Celestial Navigation in Outer Space Using Deep Learning

Nirav Madhani
Case Studies in Machine Learning
12/3/23

Contents

Introduction 1
Literature Review and Related Work 1
Some Important Definitions 1
Related Work 2
Our Contribution 3
Scope and Problem Definition 3
Methodology 4
Defining Co-ordinate System 4
Preparing dataset 5
Accuracy of Generated Dataset 7
Model Architecture 8
Training 9
Results 10
Future directions 11
Conclusion 11
Appendix 12
Calculation for pixel shift 12
Obtaining x,y,z of star from RA and DEC 12
References 12

Introduction

This paper delves into the realm of celestial navigation, a time-honored technique that has been central to human exploration since the dawn of seafaring. Originating from the mariners' need to determine their position using celestial bodies like the sun, moon, stars, and planets, this method has been a cornerstone in navigating Earth's vast oceans and has significantly shaped human history. The primary focus of this research is to extend the traditional principles of celestial navigation into the context of outer space exploration, employing deep learning as a transformative tool. While the fundamental techniques of measuring angles between celestial objects and the horizon have largely remained unchanged, this paper proposes an innovative approach that presents amalgamation of these age-old methods with advanced deep learning algorithms. By doing so, this paper aims to demonstrate the potential of this synthesis in enhancing the accuracy and effectiveness of deep learning for celestial navigation, thus offering a modern style to an ancient practice, and opening new horizons in our journey beyond Earth.

Literature Review and Related Work

Some Important Definitions

1. **Celestial navigation**: Celestial navigation, alternatively termed astronavigation, constitutes the scientific method of precise positional determination by utilizing celestial bodies (Wikipedia Contributors). This technique enables navigators to determine their exact physical coordinates in space or on Earth's terrestrial surface, circumventing reliance solely on bootstrapping the estimations using dead reckoning.
2. **Sextant**: A navigational instrument used to measure the angle between any two visible objects, particularly useful for celestial observations.

3. **True North: True north** (also called **geodetic north** or **geographic north**) is the direction along Earth's surface towards the place where the imaginary rotational axis of the Earth intersects the surface of the Earth. That place is called the True North Pole. (True north)
4. **Azimuth**: The angle between a celestial body and the observer, measured clockwise from the true north.
5. **Altitude**: The angular distance of a celestial body above the horizon.
6. **Celestial Sphere**: An imaginary sphere of arbitrarily large radius, concentric with the Earth, on which all celestial bodies are considered to lie.
7. **Sidereal Time:** Timekeeping based on the Earth's rate of rotation measured relative to the fixed stars.
8. **Dead reckoning:** It is the process of calculating the current position of a moving object by using a previously determined position, or fix, and incorporating estimates of speed, heading (or direction or course), and elapsed time.
9. **Orbital Plane:** It is a flat surface formed by the path a celestial body takes while orbiting another body, like a planet orbiting the Sun in our solar system. It's a two-dimensional representation that helps describe and visualize the motion of objects like planets or comets within the solar system by defining the geometric plane created by their orbit.
10. **Space Prob:** A space probe (commonly referred as prob in context) is an unmanned spacecraft sent into space to explore and collect data about planets, moons, asteroids, comets, or other celestial bodies, using scientific instruments to conduct experiments and gather valuable information for scientific research.

Related Work

Deep Learning-based Spacecraft Relative Navigation Methods: A Survey (Song, Rondao and Aouf) is a comprehensive survey investigates deep learning-based autonomous spacecraft relative navigation methods. It focuses on specific orbital applications like spacecraft rendezvous and landing on celestial bodies, including the Moon and other small bodies with focus on “Relative Navigation” methods. Relative navigation in space exploration involves precisely determining the position and movement of a spacecraft in relation to another object, enabling maneuvers like docking or maintaining specific formations during missions in space. Their work delves into tasks like crater detection, hazard detection and terrain Navigation. **Celestial Navigation in Deep Space Exploration Using Spherical Simplex** (Zhao, Ge and Zhang) enhances the accuracy of celestial navigation in deep space through an improved filtering algorithm, the spherical simplex unscented particle filter (SSUPF). The Spherical Simplex Unscented Particle Filter is an advanced filtering method that integrates the principles of the Unscented Kalman Filter, a technique designed to estimate the state of a non-linear system by approximating probability distributions using representative points (sigma points) and their weighted measurements, enabling accurate estimation even in the presence of non-linearities and uncertainties. In this variation, the particles used for representing the probability distribution are confined to a hypersphere, enhancing the filter's performance and robustness in handling complex systems. **A Novel Autonomous Celestial Integrated Navigation for Deep Space** (Chen, Sun and Zhang) discusses the unique navigation performance requirements for deep space exploration missions, considering the complex environmental factors like long flight distances and high communication delays. It proposes stellar spectral method for accurate velocity estimation and uses amalgamation of obtained velocity along with angle measurements to yield high accuracy. The stellar spectral method for accurate velocity estimation involves analyzing the Doppler shift in a star's spectral lines—specific wavelengths of light emitted by elements within the star. By measuring the shift towards the red or blue end of the spectrum, the program calculates the star's radial velocity relative to the object, allowing them to determine its motion along the line of sight. **Toward Automated Celestial Navigation through Deep Learning** (Tozzi) proposes utilization of Deep Regression in ***Marine Celestial Navigation***, eliminating the need for estimated positions such as in Dead Reckoning. The primary objective is to investigate potential architectures capable of deducing location solely from any random image of the sky, where azimuth and elevation remain constant while the time is known.

Our Contribution

In the existing landscape of research, celestial navigation has predominantly been examined within the context of **marine navigation** (on the surface of earth). There is some research on using Deep Learning of Navigation in space, but it is restricted to navigation related to an anchor body (**relative navigation**) like moon or asteroid and uses LIDAR for mapping surface. Other research that covers celestial navigation using image processing depends on complex methods like **SSUPF** and **SSUKF** which are very complex to implement and are prone to error as noted in the original work. Ultimately, there is no research that proposes solely using Camera input from space prob along with deep learning to estimate coordinates of the probe in space. Our study seeks to address this void by introducing an approach that relies solely on camera data from space probes coupled with deep learning for the estimation of coordinates.

