18th International Conference on Space Operations, Montreal, Canada, 26 - 30 May 2025.

SpaceOps-2025, ID # 176

Machine Learning Techniques in Autonomous Localization Based on Surrounding Terrain Outlines

Mansour Abdoli[a]*, Charles Lee[a], Catania Calavitta[a], Lauren Helt[a], Ashley Jubb[a], Erika Luz[a], Ren Padgett[a], XinXue Wang[a]

[a] *Department of Mathematics, California State University, Fullerton, California, USA*
* Corresponding Author, *mabdoli@fullerton.edu*

Abstract

This study focuses on the challenge of predicting a rover's location on the lunar surface solely based on the outline of the surrounding terrain. This capability increases the reliability of positioning and navigation of autonomous rovers in the absence of GPS, direct-with-earth communications, and information from other positioning sensors, such as IMUs and Star-Trackers.

Our study considers predicting a rover's location on an n_x by n_y grid by comparing the rover's surrounding terrain outline with those pre-computed from DEM at each grid point. Using the correlation projection, a brute-force search provides good accuracy in predicting the rover's location, but it fails in three crucial performance measures: computational time, onboard memory, and storage.

This paper investigates the feasibility of using Machine Learning techniques to quickly and accurately determine the rover's position with minimal onboard memory requirement. Our study considers the following three strategies: (1) Feature Selection: to reduce data dimensionality (PCA and Fast Fourier Transform), (2) Response Space Reduction: through clustering (e.g., K-Means), and (3) Model Selection and Enhancement: finding efficient models for predicting position clusters, the first step in the way to build a more complex decision tree. The study evaluates different prediction methods, including Logistic Regression, LDA, QDA, and Convolutional Neural Network (CNN). Discussions on limitations imposed by high-dimensional data, evaluation processes, and prediction results in the absence and presence of noises are provided. The preliminary results have shown that CNN consistently outperforms other methods.

Keywords: Machine-learning techniques, autonomous surface localization, surrounding terrain outline, horizon matching technique, positioning without PNT satellites, star-trackers, IMU sensors.

Nomenclature

Lowercase letters like x, y, α, θ are scaler values, and i, j, k, l are used as indices.
Uppercase letters like A are used to represent matrices (arrays with 2 or more dimensions).
Bold lowercase letters like $\boldsymbol{\alpha}$ and $\boldsymbol{m}$ are used to represent vectors.
$< \boldsymbol{a}, \boldsymbol{b} >$ represents the dot product of two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$; i.e., $\boldsymbol{a} \cdot \boldsymbol{b}$.

Acronyms/Abbreviations

BF: Brute Force.
CNN: Convolutional Neural Network.
DEM: Digital Elevation Map.
FFT: Fast Fourier Transformation
GLM: Generalized Linear Model.
LDA: Linear Discriminant Analysis.
LM: Linear model, a more specific type of GLM.
NNet: Neural Network.
PCA: Principal Component Analysis
QDA: Quadratic Discriminant Analysis.
STO: Surrounding Terrain Outline, also called skyline, horizon line, surrounding horizon mask.

1. Introduction

The National Aeronautics and Space Administration (NASA) has a growing interest in exploring the southern pole of the moon, specifically the South Pole–Aitken (SPA) basin. The SPA basin is one of the largest and oldest impact basins on the moon, which may allow us to better understand the origins of life on the Earth. Lunar Endurance is a proposed NASA mission that plans to send an autonomous rover to transverse thousands of kilometers on the far side of the moon to collect samples and support scientific research at the SPA [1]. Vital to an autonomous rover is the ability to find its position (localization), which is challenging on the moon without a GPS-like system or direct-to-earth communications.

Autonomous localizing has been studied in different contexts, including traversing a planned path (path traversal) [2,3], exploring a designated area (scanning/mapping) [3], and cooperating with other autonomous agents [3]. Of those approaches applicable to initial localization, some rely on the surrounding train outline (STO) obtained from panoramic view photos, and others use surface photos. [4,5] used craters as landmarks (features) for matching surface images to satellite ones. [6,7] used surface images but did not limit their feature identification to craters. [8] addressed the spatial resolution differences between nearby and distant pixels on surface images. [2] used surface images to locate obstacles in the path of the rover.

This paper focuses on the use of STO in rover localizing. [9] uses MER rover panoramic photos to estimate the rover position and orientation by extracting its observed STO (skyline) and matching it to the set of DEM-based, pre-computed STOs. They use a brute-force approach and the Euclidean distance to find the best match. [2] uses the same approach to find the location of their rover on its pre-planned path. [10] propose a brute force approach like [9] but tests the model on Lunar Orbit Laser Altimeter (LOLA) DEM, uses the correlation between STOs (surrounding horizon mask) as the similarity measure, and tests the model under some noise on orientation and elevation read.

The common thread in localizing approaches that search for the best-matching STOs is the need for a search database. Pre-computing an extensive STO database reduces the computational load on the autonomous river; however, other restrictions may remain: the space for holding the database and the computational power for searching through it. [11] suggests using cubic splines to represent STOs (captured image curve and elevation map projection curve) and using a sliding window search on the normalized curvature of each piece of the STOs being compared.

In this study, we consider localizing a rover on a 12 km by 12 km area and demonstrate some of the challenges of initial localizing in a large region. This region is broken up using a grid of size 304x304 (40m per pixel), generating 92,416 position coordinates. We use the STO database generated for all points by calculating the elevation angle at azimuths 0, 1, …, 359° [10]. We explore different machine-learning classification approaches and their limitations in this context. We also introduce fast Fourier transformation (FFT) as a feature extraction technique and compare it to the PCA method.

