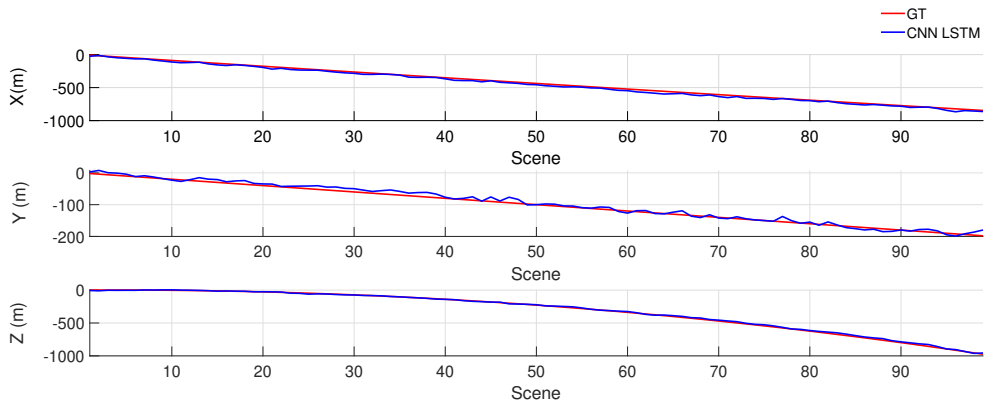


Figure 8: Spacecraft position estimation

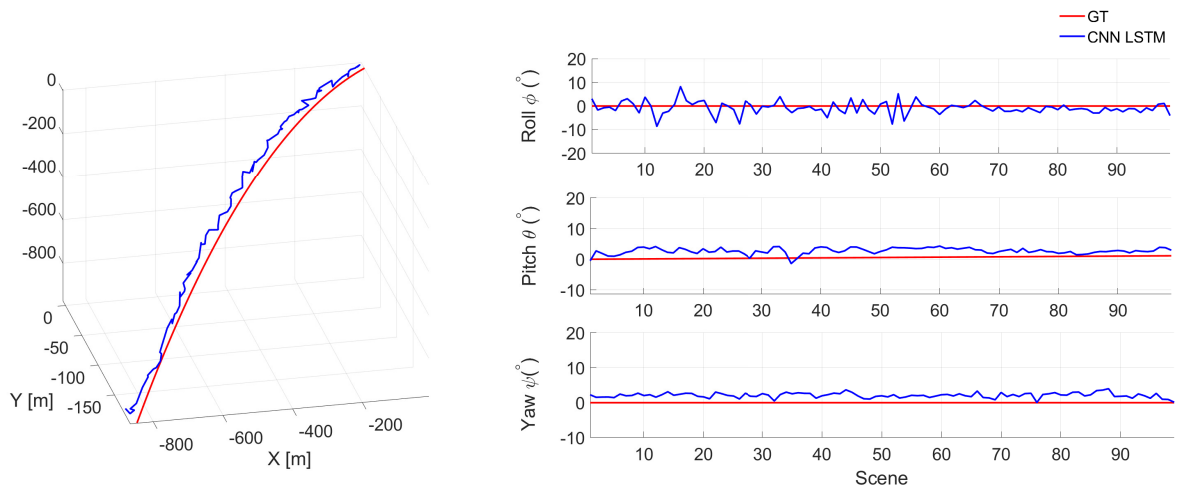


Figure 9: Orientation estimation and 3D landing trajectory prediction

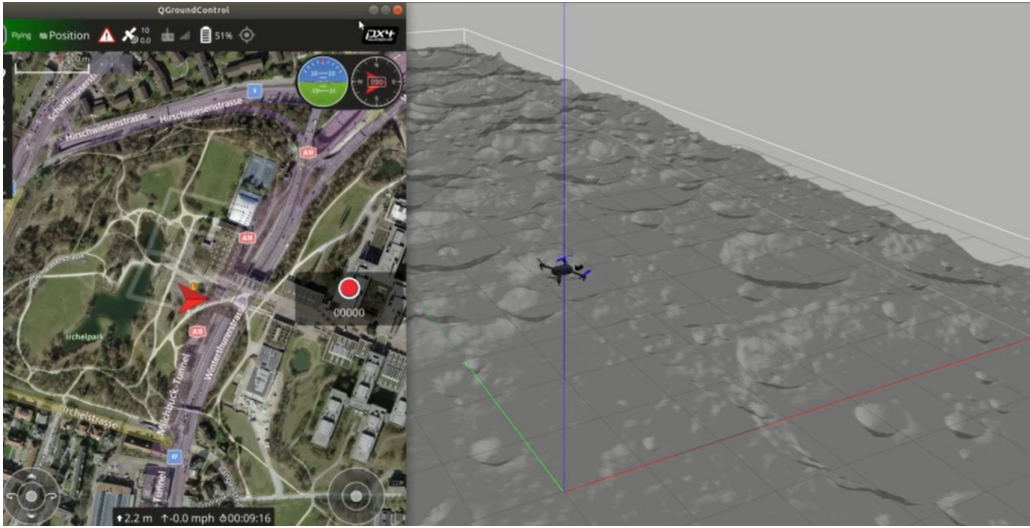
	$T_{error}(\%)$	$e^T$	t(s)
Testing Trajectory	1.53 (m)	44.461	0.0104

**Table 4**

Performance metrics of the tested data from PANGU simulation

### 3.3. Gazebo Simulation

Gazebo software is used to create an environment that contains a representative lunar terrain surface model as a ground plan of a world inside Gazebo. Thereafter, we insert a drone robot equipped with a 3D LiDAR sensor to collect the data while the drone is performing landing scenarios on the lunar terrain. Figure 10 shows the simulation environment in addition to the QGround-control station software provided for PX4 SIL simulation [65], in order to control the drone to do its planned trajectories.

**Figure 10:** Gazebo simulator environment

The data collected from the LiDAR sensor in this simulation is different from the one of PANGU, as the sensor used in Gazebo provides directly 3D point cloud data of the scene. Contrary to PANGU, this data output is different but more realistic as in real space-qualified LiDAR we expect to generate raw 3D point clouds. This enable us to test the proposed solution on a different sensor data modality as the one we get from PANGU. Figure 11 depicts few data samples taken at different landing altitudes.

#### 3.3.1. Data Processing

In the following, we present how we process the point cloud data collected from the Gazebo simulation. Deep learning can be adopted for 3D point clouds with few challenges. The most significant ones are as follows:

- Irregularity: The data is irregular since Object/scene points are not evenly distributed across different parts of a point cloud;
- Unstructured: Point cloud data is not griddable as opposed to pixels in images;
- Unorderdness: A scene's point cloud is a collection of 3D points that are typically kept as a list in a file. The sequence in which the points are kept as a set does not reflect the rendered scene;

These features of point clouds make deep learning, particularly convolutional neural networks (CNN) difficult to use. This is because convolutional neural networks are built on convolution operations, which are applied to ordered, regular, and structured inputs. Nevertheless, deep learning can be adopted straight to a point cloud by converting point cloud data to a structured one. These techniques can be classified as voxel-based or multiview-based. Although voxel-based

