JOURNAL OF SPACECRAFT AND ROCKETS
Vol. 62, No. 1, January–February 2025

Technical Notes

Real-Time Stereovision-Based Spacecraft Pose Determination Using Convolutional Neural Networks

Francis Thomas Despond* and Steve Ulrich†
Carleton University, Ottawa, Ontario K1S 5B6, Canada

https://doi.org/10.2514/1.A35973

I. Introduction

ONE of the key challenges inherent to any dual-spacecraft missions involving rendezvous and proximity operations maneuvers is related to the onboard determination of the pose (i.e., relative position and orientation) of a target object with respect to a robotic spacecraft equipped with computer vision sensors. The pose of a target represents crucial information upon which real-time path planning, pose tracking, and capture actions are planned and executed. Some of the conventional cooperative pose determination methods are based on appearance or feature tracking approaches that rely on detecting and matching known key points and features to an existing database model. Historically, a feature-based tracking pose determination system was namely used during docking maneuvers of the Space Shuttle to the International Space Station (ISS), where a light detection and ranging (LIDAR) sensor was actively tracking retroreflective concentric circular fiducial markers strategically designed and located on the ISS [1]. Similarly, Tweddle and Saenz-Otero [2] used circular fiducial markers to track a target platform in a planar environment. Specifically, the authors proposed a nonlinear iterative photogrammetric scheme to determine the pose of the markers combined with a multiplicative extended Kalman filter. Experiments conducted at MIT's Synchronized, Position, Hold, Engage, and Re-orient Experimental Satellites (SPHERES) facility demonstrated the feasibility of the stereovision approach. Another cooperative pose estimation strategy fusing computer vision data to inertial measurement units and rate gyro sensors through an extended Kalman filter was developed and verified in planar experiments at the Naval Postgraduate School's Spacecraft Robotics Laboratory [3].

However, more advanced pose determination systems have to be developed to enable future on-orbit servicing and orbital debris removal missions, such as Astroscale's Cleaning Outer Space Mission[§] and Northrop Grumman's Mission Robotic Vehicle.[‡] Indeed, in those circumstances, the robotic servicer spacecraft will need to resolve the pose of uncooperative targets, that is, targets not equipped with fiducial markers. To this end, Ruel et al. [4] developed an uncooperative real-time pose determination system, referred to as TriDAR, which was tested during the Space Shuttle ISS fly-by missions STS-128, STS-131, and STS-135. The system consisted of a LIDAR and a triangulation hybrid system that generated, up to a distance of 3 km, three-dimensional (3D) point clouds that were compared against a pregenerated 3D model through a modified iterative closest point (ICP) algorithm. With the advent of SpaceX and commercial space flight, the technology used in upcoming missions will have to be more effective in terms of size, mass, power, and cost. In this context, extensive research efforts to replace LIDAR sensors with passive components such as monocular cameras have been done in the past few years. Shi and Ulrich [5] recently proposed an uncooperative monocular pose determination system that combined a new foreground extraction technique with a level-set region-based pose estimator with improved initialization and gradient descent functionalities. The authors' method was successfully validated using synthetically generated motion sequences as well as with actual telemetry from the STS-135 ISS undocking flight segment. Rondao et al. [6] designed a monocular approach in which the observed face of the target was defined as a classification problem and where the 3D shape was learned offline using Gaussian mixture modeling. Numerical validations of the authors' approach were performed using the open-source monocular vision-based Spacecraft PosE Estimation Dataset (SPEED) dataset [7]. Both aforementioned monocular vision approaches were model-based, implying that they rely on a known 3D computer-aided design (CAD) model of the target.

Although monocular vision is the most effective option in terms of power and mass requirements, stereo cameras do not suffer from the scale invariance problem that affects monocular systems and can therefore efficiently resolve depth information at a fraction of the cost of LIDAR, yet with a significantly limited operation range. For this reason, several authors have proposed stereo-vision-based pose determination approaches. Among those, schemes that do not rely on detailed information about the target in the form of a known 3D CAD model may be advantageous. Tweddle et al. [8] approached this problem by applying a dynamic version of a simultaneous localization and mapping (SLAM)-based method, known as incremental smoothing and mapping (iSAM) [9]. Using the OpenCV implementation of speeded-up robust features (SURF) to match key points between two frames, the authors obtained the pose through Horn's absolute orientation algorithm. Although successful, the authors' SLAM-based approach was not suitable for real-time applications due to its large computation requirement. Fourie et al. [10] determined in real-time the relative position of a spinning target's center of geometry through the application of stereo depth mapping. The approach was successfully verified with the ISS-SPHERES six-degree-of-freedom (6-DOF) facility but did not resolve the full pose as the relative orientation determination was omitted. Grompone [11] approached the problem using a combination of SURF and the Kanade–Lucas–Tomasi method to track an unknown target and proposed an epipolar-constrained set of equations for pose determination but did not successfully validate it in hardware experiments. He et al. [12] demonstrated a method that identifies geometric point cloud features on a target and used a particle filter algorithm to approximate the pose. The method was tested on a simulated target with known and unknown states and was found to be effective at tracking the pose. However, the particle filtering algorithm proved to be computationally demanding. Nassir and Giorgio [13] proposed another approach, which utilized motion flow and stereo correspondences to estimate,

Received 31 December 2023; revision received 27 June 2024; accepted for publication 25 August 2024; published online 24 September 2024.

*Graduate Student, Department of Mechanical and Aerospace Engineering, 1125 Colonel By Drive.

†Associate Professor, Department of Mechanical and Aerospace Engineering, 1125 Colonel By Drive; steve.ulrich@carleton.ca. Associate Fellow AIAA (Corresponding Author).

§Astroscale, "COSMIC," https://astroscale.com/missions/cosmic/ [retrieved 24 June 2024].

‡Northrop Grumman, "Order by Intelsat Completes Manifest for Inaugural Launch in Early 2025," 20 June 2023, https://news.northropgrumman.com/news/releases/northrop-grummans-spacelogistics-continues-revolutionary-satellite-life-extension-work-with-sale-of-third-mission-extension-pod [retrieved 24 June 2024].

via the dual quaternion representation, the 6-DOF pose of an uncooperative target.

More recently, one of the most influential advancements in computer vision has been the application of deep learning methods, in particular convolutional neural networks (CNNs), to perception—the ability to interpret or understand the information within a given image. With increasing access to large amounts of data and more powerful embedded computers, CNNs, which were traditionally used primarily for the detection and segmentation of objects in an image, have begun to be considered as a powerful and viable contemporary approach to solve the pose estimation problem [14–16]. Shi et al. [17] used a pretrained network as their baseline network and further trained their model using synthetic images of small spacecraft platforms. Planar experiments conducted at Carleton University's Spacecraft Robotics and Control Laboratory demonstrated the performance and real-time capability of their CNN to detect the target platform in the camera field of view. Sharma et al. [18] showed the viability of a CNN being used entirely for pose determination from monocular camera images. Similar to Shi et al. [17], to overcome the limited amount of space image data, the authors utilized transfer learning with a pretrained network using the ImageNet dataset [19,20]. Their results obtained from a numerical testing environment showed that, even with a pretrained network, pose accuracy was significantly improved with a larger number of training data. This model classified the images that best matched a predetermined set of orientations, and based on the number of sets to match, the overall accuracy of the network changed accordingly. Sharma and D'Amico [21] recently improved their work by designing five main convolutional layers branching out to three separate output layers used to estimate the bounding box, relative attitude class, and relative attitude regression. The bounding box branch is based on the region proposal network, while the attitude branches are composed of fully connected layers. Using these outputs along with geometric constraints, the authors were able to estimate the relative position of a spacecraft using the Gauss–Newton algorithm and the attitude from the relative attitude classification and regression outputs. Cassinis et al. [22] proposed a monocular-model-based pose estimation approach using a CNN for feature detection combined with a robust Perspective-n-Points solver for pose estimation. Another monocular pose estimation system was developed by Sonawani et al. [23], where transfer learning was employed to reduce the training requirements. Their parallel neural network architecture was shown to yield good performance when evaluated with synthetically generated images. However, this particular network architecture decreases the inference speed in comparison with regular architectures.

