Relative visual navigation based on CNN in a proximity operation space mission

A. D'Ortona, G. Daddi

Politecnico di Torino, DIMEAS, corso Duca degli Abruzzi 24, Torino

antonio.dortona@polito.it, guglielmo.daddi@polito.it

Keywords: Navigation, Space, CNN

Abstract. This article explores a solution utilizing a convolutional neural network (CNN) to simulate robust monocular visual navigation during proximity operations of a space mission, where a precise determination of relative pose is crucial for mission safety. This operation involves closely observing a spacecraft with a CubeSat under challenging illumination conditions. The methodology involves generating a dataset using Blender software and training a Mask-CNN with a ResNet-50 architecture to identify relevant features representing the target's 3D model. The dataset's ground truth is obtained through an inverse Perspective-n-Point (PnP) problem. Overall, this work provides valuable insights into the potential of deep learning-based visual navigation techniques for enhancing space mission operations.

Introduction

In modern times, digital cameras have become compact, precise, non-invasive, and affordable, which has led to their widespread use in vehicle and robot navigation. Over the years, various techniques have been developed for this purpose, with visual navigation being one of the most accurate ways to estimate position and attitude, also known as *camera pose* estimation.

Visual navigation has been extensively studied in the context of robotic space exploration, including the Mars exploration rovers in 2003. However, there has been a growing interest in applying visual-based navigation techniques for on-orbit servicing missions in recent years. This is especially important for automatic rendezvous operations, which require precise determination of relative pose to ensure safe mission completion.

The traditional approach of pose estimation involves multi-view geometry, which compares two or more consecutive frames finding a set of 2D/2D correspondences to determine the camera's movement as in (Fravolini, 2010).

These methods are mostly applied in conditions where no known target object is observed. On a space mission, during proximity maneuvers, a target object can be observed and, if the geometry is known, a single image is enough to estimate the camera pose. Classical single-image pose estimation methods aim to solve the *Perspective-n-Point (PnP)* problem. PnP is the problem of estimating the pose of a calibrated camera given a set of n 3D points in the world and their corresponding 2D projections in the image. Model-based methods use a wireframe 3D model of the target spacecraft to match with 2D features extracted from the image and estimate the relative pose. Non-model-based methods compare the in-flight image with a pre-stored database of images to estimate the pose without feature extraction (al., 2012). However, these approaches lack robustness due to low signal-to-ratio, extreme illumination conditions, and dynamic Earth background in space imagery.

Recent developments in computer vision have introduced deep learning for pose estimation. Deep learning-based pose estimation methods typically use *convolutional neural networks (CNNs)* to extract features from input images or sensor data and then use these features to estimate the object's pose. By leveraging large amounts of labeled training data, deep learning methods can learn complex relationships between input data and object poses, enabling them to achieve high accuracy even in challenging conditions such as low-light environments or cluttered scenes.

The objective of this paper is to explore a solution based on a CNN for simulating robust monocular visual navigation during a space mission's proximity operation, which involves closely observing a spacecraft with a CubeSat in critical illumination conditions. The optical sensor will be evaluated as a strong option in synergy with GPS data and IMU to calculate the relative position and attitude accurately. The final part of the mission involves a docking maneuver, where visual navigation is crucial, and hence, an accurate study of the technique is necessary.