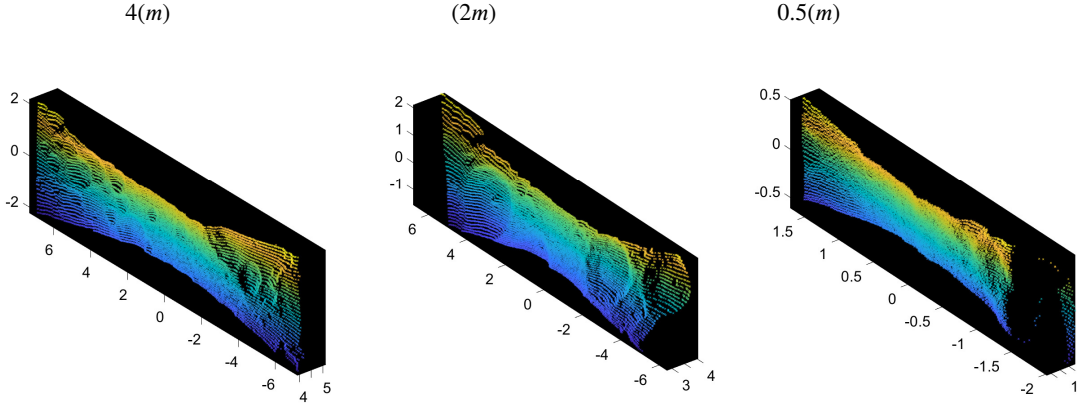


Figure 11: Samples of point cloud data from different altitude

approaches have demonstrated excellent performance, they suffer from significant memory consumption due to the sparsity of the voxels. Multiview-based networks outperform voxel-based networks because they use well-known 2D techniques, and they can hold more information due to voxelization quantization errors.

In this work, a multiview-based approach in combination with a data projection strategy onto an image plane is adopted. The proposed Deep Learning LiDAR-based architecture is then applied on the created multiple 2D depth maps to tackle the problem of relative space navigation pose estimation.

Given two consecutive point cloud frames  $\mathbf{S}_k = \{\mathbf{S}_k^1, \dots, \mathbf{S}_k^\alpha\}$  and  $\mathbf{S}_{k+1} = \{\mathbf{S}_{k+1}^1, \dots, \mathbf{S}_{k+1}^\alpha\}$ , the superscript 1 means the first point in the space, the last point superscript is  $\alpha$ . It is related to the density of the sensor and how many points in space we can get in one single frame. Our proposed solution takes advantage of both data modalities (3D and 2D) by remapping  $\mathbf{S}_k$  and  $\mathbf{S}_{k+1}$  from the 3D to the 2D domain by creating three 2D depth images. To create the three 2D depth images, first, a floating-point vertex coordinate is quantized  $\mathbf{S}_k = \{S_k^1, \dots, S_k^\alpha\}$  into  $\mathbf{S}_{Q-k} = \{S_{Q-k}^1, \dots, S_{Q-k}^\alpha\}$  with:

$$\mathbf{S}_{Q-k}(x_{Q-k}, y_{Q-k}, z_{Q-k}) = [q_f \cdot S_k(x, y, z)] \quad (4)$$

Selecting  $q_f$  is not trivial because it has a significant impact on the size of the 2D remapped images in each projection, and the amount of detail contained in these images. For our work, we set  $q_f = 20$  to preserve the point cloud topology. Figure 12 shows different quantized images using different  $q_f$  factors applied on point cloud data along YZ-plane.

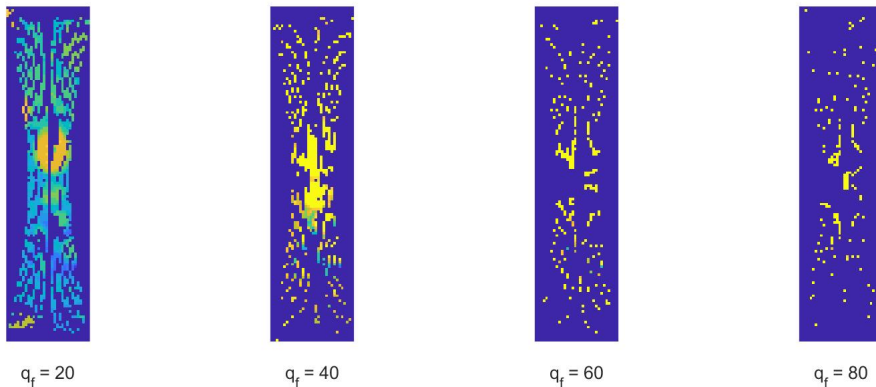


Figure 12: Effects of  $q_f$  factor on the data

Next, we use the orthographic projection process  $S_{ortho}$ , and project  $S_{Q-k}$  on each plane of the  $XYZ_{LIDAR}$  reference frame:

$$\tilde{S}_{Q-k} = \begin{bmatrix} \tilde{x}_q \\ \tilde{y}_q \\ \tilde{z}_q \\ 1 \end{bmatrix} = S_{ortho} \cdot S_{Q-k} = \begin{bmatrix} c1 & 0 & 0 & 0 \\ 0 & c2 & 0 & 0 \\ 0 & 0 & c3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{Q-k} \\ y_{Q-k} \\ z_{Q-k} \\ 1 \end{bmatrix} \quad (5)$$

Based on the three depth images of  $S_{Q-k}$ , i.e.  $\tilde{S}_{Q-k}^{XY}$ ,  $\tilde{S}_{Q-k}^{XZ}$ ,  $\tilde{S}_{Q-k}^{YZ}$ , we create a stacked image.

$$I_k = \tilde{S}_{Q-k}^{XY} || \tilde{S}_{Q-k}^{XZ} || \tilde{S}_{Q-k}^{YZ} \quad (6)$$

$||(\cdot)$  denotes the process of vertical concatenation. Subsequently, we construct  $I_{k,k+1} = I_k || I_{k+1}$  as an input to feed the proposed CNN-RNN(LSTM) network. Figure 13 exhibits the quantization, projection and concatenation processes to remap the 3D point cloud data into 2D depth images.

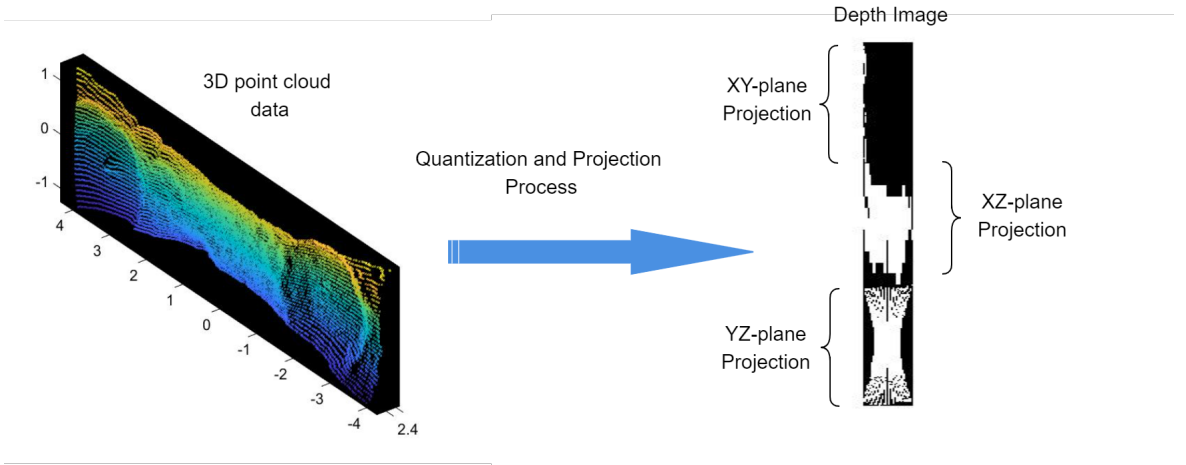


Figure 13: Projection and concatenation of a 3D point cloud data

### 3.4. Tests on Gazebo simulation LiDAR data

This evaluation is based on the data generated from Gazebo simulation, which has a different data modality (3D point clouds). Samples of simulation landing scenario conducted on Gazebo are shown in Figure 14b. Figure 14a summarizes the data processing of the data collected from Gazebo. We collect 3D point cloud frames while the drone is performing a landing operation, then the 3D data is projected into 2D depth images as previously explained. The  $768 \times 32$  pixel depth image  $I_{k,k+1}$  is the input.  $I_{k,k+1}$  is chosen to preserve point cloud information while reducing processing burden and keep providing an appealing pose estimation.