Existing monocular datasets for training CNNs, such as SPEED [7], have originally mostly relied on synthetic images for both training and validation. However, such an approach fails to accurately model realistic visual features and illumination conditions of target spacecraft. To solve this problem, the SPEED+ [24] spacecraft pose dataset, which consists of both synthetic images for training and hardware-in-the-loop images of a spacecraft model captured from the Stanford's Testbed for Rendezvous and Optical Navigation facility, was proposed. This dataset was notably used in the second Satellite Pose Estimation Challenge cohosted by Stanford and the European Space Agency to evaluate and compare the robustness of neural-network-based models trained on synthetic images. Subsequent datasets from Stanford that build on SPEED/SPEED+ include the Satellite Hardware-in-the-Loop Rendezvous Trajectories (SHIRT) [25] dataset, the Spacecraft PosE Estimation Dataset of a 3U CubeSat using Unreal Engine (SPEED-UE-Cube) [26], and the Spacecraft Pose Estimation and 3D Reconstruction (SPE3R) [27] dataset. Musallam et al. [28] introduced a dataset with images from multiple 6-DOF trajectories created in a laboratory setting and evaluated a CNN on this dataset. However, all aforementioned datasets consist of monocular images, which, in turn, cannot be used in the context of stereovision-based relative pose determination that this paper focuses on.

In this context, the original contributions of this work are 1) the development of a novel stereo-camera-based CNN architecture for spacecraft relative pose estimation, and 2) its real-time embedded implementation and experimental validation in a planar 3-DOF laboratory facility.

More precisely, at the fundamental level, the proposed stereovision approach is similar to other CNNs [29,30] developed in the context of stereo disparity mapping in which grayscale stereo images are combined, yet herein applied to solve the relative pose determination problem. Overlaying the left and right images to create a single two-channel image allows for the disparity between both images to be considered into the architecture design of the network without any additional work, thereby increasing the inference speed. This feature is, to the best of the authors' knowledge, a novel approach to stereo-based pose determination. Furthermore, compared to some of the current state-of-the-art CNN-based pose determination architectures [18,21] that feed bounding box information into separate layers to resolve the relative pose, the network developed in this paper leverages a multi-output approach for the relative pose and target detection confidence score to output all the values using a single network in a single step. Finally, no real-time laboratory experimental work has been reported in the literature in the area of CNN-based spacecraft relative pose determination, which further contributes to the relevance of this work.

II. Network Architecture

This section presents the design and architecture of a novel CNN to determine the relative planar pose (x, y, ψ) of a chaser platform with respect to an uncooperative target platform. This network was developed using Python using Tensorflow 2.0.

The novel deep learning architecture used to determine the relative pose is developed to be computationally efficient so that it can be deployed in real-time applications and executed on an embedded hardware platform. The authors' earlier attempts at designing a network using transfer learning to leverage pretrained network weights proved to be unsuccessful [31]. Indeed, pretrained networks are not transferable efficiently to stereo images since they need to be duplicated to consider stereo images. As such, a custom architecture was created. Drawing inspiration from some of the most influential CNNs in computer vision, the proposed network consists of four major subsections that contribute to the overall performance of the pose determination system. As detailed in Table 1 and illustrated in Fig. 1, the first main section is the image capture and preprocessing step, the second is the multifilter input layer, followed by the main convolutional layers, and the final layers are the fully connected output layers. Compared to state-of-the-art spacecraft relative pose determination CNNs [18,21], this approach is original as it uses and stacks two grayscale stereo images as the first input layer into the network and processes them through the multifilter input layer via three varying-sized filters (1×1, 3×3, and 5×5 kernels) that operate on the stacked input image to produce a single concatenated tensor. In addition, the proposed deep-learning-based method leverages a multi-output approach where a single network produces the relative planar pose (x, y, ψ), as well as the target detection confidence score. In comparison, the most recent work of Sharma and D'Amico [21] leveraged a regional proposal network that outputted bounding box information, which was then fed into separate layers to resolve the relative pose. In the remainder of this section, the four main sections of the network are detailed.

A. Image Capture and Processing

The image capture layer has a great impact on the overall size and speed of the network. Indeed, the larger the input, the more the memory required to process the network, thereby reducing the computational efficiency. The initial preprocessing stage took inspiration from existing non-machine-learning approaches when processing a stereo image. Specifically, in traditional stereovision techniques, the kernel applied to the left image is simultaneously applied to the right image to generate a disparity map and infer depth information. Since CNNs apply their kernels to the multichannel inputs in a similar fashion, the stereo images were first downsized and converted to grayscale and were then overlaid to create a single two-channel image. This allowed for the disparity between both input images to

Table 1 Convolutional network architecture, where a convolutional layer (conv layer) consists of a series of convolutional filters and a leaky ReLU activation layer

Layer	Layer details			
1	Input layer 2 channel left and right image grayscale			
2	$1 \times 1 \times 32$ conv layer	$3 \times 3 \times 32$ conv layer	$5 \times 5 \times 32$ conv layer	
3	Concatenate previous layers to create a single tensor			
4	$3 \times 3 \times 128$ conv layer			
5	Max pooling 2×2			
6	$3 \times 3 \times 256$ conv layer			
7	$1 \times 1 \times 128$ conv layer			
8	$3 \times 3 \times 256$ conv layer			
9	Max pooling 2×2			
10	$3 \times 3 \times 512$ conv layer			
11	$1 \times 1 \times 256$ conv layer			
12	$3 \times 3 \times 512$ conv layer			
13	Max pooling 2×2			
14	$3 \times 3 \times 1024$ conv layer			
15	Max pooling 2×2			
16	$3 \times 3 \times 1024$ conv layer			
17	Max pooling 2×2			
18	$3 \times 3 \times 512$ conv layer			
19	Average pooling 2×2			
20	Flatten			
21	Fully connected layer (128 nodes)	Fully connected layer (128 nodes)	Fully connected layer (256 nodes)	Fully connected layer (32 nodes)
22	x (linear)	y (linear)	ψ (softmax)	Target detection confidence (sigmoid)

All fully connected layers use the ELU activation function.

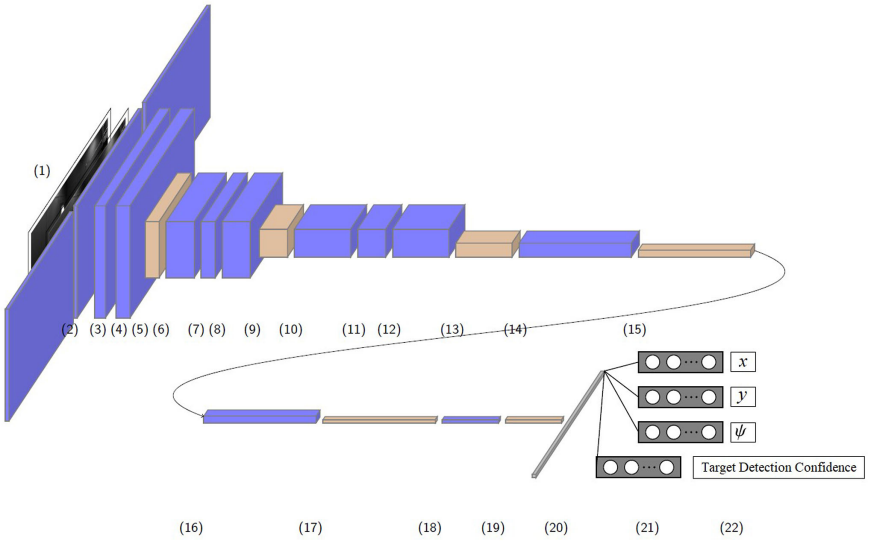


Fig. 1 Visual representation of the CNN architecture. Blue boxes are convolutional layers with leaky ReLU activations, and tan boxes are pooling layers.

be considered into the architecture without any additional work. This, to the best of the authors' knowledge, is a novel feature of stereo-based CNNs relative pose determination. Furthermore, the conversion to grayscale instead of considering three-channel color images allowed for a faster and relatively smaller network. The image color conversion and downsizing were done in OpenCV using the *imread* and *resize* functions, respectively. To determine the most appropriate input image pixel (px) dimensions, the network was evaluated using three different sizes: 320 px × 320 px, 448 px × 448 px, and 576 px × 576 px. However, in this paper, only the results from the smaller, yet faster, network are included for conciseness.