Presently, interplanetary space probes determine their locations with the assistance of the **D**eep **S**pace **N**etwork (NASA, JPL). The **DSN**'s antennas function as the primary measurement system for these probes, transmitting radio signals to the spacecraft (Jones). Upon reception, the probe slightly modifies the frequency of these signals before sending them back to the ground station. Analyzing the difference between the transmitted and received signals allows for the precise calculation of the probe's distance and velocity along the antenna's path. This accuracy is facilitated by the high-frequency nature of the signals and the use of an exceptionally precise atomic clock to measure these subtle frequency alterations.

As the probe ventures farther from Earth, there is an increased latency between the transmission and reception of signals, posing challenges in real-time coordinate measurements. Consequently, the probe often resorts to Dead Reckoning methods and heavily relies on knowledge of its state, which is derived from previously known states. This reliance may result in cumulative errors over time between signal transmissions and receptions, potentially jeopardizing missions and incurring substantial financial losses.

Utilizing Celestial Navigation-based position estimation offers a solution for interplanetary probes to be more certain about their real-time positions regardless of their distance from Earth or the availability of DSN antennas. This approach ensures more reliable positioning without solely depending on signal transmissions and receptions.

Scope and Problem Definition

In this research, we primarily utilize RGB camera images as input to determine the coordinates of a space probe, framing this as a **multivariate regression problem** from a machine learning perspective. We propose the assumption that the probe is situated in an orbital plane, an assumption closely aligned with reality for several reasons.

The trajectories of space missions typically conform to the plane of the planets' orbits around the Sun, facilitating efficient launch opportunities that minimize both fuel consumption and travel time. Optimal transfer orbits, such as the Hohmann transfer orbit, are generally situated near the ecliptic plane, offering the most fuel-efficient routes between celestial bodies. Furthermore, being in proximity to this plane simplifies the planning of gravitational assists from planets or other bodies, aiding in trajectory adjustments without excessive fuel expenditure. These trajectories also enhance predictability, stability, and facilitate smoother communication between spacecraft and Earth-based control systems, optimizing navigation and operational efficiency throughout interplanetary missions.

To further refine the problem, we assume that the probe possesses only one degree of freedom for rotation around an axis perpendicular to the orbital plane. Consequently, the output will be a 3-Dimensional Vector, yielding the x and z position and y orientation in a left-hand coordinate system (Weisstein), with the x-z plane aligned to the orbital plane. This assumption streamlines the complexity of the problem, allowing us to generate meaningful results with limited computational resources.

Methodology

Defining Co-ordinate System

The **Equatorial Coordinate System** is a fundamental celestial reference framework utilized in astronomy to precisely locate celestial objects in the sky. It operates akin to Earth's latitude and longitude system, providing a standardized method to pinpoint positions of stars, planets, and other celestial entities. Central to this system are two key celestial features: Celestial Equator and Vernal Equinox

1. **Celestial Equator**: The celestial equator is an imaginary circle projected onto the celestial sphere, much like Earth's equator. It extends outward from Earth's equator into space and is positioned such that it intersects the sky's celestial sphere. This line divides the celestial sphere into northern and southern hemispheres, similar to how Earth's equator separates its hemispheres.

2. **Vernal Equinox**: The vernal equinox marks a crucial point in the sky where the celestial equator intersects the plane of the Earth's orbit around the Sun. It denotes the beginning of spring in the northern hemisphere and is an essential reference point for measuring Right Ascension in the Equatorial Coordinate System. This point's location remains fixed in the sky and serves as the starting reference for measuring celestial longitude or Right Ascension.

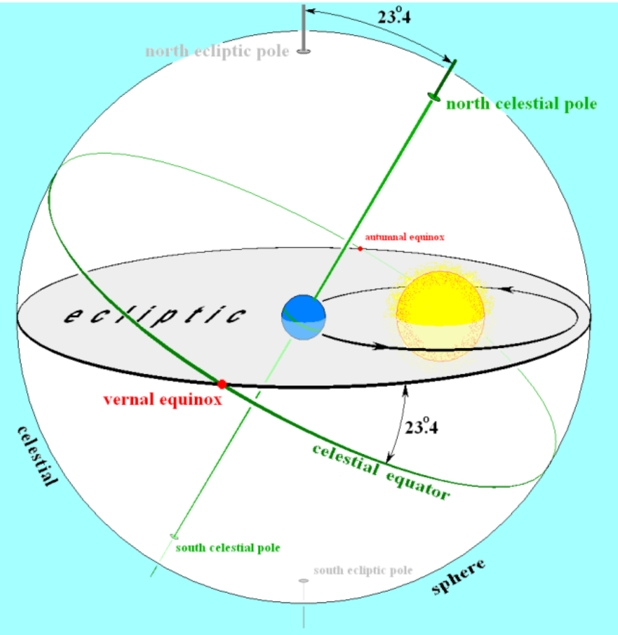


Figure 1 Diagram Explaining Celestial Equator and Vernal Equinox on Celestial Sphere. Source: (Tfr000)

Any point in night sky can be described using 2 coordinates, Right Ascension, and Declination which can be described after we understood vernal equinox and celestial equator.

Right Ascension (RA): Like terrestrial longitude, Right Ascension measures the angular distance eastward along the celestial equator from a defined point known as the vernal equinox. The vernal equinox is the point in the sky where the celestial equator intersects the plane of the Earth's orbit around the Sun, marking the beginning of spring in the northern hemisphere.

Declination (Dec): Comparable to terrestrial latitude, Declination measures the angular position of a celestial object north or south of the celestial equator. It serves as a reference point for vertical positioning in the celestial sphere.