Based on the training dataset, the parameters of the proposed architecture are tuned for optimal pose estimation. Our trials are evaluated in terms of  $T_{error}$  which is the metric that represents the overall translation error as a percentage of the GT distance travelled.  $e^T$  denotes the root mean square error of the actual and estimated positions along the landing trajectory.  $t$  is the processing time necessary for each frame. The desktop used to run our algorithm solution is equipped with an Intel i7, an NVIDIA Quadro P2000 GPU, 16 GB of RAM, and is running Linux (Ubuntu 18.04) and MATLAB 2020b.

Figure 15 depicts the curves of the errors between the pose estimation  $\mathcal{R}^*$  and the ground truth. The performances are compared to an estimation obtained by using only a CNN network. It can be observed that the hybrid CNN-RNN(LSTM) network achieves better accuracy because of the integrated RNN(LSTM) module. It is noted from the performances in Table 5 that the network CNN-RNN(LSTM) offers a promising accuracy.

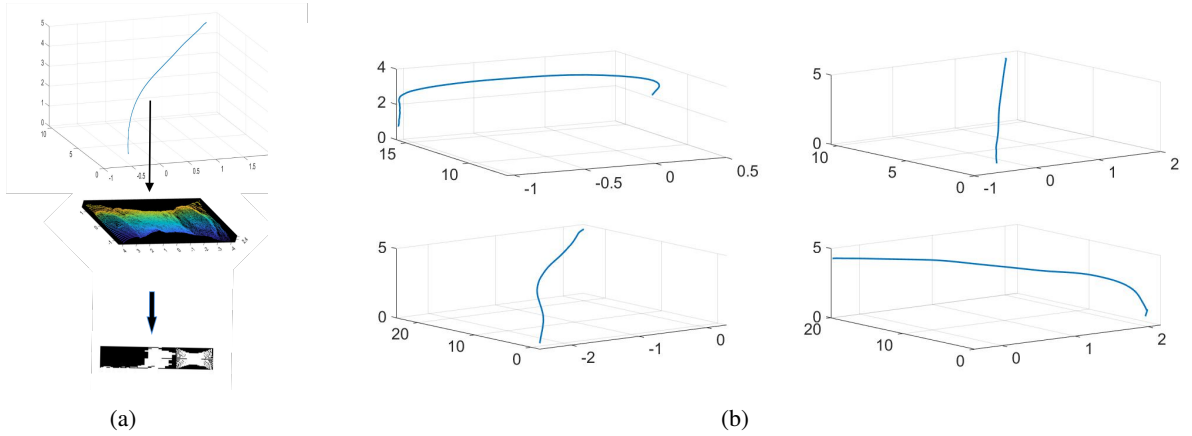


Figure 14: Gazebo Landing scenario samples

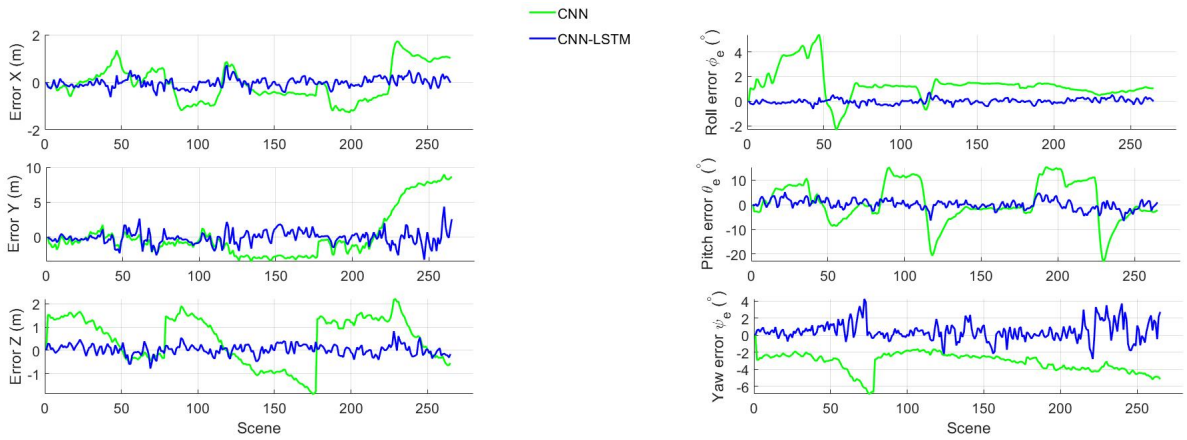


Figure 15: Pose estimation of landing scenario in Gazebo

	$T_{error}(\%)$	$e^T$	t(s)
CNN	6.4357	0.34984	0.0150
CNN-RNN(LSTM)	2.9836	0.14794	0.0220

Table 5

Performance metrics of the tested trajectory from Gazebo simulation

## 4. Experiment and Validation

The following details the performed experiment to obtain real data from a Velodyne VLP-16 Puck Lite LiDAR sensor including the installation of the LiDAR sensor on a flying UAV with respect to a scaled lunar terrain surface. The suggested CNN-RNN(LSTM) is trained and evaluated using several landing scenario trajectories. We compare the accuracy of the estimated pose to the flying robot's ground truth, which is obtained using an OptiTrack system that tracks the drone inside our arena's workspace.