B. Multikernel Layer

The first convolution layer in the CNN consists of three separate convolution operations that are concatenated together into a single tensor, similar to that used by the Google Inception network [32]. Such a multikernel approach is known to improve the performance when compared to a single filter of equal size. The objective of this layer is to extract additional spatial information from the raw image early on. This contributes to reducing the overall network size. These filter sizes are $1 \times 1 \times 32$, $3 \times 3 \times 32$, and $5 \times 5 \times 32$, with the strides and padding set such that the output image size is equal to the input image dimensions. Stacking the output of the filters produces a single tensor of size $A \times A \times 96$, where A is the input image size.

C. Convolutional Layers

After the multikernel layer, the main computations of the CNN are performed. These layers are inspired by the You-Look-Only-Once (YOLO) architecture [33], consisting of 10 convolutional layers with increasing and decreasing filter sizes and a leaky rectified linear unit

(ReLU) activation function after each layer. Leaky ReLU, which is a type of activation function based on ReLU but with a small slope for negative input values (instead of a flat slope), is given by

$$f_x = \begin{cases} \alpha x & x < 0 \\ x & x \geq 0 \end{cases} \tag{1}$$

where f_x, α, and x denote the activation function, slope, and input of the activation function, respectively. Leaky ReLU activation functions were used to specifically reduce to the well-known vanishing gradient problem [33]. One of the most important considerations for the main convolutional layer section of the network is to reduce the number of connections to the subsequent fully connected multi-output layer. The fully connected layers are memory-intensive due to every node having to be directly connected to the input layer. This final layer of the convolutional layer section ultimately controls the overall network size and provides a balance between keeping enough information while reducing the overall memory size. Thus, controlling the total number of outputs from the convolutional layer section proved to be very important to limit the size of the network. For example, changing the last layer of this section from 512 to 1024 filters increases the number of parameters to train by 28.6% from 29,101,795 to 40,784,099 for the smallest 320×320 model.

D. Fully Connected Multi-Output Layer

The final section of the CNN contains four separate, fully connected output layers. Three of these calculate the relative planar pose, while the fourth layer produces a target detection confidence score, which quantifies whether or not the target spacecraft is detected within the field of view of the stereo camera. This approach contributes to decoupling each component of the outputs so that if tuning or additional training needs to be done, it could be done selectively on one layer while holding the other layers constant. This is useful when training the confidence score since it requires passing images of the target as well as images where there is no target in the field of view. In the latter case, the values produced from the relative pose layers are nonsensical, and allowing backpropagation on those values would cause the network to diverge. To get around this issue, the target detection confidence score is thus trained separately from the other three layers.

The relative position components, x and y, are obtained from two successive layers. The first layer consists of 128 nodes with an exponential linear unit (ELU) activation function described by

$$f_x = \begin{cases} e^x - 1 & x < 0 \\ x & x \geq 0 \end{cases} \tag{2}$$

These nonlinear activation layers allow for time-varying derivatives, thereby allowing the network to train properly. This layer is connected to a second single node that uses a linear activation function. The resulting output of this single node layer is desired to be linear since the relative x and y components are unbounded, while the objective is to resolve positions not seen in training.

Similarly, the relative orientation component ψ is determined from two consecutive layers. The first layer consists of 256 nodes with ELU activation functions that are connected to a second layer that comprises 128 discretized outputs calculated through *softmax* activation functions given by

$$f(\boldsymbol{x})_i = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \tag{3}$$

which outputs the probability of the relative orientation, with a resolution of $2\pi/128$ rad. While increasing the number of outputs would improve the relative orientation determination performance, the number of nodes in the previous layer would also need to be increased.

Finally, the last output of the CNN is the target detection confidence score obtained through two layers again. The primary objective of this layer is to determine when the outputs from the other layers can be considered valid and therefore usable by the main guidance and control loop. The first layer consists of 32 nodes with an ELU activation function and a single node with a sigmoid activation function:

$$f_x = \frac{1}{1 + e^{-x}} \tag{4}$$

The sigmoid activation function outputs a bounded value between 0 and 1, allowing for the calculation of a binary confidence score, with 1 indicating that the target is within the field of view of the stereo camera.

Overall, the developed architecture is different from the current trend in CNNs, e.g., YOLO [33], in which multiple objects can be tracked and identified based on the training data. The ability to simultaneously track and localize multiple objects was not deemed essential for this particular application. Creating a smaller and more efficient network for the specific task of planar pose determination allowed the architecture to leverage fully connected layers in this way. This reduced both the training complexity and network size.

III. Testbed and Data Collection

This section first provides an overview of the experimental facility used to generate and collect experimental data. Then, the data collection approach and methodology to gather training and testing data for the developed convolutional network are described.

A. Testbed

The data collection campaign was conducted at the Spacecraft Robotics and Control Laboratory of Carleton University, using the Spacecraft Proximity Operations Testbed (SPOT) facility. This free-floating testbed, pictured in Fig. 2, is used by researchers to investigate robotics, control, path planning, and computer vision technologies enabling spacecraft proximity operation tasks, such as inspection maneuvers, rendezvous and docking, and robotic capture of a spinning target. SPOT consists of three air-bearing spacecraft platforms operating in close proximity on a 2.4×3.5 m granite surface. The use of air bearings on the platforms reduces the friction to a negligible level. All platforms have dimensions of $0.3\ \text{m} \times 0.3\ \text{m} \times 0.3\ \text{m}$ and are actuated by expelling compressed air at 550 kPa (80 psi) through eight miniature air nozzles distributed around each platform, thereby providing full planar control authority. Each thruster generates approximately 0.25 N of thrust and is controlled at a frequency of 10 Hz. The guidance and control functions of each platform are autocoded from MATLAB/Simulink and executed on the Raspberry Pi3 in charge of issuing commands to actuate the thrusters.

During experiments, ground-truth knowledge of the platforms' position and attitude on the granite surface is obtained through a 10-camera PhaseSpace™ motion capture system. The motion-capture infrared cameras track the position of active LEDs positioned at the four corners of each platform. Using these data, the planar position and orientation of all platforms are calculated and sent to the ground computer for postprocessing data analysis purposes. The cameras can update the position of all spacecraft at a user-defineable frequency, up to 960 Hz, with a standard deviation less than 0.1 mm.

Fig. 2 Carleton University's Spacecraft Proximity Operations Testbed (SPOT).

This information is also relayed to the platforms' onboard Raspberry Pi3 computer to be used by the main GNC algorithms whenever a particular experiment does not involve real-time computer vision techniques and sensors.

To enable more demanding tasks such as real-time deep-learning-based pose determination, the chaser platform is further equipped with an NVIDIA Jetson TX2–embedded computer. This 15 W module consists of a 256-core NVIDIA Pascal graphical processing unit (GPU), a Dual-Core NVIDIA Denver 64-bit computing processing unit (CPU), a Quad-Core ARM Cortex A57 MPCore processor, 8 GB of LPDDR4 memory, and 32 GB of internal storage. This computer dedicated to computer vision and machine learning processes is interfaced with a ZED stereo camera by StereoLabs™, which is capable of capturing up to 2K video at 15 frames per second. However, for this work, the resolution was set to 720p and the capture rate mode at 30–60 fps. The outputs of the CNN are returned to the Raspberry Pi-3 that executes the main guidance and control functions for motion planning and tracking, respectively.

B. Data Collection

Data collection is arguably the most important part of the proposed machine learning approach. Indeed, a CNN model performance is only as good as the data given to it. Since there are no pre-existing spacecraft stereo imagery datasets, a custom dataset had to be created specifically for this research. Leveraging the PhaseSpace motion capture camera system's ability to record positioning of the platforms with sub-millimeter accuracy and the ZED stereo camera, several thousands of images along with positioning ground-truth data were recorded. Of these, some were used for the training and validation of the proposed deep learning architecture, while an additional test dataset was used exclusively to test the model's generalization.

Synchronizing this motion capture positional data with the camera images allows for an accurate data set to be used during training. More specifically, recording this ground truth data while the target platform performs translational and rotational maneuvers in front of the chaser platform allows for the relative pose data and the respective camera image to be collected and used as training data for the CNN model.