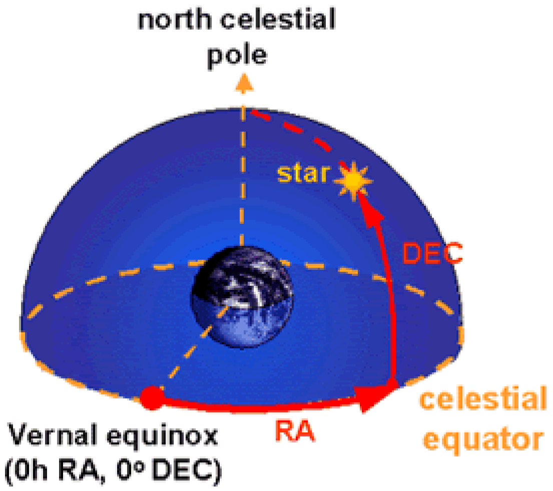
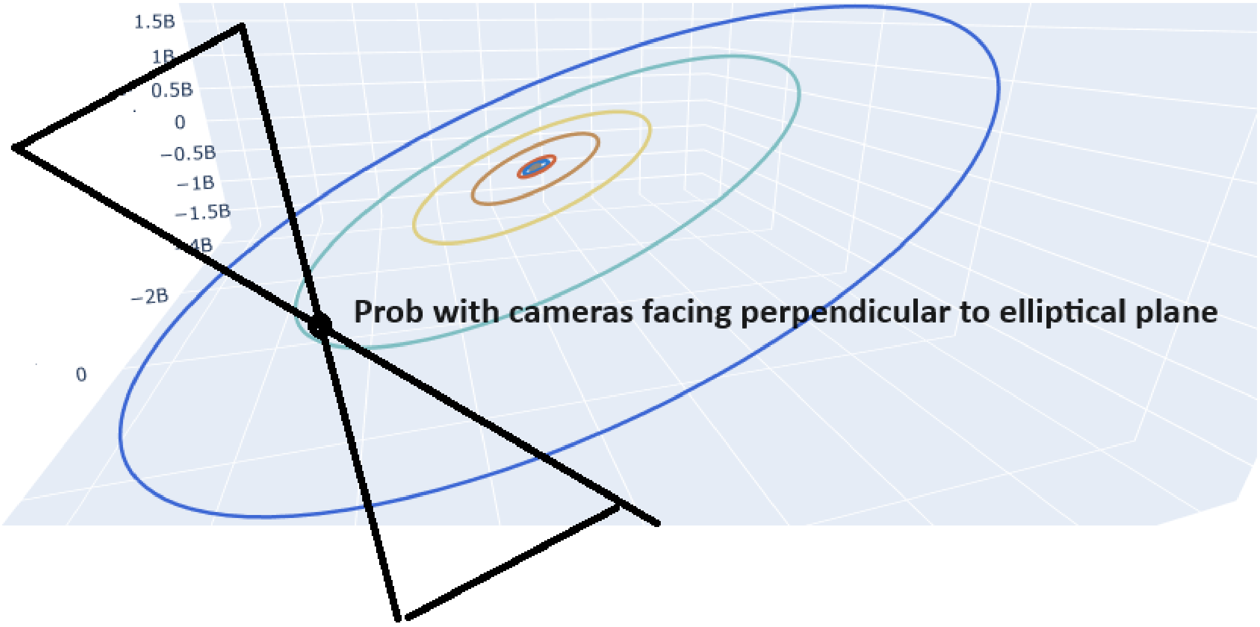


Figure 2: How RA and DEC are measured with respect to Vernal Equinox and Celestial Equator. Source: (Swinburne University of Technology)

The advantage of using equatorial coordinates is that they remain practically invariant over scope of time and space for space prob missions. RA and DEC vary by 1° over 72 years. If we consider camera resolution to be 1024px (twice of what we used) and FOV to be 90 degrees, position of star in pixels remains constant over period of 5 years as demonstrated by ***Calculation for pixel shift***. It could be mathematically proven RA and Dec co-ordinates of stars remain practically invariant over any point in solar system. Rather that providing complex mathematical proof here, we choose to provide a simple argument that *RA and DEC doesn't depend on time of year or position of earth around sun and hence it remains constant* over space. Therefore, this co-ordinate system simplifies representation of star's co-ordinates by removing the need for computing transformations over space and time. Beauty of the solution lies in the fact that even though RA and DEC remain similar, the image formed by parallax noticeably changes with position (Stellar Prallax), which enables for learning from images as evidenced in results section.