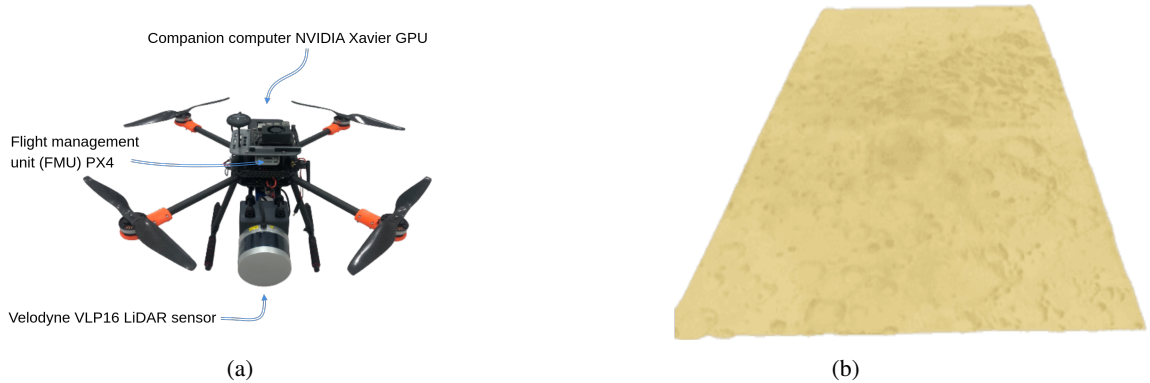
### 4.1. Main components

#### 4.1.1. Platform test bench

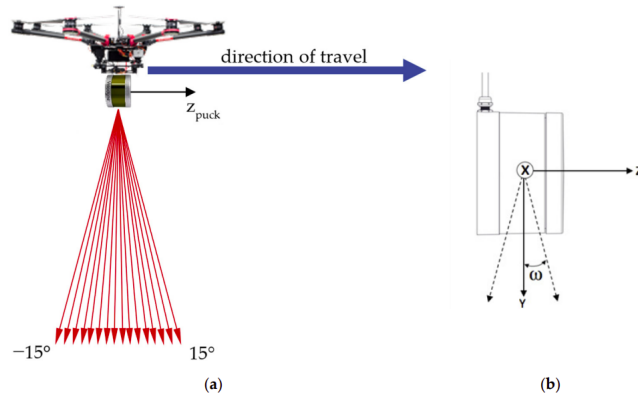
Figure 16a shows the platform adopted for the experiment. It includes the main core of the drone, a companion computer, and a Velodyne VLP16 LiDAR sensor. Initially, we investigate mounting the LiDAR in different orientations



in order to find the appropriate mounting position and orientation for better data reading and collection of the area of interest. The best LiDAR mounting configuration used for this work [66] is shown in Figure 17.



**Figure 16:** (a) Test bench hardware, (b) Lunar terrain DEM model



**Figure 17:** (a) Typical orientation of the VLP-16 scanner aboard a drone. (b) Scanner-oriented coordinate system of the VLP-16 [66]

The Velodyne is connected to the computer companion NVIDIA Xavier GPU Board, which is the brain of our platform. The computer is responsible of several tasks, mainly, receiving the tracking data from an exterior system (Optitrack system), reading the 3D point cloud data acquired by the Velodyne, and producing the estimated pose by our proposed algorithm.

#### 4.1.2. Lunar terrain DEM model

An accurate digital elevation model (DEM) improves our knowledge of the lunar surface, aids in the identification and characterization of the topography, and helps us to understand the impact history by evaluating the relative ages of the geological units [67, 68]. In addition, illumination and communication maps can be simulated based on a DEM for optimizing the potential lunar landing sites [69, 70]. Laser altimeters are commonly used to map the lunar topography. The Apollo 15 to Apollo 17 missions provided preliminary laser altimeter data for creating maps of the lunar topography [71]. In our experiments, the DEM model of the lunar terrain is provided by ESA and it is printed using a dedicated and special CNC machine. The picture in Figure 16b, shows the produced model used in our work. The model dimension are as follow:  $300 \times 150 \times 10$  (cm).

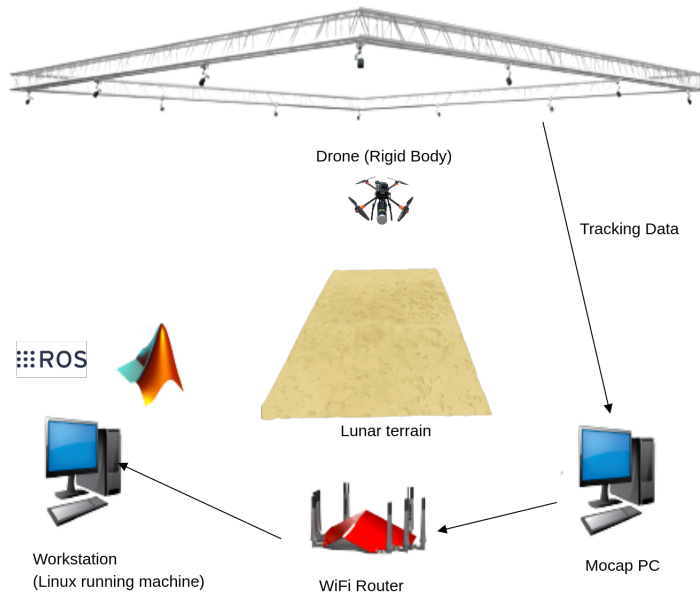
#### 4.1.3. Motion Capture (MoCap) system

MoCap involves measuring the position as well as the orientation of objects or people in a physical space [72]. A single OptiTrack motion capture system tracks the 6 degrees of freedom (DOF) pose of one or more objects in

the workspace. Tracked objects need to be defined as rigid bodies. Each rigid body is a cluster of reflective markers arranged in a unique configuration when appropriately mounted on the body. Our OptiTrack real-time tracking system is worldwide used for Robotics applications due to its low latency and precise 6DoF indoor tracking poses of ground and aerial robotics (UAVs).

## 4.2. Experiment

The validation and testing task concerns the utilisation of the hardware prepared for data reading and data collection, to train and test the developed and implemented Deep Learning pose estimation solution. Testing and evaluation are performed offline. Figure 18 shows the conducted experiment including all the involved components. Basically, the lunar terrain model is placed in a workspace monitored by the OptiTrack system. The OptiTrack cameras are connected via an Ethernet switch to the desktop PC. The motive software, installed on this PC, receives the tracking data of the drone inside the workspace and streams, over a WIFI network, the data to use in another workstation machine. This latter monitors the drone status via the QGroundcontrol software supported by the PX4. The workstation uses the ROS layer, to communicate with the drone and remotely accesses the drone's on-board computer in order to execute the control algorithm that run the waypoints navigation and move the drone according to the planned landing trajectory. The workstation machine is also used for data recording, visualisation, and monitoring purposes.