To ensure that the camera data being recorded has the target platform within the field of view of the stereo camera system, both platforms were randomly moved around on the table while ensuring adequate coverage of the workspace with different relative orientations and positions. If there was no target spacecraft in the frame, the data were labeled as NULL data. These NULL data were considered in training to handle situations where the target platform is not in the camera frame, as shown in Fig. 3. To help with the network robustness of the training and to add variations to the dataset, the chaser platform had its robotic arm installed and visible in the camera frame in both a deployed and stowed positions, as illustrated in Figs. 4 and 5. All these variations were collected in the attempt to create a more generalizable model that is less prone to failure in any future experiments, whenever the developed CNN-based pose determination system is used in combination with the three-link robotic arm. Throughout all the data collection, a total of 151,101 frames were

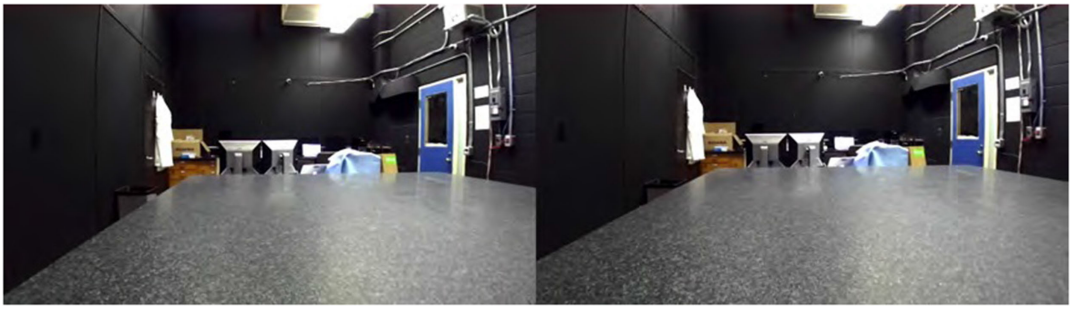


Fig. 3 Example NULL data.

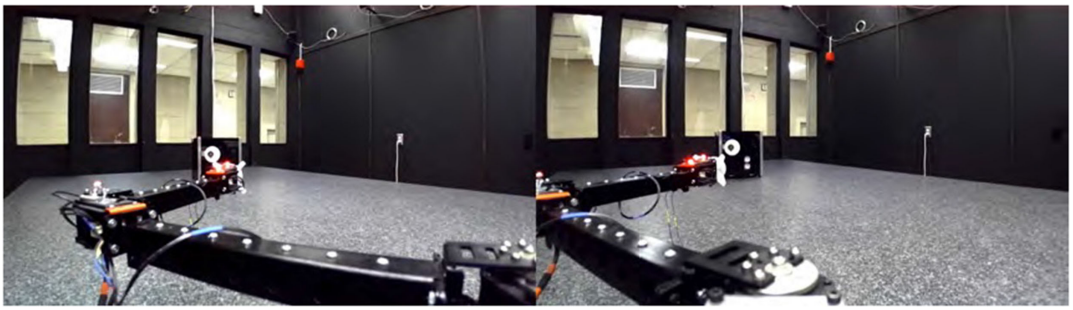


Fig. 4 Example stereo camera imagery with robotic arm extended partially occluding the frame.

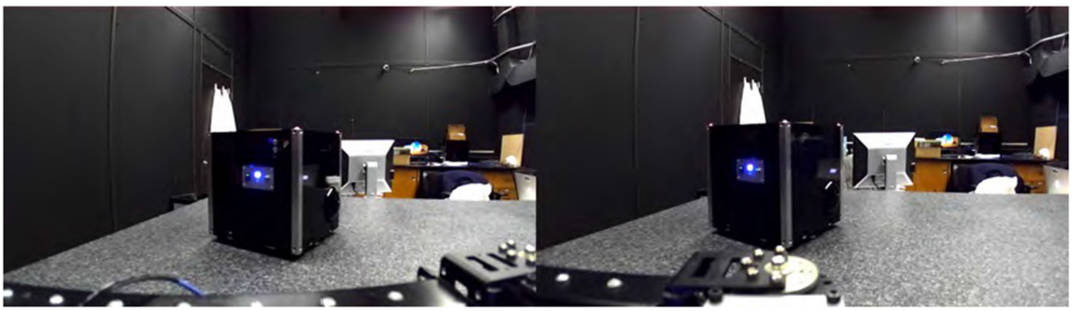
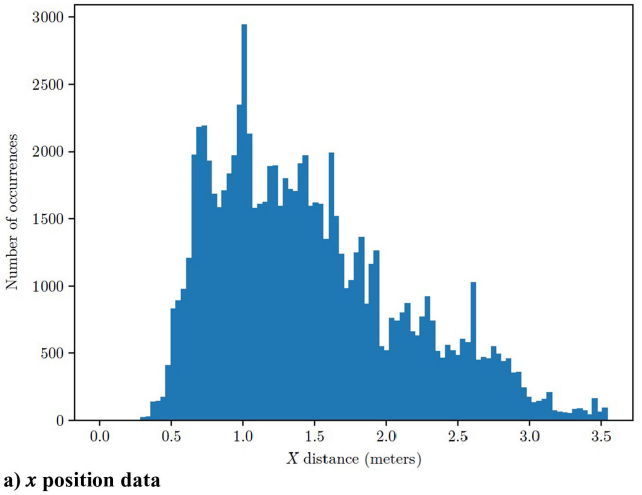


Fig. 5 Example stereo camera imagery with robotic arm retracted.

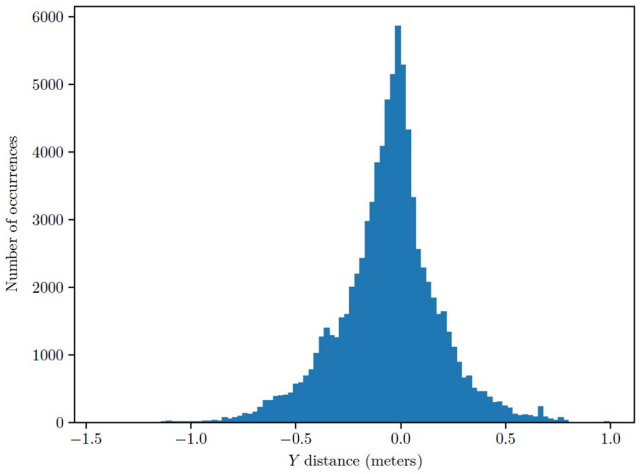
recorded. Any data that were acquired and were known to be invalid were discarded and are not included in this total. However, due to the large scale of this dataset, there is a possibility of inaccurate data being nonetheless included. That being said, the vast majority of images and positional data is accurate and outweighs any negatives that may arise from some outliers.

Of the 151,101 frames recorded, a total of 132,663 were used for training data, with 86,143 specifically being used for the relative pose training and validation since they contained the target spacecraft in the frame. The remaining 46,520 were used as NULL data for the spacecraft detection output. The abundance of NULL data was collected to ensure that an adequate amount of NULL data was present for early design ideas as well as for future designs that may benefit from NULL data.

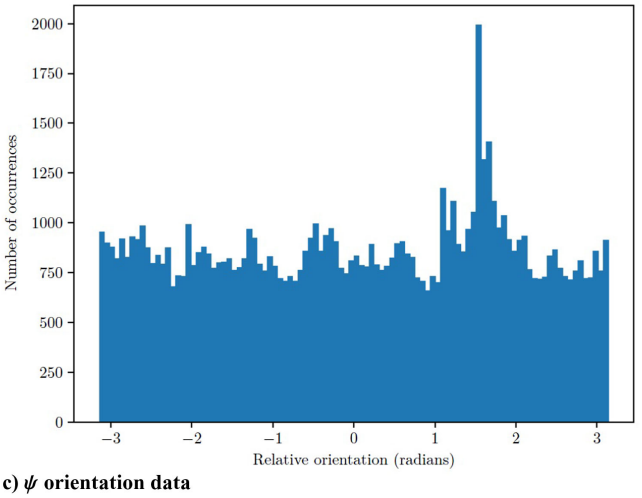
The graphs of the training/validation and test data provided in Figs. 6 and 7, respectively, show that they both have very similar coverage of the test area. This is due to the limited size of the floating granite surface, and so the test data, although unique, will appear similar to the training data. This is also seen in the x position where there is more space on the table where both platforms can be within