2. Horizon-Based Position Discovery

In traditional localization methods, when a Positioning, Navigation, and Timing (PNT) system does not exist, the location of a rover is identified by comparing a panoramic photo of its surroundings (see Fig. 1) to a collection of aerial photos enriched with landmark identifications (see Fig. 2, for an example). This time-consuming process requires ground-in-the-loop interactions, making it impractical for autonomous rovers.

A common approach to localizing a rover is to (1) use the horizon line, i.e., the surrounding terrain outline (STO), get a panoramic photo, extract the rover's observed STO, and search a pre-constructed STO database to find the pixel of the best match. To formulate this approach, let $\boldsymbol{\alpha}_{x,y} = \left[\alpha_{x,y\phi}\right]_{\phi=\phi_1,\phi_2,\ldots,\phi_{n_z}}$ denote an STO, a set of horizon masks α measured in different azimuth directions ϕ in point (x, y) (see Fig. 3).

The STO database, a collection of horizon masks measured in n_z azimuth directions at each grid point (i, j) in a region of interest with a $n_x \times n_y$ grid can be presented as $\mathrm{A} = \left[\alpha_{x_i,y_j\phi_k}\right]_{n_x\times n_y\times n_z}$, where $i = 1,..,n_x$, $j = 1,\ldots,n_y, k = 1,\ldots,n_z$. A visual presentation of A is displayed in Figure 4. Fig. 5 shows a 2D representation of the STO database where a XY-position index represents the position of the grid point (x_i, y_j). The XY-position index is determined by:

$$XY\text{-Position Index} = i + (j-1)n_x \qquad (1)$$

where $i = 1,\ldots,n_x$ and $j = 1,\ldots,n_y$.

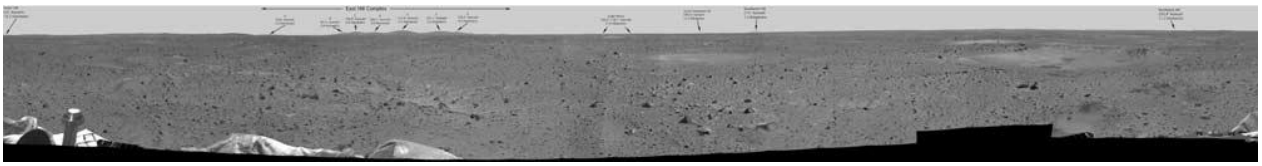

Fig. 1. A panoramic photo taken by a rover.

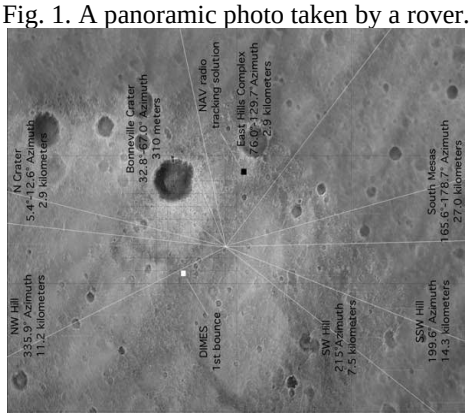

Fig. 2. Arial photo with identified landmarks.

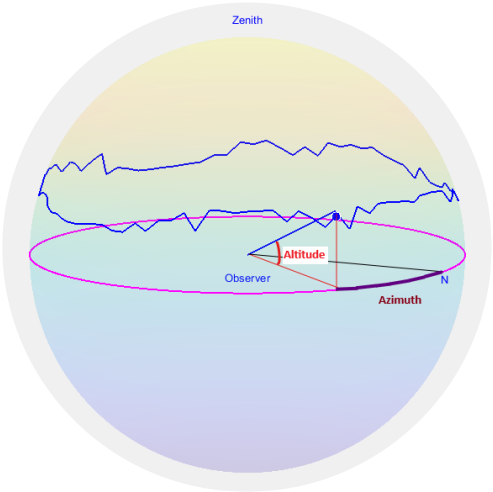

Fig. 3. Rover's STO is visualized in a topocentric coordinate system.

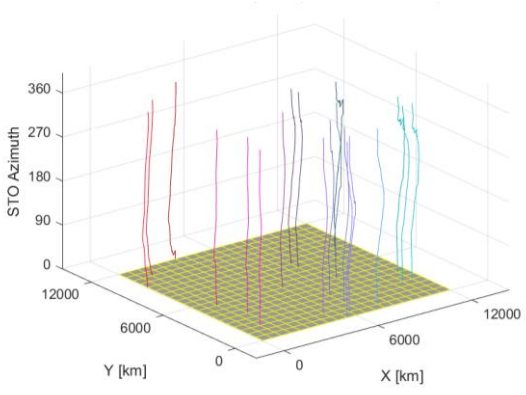

Fig. 4. The 3D view of 20 random STOs in an STO database on a 12km-by-12km region.

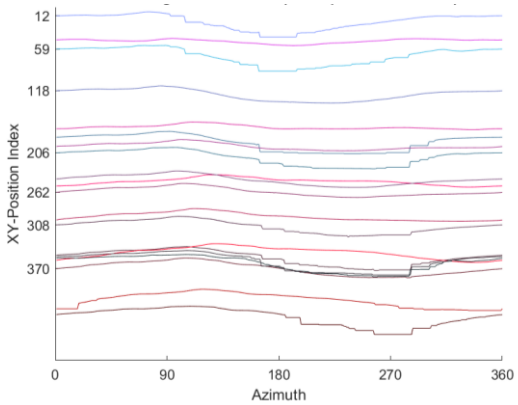

Fig. 5. The 2D representation of 20 random STOs in the STO database is shown in Fig. 4.