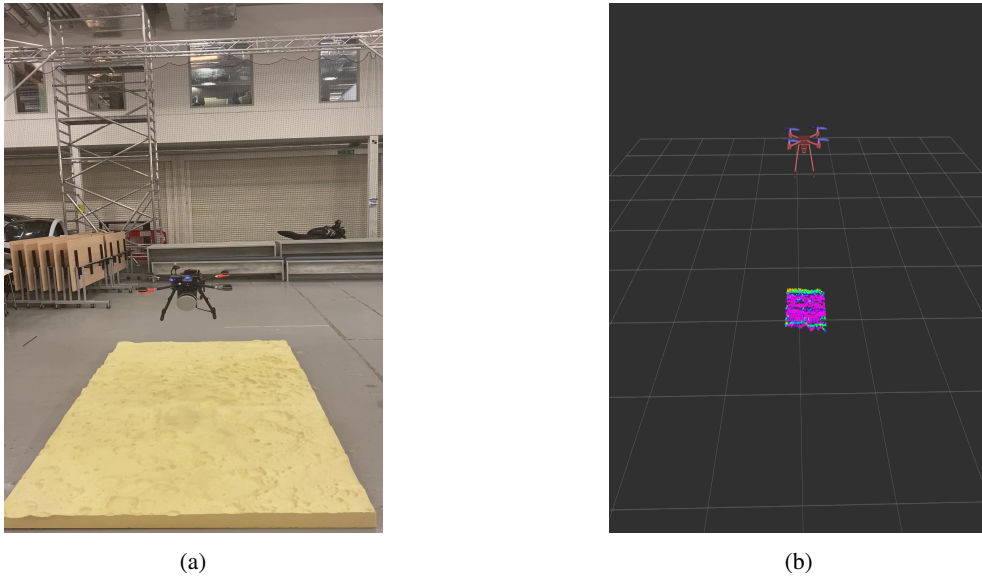


**Figure 18:** Experiment setup

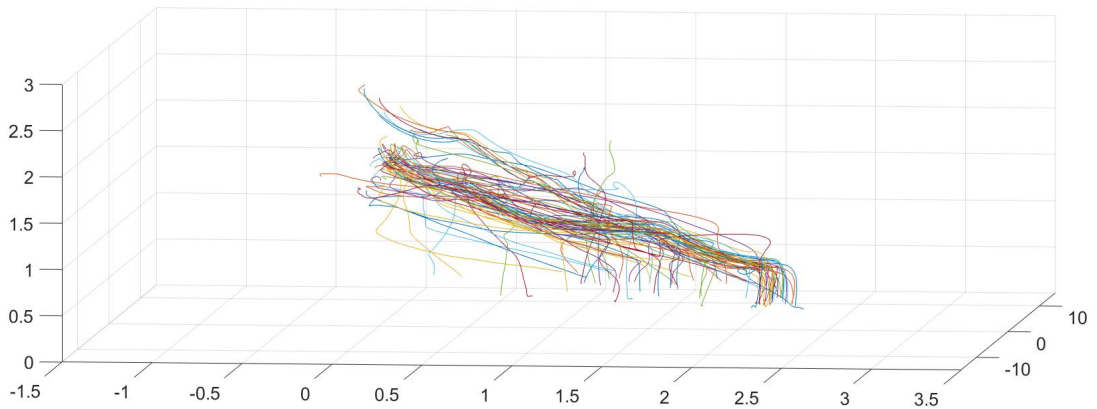
In order to simulate the landing scenario and collect realistic LiDAR data, a predefined trajectory is implemented on-board the drone's companion computer and executed from the workstation machine. The data recording process starts after the drone takes off and reaches the maximum altitude defined in the trajectory. Thus, the point cloud data is recorded only during the landing operation. Figure 19a shows the platform during the landing operation, while Figure 19b is a visualization tool to follow the data acquisition from the available sensors.

## 4.3. Tests on real LiDAR data

The objective of this work is to study the feasibility of developing for the first time an advanced deep learning strategy based on LiDAR data to space landing operation. Thus, our experimental evaluation is mainly focused on assessing the methodologies that we develop in the case of real point cloud data. We also compare the performances of the proposed hybrid CNN-LSTM with respect to a CNN network. The data set created for the training is 120 trajectories (including simulation trajectories and real experimental trajectories). In the experimental case the landing site is distributed along the Moon mock-up as shown in Figure 20. The initial conditions are different for all the trajectories and simulated scenarios.



**Figure 19:** (a)Drone during landing, (b)Point cloud data acquisition



**Figure 20:** Experiment landing trajectories

	$T_{error}(\%)$	$e^T$	t(s)
CNN	7.2536	0.69631	0.0155
CNN-RNN(LSTM)	2.8142	0.15123	0.0210

**Table 6**

Performance metrics of the tested trajectory from real experiment

Figure 21b shows separately few samples of trajectories performed to collect the data from our real drone experimental setup. The objective is to collect point cloud data of the representative lunar terrain and test our deep learning estimator solution on those real data. Figure 22 presents the pose estimation delivered by the CNN-RNN(LSTM) applied to real point cloud data. Figure 23 presents the 3D pose estimation of one of the test trajectories. Performance results for the real landing trajectories tests are given in Table 6. The mean absolute error and the standard deviation are shown in the Table 7.