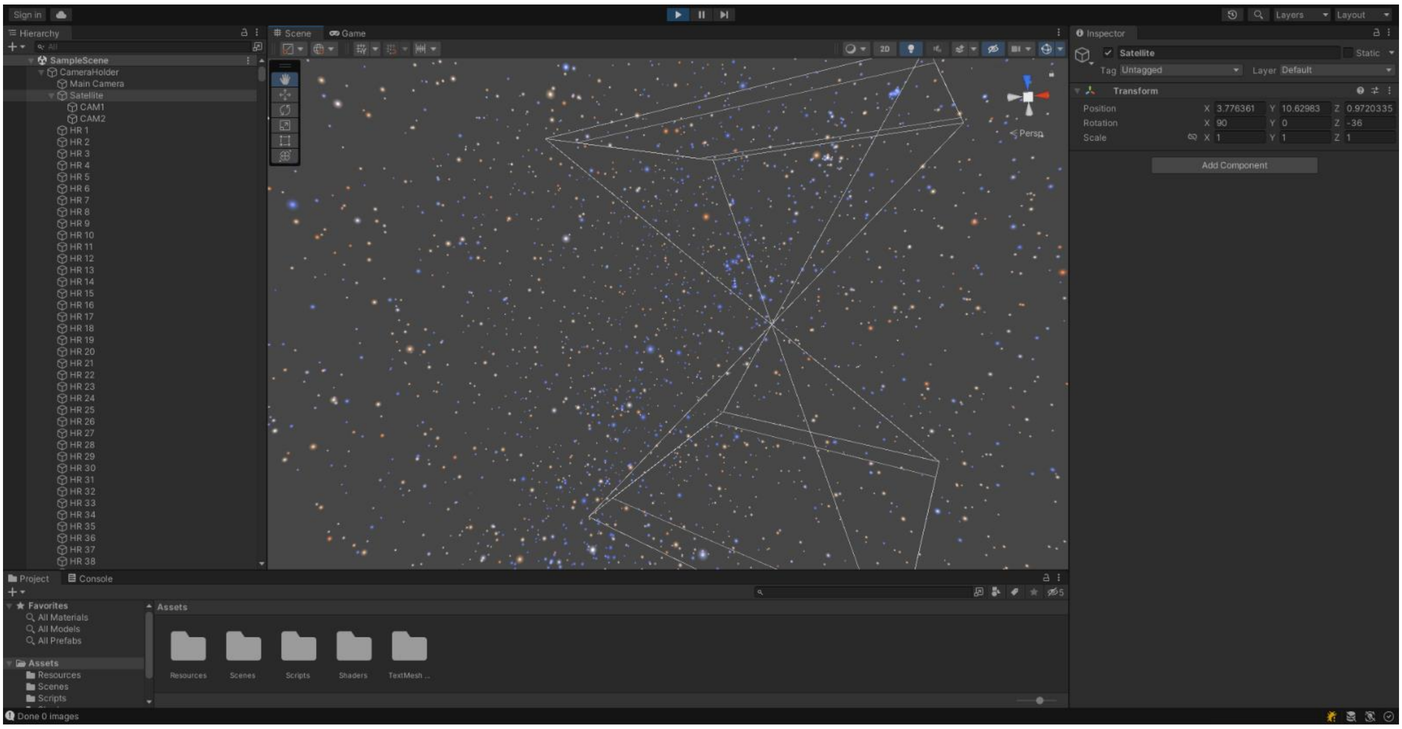
Preparing dataset

To prepare the dataset, we first simulate the view of outer space (stars) from the frame of view of solar system. We us Yale Bright Star CatLog (Hoffleit and Warren) as data source and Unity 3D for rendering the stars. Unity 3D is a versatile game development platform renowned for its capacity to create not only games but also sophisticated simulations. It offers developers an intuitive interface and comprehensive tools to design, code, and deploy interactive experiences for gaming, virtual reality, and complex simulations across diverse platforms. This makes Unity 3D a good choice for creating realistic and immersive simulations, allowing for the replication of real-world scenarios in fields like training, education, and engineering. To draw this points in Unity3D, we need to obtain *(x,y,z)* co-ordinates from RA and DEC. This can be achieved by using algorithm as described in ***Obtaining x,y,z of star from RA and DEC***. This algorithm is inspired the GitHub repository called Starry Sky by (Firnox) . Colors and brightness of stars are parsed according to (Harre and Heller). Other bright objects of solar system like Sun or planets are not rendered because in our arrangement cameras of prob are pointing in directions perpendicular to the **orbital plane**. It is important to mention that approach of using ***Stellarium*** (Zotti, Hoffmann and Wolf) used by (Tozzi) does not work in this case because images obtained from Stellarium are night sky images from any point on earth and are dependent on time and co-ordinate from where images are taken. Prob is placed at random co-ordinates in solar system along orbital plane with controlled (about y-axis) random orientation. Images are captured from 2 cameras of prob and its position and orientation is recorded. Each Image is 24-bit RGB of dimensions 512x512. The unity program is exported as standalone and can be called via command line to generate dataset. It accepts 'number of images' and 'offset' as parameter and for each datapoint it generates 2 images, one for top and bottom camera each and comma separated values for position and orientation. The offset option is used to parallelize generating of image by invoking multiple instances of unity application. Statistics of the dataset can be found in the table below. We have chosen the position co-ordinate to be between -100 to 100 where each unit represents 1 million km. Selection is appropriately made to reflect the scale of distance which could be expected in actual space probe missions. As seen in Figure 6, datapoints are uniformly distributed.



Prob with cameras facing perpendicular to elliptical plane

Figure 3: Representation of positioning of prob in solar system while generating datapoints



Satellite ☐ Static
Tag Untagged Layer Default
Transform
Position X 3.776361 Y 10.62983 Z 0.9720335
Rotation X 90 Y 0 Z -36
Scale X 1 Y 1 Z 1

Figure 4: Screenshot of how simulation looks in scene view of unity engine.

index	**x**	**z**	**theta**
count	10000.0	10000.0	10000.0
mean	-0.0182037608691001	0.26725655571533996	0.6352146190392
std	33.27355697611877	33.275819993915356	0.30890961894234
min	-99.37161	-99.41784999999999	0.0
25%	-18.947002500000004	-18.3740825	0.3826835
50%	-0.034250215	-0.06236553	0.7071068
75%	19.105345	19.170135000000002	0.9205049
max	99.04765	97.42924000000001	0.999962

Table 1: Stats of dataset points generated using Unity3D tool we designed

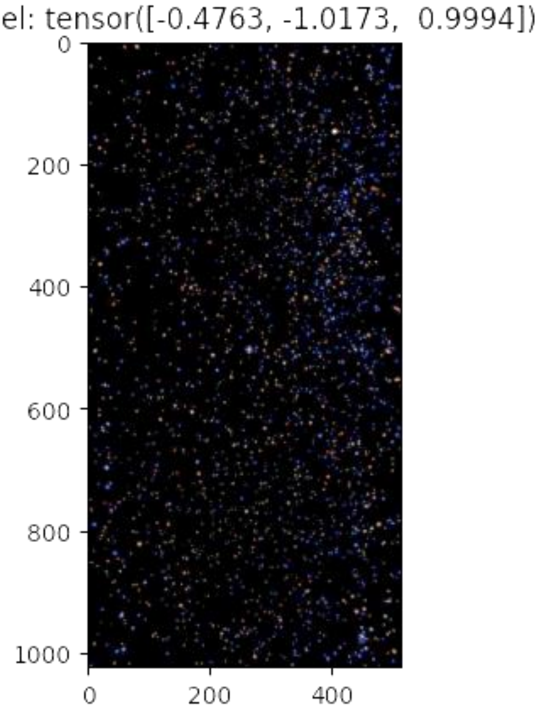


Figure 5: Sample Dataset point showing vertically stacked image along with output label (Contrast increased by 40% for better visibility.)