a) x position data

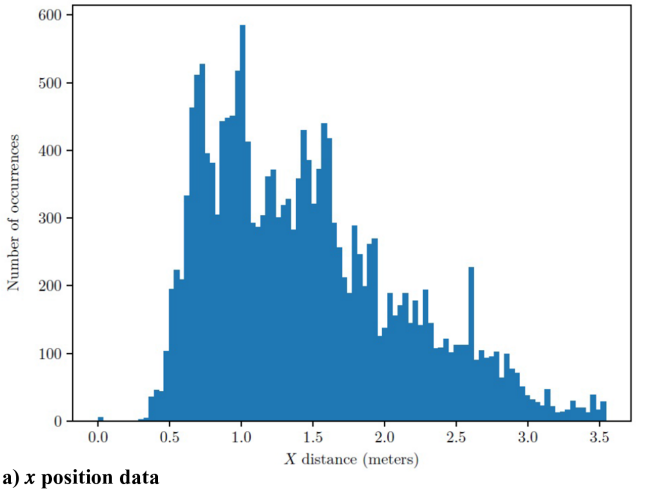


b) y position data

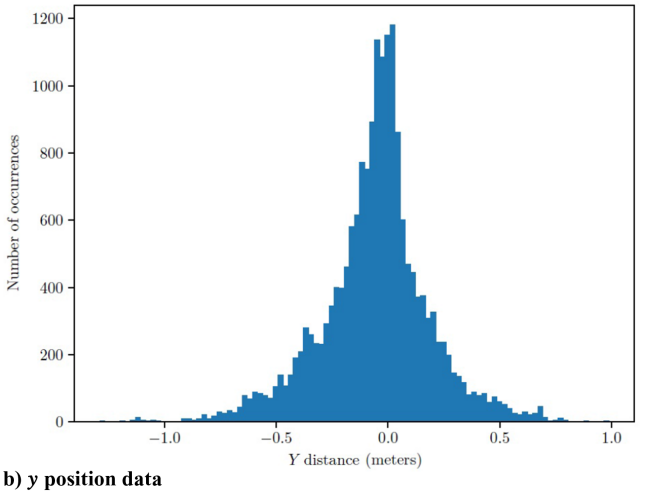


c) ψ orientation data

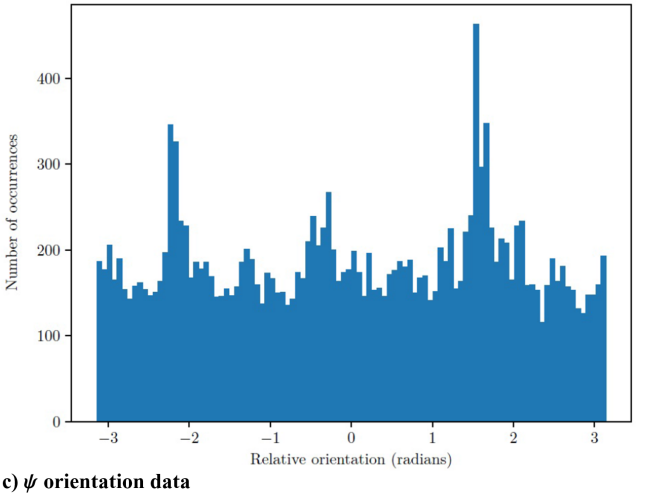
Fig. 6 Breakdown of data for training and validation.



a) x position data



b) y position data



c) ψ orientation data

Fig. 7 Breakdown of data for testing.

0.5–1.5 m compared to distances greater than 3.0 m. Likewise, along the y direction, the table constraints cause more positions to be centered on each other than apart. The other area to note is that in both datasets there is a spike of orientation data around 1.5 rad. This was unintentional but was from the method of gathering data, presumably from the starting orientation more often than not being at that orientation, causing a spike relative to all other orientations.

IV. Training

Training of the CNN was spread across two servers provided by Carleton's Research Computing and Development Cloud services. The first server has a 32-core processor with 32 GB of RAM, while the second provides an additional NVIDIA V100 16 GB GPU. Although both servers were used, the majority of the training was done on the GPU server. The first stage trained the three relative pose outputs while holding the fully connected layer outputs of the target detection confidence layers constant. In the second stage, the inverse is applied, and only the target detection confidence layers are trained while everything else in the network is held constant. This allowed for two major benefits: first, the network could be independently trained on two separate datasets, one containing only information with the target spacecraft in frame and the second containing images of both the target spacecraft and without the target spacecraft; second, it allows for the first stage to train the upper layers based on the relative pose estimation requirements and forces the target detection confidence layers to learn when the target is in frame based on the outputs from those trained layers.

Training was done with dropout layers in between each layer of the fully connected layers operating at a 40% dropout rate. The images were trained in mini batches of 8–16 (due to memory constraints), and the training was allowed to run until the validation error no longer improved after two iterations. The images were subjected to image augmentation, where only the brightness of the images was randomly adjusted by a maximum of 25% of their original values. This allowed for different training epochs to achieve slightly different images and improve robustness in the scene with regards to ambient lighting.

To measure how well the CNN models the training data, regression and classification loss functions were used. Specifically, the mean absolute error (MAE) loss function was adopted for the relative x and y positions, as follows:

$$\mathrm{MAE}_x = \frac{1}{n}\sum_{i=1}^{n}|x_i - \hat{x}_i| \tag{5}$$

$$\mathrm{MAE}_y = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{6}$$

where x and y denote the ground truth relative position of the target platform with respect to the chaser platform, calculated based on the PhaseSpace motion capture system, and where $\hat{x}$ and $\hat{y}$ denote the determined relative position.

For the relative orientation ψ, the cross entropy (CE) loss function was selected, whereas the binary cross entropy (BCE) loss function was used for the target detection confidence score S, as follows:

$$\mathrm{CE}_\psi = -\sum_{i=1}^{128}\psi_i \log\hat{\psi}_i \tag{7}$$

$$\mathrm{BCE}_S = -\left(S\log\hat{S} + (1-S)\log(1-\hat{S})\right) \tag{8}$$

The learning rate was set to 5E−5 and the total loss was the summation of the relevant output layer losses. Training was done as an 80–20 training–validation split with a final additional test dataset to further validate the model. The total number of images used in relative pose training and validation were 68,914 and 17,229, respectively. Adaptive Moment Estimation (ADAM) [34] was selected as the training optimizer as it is known to overcome local minima easier than the widely used stochastic gradient decent approach. The parameters used for ADAM were a learning rate of 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 0.1$. A dropout rate of 30% was used as well to improve the connections between the layers.

V. Experimental Validations

To evaluate the pose determination performance, the CNN is implemented on the SPOT located at Carleton University's Spacecraft Robotics and Control Laboratory. Several maneuvers aimed at generating relative motion between a chaser platform equipped with the stereo camera system and a target platform were performed. These maneuvers were executed on the testbed's 3-DOF planar surface. During the experiments, the relative pose was simultaneously determined in real-time by the proposed deep-learning-based vision system and independently measured by a high-accuracy PhaseSpace motion capture system. As mentioned earlier, while all three network sizes were validated, only the results from the smallest network, i.e., the 320 px × 320 px model, which was found to be most appropriate for real-time applications, are reported in this paper.

A video that illustrates the CNN-determined pose against real-time footage during some of the experiments can be found here. Specifically, the 3-DOF-determined pose is depicted as a green overlay on top of the real-time footage captured by one of the stereo cameras, similar to what shown in Fig. 8. Note that this overlay is only meant to provide a qualitative assessment of how close the determined relative pose is to the actual one, as opposed to qualitatively determining the accuracy of the CNN.

For each experiment scenario, the target platform followed a predefined trajectory over a duration of 120 s, while the chaser remained stationary for all but the final experiment. In total, eight different scenarios were validated, three of which are presented here. The others can be found in Ref. [31].

A. Experiment 1: Translation Along y

For this experiment, a translational maneuver along the y axis over a range of 1.5 m was performed while maintaining the x position and attitude constant. Note that the y axis, based on the lab configuration and relative initial states of both platforms on the table, corresponds to the axis perpendicular to the boresight axis of the camera. The results are reported in Fig. 9, which shows an average error between the CNN and the PhaseSpace ground truth system of 0.88 ± 0.60 cm, 3.90 ± 2.05 cm, and 2.84 ± 1.23 deg, along the x and y axes and about the ψ axis, respectively. For this experiment, the average inference speed was found to be 0.48 s.