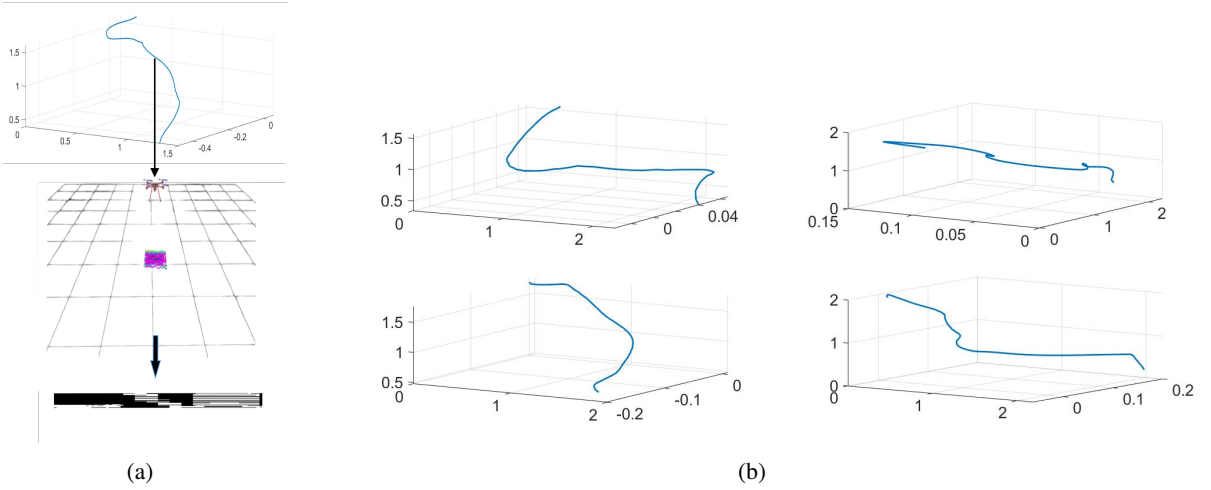


Figure 21: Samples from experiment trajectories

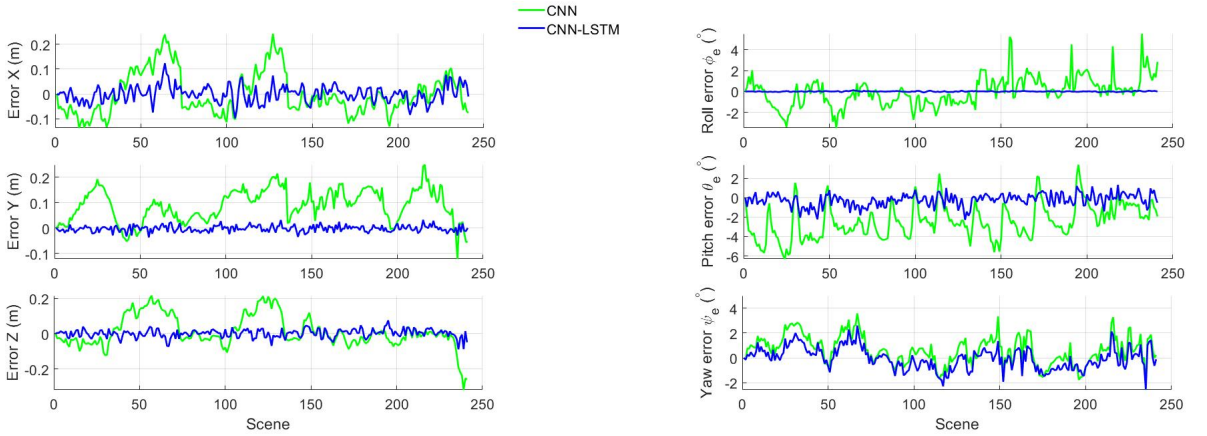


Figure 22: pose estimation on real point cloud data

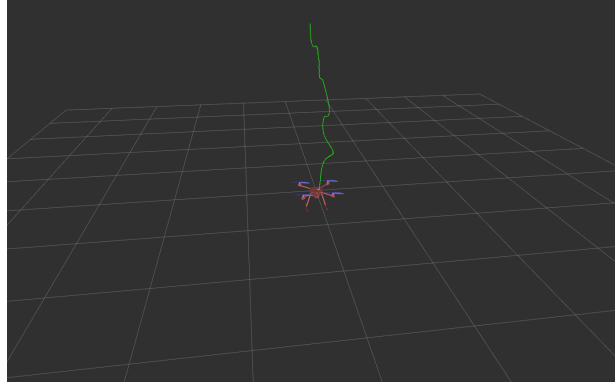
Mean Absolute Error	Traj <sub>1</sub>	Traj <sub>2</sub>	Traj <sub>3</sub>
Roll	0.5253	0.9128	0.5536
Pitch	0.4994	0.6392	1.2069
Yaw	2.2592	0.2932	0.3312
Standard deviation			
Roll	0.0826	0.0536	0.0488
Pitch	0.1624	0.2322	0.6854
Yaw	0.0995	0.1446	0.1531

Table 7

Mean absolute error with the standard deviation of the orientation

#### 4.4. Algorithm's parameters analysis

We propose in this sub-section a parametric variations evaluation in the landing scenario to explore the envelope of the algorithm's performance. Given the main of the presented architecture in Figure 4, we propose to evaluate how varying the parameters  $I_{k,k+1}$  input size, the interpolation scheme and several CNN-RNN(LSTM) bridge layer configurations have an effect on the  $T_{error}$  metric.



**Figure 23:** Estimated landing position

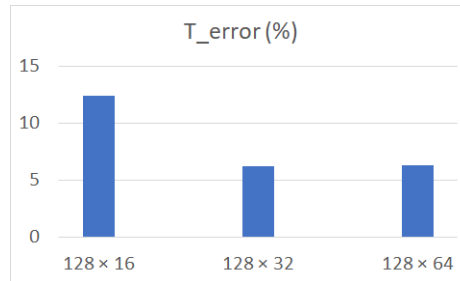
	$T_{error}(\%)$	$t(s)$
nearest neighbor	0.34984	0.0150
bilinear	0.58723	0.0150
bicubic	0.49560	0.0150

**Table 8**

Performance and computation results with different interpolation techniques

#### 4.4.1. Input image sizes

The quantization factor ( $q_f = 20$ ) is playing a key role in finding the optimal height/width ratio, affecting substantially the deep learning pose estimation. It is also critical to keep the  $I_k$  depth variation preserved, as these parameters trigger the responses within each CNN-RNN(LSTM) layers Figure 24 summarizes our tests using different input image sizes and evaluated by  $T_{error}$ . It is observed that the accuracy of pose estimation has improved while increasing the width of the input from 16 to 32. Further increase of the image width to 64 provides almost similar accuracy to 16 but with higher computational time. We conclude then that the optimal input image size, which can be used is  $128 \times 32$  as it achieves good pose estimation accuracy while having the low computational time.



**Figure 24:** Performance for various input image sizes

Another important parameter affecting the details available in  $I_k$  is the method of interpolation used to quantize and resize the depth images to produce the  $I_k$  input image. We evaluate different methods of interpolations: the first one is the nearest neighbor, the second is the bilinear, and the last one is the bicubic. The nearest neighbor is the best selected technique (see Table 8).

#### 4.4.2. Bridge-layer configurations

The bridge layer that connects the CNN and RNN(LSTMs) modules has a significant role in the CNN-RNN(LSTMs) network performance. The minimum bridge-layer size of the fully Connected (FC) layer is  $16 \times 1$  expressed in a form of a vector. Altering that impacts on the RNN(LSTM)'s input and output. Although using an

FC-16 layer with size of a  $16 \times 1$  vector is appealing, it is not the optimal choice. It may be beneficial to increase the FC layer size elements and exploit the first 16 element whereas the FC layer size is at least 1024. This is confirmed by the figure below showing how the error in terms of pose estimation can get reduced in this case.

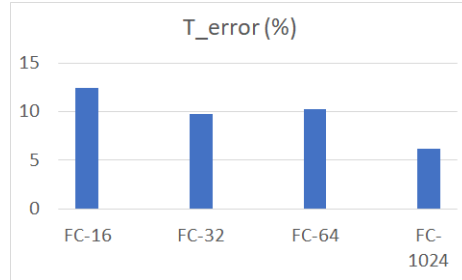


Figure 25: Performance using various configuration of the bridge-layer

## 5. Conclusion

In this work, we present an innovative and effective AI based relative navigation solution for space landing by proposing a Deep Recurrent Convolutional Neural Network (DRCNN) architecture. Both CNNs and RNN(LSTMs) are leveraged in this architecture. In particular, CNNs are adopted for feature extraction and learning and RNN(LSTMs) providing a robust navigation motion modelling. Tests of the solution have shown that the hybrid CNN-RNN(LSTM) architecture delivers high accuracy with a low processing burden. Furthermore, the architecture's adaptability allows it to be trained and tested on a variety of data types while still providing accurate pose estimation.

The next phase of the project will focus on the implementation of the designed solution in real-time. Thus, the right on-board computation hardware must be carefully selected to meet the real-time requirements. The proposed solution will also be trained on different planetary terrains and surfaces, and tested on unseen planetary terrain models.

## Acknowledgment

This work is supported by the European Space Agency (ESA), with reference ESA-TECSAG-SOW-019621

## References

- [1] Marc Steven Krämer, Simon Hardt, and Klaus-Dieter Kuhnert. Image features in space-evaluation of feature algorithms for motion estimation in space scenarios. In *ICPRAM*, pages 300–308, 2018.
- [2] Duarte Ronda, Nabil Aouf, Mark A Richardson, and Vincent Dubanchet. Robust on-manifold optimization for uncooperative space relative navigation with a single camera. *Journal of Guidance, Control, and Dynamics*, 44(6):1157–1182, 2021.
- [3] Linhui Li, Jing Lian, Lie Guo, and Rongben Wang. Visual odometry for planetary exploration rovers in sandy terrains. *International Journal of Advanced Robotic Systems*, 10(5):234, 2013.
- [4] Tommi Tykkälä and Andrew I Comport. A dense structure model for image based stereo slam. In *2011 IEEE International Conference on Robotics and Automation*, pages 1758–1763. IEEE, 2011.
- [5] Yang Cheng, Mark Maimone, and Larry Matthies. Visual odometry on the mars exploration rovers. In *2005 IEEE International Conference on Systems, Man and Cybernetics*, volume 1, pages 903–910. IEEE, 2005.
- [6] Mark Maimone, Yang Cheng, and Larry Matthies. Two years of visual odometry on the mars exploration rovers. *Journal of Field Robotics*, 24(3):169–186, 2007.
- [7] O Yilmaz, Nabil Aouf, L Majewski, M Sanchez-Gestido, and G Ortega. Using infrared based relative navigation for active debris removal. 2017.
- [8] Roberto Opromolla, Marco Zaccaria Di Fraia, Giancarmine Fasano, Giancarlo Rufino, and Michele Grassi. Laboratory test of pose determination algorithms for uncooperative spacecraft. In *2017 IEEE International Workshop on Metrology for AeroSpace (MetroAeroSpace)*, pages 169–174. IEEE, 2017.
- [9] John O Woods and John A Christian. Lidar-based relative navigation with respect to non-cooperative objects. *Acta Astronautica*, 126:298–311, 2016.
- [10] Renato Volpe, Giovanni B Palmerini, and M Sabatini. Monocular and lidar based determination of shape, relative attitude and position of a non-cooperative unknown satellite. In *Proceedings of the 68th International Astronautical Congress*, volume 1, 2017.