The rover's observed STO at point. (x_o, y_o) can be formulated by $\boldsymbol{m}_{x_o,y_o}(\theta_o) = \left[m_{x_o,y_o(\phi-\theta_o)}\right]_{\phi=\phi_1,\phi_2,\ldots,\phi_{n_z}}$, a mask generated from a panoramic photo at a point (x_o, y_o), and a direction deviation (horizontal orientation shift) of θ_o, which could be any number between 0° and 360°. Then, the brute force approach used in [10] predicts (x_o, y_o, θ_o) by minimizing the correlation (covariance) between the observed STO and all grid points in all directions:

$$\underset{i,j,\theta}{arg.min}\, f(x_i, y_j, \theta) = < \boldsymbol{\alpha}_{x_i,y_j},\ \boldsymbol{m}_{x_o,y_o}(\theta) > \qquad (2)$$

where $i = 1,..,n_x$, $j = 1,\ldots,n_y$ and $\theta = \phi_k - \phi_1$ for $k = 1,\ldots,n_z$.

This study considers a 12km-by-12km region, divided by a 304-by-304 grid into 92,416 pixels ($n_x = n_y = 304$). We consider $n_z = 360$, $\phi_1 = 0$, and $\phi_{360} = 359$ when constructing the STO database, which leads to this simplifying relation: $\theta = \phi_k = k - 1$.

The pre-computed STO database includes 92,416 entries and takes around 240MB of space in a MAT file format. If θ is unknown, searching the dataset means evaluating the inner product of each entry with the rover's observed STO 360 times. These requirements are only for 40m wide grid steps (pixels). As the grid steps are reduced to attain a reasonable accuracy, the database size and the computational requirement for searching it becomes increasingly unmanageable by a rover's onboard computer; for example, using a step size of 0.5m (as used in [2]) would lead to 2.1×10^{12} inner product calculations on a database that requires more than 1 TB of memory space.

3. Machine Learning Approach to Horizon-Based Position Discovery

The ultimate goal of using machine learning (ML) techniques to predict the location of a rover is to train a model with a smaller memory footprint and computational needs so that it can be loaded on the rover's computer. Such models can be extensively trained and tested using available data, an ideal case for *supervised learning* techniques (see [12] for an overview of ML techniques and a survey of ML methods used in localization). However, the choice of supervised ML techniques depends on the problem definition characterized by *response* (dependent variable) and data *features* (predictors or independent variables).

3.1 Response

One natural choice for the response variable is (x, y, θ), the real-valued vector of location and horizontal orientation of the rover. Multivariate regression models are the proper form of ML approaches for this choice. Considering the problem's high dimensionality and its non-linear nature, this study does not explore traditional multivariate regression models. We suggest using Neural Network models to explore localization with a real-valued multivariate response.

Another candidate for a response variable is (i, j, k), where $i = 1,\ldots,n_x$, $j = 1,\ldots,n_y$, and $k = 1,\ldots,n_z$. In this case, classification ML methods are more appropriate. Finding (i, j, k) in a classification problem is equivalent to finding a position index l in an equivalent 2D representation of the search database:

$$l = i + (j-1) * n_x + (k-1) * n_x * n_y \qquad (3)$$

where $l \in \{1, 2, 3, \ldots, n_x n_y n_z\}$. In this study, we considered the case that there is no orientation shift, i.e. $\theta = 0$, which simplifies the search and reduces the response space:

$$l = i + (j-1) * n_x \qquad (4)$$

where $l \in \{1, 2, 3, \ldots, n_x n_y\}$.

Finally, the position indices can be grouped into clusters based on the similarity of their STOs. The resulting clusters can be used as the response for the localizing, reducing the search database size albeit at the cost of diminishing the localization accuracy. However, this approach provides the basis for developing decision trees that gradually fine-tune the rover's location.

3.2 Features

The n_z horizon masks in $\boldsymbol{\alpha}_{x_i,y_j}$ are raw features in the STO database A. A large n_z increases a predictive model's computational need, making it sensitive to noise. The use of cubic spline to eliminate curvature noise in [11] is an example of feature extraction, where instead of a large number of raw (original) features, a smaller set of transformed features is used with limited loss of accuracy.

3.2.1 Principal Component Analysis (PCA)

We consider the PCA (Principal Component Analysis) approach to feature extraction, where raw features are linearly transformed into an orthogonal space from which a few critical transformed features, called PCA modes, are selected for use. The importance of the extracted features is determined by the variability explained by those features, which is also used to determine the optimal number of PCA modes to be used.

3.2.2 Fast Fourier Transformation (FFT)

We propose a novel approach of using the Fourier Transformation (FT) of STOs as a feature extraction technique in horizon-based localization. The motivation for using FT is to decouple position prediction from the prediction of direction deviance (horizontal orientation shift), where frequency domain magnitudes predict the position and phase shifts predict the orientation shift. We use the FFT function to calculate the frequency domain representation of STOs.

3.3 Models

In this study, we focus on classification ML models. Some of the typical classification models used are GLM, LDA, and QDA. Neural Network (NNet) and Convolutional Neural Network (CNN) are advanced machine learning that can also be used for classifications.

3.3.1 GLM

The Generalized Linear Model (GLM) extends linear regression to categorical responses by linking a linear function of the predictors to a response function. A GLM with an identity link function is a linear model (LM), and a logistic link function is a logistic regression (LR) [13]. The common LR approach applies to binary categorical responses. For a multi-level categorical response, like what we deal with, a multinomial logistic regression (MNLR) needs to be used; however, our experience shows that MNLR does not work for a high number of response levels and fails to converge on a low number of responses.