Methodology

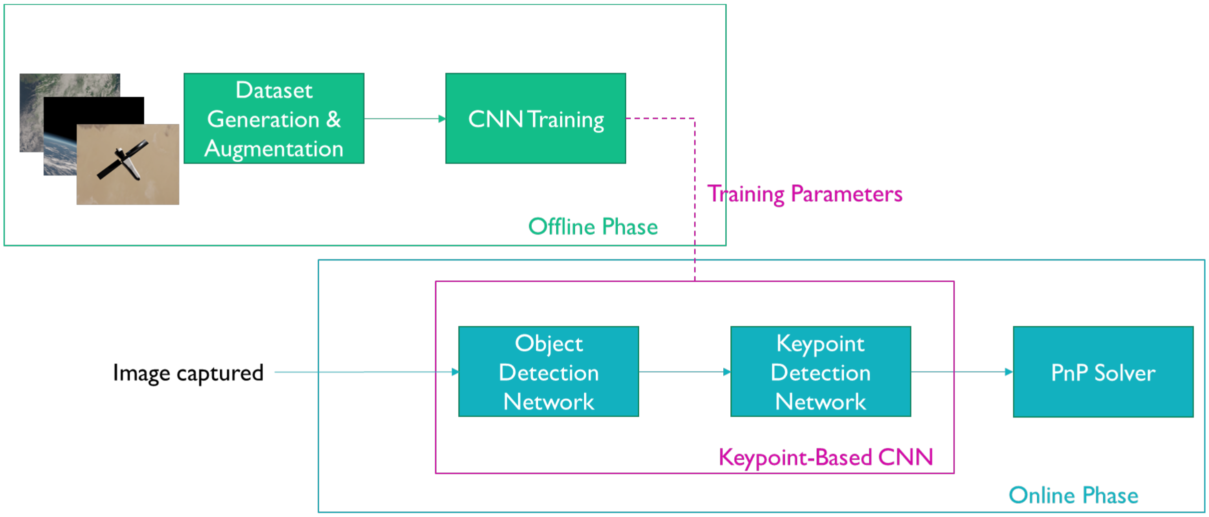

Figure 1 High-level architecture

The high-level architecture first involves the generation of the dataset using the 3D modeling software Blender. We placed in a blender environment a CAD model of SR and used blender’s built-in camera model to visualize the camera’s field of view, sensor size and resolution. Although this method is not physically or radiometrically accurate, it is a good way to test feature extraction algorithms under ideal circumstances. The environment is built by introducing a realistically scaled Earth, a moving sun, and trajectory import from STK or MATLAB simulations. However, the vast scale differences between Space Rider and Earth introduced core renderer issues that were addressed via workarounds. Three objects - terrain, atmosphere, and clouds - each with their own custom shader, made up the Earth. The terrain was composed of high-resolution satellite imagery, and a layer mask was used to increase roughness and create specular reflections on the water. The atmosphere was modeled to capture Rayleigh scattering and atmospheric pressure decay, which gives the sky its colour during daytime and tinge it red during sunset. The cloud layer imitated volumetric effects through a semi-transparent texture wrapped around the globe, and the sun mesh was placed according to the sun vector, with Blender's lens flare effects overlayed when visible from the camera. Blender's environment can read coordinate files in .csv form taken from STK or MATLAB and use them to procedurally place objects throughout the scene. Overall, this detailed environment provided a good testbed for feature extraction algorithms before introducing complications such as an illuminated moving background, real-world effects such as amplification and radiation noise, motion blur, bloom, or optic-induced blur.

The second component of a visual navigation simulator involves training a CNN to recognize relevant features in the images. In this case, a Mask-CNN with a ResNet-50 architecture is used (He, Gkioxari, Dollar, & Girshick, 2018). The Mask-CNN is a variant of the standard CNN that includes an additional output layer that predicts object masks. This is useful for tasks such as object segmentation, where the goal is to identify the pixels corresponding to specific objects in an image. In our case, the specific objects are 6 keypoint representing the 3D model of our target.

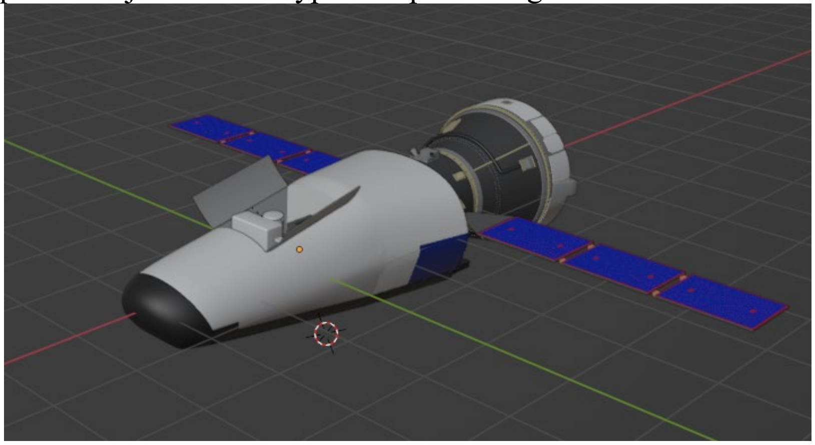

Figure 2 3D Model in Blender environment

The ground truth of the dataset (the 6 keypoints) is calculated through an inverse PnP problem; given the relative attitude and the relative position, the u and v coordinates of the keypoints on the image captured are automatically calculated:

$$(u, v) = \left(\frac{fX_{cam}}{Z_{cam}} + u_0, \frac{fY_{cam}}{Z_{cam}} + v_0\right)$$

$$\begin{bmatrix} fX_{cam} \\ fY_{cam} \\ Z_{cam} \end{bmatrix} = \begin{bmatrix} f & & u_0 \\ & f & v_0 \\ & & 1 \end{bmatrix} \begin{bmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \\ 1 \end{bmatrix} = \begin{bmatrix} f & & u_0 & 0 \\ & f & v_0 & 0 \\ & & 1 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \qquad (1)$$

In which X_{cam} and Y_{cam} are the coordinates of the keypoints in the camera reference frame, f is the focal length, u_0 and v_0 are the coordinates of the principal point of the camera, X Y and Z are the 3D coordinates in world coordinates frame (we considered a body reference frame).