- [11] Joseph Galante, John Van Eepoel, Matt Strube, Nat Gill, Marcelo Gonzalez, Andrew Hyslop, and Bryan Patrick. Pose measurement performance of the argon relative navigation sensor suite in simulated-flight conditions. In *AIAA guidance, navigation, and control conference*, page 4927, 2012.
- [12] Jordan L Sell, Andrew Rhodes, John Woods, and John A Christian. Pose performance of lidar-based navigation for satellite servicing. In *AIAA/AAS Astrodynamics Specialist Conference*, page 4360, 2014.
- [13] Roberto Opromolla, Giancarmine Fasano, Giancarlo Rufino, and Michele Grassi. Spaceborne lidar-based system for pose determination of uncooperative targets. In *2014 IEEE Metrology for Aerospace (MetroAeroSpace)*, pages 265–270. IEEE, 2014.
- [14] Roberto Opromolla, Giancarmine Fasano, Giancarlo Rufino, and Michele Grassi. A model-based 3d template matching technique for pose acquisition of an uncooperative space object. *Sensors*, 15(3):6360–6382, 2015.
- [15] Odysseas Kechagias-Stamatis and Nabil Aouf. Hoo lidar odometry for spacecraft relative navigation. *IET Radar, Sonar & Navigation*, 13(5):771–775, 2019.
- [16] Andrew P Rhodes, John A Christian, and Thomas Evans. A concise guide to feature histograms with applications to lidar-based spacecraft relative navigation. *The Journal of the Astronautical Sciences*, 64(4):414–445, 2017.
- [17] Ann Dietrich and Jay W McMahon. Orbit determination using flash lidar around small bodies. *Journal of Guidance, Control, and Dynamics*, 40(3):650–665, 2017.
- [18] Ann B Dietrich and Jay W McMahon. Robust orbit determination with flash lidar around small bodies. *Journal of Guidance, Control, and Dynamics*, 41(10):2163–2184, 2018.
- [19] Odysseas Kechagias-Stamatis, Nabil Aouf, and Mark A Richardson. High-speed multi-dimensional relative navigation for uncooperative space objects. *Acta Astronautica*, 160:388–400, 2019.
- [20] Bong Wie, Vaios Lappas, and Jesús Gil-Fernández. *Attitude and Orbit Control Systems*, pages 323–369. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.
- [21] Roberto Opromolla, Giancarmine Fasano, Giancarlo Rufino, and Michele Grassi. A review of cooperative and uncooperative spacecraft pose determination techniques for close-proximity operations. *Progress in Aerospace Sciences*, 93:53–72, 2017.
- [22] Timothy P Setterfield, Robert A Hewitt, Po-Ting Chen, Antonio Teran Espinoza, Nikolas Trawny, and Anup Katake. Lidar-inertial based navigation and mapping for precision landing. 2021.
- [23] Tanner Campbell, Roberto Furfaro, Richard Linares, and David Gaylor. A deep learning approach for optical autonomous planetary relative terrain navigation. *Spaceflight Mechanics*, 160:3293–3302, 2017.
- [24] Lena M Downes, Ted J Steiner, and Jonathan P How. Lunar terrain relative navigation using a convolutional neural network for visual crater detection. In *2020 American Control Conference (ACC)*, pages 4448–4453. IEEE, 2020.
- [25] Leena Singh and Sungyung Lim. On lunar on-orbit vision-based navigation: Terrain mapping, feature tracking driven ekf. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, page 6834, 2008.
- [26] Chengchao Bai, Jifeng Guo, Linli Guo, and Junlin Song. Deep multi-layer perception based terrain classification for planetary exploration rovers. *Sensors*, 19(14):3102, 2019.
- [27] Chengchao Bai. *Research on Vision/Vibration Based Terrain Perception for Rovers*. Ph.D. thesis, 2019.
- [28] Chengchao Bai, Jifeng Guo, Linli Guo, and Junlin Song. Deep multi-layer perception based terrain classification for planetary exploration rovers. *Sensors*, 19(14), 2019.
- [29] Roberto Furfaro and Richard Linares. Deep learning for autonomous lunar landing. 08 2018.
- [30] Masahiro Ono, Brandon Rothrock, Kyohei Otsu, Shoya Higa, Yumi Iwashita, Annie Didier, Tanvir Islam, Christopher Laporte, Vivian Sun, Kathryn Stack, et al. Maars: Machine learning-based analytics for automated rover systems. In *2020 IEEE Aerospace Conference*, pages 1–17. IEEE, 2020.
- [31] Neil Abcouwer, Shreyansh Daftry, Tyler del Sesto, Olivier Toupet, Masahiro Ono, Siddarth Venkatraman, Ravi Lanka, Jialin Song, and Yisong Yue. Machine learning based path planning for improved rover navigation. In *2021 IEEE Aerospace Conference (50100)*, pages 1–9. IEEE, 2021.
- [32] Vivek Kothari, Edgar Liberis, and Nicholas D. Lane. The final frontier: Deep learning in space. In *Proceedings of the 21st International Workshop on Mobile Computing Systems and Applications*, HotMobile '20, page 45–49, New York, NY, USA, 2020. Association for Computing Machinery.
- [33] Charles Oestreich, Tae W Lim, and Randy Broussard. On-orbit relative pose initialization via convolutional neural networks. In *AIAA Scitech 2020 Forum*, page 0457, 2020.
- [34] Lorenzo Pasqualetto Cassinis, Robert Fonod, Eberhard Gill, Ingo Ahrns, and Jesus Gil Fernandez. Cnn-based pose estimation system for close-proximity operations around uncooperative spacecraft. In *AIAA Scitech 2020 Forum*, page 1457, 2020.
- [35] Sumant Sharma, Connor Beierle, and Simone D’Amico. Pose estimation for non-cooperative spacecraft rendezvous using convolutional neural networks. In *2018 IEEE Aerospace Conference*, pages 1–12. IEEE, 2018.
- [36] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.
- [37] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.
- [38] Staffan Persson, Per Bodin, Eberhard Gill, Jon Harr, and John Jörgensen. Prisma-an autonomous formation flying mission. 2006.
- [39] Sumant Sharma, Connor Beierle, and Simone D’Amico. Pose estimation for non-cooperative spacecraft rendezvous using convolutional neural networks. In *2018 IEEE Aerospace Conference*, pages 1–12. IEEE, 2018.
- [40] Sumant Sharma, Jacopo Ventura, and Simone D’Amico. Robust model-based monocular pose initialization for noncooperative spacecraft rendezvous. *Journal of Spacecraft and Rockets*, 55(6):1414–1429, 2018.
- [41] Daichi Hirano, Hiroki Kato, and Tatsuhiko Saito. Deep learning based pose estimation in space. In *Proceedings of the International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*, Madrid, Spain, pages 4–6, 2018.