3.3.2 LDA and QDA

These methods, which can be used for multilevel responses, assume features are normally distributed and find a combination of the features that best separate response levels. LDA creates a linear separation boundary based on a linear combination of features, while QDA uses a quadratic separation boundary.

3.3.3 NNet and CNN

Neural network models mimic the human neural system by forming an interconnected network of neurons, each sending a signal forward based on received signals and their parameters. A NNet is commonly formed from an input layer where features are input, an output layer where the response value is produced, and one or more hidden layers that transform the features into responses. NNet models are powerful tools for modeling complex, nonlinear relationships between features and responses.

Convolutional neural networks are specialized NNet models that are most appropriate for processing images. These models include additional layers for extracting important local features of images before passing them to a NNet model.

While an STO is not an image, it is a good candidate for CNN as the STO's unique local features can be used to quickly identify its location. Our preliminary exploration of these models showed that CNN outperforms NNet models.

4. Exploration, Simulation, and Results

This section first explores feature extraction methods to decide on the simulation parameters. Later, we present and discuss the simulation results.

To make the exploration efficient, we use a smaller STO database, where the original 304-by-304 grid is sampled 1 in 15, reducing the grid size to 21-by-21 and increasing the pixel size to 600m. We also assume $\theta = 0$ to reduce computational load.

4.1 Feature Extraction and Dimensionality Reduction

We can reduce the dimensionality of a localizing problem both in the feature space and response space; we will consider PCA and FFT methods for feature extraction and then use selected features to reduce the response space. Using a larger grid size is a rudimentary way of lowering the response space, where a large region is represented by one response value. We later use the K-Means algorithm on selected features as a more advanced method of grouping responses based on feature similarities.

4.1.1 PCA Feature Extraction

A linear transformation creates a new set of features in PCA. The latest set of features, each called a principal component (PC), contains the same amount of information, or lack thereof, measured in variability in data. Feature extraction using PCA is based on selecting the most critical PCA modes to maintain most of the variability. Fig. 6 shows that adding PCA modes sharply increases the proportion of explained variability but quickly flattens after 5 PCA modes.

For two randomly selected STOs, Fig. 7 shows as more PCA modes are used, the reconstructed STOs get closer to the original ones. Fig. 8 shows the overall reconstruction error drops as the number of PCA modes increases. We choose to use 15 PCA modes in our simulation.

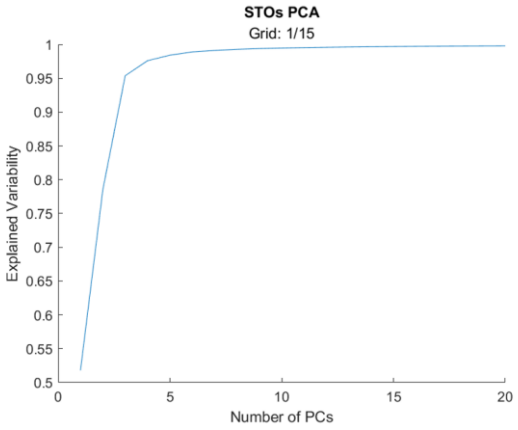

Fig. 6. Impact of number of PCA modes on the explained variability in the extracted features.

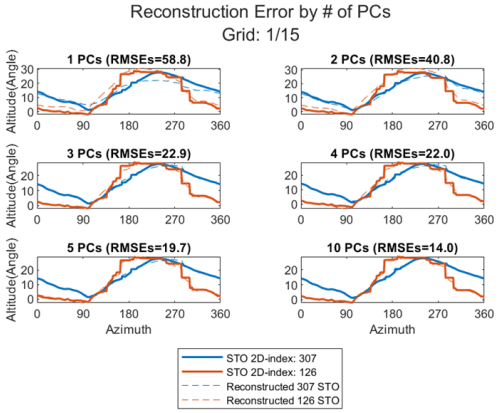

Fig. 7. Reconstruction of STOs with different numbers of PC for at two locations.

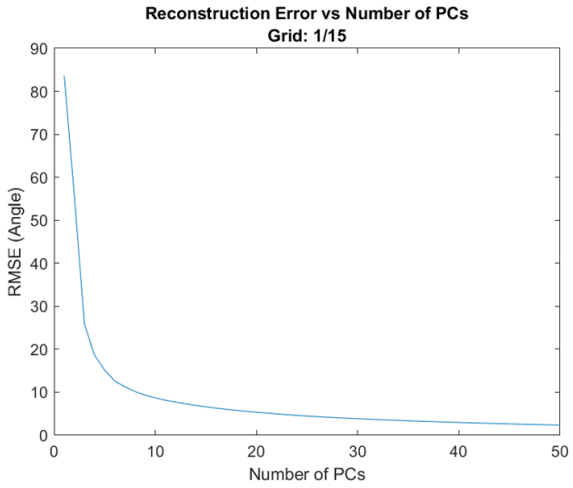

Fig. 8. Overall RMSE of reconstructing STOs as a function of the number of PCA modes.

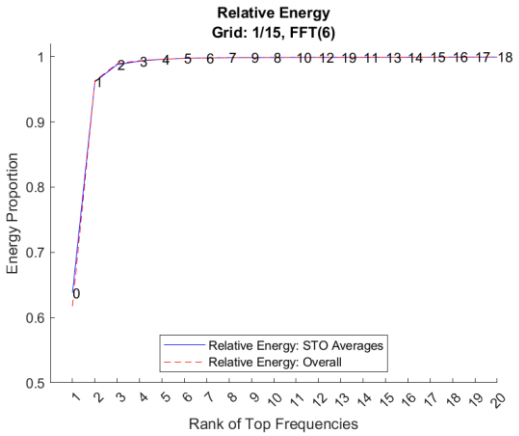

Fig. 9. Impact of critical frequencies on the explained energy for the top 6 contributing frequencies.