The CNN architecture was designed with Pytorch framework on Python with a texture-randomized to increase accuracy and robustness. The Resnet 50 architecture is already implemented in the standard Pytorch libraries.

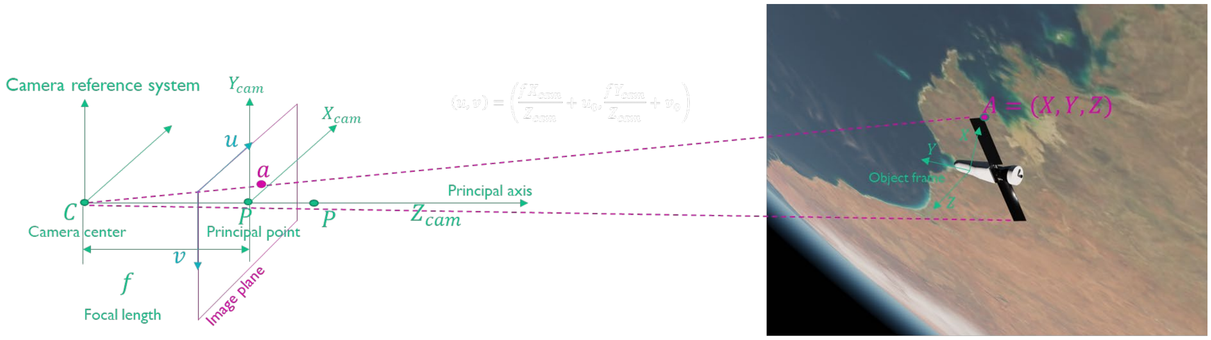

Figure 3 PnP problem depicted

The third component of a visual navigation simulator involves solving the PnP problem to find the relative pose of the target with respect to camera reference. The PnP problem is a classic computer vision problem that involves estimating the position and orientation of an object relative to a camera, given a set of 2D image points and their corresponding 3D points in the object coordinate system. In the context of visual navigation, the PnP problem is used to determine the position and orientation (R and t in

Equation 1) of the autonomous agent relative to the environment it is navigating in. This information is critical for the agent to make informed decisions on how to navigate through the environment safely and efficiently. The problem is solved with RANSAC (Random Sample Consensus) (Fischler, 1981) algorithm, which is robust to outliers and can handle noise in the data. The RANSAC algorithm works by randomly selecting a subset of 3D-2D point correspondences to estimate the camera pose.

Overall, the architecture of a visual navigation simulator is designed to enable the creation of realistic environments and provide the necessary tools for training and testing autonomous navigation algorithms. The use of realistic illumination conditions, the training of a CNN, and the solution of the PnP problem are all critical components of a visual navigation simulator that contribute to its effectiveness in training and testing autonomous navigation algorithms.

Results & discussion

The model was trained with 4500 synthetic images generated with Blender as explained in the previous section. The synthetic images are generated in a random position of our chaser in a range from 100 m to 10 m around the target. This dataset replicates the maneuver carried out which is an observation maneuver around the target, and the distance varies as it is a spiral-shaped maneuver with an elliptical-shaped base.

The learning rate is initially set to 0.001 and decays exponentially by a factor of 0.98 after every epoch. The network is trained on an NVIDIA GeForce RTX 3090 for 200 epochs. Our aim was to evaluate the performance of the model in predicting the pose of objects in the scene. We used two metrics to evaluate the error in the predicted pose: E_R for the quaternion error and E_T for the translation error (Sharma & D'Amico, 2019).

$$E_T = |\tilde{t} - t|$$

$$E_R = 2\arccos|q \ \cdot \tilde{q}| \qquad (2)$$

$$E_{TN} = \frac{|\tilde{t} - t|}{|t|}$$

Where t, q are the predicted unit quaternion and translation vector aligning the target body reference frame and the Camera frame, and t,q are the ground-truth unit quaternion and translation vector. ER corresponds to the angle of the smallest rotation that aligns q

and q. ETN is this distance normalized by the ground truth distance between the target and the camera. A final metric combines the two errors:

EC=ETN+ER

Table 1 CNN Scores

Metrics	**Score**
Mean E_T [m]	[0.2978 0.2131 0.3376]
Mean E_R [deg]	4.302
Mean E_C	0.1345

Table 1 reports the CNN's performances (Park, Sharma, & D'Amico, 2019) tested on a validation dataset of 100 synthetic images.