B. Experiment 2: Translation Along x and Rotation ψ

For this experiment, a translational maneuver along the x axis over a range of 1.5 m while simultaneously rotating about the z axis over four complete rotations was performed. During this maneuver, the y position of the target platform was maintained constant. The x axis corresponds to the axis parallel to the boresight axis of the camera. The results are on the same order of magnitude as those of the first experiment and are reported in Fig. 10. Specifically, the resulting average

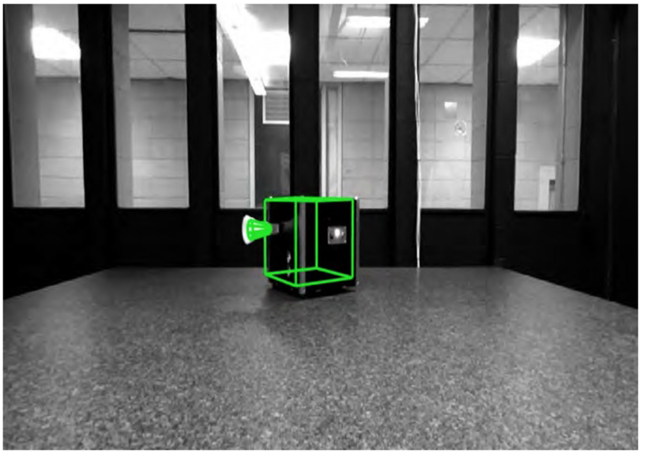


Fig. 8 Image from camera footage with visual overlay.

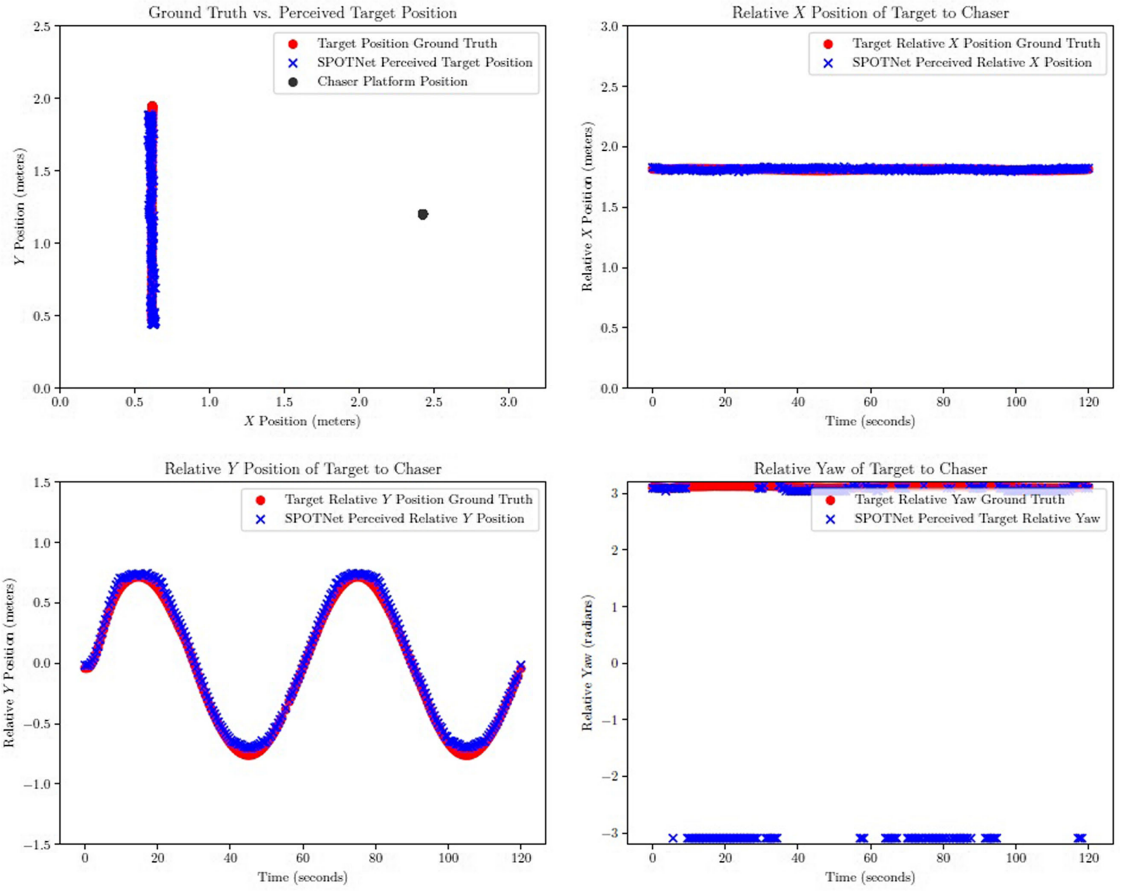


Fig. 9 Experiment 1 results: translation along y.

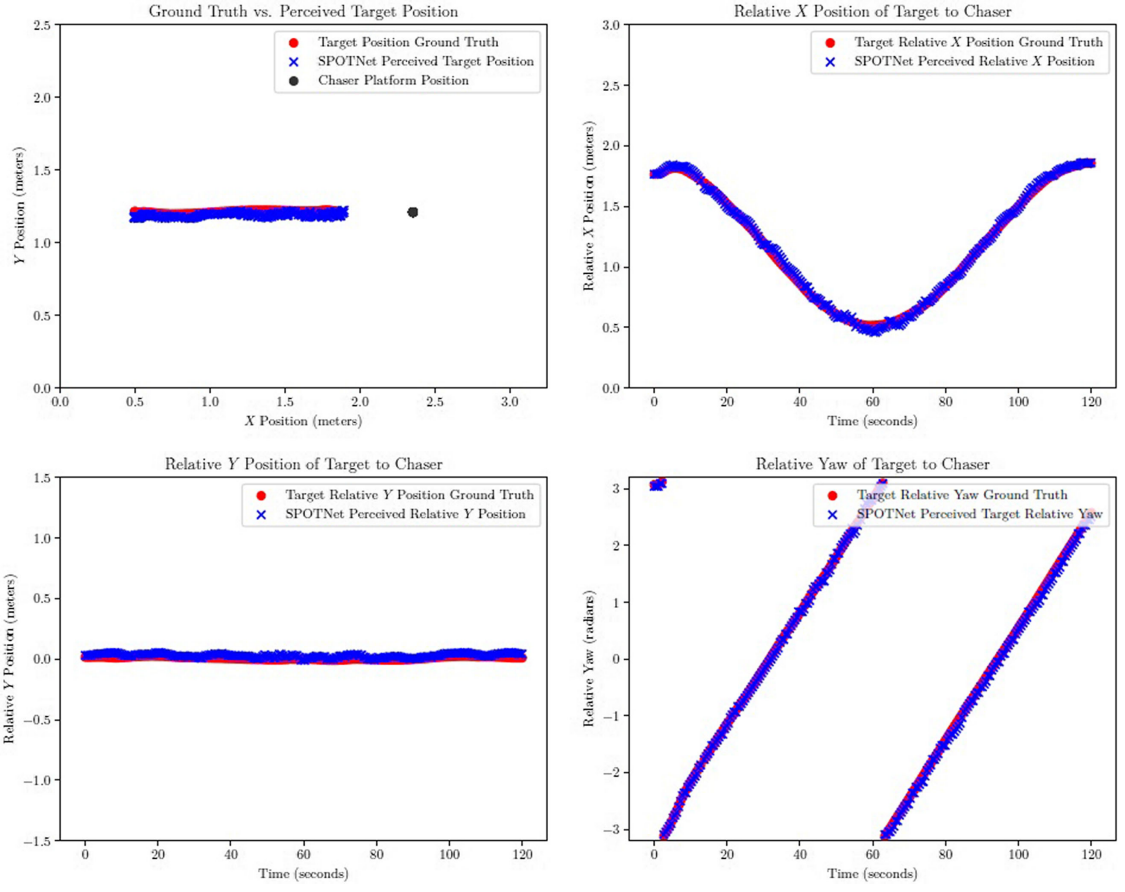


Fig. 10 Experiment 2 results: translation along x and rotation ψ.