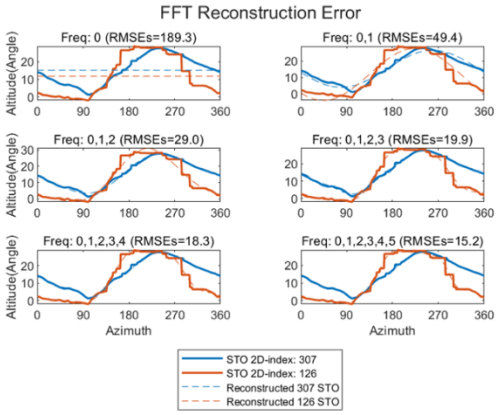

Fig. 10. Impact of critical frequencies on the explained energy in the extracted features.

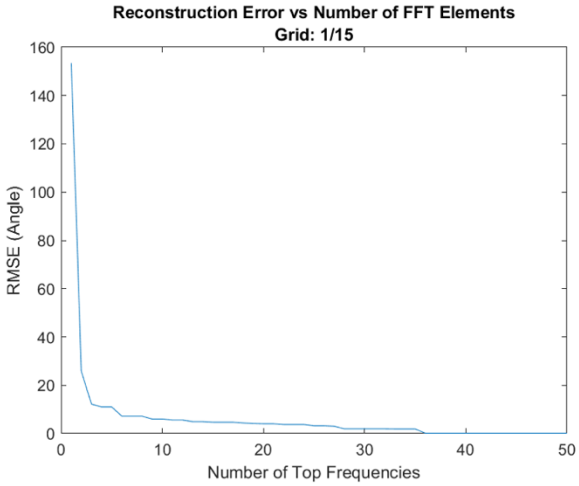

Fig. 11. Overall RMSE of reconstructing STOs as a function of the number of top contributing frequencies.

4.1.2 FFT Feature Extraction

Considering an STO as a signal, we can apply the FFT function to find the Fourier spectrum representation of the signal. The spectral representation of a signal has the same number of features, half or more in the form of magnitude values and the rest in the form of phase values.

A desired characteristic of the spectral representation of a signal is that if the signal is shifted, the magnitude features remain the same, and only the phase values change. This suggests that we can identify an STO regardless of the value of direction deviance θ, making magnitudes in the spectral domain a good candidate for localizing a rover based on the horizon line.

To further reduce the dimensionality of the database, we considered selecting the most essential frequencies based on their spectral magnitudes. However, since each STO may have a different set of most important frequencies, we used Borda (a vote-counting method [14]) to rank frequencies based on their STO-level rankings. Fig. 9 shows that the top 6 contributing frequencies for the whole database cover all frequencies from 0 to 19 but not in the same order. It also shows that the amount of relative energy (whether calculated at the database level or averaged over STO-level values) is mainly carried by the first few critical frequencies. Fig. 10 shows that the reconstruction error diminishes as the features of more frequencies are kept in the spectral representation of an STO. Finally, Fig. 11 shows how increasing the number of top contributing frequencies reduces the overall reconstruction error. It should be noted that the final number of frequencies kept in the model would always be larger than the number of top-contributing frequencies, as seen in the case of the top 6 contributing frequencies, where 20 frequencies (0 to 19) formed the top 6 contributing frequencies at STO level.

4.2 Clustering Responses

Grouping responses using a larger grid is a simple way of reducing the response space. More advanced clustering methods like K-Means use feature similarities, which are more practical because features are the only available data for predicting responses.

K-Means method can be applied directly to STOs, or it can be applied to extracted features. Results show that clusters found using PCA-generated features closely resemble those generated by STOs. Fig. 12 shows clustering using 15 PCA modes for different numbers of clusters. The image does not help identify a desirable number of clusters. In practice, the number of optimal clusters is found using the proportion of variability explained by clustering the data. Fig. 13 shows that the proportion of explained variability grows somewhat slowly, with a sensible rate dropping around 6 clusters and then around 10.

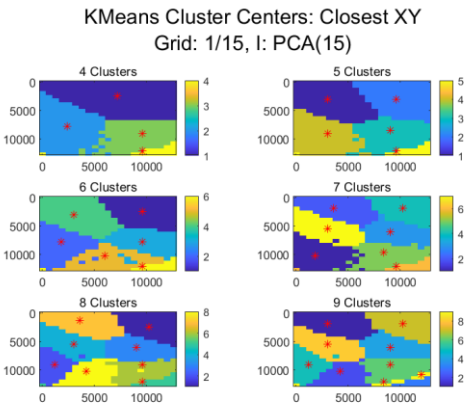

Fig. 12. Clustering (x, y) response region using 15 PCA modes for different numbers of clusters.

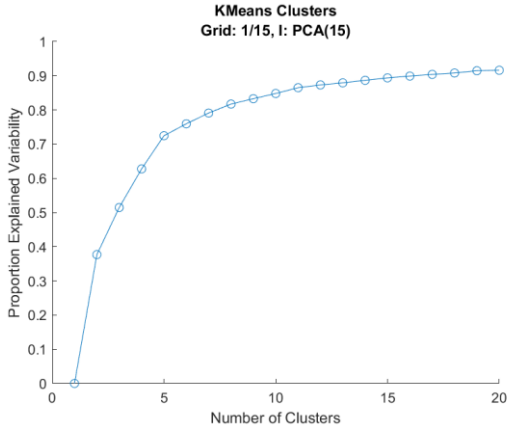

Fig. 13. The proportion of variability is explained as a function of a number of clusters generated by PCA features.