- [42] N. Koenig and A. Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, volume 3, pages 2149–2154 vol.3, 2004.
- [43] Ryohei Arakawa, Yuri Matsushita, Toshiya Hanada, Yasuhiro Yoshimura, and Shuji Nagasaki. Attitude estimation of space objects using imaging observations and deep learning. In *Proc. AMOS*, page 21, 2019.
- [44] Shubham Sonawani, Ryan Alimo, Renaud Detry, Daniel Jeong, Andrew Hess, and Heni Ben Amor. Assistive relative pose estimation for on-orbit assembly using convolutional neural networks. *arXiv preprint arXiv:2001.10673*, 2020.
- [45] Thaweerath Phisannupawong, Patcharin Kamsing, Peerapong Torteeka, Sittiporn Channumsin, Utane Sawangwit, Warunyu Hematulin, Tanatthep Jarawan, Thanaporn Somjit, Soemsak Yooyen, Daniel Delahaye, et al. Vision-based spacecraft pose estimation via a deep convolutional neural network for noncooperative docking operations. *Aerospace*, 7(9):126, 2020.
- [46] Kiruki Cosmas and Asami Kenichi. Utilization of fpga for onboard inference of landmark localization in cnn-based spacecraft pose estimation. *Aerospace*, 7(11):159, 2020.
- [47] Lorenzo Pasqualetto Cassinis, Robert Fonod, and Eberhard Gill. Review of the robustness and applicability of monocular pose estimation systems for relative navigation with an uncooperative spacecraft. *Progress in Aerospace Sciences*, 110:100548, 2019.
- [48] Duarte Rondao, Nabil Aouf, and Mark A Richardson. Chinet: Deep recurrent convolutional learning for multimodal spacecraft pose estimation. *arXiv preprint arXiv:2108.10282*, 2021.
- [49] Jianing Song, Duarte Rondao, and Nabil Aouf. Deep learning-based spacecraft relative navigation methods: A survey. *Acta Astronautica*, 191:22–40, 2022.
- [50] Andrew E Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on pattern analysis and machine intelligence*, 21(5):433–449, 1999.
- [51] Andrew Rhodes, Eric Kim, John A Christian, and Thomas Evans. Lidar-based relative navigation of non-cooperative objects using point cloud descriptors. In *AIAA/AAS Astrodynamics Specialist Conference*, page 5517, 2016.
- [52] Odysseas Kechagias-Stamatis, Nabil Aouf, and Vincent Dubanchet. Evaluating 3d local descriptors and recursive filtering schemes for lidar-based uncooperative relative space navigation. *Journal of Field Robotics*, 37(5):848–888, 2020.
- [53] Odysseas Kechagias-Stamatis, Nabil Aouf, Vincent Dubanchet, and Mark A Richardson. Deeplo: Multi-projection deep lidar odometry for space orbital robotics rendezvous relative navigation. *Acta Astronautica*, 177:270–285, 2020.
- [54] P.J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.
- [55] Farzin Amzajerdian, Diego Pierrotet, Larry B Petway, Glenn D Hines, Vincent E Roback, and Robert A Reisse. Lidar sensors for autonomous landing and hazard avoidance. In *AIAA Space 2013 Conference and Exposition*, page 5312, 2013.
- [56] Andrew E Johnson and James F Montgomery. Overview of terrain relative navigation approaches for precise lunar landing. In *2008 IEEE Aerospace Conference*, pages 1–10. IEEE, 2008.
- [57] Andrew Johnson and Tonislav Ivanov. Analysis and testing of a lidar-based approach to terrain relative navigation for precise lunar landing. In *Aiaa guidance, navigation, and control conference*, page 6578, 2011.
- [58] Jean de Lafontaine, David Neveu, and Jean-François Hamel. Pseudo-doppler velocity navigation for lidar-based planetary exploration. In *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, page 6664, 2006.
- [59] Robert A Hewitt, Timothy P Setterfield, and Nikolas Trawny. Lidar-based map relative localization performance analysis for landing on europa. In *2021 IEEE Aerospace Conference (50100)*, pages 1–13. IEEE, 2021.
- [60] Roberto Furfaro, Ilaria Bloise, Marcello Orlandelli, Pierluigi Di Lizia, Francesco Topputo, and Richard Linares. A recurrent deep architecture for quasi-optimal feedback guidance in planetary landing. 2018.
- [61] Luca Ghilardi, Andrea D’ambrosio, Andrea Scorsoglio, Roberto Furfaro, Richard Linares, and Fabio Curti. Image-based optimal powered descent guidance via deep recurrent imitation learning. In *AIAA/AAS Astrodynamics Specialist Conference*, 2020.
- [62] Duarte Rondao and Nabil Aouf. Multi-view monocular pose estimation for spacecraft relative navigation. In *2018 AIAA Guidance, Navigation, and Control Conference*, page 2100, 2018.
- [63] Planet and asteroid natural scene generation utility. <https://pangu.software/>. Accessed: 2022-01-07.
- [64] Ronald R Sostaric, Sam Pedrotty, John M Carson, Jay N Estes, Farzin Amzajerdian, Alicia M Dwyer-Cianciolo, and James B Blair. The splice project: Safe and precise landing technology development and testing. In *AIAA Scitech 2021 Forum*, page 0256, 2021.
- [65] Px4 sil gazebo simulation. <https://docs.px4.io/master/en/simulation/gazebo.html>. Accessed: 2022-01-17.
- [66] H Andrew Lassiter, Travis Whitley, Benjamin Wilkinson, and Amr Abd-Elrahman. Scan pattern characterization of velodyne vlp-16 lidar sensor for uas laser scanning. *Sensors*, 20(24):7351, 2020.
- [67] James W Head, Caleb I Fassett, Seth J Kadish, David E Smith, Maria T Zuber, Gregory A Neumann, and Erwan Mazarico. Global distribution of large lunar craters: Implications for resurfacing and impactor populations. *science*, 329(5998):1504–1507, 2010.
- [68] CI Fassett, JW Head, SJ Kadish, E Mazarico, GA Neumann, DE Smith, and MT Zuber. Lunar impact basins: Stratigraphy, sequence and ages from superposed impact crater populations measured from lunar orbiter laser altimeter (lola) data. *Journal of Geophysical Research: Planets*, 117(E12), 2012.
- [69] P Gläser, F Scholten, D De Rosa, R Marco Figuera, J Oberst, E Mazarico, GA Neumann, and MS Robinson. Illumination conditions at the lunar south pole using high resolution digital terrain models from lola. *Icarus*, 243:78–90, 2014.
- [70] Wei-Feng Hao, Fei Li, Jian-Guo Yan, Jie Zhang, and Xiao-Li Su. Lunar polar illumination based on chang’e-1 laser altimeter. *Chinese Journal of Geophysics*, 55(1):46–52, 2012.
- [71] William M Kaula, Gerald Schubert, Richard E Lingenfelter, W Lu Sjogren, and WR Wollenhaupt. Apollo laser altimetry and inferences as to lunar structure. In *Lunar and Planetary Science Conference Proceedings*, volume 5, pages 3049–3058, 1974.
- [72] What is motion capture?: What can i use motion capture for?, Apr 2020.