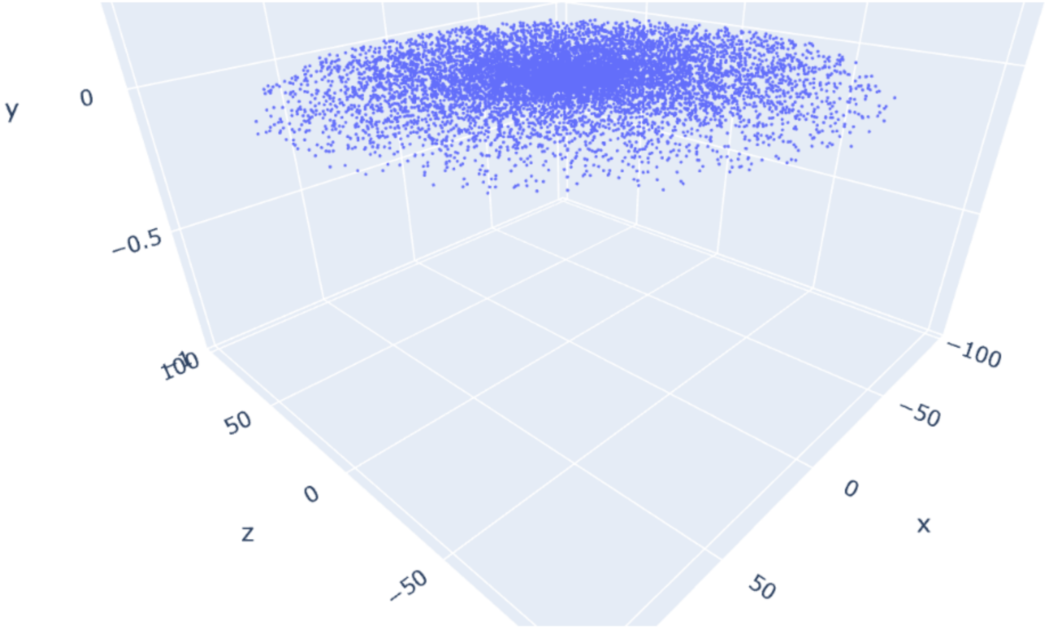


Figure 6:Plot of uniformly distributed dataset points after running unity program and importing dataset

Accuracy of Generated Dataset

One might argue that since we are using bright star catalog, we are not rendering dimmer star while creating our dataset which camera would see in real world. Since we have measure of how bright the stars in catalogue appear, we can set our camera shutter such that it does not capture fainter than certain intensity of light. Thus, stars not included in Yale Bright Star Catalogue are also not visible to camera in real world as well. One might ask how closely do these images reflect real world or whether this model can be used for real world scenarios. To answer these questions, we first look at existing works in literature that have used models trained on simulations and extended it to real world use cases. One domain specific example is **Transfer from Simulation to Real World through Learning Deep Inverse Dynamics Model** (Christiano, Shah and Mordatch) where model learns what action to take and in addition what is the next most likely-state and uses its knowledge to Dynamics to perform action. However, this approach only works for specified domain. Approach like **Transfer learning** as proposed by (Tercan, Guajardo and Heinisch) can work on wide variety of domains. In existing literature this idea has been extended further. What if simulation inaccuracy is another parameter to learn, for example if we cannot model friction between gripper and robot accurately, we take friction as something to be learned by robot based on its interaction with object. This is achieved by providing a range of friction value in simulation over single constant value which we think most accurately reality. This idea is called **Domain Randomization**. Authors of **Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World** (Tobin, Fong and Ray) have used domain randomization to train on simulated image and transfer skill to read image. This approach is very similar to what we are looking for. The question remains how inaccurate the model trained on simulation is and what causes inaccuracies to rise. The latter is very important to answer as it is the basis of implementing our domain randomization system. Without knowing what in simulation does not reflect the real world, we cannot randomize it. In our case, here are the items that introduce difference in simulation vs real world: -

1. **Parallax of stars**: We assigned an arbitrary large value to parallax for simplification which still provides accurate position to stars but does not account for relative movement of one star to another. This value needs to be properly calibrated.
2. **Intensity of star**: We have used simple linear approximations for brightness of star based on information from catalogue which is good approximation may not produce same image as those captured by real camera.

Although these techniques can tackle the problem of discrepancies between simulator and real world, they are not implemented here due to a variety of constraints. First, domain randomization requires an exponential increase in the amount of training data. The model needs to be able to capture significantly complex relationships and requires much more training. Domain randomization experiments need to be carefully engineered and observed as training progresses. It is important to address that though points of difference between simulation and reality are listed, there is no data or empirical result on how much these would affect model's accuracy. All in all, to computation and time constraints, and without suitable data to indicate need for requiring model to adapt to real world, it is not worthwhile to delve into it for scope of our paper.

Model Architecture

As noted above, the problem is multivariate regression problem. We are given 2 images and as input and we need to regress position and orientation of prob from the same. For multivariate regression multiple architectures can be considered. They can be categorized as:

1. **CNN-Based Architectures**:
   - AlexNet (Krizhevsky, Sutskever and Hinton)
   - VGG (Simonyan and Zisserman)
   - ResNet (He, Zhang and Ren)
2. **Transformers-Based Architectures**:
   - Vision Transformer (Dosovitskiy, Beyer and Kolesnikov)
   - Swin Transformer (Liu, Lin and Cao)
3. **Hybrid Architectures**:
   - ConvMixer (Trockman and Kolter)
   - EfficientNet (Tan and Le)

Among the architectures presented, Resnet (Resnet50) emerges as a particularly suitable choice for the multivariate regression problem being addressed. This preference is grounded in several key attributes of Resnet. Firstly, its utilization of residual blocks effectively counters the vanishing gradient problem, a common challenge in deep deep-learning models. Furthermore, the adaptability of Resnet is evident through its compatibility with transfer learning techniques, enabling seamless modifications tailored to our specific regression task. Empirical evidence, as highlighted in studies by (Ajal, Harikrishnan and Nai) and (Nair), reinforces Resnet's efficacy, demonstrating its efficacious performance in regression tasks. However, we do not posit ResNet as the unequivocally best choice for all (or even this) scenario(s). Rather, the objective is to substantiate that ResNet is a robust and capable option for the regression problem at hand, offering a blend of attributes that align well with the requirements of this specific task. Finding the best possible model architecture is outside scope of this work. First and last layer of ResNet are modified to implement transfer learning.