Fig. 14 shows K-Means clusters for FFT-generated features. The shape of clusters is more meaningful than the ones created using PCA-generated features. Fig. 15 also shows that around 6 clusters, the rate of increase drops, but the explained variability with 6 clusters is higher in this case compared to those of PCA (see Fig. 13).

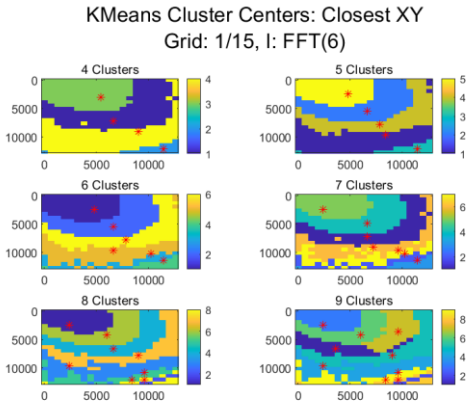

Fig. 14. Clustering (x, y) response region using the top 6 contributing frequencies for different numbers of clusters.

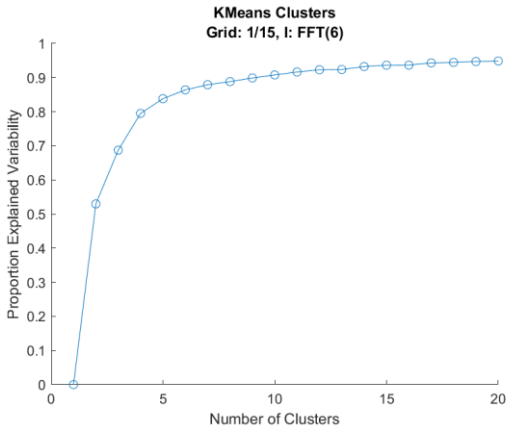

Fig. 15. The proportion of variability is explained as a function of several clusters generated by FFT features.

4.3 Simulation

We have conducted a simulation study to investigate the effectiveness of both feature extraction and prediction methods. The simulation was programmed in MATLAB and ran on an x-64-based laptop with four physical Intel® i-7, 2.8GHz cores, 16GB installed RAM, and running the Windows 11 OS.

4.3.1 Simulation Parameters

Table 1 shows the simulation parameters, which include information about the region of interest, features, responses, and models used in the simulation. The table also shows that we have considered adding noise in two places: one to the raw STO, whose impact carries through feature extraction, training, and evaluation, and the other to the test STOs.

The simulation trains models on all available data (determined by the Grid Size Reduction value). Then, it tests those models on batches of STOs from the database, similar to a K-fold cross-validation approach. The main difference is that the model is trained on all data (the test data is not left out) because, in practice, the models are trained in advance when all data is available. Adding noise to the STOs before testing better mimics the practical use of these models.

Table 1. Simulation Parameters

Parameters	Options	Values
Region	Dimension	12km x 12km
	Initial Grid	304 x 304
	Grid Size Reduction	1/5, 1/15, 1/30
Features	STO	
	PCA	PCA: 15 PCA modes
	FFT	FFT: Frequencies in the Top 6 Contributors
Responses	Position Index	
	STO-Based Clusters	6 and 10 clusters
	PCA-based Clusters	6 and 10 clusters
	FFT-based Clusters	6 and 10 clusters
Models	Brute Force	
	GLM	LM (identity link and normal distribution)
	LDA	Linear with a fallback on 'pseudolinear'
	QDA	Quadratic with a fallback on 'diagquadratic'
	CNN	
Noise Profile	STO	NoNoise and Noise1
	Test STO	NoNoise and Noise1

4.3.2 Performance Measures

Since the goal is to develop localization for autonomous rovers, we consider two accuracy measures: prediction time and memory footprint (see Table 2). The accuracy measures (MCR and RMSE) are defined for a test set of size m, where $\hat{l}_i$ and $(\hat{x}_i, \hat{y}_j)$ are predicted values for a response with an index position of l_i and a position of (x_i, y_i). The memory footprint is measured based on the file size of the model object saved as a MAT file, and the time is calculated only for the test prediction.

Table 2. Performance Measures

Measure	Definition
Miss-Classification Rate	MCR=$\frac{1}{m}\sum_i 1_{[l_i \neq \hat{l}_i]}$
Root Mean Square Error	RMSE = $\sqrt{\frac{1}{m}\sum_i (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2}$
Time	CPU Time in seconds
Space	Memory size of the trained model object saved as a MAT file

4.4 Results

Table 3 shows the mean and standard deviation of file sizes (in gigabytes) of the model objects saved as a MAT file. The table shows that as the Grid size increases, the file sizes decrease, but the change is not uniform: BF, followed by GLM, shows the largest increase, and CNN shows the least amount. It should be noted that CNN size was moderately high but stayed almost the same as the grid size increased.

The table also reveals that BF is only used for two cases: when the response is the position index in a 2D representation of the data and features are either the original STOs or the STOs plus noise. Another information provided in the table is that LDA and QDA models have failed to be trained or tested on data in 6 scenarios.

The other performance measures are presented in the following figures. Fig. 16 compares performance measures for all five methods at different grid sizes: 1/30, 1/15, and 1/5. Notably, BF performs best except for processing time when the grid size is the highest. Considering that BF also requires the highest memory space, it is clear that its ability is minimal and not a good choice for use on an autonomous rover. The figure also shows that GLM performs significantly worse when the grid size increases. It should be noted that the implementation of GLM was not an actual multinomial logistic regression, and its poor performance was expected. The fact that the GLM produces some results is a good case for representing that an improper model can produce results that may seem correct.

Table 3. File Size mean and standard deviation (in GB) for different models and grid sizes.