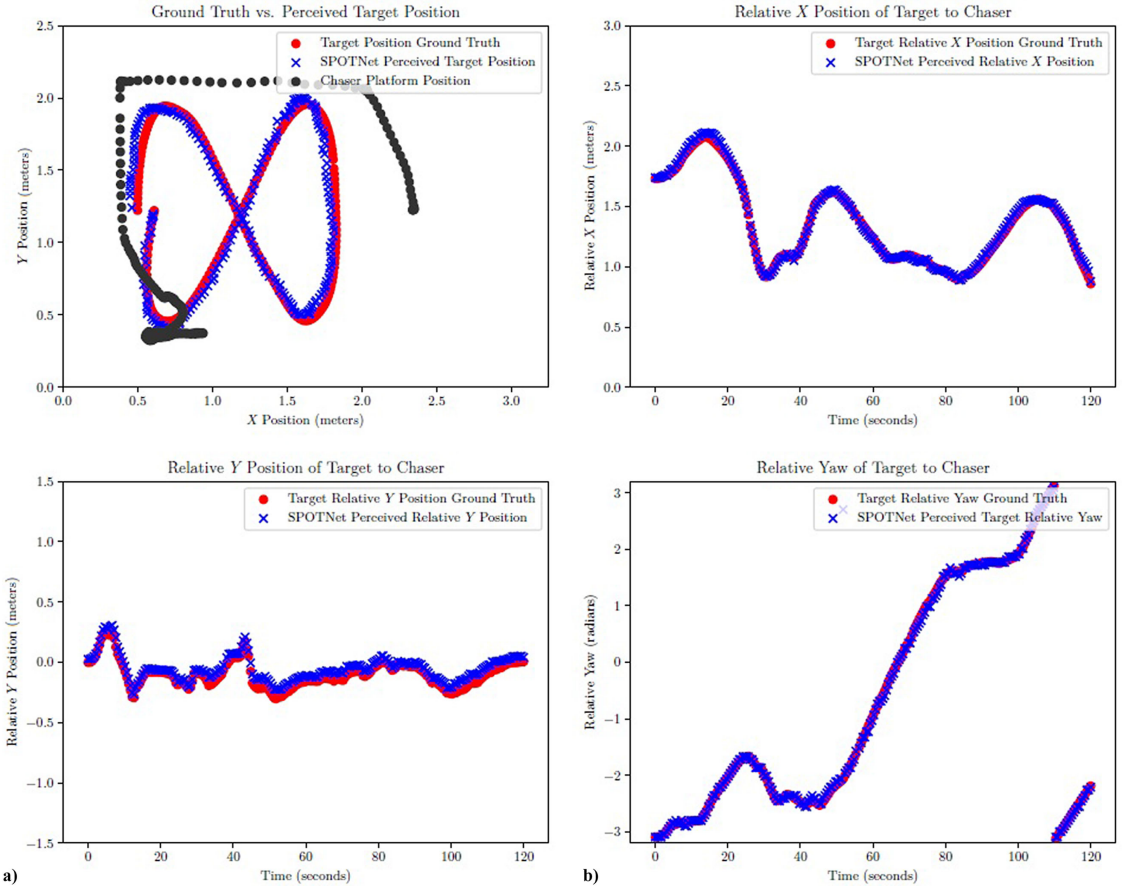


Fig. 11 Experiment 3 results: translation along x and y, and rotation ψ with chaser in motion.

pose determination errors are 1.83 ± 1.31 cm, 2.12 ± 1.17 cm, and 2.23 ± 1.61 deg, along the x and y axes and about the ψ axis, respectively. For this experiment, the average inference speed was found to be 0.49 s, which is almost identical to the one obtained from the first experiment.

C. Experiment 3: Translation Along x and y and Rotation ψ with Chaser in Motion

The pose determination errors associated with this particular experiment are the largest among all eight experiments and across all three input image sizes considered in this validation campaign (i.e., a total of 24 experimental runs). This is due to the highest complexity of the scenario considered, where both platforms were undergoing 3-DOF motion at relative high speed. The results are reported in Fig. 11, and the average pose determination errors are 1.69 ± 1.32 cm, 4.56 ± 1.81 cm, and 2.01 ± 5.691 deg, along the x and y axes and about the ψ axis, respectively. For this experiment, the average inference speed was found to be 0.48 s, which is representative of the other experiments.

D. Discussion

The performance of the developed CNN architecture shows its ability to yield comparable and improved results in some cases to that of the spacecraft relative pose approach by Sharma et al. [18] and Sharma and D'Amico [21]. In their work, the authors' system produced an average error on the order of 3 cm along the x and y axes and 5 deg about ψ. In comparison, the smaller network whose results are provided in this section results in an average error of 1.8 cm, 3.2 cm, and 1.23 deg along x, y, and about ψ, respectively. This demonstrates that the new stereo CNN architecture compares favorably against the current state-of-the-art relative spacecraft pose estimation network model. It is important to note, however, that this comparison is only meant to provide a qualitative perspective to the results presented in this paper. Indeed, the test and laboratory facilities as well as input data considered are quite different.

The experimental validation campaign also highlighted some limitations of the proposed deep learning architecture. For example, during parts of the third experiment reported in this paper, the attitude and position of both platforms were such that the stereovision camera could not maintain the entirety of the target in the sensor field of view. In turn, a noticeable instantaneous decrease in pose determination accuracy could be observed whenever the target platform reached the extremes of the field of view, even if a relatively high target detection confidence score (>0.8) was calculated by the network. The reason for this is attributed to the fact that the training data consisted primarily of images of the target platform being centered in frame. In addition, due to the ambiguity between $-\pi$ and π, the relative orientation ψ graph for the first experiment reported larger differences than what is the actual error. This is a limitation of the proposed pose determination method, which does not prevent discontinuities between two subsequent determined data points. One solution would be to use a Kalman filter in the pose estimation architecture to not only further improve the determination performance, but also prevent any discontinuities in the relative pose solution.

VI. Conclusions

This paper presents the development of a novel real-time spacecraft relative pose determination architecture that relies on stereovision. Some key features of this CNN-based system are that it does not need a CAD model of the target object, it stacks both right and left images as the first layer of the network, and it directly outputs the relative pose as well as the target detection score from input images in a single step without any intermediate functions such as segmentation and corners and edges detection. This results in a computationally efficient pose determination architecture that can be implemented on embedded spacecraft computing hardware, thereby enabling

real-time operations. Although limited to 3-DOF, the laboratory experiments demonstrate that the proposed deep-learning-based stereo pose determination approach resulted in pose errors that compared favorably to current state-of-the-art spacecraft pose determination CNNs.

As future work, expanding the neural network to operate in a 6-DOF environment will be considered. Furthermore, additional research efforts will focus on exploring various ways to reduce the volume of training data while simultaneously improving the generalization of the network to target objects not encountered during training.

Acknowledgments

This work was supported in part by the Ontario Graduate Scholarship program and the Natural Sciences and Engineering Research Council of Canada under the Discovery Grant program.

References

[1] Christian, J. A., and Cryan, S., "A Survey of LIDAR Technology and Its Use in Spacecraft Relative Navigation," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2013-4641, 2013. https://doi.org/10.2514/6.2013-4641

[2] Tweddle, B. E., and Saenz-Otero, A., "Relative Computer Vision-Based Navigation for Small Inspection Spacecraft," *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 5, 2015, pp. 969–977. https://doi.org/10.2514/1.G0006875

[3] Romano, M., Friedman, D. A., and Shay, T. J., "Laboratory Experimentation of Autonomous Spacecraft Approach and Docking to a Collaborative Target," *Journal of Spacecraft and Rockets*, Vol. 44, No. 1, 2007, pp. 164–173. https://doi.org/10.2514/1.22092

[4] Ruel, S., Luu, T., and Berube, A., "Space Shuttle Testing of the TriDAR 3D Rendezvous and Docking Sensor," *Journal of Field Robotics*, Vol. 29, No. 4, 2012, pp. 535–553. https://doi.org/10.1002/rob.20420

[5] Shi, J., and Ulrich, S., "Uncooperative Spacecraft Pose Estimation Using Monocular Monochromatic Images," *Journal of Spacecraft and Rockets*, Vol. 58, No. 2, 2021, pp. 284–301. https://doi.org/10.2514/1.A34775

[6] Rondao, D., Aouf, N., Richardson, M. A., and Dubanchet, V., "Robust On-Manifold Optimization for Uncooperative Space Relative Navigation with a Single Camera," *Journal of Guidance, Control, and Dynamics*, Vol. 44, No. 6, 2021, pp. 1157–1182. https://doi.org/10.2514/1.G004794

[7] Kisantal, M., Sharma, S., Park, T. H., Izzo, D., Märtens, M., and D'Amico, S., "Satellite Pose Estimation Challenge: Dataset, Competition Design and Results," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 56, No. 5, 2020, pp. 4083–4098. https://doi.org/10.1109/taes.2020.2989063