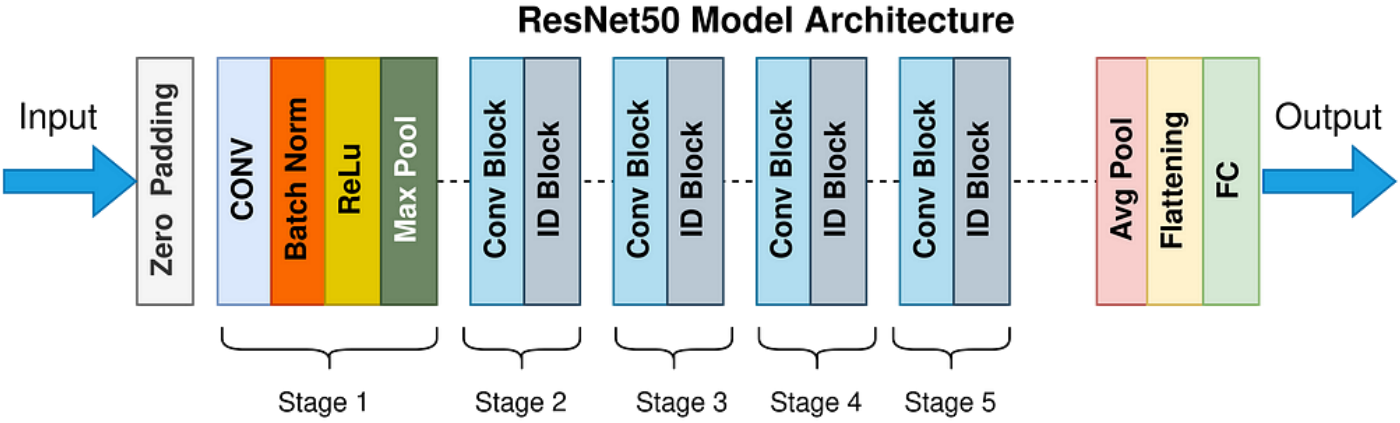


Figure 7: Architecture of ResNet50 model. Source: (Praveen)

First Layer	Conv2d(3, 64, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3), bias=False)
Last Layer	Linear(in_features=2048, out_features=3, bias=True)

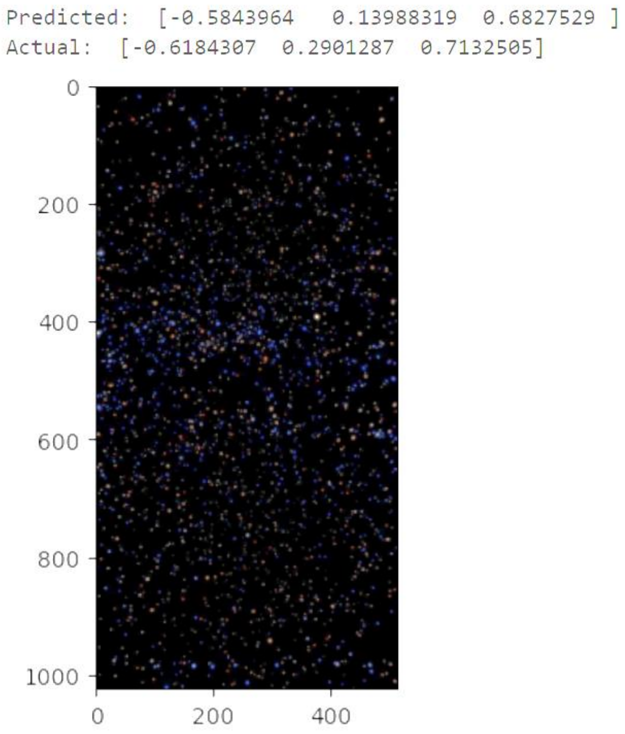
Table 2: Showing modification of first and last layer in Resnet50 Architecture for transfer learning.

Training

The training procedure begins by directly loading the dataset without applying any transforms as part of preprocessing steps, although the images need to be vertically stacked. As referenced previously, transfer learning method (Gupta, Pathak and Kumar) is used to speed-up learning. Mean Squared Error (MSE) is employed as the preferred loss function as it directly measures loss in the form of distance, providing a robust evaluation metric for regression models. Additionally, *Adam optimizer* is used due to advantages as it offers efficient gradient-based optimization, adapting learning rates for each parameter individually and often converging faster compared to other optimization algorithms.

First, we validate that the dataset's images offer sufficient information for the model to accurately learn and predict positions and orientations. We confirm this by training model on 15 epochs and see observing training and validation losses. As evidenced from Figure 9, model intendeds learns as we have both training and validation loss steadily decreasing.

Following this, training is extended further to enhance the model's accuracy until it starts to overfit. Checkpoints are created after each epoch. Training / Validation Loss graph is monitored, and potential overfitting is identified by observing rise in validation loss. The selection of the most suitable checkpoint as the final model aims to mitigate overfitting risks and ensures the model's generalizability beyond the training data. This method is referred as early stopping. Other methods that could address overfitting are regularization and dropout (Salakhutdinov). However these methods *do not add much value when used with larger datasets* (Goodfellow, Bengio and Courville) and hence are not employed here.