Overall, our results demonstrate the effectiveness of the Mask-CNN Resnet 50 model in accurately predicting the pose of objects in synthetic images. Specifically, the mean ET is about 50 cm, while the mean ER is around 4.3 degrees.

While initial results show promise, they are not as robust as those reported in seminal works in the field (Park, Sharma, & D'Amico, 2019) (Black, 2021). However, it is important to note that this work represents an early iteration in the development of new approaches to pose estimation using CNNs.

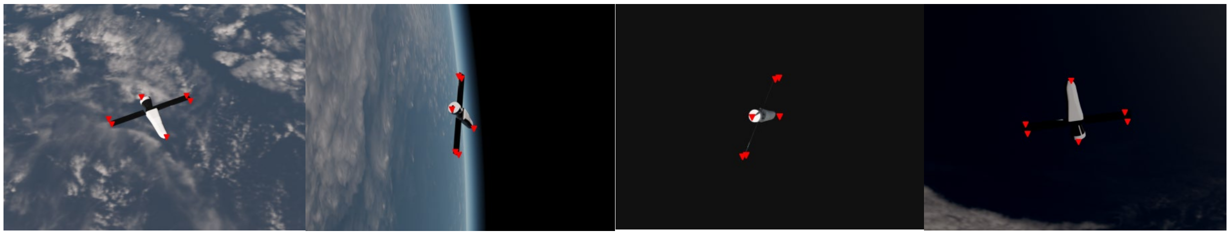

Figure 4 Keypoint detection results

Conclusion

In conclusion, this paper presents a solution based on a convolutional neural network (CNN) for simulating robust monocular visual navigation during a space mission's proximity operation. The proposed approach involves the generation of a dataset using 3D modeling software and training a Mask-CNN with a ResNet-50 architecture to recognize 6 keypoint features in images. The ground truth of the dataset is calculated through an inverse Perspective-n-Point (PnP) problem. The CNN's output is used to estimate the relative pose of a target spacecraft with respect to a CubeSat using a monocular camera in critical illumination conditions.

The proposed approach has several advantages over traditional methods for visual navigation. Deep learning-based pose estimation methods can learn complex relationships between input data and object poses, enabling them to achieve high accuracy even in challenging conditions such as low-light environments or cluttered scenes. The use of a monocular camera in critical illumination conditions reduces the complexity and cost of the system while maintaining high accuracy. The proposed approach can be used for automatic rendezvous operations, which require precise determination of relative pose to ensure safe mission completion.

The present findings underscore the need for further research and refinement of the proposed method to achieve more robust and accurate results. Expanding the dataset, improving its quality evaluating the use of other software and adjusting training parameters may hold promise as solutions to the current limitations of the method.

Another future work involves the integration of the proposed approach with GPS data and Inertial Measurement Unit (IMU) data to calculate the relative position and attitude accurately. Overall, the proposed approach shows promise for improving visual navigation in space missions and could be an essential tool for future on-orbit servicing missions.

References

[1] al., D. A. (2012). Solving the PnP Problem for Visual Odometry – An Evaluation of Methodologies for Mobile Robots. *Conference: Conference Towards Autonomous Robotic Systems*, 451-452. https://doi.org/10.1007/978-3-642-32527-4_54

[2] Black, K. &. (2021). Real-Time, Flight-Ready, Non-Cooperative Spacecraft Pose Estimation Using Monocular Imagery.

[3] Fischler, M. A. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM 24.6*, 381-395. https://doi.org/10.1145/358669.358692

[4] Fravolini, S. F. (2010). A Robust Monocular Visual Algorithm for Autonomous Robot Application. *IFAC Proceedings Volumes 43.16*, 551–556. https://doi.org/10.3182/20100906-3-IT-2019.00095

[5] He, K., Gkioxari, G., Dollar, P., & Girshick, R. (2018). Mask R-CNN. *CoRR*. https://doi.org/10.1109/ICCV.2017.322

[6] Park, T. H., Sharma, S., & D'Amico, S. (2019). Towards Robust Learning-Based Pose Estimation of Noncooperative Spacecraft. *ArXiv*.

[7] Sharma, S., & D'Amico, S. (2019). Pose estimation for non-cooperative spacecraft.