[8] Tweddle, B. E., Saenz-Otero, A., Leonard, J. J., and Miller, D. W., "Factor Graph Modeling of Rigid-Body Dynamics for Localization, Mapping, and Parameter Estimation of a Spinning Object in Space," *Journal of Field Robotics*, Vol. 32, No. 6, 2015, pp. 897–933. https://doi.org/10.1002/rob.21548

[9] Kaess, M., Ranganathan, A., and Dellaert, F., "iSAM: Incremental Smoothing and Mapping," *IEEE Transactions on Robotics*, Vol. 24, No. 6, 2008, pp. 1365–1378. https://doi.org/10.1109/TRO.2008.2006706

[10] Fourie, D., Tweddle, B. E., Ulrich, S., and Saenz-Otero, A., "Flight Results of Vision-Based Navigation and Control for Autonomous Spacecraft Inspection of an Unknown Object," *Journal of Spacecraft and Rockets*, Vol. 51, No. 6, 2014, pp. 2016–2026. https://doi.org/10.2514/1.A32813

[11] Grompone, A., "Vision-Based 3D Motion Estimation for On-Orbit Proximity Satellite Tracking and Navigation," Master's Thesis, Naval Postgraduate School, Monterey, CA, 2015.

[12] He, Y., Liang, B., He, J., and Li, S., "Non-Cooperative Spacecraft Pose Tracking Based on Point Cloud Feature," *Acta Astronautica*, Vol. 139, Oct. 2017, pp. 213–221. https://doi.org/10.1016/j.actaastro.2017.06.021

[13] Nassir, W. O., and Giorgio, P., "3D Point Tracking and Pose Estimation of a Space Object Using Stereo Images," *21st International Conference on Pattern Recognition*, Inst. of Electrical and Electronics Engineers, New York, 2012, pp. 796–800.

[14] Sharma, S., and D'Amico, S., "Pose Estimation for Non-Cooperative Spacecraft Rendezvous Using Neural Networks," *29th AAS/AIAA Space Flight Mechanics Conference*, AAS Paper 19-350, 2019.

[15] Su, H., Qi, C. R., Li, Y., and Guibas, L. J., "Render for CNN: Viewpoint Estimation in Images Using CNNs Trained with Rendered 3D Model Views," *IEEE International Conference on Computer Vision*, Inst. of Electrical and Electronics Engineers, New York, 2015, pp. 2686–2694. https://doi.org/10.1109/iccv.2015.308

[16] Tulsiani, S., and Malik, J., "Viewpoints and Keypoints," *IEEE Conference on Computer Vision and Pattern Recognition*, Inst. of Electrical and Electronics Engineers, New York, 2015, pp. 1510–1519. https://doi.org/10.1109/cvpr.2015.7298758

[17] Shi, J.-F., Ulrich, S., and Ruel, S., "CubeSat Simulation and Detection Using Monocular Camera Images and Convolutional Neural Networks," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2018-1604, 2018. https://doi.org/10.2514/6.2018-1604

[18] Sharma, S., Beierle, C., and D'Amico, S., "Pose Estimation for Non-Cooperative Spacecraft Rendezvous Using Convolutional Neural Networks," *IEEE Aerospace Conference*, Inst. of Electrical and Electronics Engineers, New York, 2018, pp. 1–12. https://doi.org/10.1109/AERO.2018.8396425

[19] Deng, J., Dong, W., Socher, R., Li, L., Li, K., and Fei-Fei, L., "Imagenet: A Large-Scale Hierarchical Image Database," *IEEE Conference on Computer Vision and Pattern Recognition*, Inst. of Electrical and Electronics Engineers, New York, 2009, pp. 248–255. https://doi.org/10.1109/CVPR.2009.5206848

[20] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L., "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, Vol. 115, No. 3, 2015, pp. 211–252. https://doi.org/10.1007/s11263-015-0816-y

[21] Sharma, S., and D'Amico, S., "Neural Network-Based Pose Estimation for Noncooperative Spacecraft Rendezvous," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 56, No. 2, 2020, pp. 4638–4658. https://doi.org/10.1109/TAES.2020.2999148

[22] Cassinis, L. P., Fonod, R., Gill, E., Ahrns, I., and Fernandez, J. G., "CNN-Based Pose Estimation System for Close-Proximity Operations Around Uncooperative Spacecraft," *AIAA Scitech Forum*, AIAA Paper 2020-1457, 2020. https://doi.org/10.2514/6.2020-1457

[23] Sonawani, S., Alimo, R., Detry, R., Jeong, D., Hess, A., and Amor, H. B., "Assistive Relative Pose Estimation for On-Orbit Assembly Using Convolutional Neural Networks," *AIAA Scitech Forum*, AIAA Paper 2020-2096, 2020. https://doi.org/10.2514/6.2020-2096

[24] Park, T. H., Märtens, M., Lecuyer, G., Izzo, D., and D'Amico, S., "SPEED+: Next-Generation Dataset for Spacecraft Pose Estimation Across Domain Gap," *IEEE Aerospace Conference*, Inst. of Electrical and Electronics Engineers, New York, 2022, pp. 1–15. https://doi.org/10.1109/AERO53065.2022.9843439

[25] Park, T. H., and D'Amico, S., "SHIRT: Satellite Hardware-in-the-loop Rendezvous Trajectories Dataset," stanford.edu/zq716br5462, 2022.

[26] Ahmed, Z., Park, T. H., Bhattacharjee, A., Fazel-Rezai, R., Graves, R., Saarela, O., Teramoto, R., Vemulapalli, K., and D'Amico, S., "SPEED-UE-Cube: A Machine Learning Dataset for Autonomous, Vision-Based Spacecraft Navigation," *46th Rocky Mountain AAS Guidance, Navigation and Control Conference*, AAS Paper 24-027, 2024.

[27] Park, T. H., and D'Amico, S., "SPE3R: Synthetic Dataset for Satellite Pose Estimation and 3D Reconstruction," stanford.edu/pk719hm4806, 2024.

[28] Musallam, M. A., Rathinam, A., Gaudillière, V., de Castillo, M. O., and Aouada, D., "CubeSat-CDT: A Cross-Domain Dataset for 6-DoF Trajectory Estimation of a Symmetric Spacecraft," *Computer Vision—ECCV 2022 Workshops*, edited by L. Karlinsky, T. Michaeli, and K. Nishino, Springer Nature, Cham, Switzerland, 2023, pp. 112–126. https://doi.org/10.1007/978-3-031-25056-9_8

[29] Žbontar, J., and LeCun, Y., "Computing the Stereo Matching Cost with a Convolutional Neural Network," *IEEE Conference on Computer Vision and Pattern Recognition*, Inst. of Electrical and Electronics Engineers, New York, 2015, pp. 1592–1599. https://doi.org/10.1109/CVPR.2015.7298767

[30] Renteria-Vidales, O. I., Cuevas-Tello, J. C., Reyes-Figueroa, A., and Rivera, M., "ModuleNet: A Convolutional Neural Network for Stereo Vision," *Pattern Recognition*, edited by K. M. Figueroa Mora, J. Anzurez Marín, J. Cerda, J. A. Carrasco-Ochoa, J. F. Martínez-Trinidad,

and J. A. Olvera-López, Springer International, Cham, Switzerland, 2020, pp. 219–228.
https://doi.org/10.1007/978-3-030-49076-8_21

[31] Despond, F. T., "Non-Cooperative Spacecraft Pose Estimation Using Convolutional Neural Networks," Master's Thesis, Carleton Univ., Ottawa, 2022.
https://doi.org/10.22215/etd/2021-14867

[32] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A., "Going Deeper with Convolutions," *IEEE Conference on Computer Vision and Pattern Recognition*, Inst. of Electrical and Electronics Engineers, New York, 2015, pp. 1–9.
https://doi.org/10.1109/CVPR.2015.7298594

[33] Redmon, J., and Farhadi, A., "YOLO9000: Better, Faster, Stronger," *IEEE Conference on Computer Vision and Pattern Recognition*, Inst. of Electrical and Electronics Engineers, New York, 2017, pp. 6517–6525.
https://doi.org/10.1109/CVPR.2017.690

[34] Kingma, D. P., and Ba, J., "Adam: A Method for Stochastic Optimization," arXiv:1412.6980, 2014.

J. A. Christian
Associate Editor