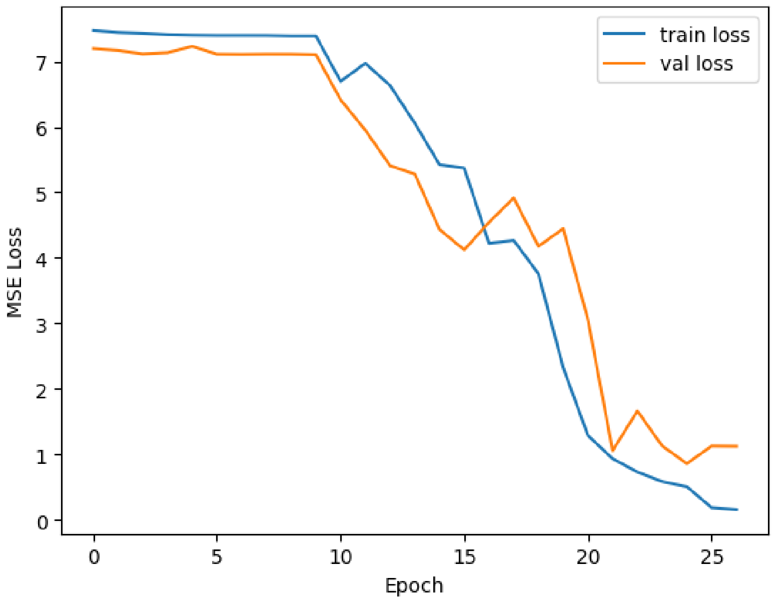


```
Predicted:  [-0.5843964   0.13988319  0.6827529 ]
Actual:  [-0.6184307  0.2901287  0.7132505]
```

Figure 8: Sample prediction on test dataset. (Contrast increase by 40% for better visibility)

Results

The model's learning progress, evident within the first 15 epochs, is highlighted by a decreasing trend in training and validation losses. Initially, the validation loss is notably lower than the training loss. This phenomenon could be attributed to several factors. Batch normalization, for instance, might introduce noise when the batch size is insufficient to represent the dataset adequately. Our batch size was limited to 12, constrained by GPU memory, in contrast to a validation set comprising 1500 images. Also, the use of pretrained weights in transfer learning may result in an initially lower validation loss, as these weights are often effective in making predictions. The Adam optimizer, adjusting learning rates based on the gradients' first and second moments, might also contribute to early improvements on the validation set. The lowest combined training and validation loss was observed at epoch 24, beyond which the validation loss started increasing. The Mean Squared Error (MSE) loss for the validation set at this point was 0.858. This is a significant amount considering the scale presented in the dataset and practical application. For example in our test datapoint, model predicted x position by offset of 30,000 km and z position by offset of 150,000 km. However, this accuracy can be improved by methods discussed in Future directions section. Overall, this research underscores the potential of using deep learning for celestial navigation in space, with room for improvements through advanced modeling and richer datasets.



.

Figure 9: Training and Validation loss while training (losses after epoch 26 truncated for better visiblity)

Future directions

This paper lays groundwork for potential application of Deep Image Regression using Prob Cameras for Celestial Navigation. Here, theory and mathematical concepts supporting the idea have been presented. We only used input from Camera to regress co-ordinates. But in practice, we can augment our input with previous position data and images of outer space taken during that time. This can provide a richer set of information to model which can yield more accurate results. It can work something similar to example of work by (Christiano, Shah and Mordatch). Here, only data from Yale bright star catalog is used which can be further augmented with other data to obtain more accurate parameters like parallax. Some other source that could be used are **Henry Draper Catalogue** (Cannon and Pickering)**, The HIPPARCOS catalogue** (Perryman, Lindegren and Hoeg), **The Guide Star Catalogue** (Morrison, Roser and McLean) ,etc. Here, we used (x,z) coordinates and (y) orientation for learning, but this can be extended to learning coordinates along all 6 Degrees of Freedom as well as linear and orbital velocity of prob. More complex architectures such as ResNet-101, ResNet-152, Inception-v4, or EfficientNet-B7 could be considered. These advanced models, with their greater depth and sophisticated design, are likely to more effectively capture the complexities we introduced. Additionally, the scalability of our dataset generation approach presents a significant advantage. Given that the dataset is generated through simulation, it is feasible to expand its size substantially, even by 10 times, providing a much more diverse and extensive training environment. This increase could lead to a more robust and accurate model, better suited for practical applications in space navigation. Model can be tested on existing infrastructure (for example cameras on ISS) we can collect data of accuracy of model. Finally, if needed things like Domin Randomization and transfer learning can be introduced to practical deployment of model.

Conclusion

This research has innovatively showcased the potential of deep learning and celestial navigation in the context of deep (outer) space exploration. By focusing on celestial navigation through camera inputs and multidimensional regression, we extended a way for extending use classical celestial navigation methods performed on earth to outer space, transcending dependence on the Deep Space Network (DSN) and combating the limitations of long-distance signal transmission and reception.

Our methodology, grounded in the Equatorial Coordinate System, facilitates a pragmatic approach to celestial navigation. The decision to forgo domain randomization was rooted in computational and time constraints, prioritizing practical feasibility in this initial phase. The ResNet architecture, selected for its proficiency in handling multi-dimensional regression problems, showcased promising results. Despite the challenge of achieving practical-level accuracy due to the complexity and novelty of the task, the research confirmed the viability of celestial navigation through camera inputs in outer space. The model's learning trajectory, as evidenced by the training and validation loss patterns, underscores its potential for further refinement and application. This work lays a strong foundation for future advancements. The integration of additional data sources, the exploration of all six Degrees of Freedom, and the testing of the model in real-world scenarios, are proposed as logical next steps. The application of **domain randomization** and **transfer learning techniques** could bridge the gap between simulation and real-world application, enhancing the model's accuracy and practical utility.

In conclusion, this research not only demonstrates the feasibility of celestial navigation using deep image regression but also opens avenues for more sophisticated, accurate, and autonomous navigation systems for future space exploration endeavors.

Appendix

Calculation for pixel shift

Here we consider simple cases of deviation along either horizontal or vertical axis. Any case between these 2 extremes will result in deviation less than obtained deviation.

- Angular deviation in 1 year $= \frac{1}{72}$ degrees.
- Image Width / Height = 1024 px.
- Total pixels in 90 degrees $=$ 1024 px.