Method	N (Test Sets)	Grid Size →	**one1/30**	**1/15**	**1/5**
BF	2	Mean (SD) [in GB] →	0.3 (0.0)	1.2 (0.0)	9.8 (0.0)
CNN	2		1.9 (0.9)	1.9 (1.0)	2.2 (1.4)
GLM	36		0.3 (0.3)	1.4 (1.5)	9.1 (8.9)
LDA	30		0.8 (0.9)	1.4 (1.5)	5.2 (5.3)
QDA	30		0.5 (0.5)	1.2 (1.3)	7.0 (7.4)

Finally, Fig. 16 suggests that CNN generally works better than others as the grid size increases, except for the processing time. Considering the above observations, we focus on the 1/5 grid size and LDA, QDA, and CNN models as we move forward.

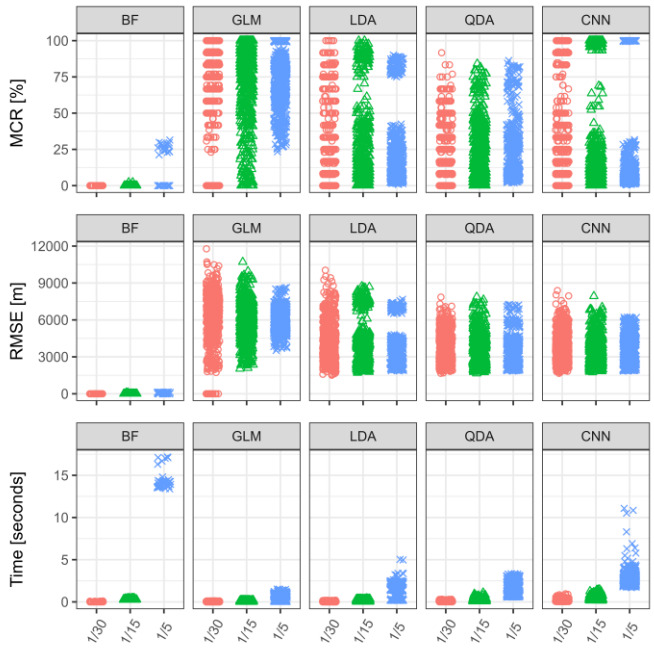

Fig. 16. The impact of grid size on MCR, RMSE, and Time performance measures.

Fig. 17 shows the impact of response type (Index, 10 Clusters, and 6 Clusters) and noise in the test STOs. The figure reveals that LDA and QDA could not fit a model when the position index was used as the response. In this case, each response has only one associated STO, and these methods cannot estimate a covariance matrix based on one observation. On the other hand, CNN can produce predictions for position index responses, albeit they are inferior. It is also notable that LDA and QDA are more sensitive to noisy test features. Moving forward, we further limit our focus on noisy test features, as it is more realistic that rover readings would include some noise.

Fig. 18 summarizes the impact of cluster types and features for predicting responses with 6 clusters. The figure shows that LDA does not do well when STOs (rad data) are used to predict the location. It also indicates that STOs generally take longer to predict an expected result because they have the most significant dimension compared to using 15 PCA modes or 20 frequency magnitudes (in the top 6 contributing frequencies).

The figure also shows that PCA generally performs better. Some QDA measurements behave differently and have a performance that degrades more when FFT features or FFT-based clusters are used. A closer look revealed that those relate to features without noise. This surprising result could be due to the high collinearity expected for adjacent STOs and its impact on estimated covariance in QDA. Adding noise can reduce the collinearity and the covariance sensitivity.

Fig. 17. The impact of Test Noise and Response Type on MCR, RMSE, and Time performance measures.

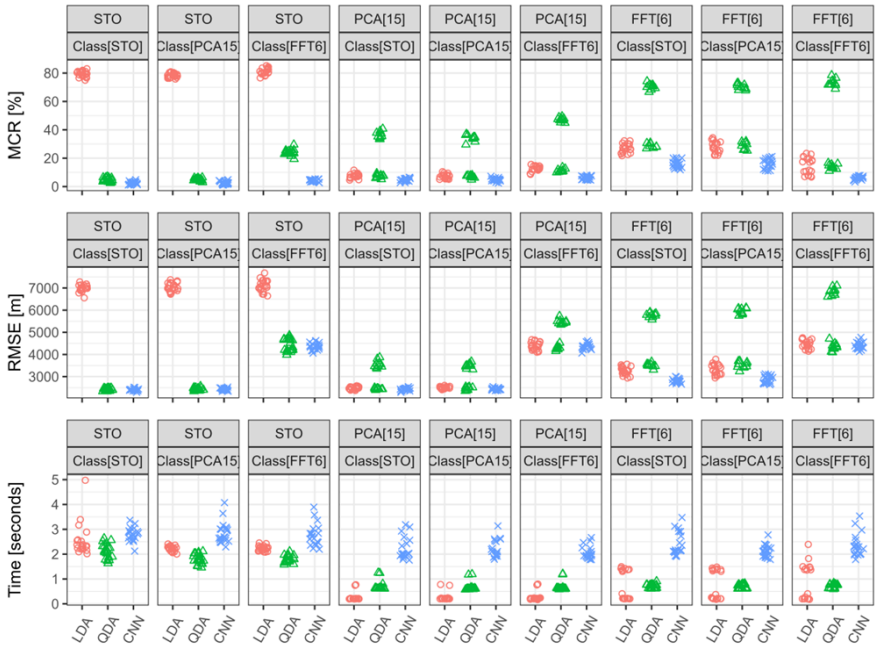

Fig. 18. The impact of Feature and Cluster Types on MCR, RMSE, and Time for 6-Cluster Response.