We calculate the total pixel shift in $\frac{1}{72}$ degrees as follows:
Pixel shift in $\frac{1}{72}$ degrees $= \left(\frac{1}{72}\right) \times \frac{1024\text{px}}{90}$

Solving for the pixel shift:

$$\text{Pixel shift } = \frac{1024}{72 \times 90} \approx 0.1580\text{px}$$

Then, the total pixel shift in 5 years is:
Total pixel shift in 5 years $= 0.1580 \times 5 = 0.7900\text{px}$

Obtaining x,y,z of star from RA and DEC

1. Input: Right Ascension (α), Declination (δ)
2. Set Scale factor S to an arbitrarily large number.
3. $x = \cos(\alpha) \cdot \cos(\delta)$
4. $y = \sin(\delta)$
5. $z = \sin(\alpha) \cdot \cos(\delta)$
6. Return vector $\mathbf{f} = (x, y, z)$
   Here, α represents the right ascension and δ represents the declination. The function returns the Cartesian coordinates as a vector **f**.

References

Ajal, Sunday, et al. "Comparing machine learning and deep learning regression frameworks for accurate prediction of dielectrophoretic force." *Scientific Reports*. Nature, n.d.

Cannon, Annie Jump and Edward Charles Pickering. *The Henry Draper Catalog*. The Observatory, 1918.

Chen, Xiao, et al. "A Novel Autonomous Celestial Integrated Navigation for Deep Space Exploration Based on Angle and Stellar Spectra Shift Velocity Measurement." *Sensors (Basel)* (2019).

Christiano, Paul, et al. "Transfer from simulation to real world through learning deep inverse dynamics model." *Robotics* (n.d.).

Contributors, Wikipedia. *Stellar Prallax*. 4 November 2023. 28 November 2023. <https://en.wikipedia.org/wiki/Stellar_parallax>.

Dosovitskiy, Alexey, et al. "An image is worth 16x16 words: Transformers for image recognition at scale." *arXiv preprint arXiv:2010.11929* (2020).

Firnox. "Starry Sky." n.d. *GitHub*. 15 11 2023. <https://github.com/Firnox/StarrySky>.

Goodfellow, Ian, Yoshua Bengio and Aaron Courville. "REGULARIZATION FOR DEEP LEARNING." Goodfellow, Ian, Yoshua Bengio and Aaron Courville. *Deep Learning (Adaptive Computation and Machine Learning series)*. 2016. 265.

Gupta, Jaya, Sunil Pathak and Gireesh Kumar. "Deep learning (CNN) and transfer learning: a review." *Journal of Physics: Conference Series* (2022).

Harre, Jan-Vincent and René Heller. *Digital color codes of stars*. 11 January 2021.

He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016): 770-778.

Hoffleit, D. and Jr., W.H Warren. *Bright Star Catalogue*. Yale Bright Star Catalog, 1991.

Jones, Jeremy. "How do space probes navigate large distances with such accuracy and how do the mission controllers know when they've reached their target?" n.d. *Scientific American*. <https://www.scientificamerican.com/article/how-do-space-probes-navig/>.

Krizhevsky, Alex, Ilya Sutskever and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks." *Advances in Neural Information Processing Systems 25 (NIPS 2012)* (n.d.).

Liu, Ze, et al. "Swin transformer: Hierarchical vision transformer using shifted windows." *Proceedings of the IEEE/CVF international conference on computer vision* (2021): 10012-10022.

Morrison, JE, et al. "The guide star catalog, Version 1.2: An astrometric recalibration and other refinements." *The Astronomical Journal* (2001).

Nair, A., Lin, CY., Hsu, FC. et al. "Categorization of collagen type I and II blend hydrogel using multipolarization SHG imaging with ResNet regression." *Scientific Reports*. Nature, n.d.

NASA, JPL. *NASA*. n.d. 27 November 2023. <https://www.jpl.nasa.gov/missions/dsn>.

Perryman, Michael AC, et al. "The HIPPARCOS catalogue." *Astronomy and Astrophysics, Vol. 323* (1997): L49-L52.

Praveen, Gorla. "ResNet50.png." 21 October 2021. *Wikimedia Commons*. <https://commons.wikimedia.org/wiki/File:ResNet50.png>.

Salakhutdinov, Nitish Srivastava and Geoffrey Hinton and Alex Krizhevsky and Ilya Sutskever and Ruslan. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting." *Journal of Machine Learning Research}* (2014): 1929-1958.

Simonyan, Karen and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition." *arXiv preprint arXiv:1409.1556* (n.d.).

Song, Jianing, Duarte Rondao and Nabil Aouf. "Deep Learning-based Spacecraft Relative Navigation Methods: A Survey." *Acta Astronautica* (2023).

Tan, Mingxing and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks." *International conference on machine learning* (2019): 6105-6114.

—. "Efficientnet: Rethinking model scaling for convolutional neural networks." *International conference on machine learning* (2019): 6105--6114.

Technology, Swinburne University of. "Equatorial Coordinate System." n.d. *Study Astronomy Online at Swinburne University.*

Tercan, Hasan, et al. "Transfer-Learning: Bridging the Gap between Real and Simulation Data for Machine Learning in Injection Molding." *ScienceDirect* (n.d.).

Tfr000. "File:Earths orbit and ecliptic.PNG." 14 march 2012. *Wikimedia.org.* 29 Nov 2023.

Tobin, Josh, et al. "Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World." *Robotics* (n.d.).

Tozzi, Greg. *Toward Automated Celestial Navigation with Deep Learning*. Aug 2020. <https://github.com/gregtozzi/deep_learning_celnav>.

Trockman, Asher and J Zico Kolter. "Patches are all you need?" *arXiv preprint arXiv:2201.09792* (2022).

True north. 3 July 2023. 18 Nov 2023. <https://en.wikipedia.org/w/index.php?title=True_north&oldid=1163138156>.

Weisstein, Eric. *Left-Handed Coordinate System*. n.d. <https://mathworld.wolfram.com/Left-HandedCoordinateSystem.html>.

Wikipedia. *Celestial navigation*. 14 October 2023. <https://en.wikipedia.org/wiki/Celestial_navigation>.

Zhao, Fangfang, et al. "Celestial navigation in deep space exploration using spherical simplex unscented particle filter." *IET Signal Processing* (2018).

Zotti, Georg, et al. "The Simulated Sky: Stellarium for Cultural Astronomy Research." *Journal of Skyscape Archaeology* (2021).