5. Conclusions

In this study, we investigated the challenges of implementing localization by an autonomous rover in the absence of PNT satellite infrastructure. We showed the relation between desired accuracy, required memory space, and processing time to achieve that accuracy under a BF search approach. We also systematically characterized localization problems based on features (predictors), responses, and applicable methods. We considered feature extraction and clustering techniques as solutions to the high dimensionality problem for localization in vast regions. We then proposed using FFT as an alternative to the commonly used PCA method for feature extraction. Considering position indices as

the response, we investigated the application of relevant ML approaches like GLM, LDA, QDA, and CNN through simulation. Our simulation results showed shortcomings of using GLM, LDA, and QDA in localization problems and surfaced CNN as a more reliable method; however, the results showed that while CNN memory requirement and time restrictions are less dependent on other parameters (like feature and response type), they are not always the lowest values. We also found that PCA works well and that FFT is a promising contender, particularly considering that we did not explore a search database exploded with non-zero values for direction deviance, the unknown horizontal orientation shifts at the time the rover takes a panoramic photo of its surroundings.

We consider three areas of future research: 1) finding more effective feature extraction methods, 2) considering alternative response structures, like using numerical responses or hierarchical structure for increasing resolution of categorical responses, and 3) more effective ML methods or fine-tuning existing ones.

Acknowledgments

The authors gratefully acknowledge the support provided by California State University, Fullerton (CSUF) through computational and financial resources for preparing and presenting the manuscript at the conference.

We also acknowledge using Grammarly and generative AI tools (such as Google NotebookLM and Claude) to assist with sentence improvement, literature analysis and thought organization while preparing this manuscript. These tools were used as writing aids while all intellectual content, studies, and conclusions remain the original work of the authors.

References

[1] J.T. Keane, S.M. Tikoo and J. Elliott, Endurance: Lunar South Pole–Aitken Basin Traverse and Sample Return Rover. (date not available) https://science.nasa.gov/wp-content/uploads/2023/11/endurance-spa-traverse-and-sample-return.pdf (Accessed: 31 March 2025).

[2] H. Zhang, A. Mao, Z. Tian, Y. Zhang, Y. Sun and Q. Hu, Localization and Autonomous Exploration for Lunar Rover with Only RGBD Sensors, 2023 IEEE International Conference on Unmanned Systems (ICUS), Hefei, China, 2023, pp. 972-977.

[3] J. Kim, Autonomous Lunar Rover Localization while Fully Scanning a Bounded Obstacle-Rich Workspace. Sensors 2024, 24, 6400. https://doi.org/10.3390/s24196400 (Accessed: 01.04.25).

[4] L. Matthies et al., Lunar Rover Localization Using Craters as Landmarks, 2022 IEEE Aerospace Conference (AERO), Big Sky, MT, USA, 2022, pp. 1-17, doi: 10.1109/AERO53065.2022.9843714.

[5] S. Daftry et al., "LunarNav: Crater-based Localization for Long-range Autonomous Lunar Rover Navigation," 2023 IEEE Aerospace Conference, Big Sky, MT, USA, 2023, pp. 1-15, doi: 10.1109/AERO55745.2023.10115640.

[6] Chen, Z., Li, K., Li, H. et al. Metric localization for lunar rovers via cross-view image matching. Vis. Intell. 2, 12 (2024). https://doi.org/10.1007/s44267-024-00045-y (Accessed: 01.04.25).

[7] X. Zhao, L. Cui, X. Wei, C. Liu and J. Yin, Lunar Rover Cross-View Localization Through Integration of Rover and Orbital Images, in IEEE Transactions on Geoscience and Remote Sensing, vol. 62, pp. 1-14, 2024, Art no. 5642414, doi: 10.1109/TGRS.2024.3462487.

[8] W. Liu, C. Liu, Z. Cheng, Z. Zhang, X. Liu and C. Huang, A General Model and Application Analysis of Spatial Resolution on Large Space Navigation Imaging of Yutu 2 Lunar Rover, in IEEE Transactions on Aerospace and Electronic Systems, vol. 60, no. 4, pp. 4023-4034, Aug. 2024, doi: 10.1109/TAES.2024.3375279.

[9] S. Chiodini, M. Pertile, S. Debei, et al., Mars rovers localization by matching local horizon to surface digital elevation models, 2017 IEEE International Workshop on Metrology for AeroSpace (MetroAeroSpace), Padua, Italy, 2017, pp. 374-379.

[10] C. H. Lee, K-M. Cheung, A Novel Autonomous Lunar Self-Localization Technique Based on Local Surrounding Horizon Mask, 2024 IEEE Aerospace Conference, Big Sky, MT, USA, 2024, pp. 1-7, doi: 10.1109/AERO58975.2024.10521112.

[11] Z. Tian, H. Zhang and Q. Hu, Global Localization Technology of Lunar Rover by Horizon Line Matching, in IEEE Transactions on Aerospace and Electronic Systems, vol. 60, no. 6, pp. 8744-8756, Dec. 2024, doi: 10.1109/TAES.2024.3433838.

[12] D. Burghal, A. T. Ravi, V. Rao, A. A. Alghafis, A. F. Molisch, A Comprehensive Survey of Machine Learning Based Localization with Wireless Signals, (Dec 2020) https://arxiv.org/abs/2012.11171 (Accessed: 01.04.25).

[13] MATLAB R2024b, fitglm, https://www.mathworks.com/help/stats/fitglm.html (Accessed: 01.04.25).

[14] N. B. Fox, B. Bruyns, An Evaluation of Borda Count Variations Using Ranked Choice Voting Data, (Jan 2025) https://arxiv.org/abs/2501.00618 (Accessed: 01.04.25).