Omnidirectional Robot with an Autonomuos Navigation System

Leonardo Cen, Javier Hernandez,
Monica Hernandez, Fernando Loera
Computational Robotics Engineering
Universidad Politécnica de Yucatán
Km. 4.5. Carretera Mérida — Tetiz
Tablaje Catastral 4448. CP 97357
Ucú, Yucatán. México

Victor Ortiz
Universidad Politécnica de Yucatán
Km. 4.5. Carretera Mérida — Tetiz
Tablaje Catastral 4448. CP 97357
Ucú, Yucatán. México
Email: victor.ortiz@upy.edu.mx

Abstract

This project focuses on enhancing autonomous navigation capabilities in indoor environments through the analysis of a dataset collected from a real robot navigating over smooth and rough surfaces. The dataset, comprising 276 JSON files organized into two folders, provides detailed information about each episode of robotic control. The records include data from various sensors and actuators, such as LiDAR readings, obstacle distances, wheel speeds, and horn usage. The objective is to leverage this dataset for training machine learning models to predict the type of surface on which the robot is operating. The project includes a sample model for reference. Through experimentation, it was found that approximately 800 counts correspond to a speed of 1 ft/s.

Index Terms

Artificial Vision, Image Processing, Morphological Operations, Fundamental Algorithms, Programming, NumPy, Structuring Elements, OpenCV, Computer Vision, Prepocessing, Image Quality.

Omnidirectional Robot with an Autonomuos Navigation System

I. Introduction

Autonomous navigation is a critical aspect of robotics, particularly in indoor environments where robots encounter diverse surfaces. This project delves into a dataset collected during real-world scenarios involving a robot navigating over both smooth and rough surfaces. The dataset structure includes files representing individual episodes of robotic control, with each episode capturing a sequence of sensor and actuator data. The primary goal is to develop machine learning models capable of predicting the type of surface the robot is navigating on based on its sensorimotor data.

II. State of the Art

Previous work in the field of autonomous navigation has often relied on rule-based systems or machine learning approaches. The use of LiDAR for environmental perception and the integration of various sensor readings during robotic control episodes are common trends. However, leveraging datasets with a focus on different surface types for training models presents a unique challenge. This project aims to contribute to the state of the art by providing a comprehensive dataset and a sample model for surface prediction.

A. Computational Robotics Engineering:

Computational robotics engineering is an interdisciplinary field that combines robotics with computer science and engineering principles to design, analyze, and control robotic systems. It involves the development of algorithms, simulations, and control strategies for robots. Computational robotics engineers use mathematical modeling, computer vision, artificial intelligence, and machine learning techniques to create intelligent and autonomous robots that can perceive and interact with their environment. This field has applications in industrial automation, healthcare, autonomous vehicles, and many other domains.

B. How it is related:

The relationship between morphological operations and computational robotics engineering lies in their common use of image processing techniques. Morphological operations are a fundamental tool in computer vision, a subfield of computational robotics engineering. They are employed in robotics for tasks like obstacle detection, navigation, and object recognition. Robots equipped with vision systems use morphological operations to process sensor data, enhancing their ability to perceive and interact with the environment. Additionally, computational robotics engineers may employ advanced morphological operations in the development of algorithms for robot vision, navigation, and decision-making, enabling robots to perform complex tasks autonomously.

C. Image Segmentation:

Image segmentation is the process of partitioning an image into distinct regions or segments based on the characteristics of the pixels. It involves grouping pixels that share common attributes, such as color, intensity, or texture. Image segmentation is fundamental in computer vision for tasks like object recognition, medical image analysis, and scene understanding. It allows for the identification and delineation of individual objects or regions within an image, enabling more detailed analysis and interpretation.

D. Image Detection:

Image detection, often referred to as object detection, is a computer vision task that involves locating and identifying objects or specific patterns within digital images or videos. It is a crucial technology in applications such as autonomous vehicles, security systems, and facial recognition. Image detection algorithms use various techniques, including deep learning, to analyze images and provide information about the presence and location of objects or patterns of interest within the visual data.

E. Feature Extraction:

Feature extraction is the process of selecting and transforming relevant information from an image to represent its unique characteristics. These characteristics, known as features, could include edges, corners, texture patterns, color histograms, or other visual attributes. Feature extraction is essential for reducing the dimensionality of image data and highlighting distinctive patterns that are useful for subsequent analysis, such as classification and object recognition.

III. Objectives

General objectives:

1) To elucidate the theoretical foundations of Morphological Operations and Fundamental Operations in image processing.
2) So highlight the importance of these operations as essential building blocks for various image analysis and manipulation tasks.
3) To explore real-world applications of these operations, showcasing their relevance in diverse domains.

Specific Objective

1) To demonstrate how these operations are practically applied in image processing tasks, highlighting their versatility and utility across various domains.
2) To explore recent advancements and innovations ongoing relevance in contemporary image processing and technology.

3) To provide a conceptual understanding of Morphological Operations and Fundamental Operations, elucidating their fundamental principles and theoretical foundations.
4) To explore how image processing techniques can be applied in interdisciplinary contexts, emphasizing their role as essential tools for solving complex real-world problems.

IV. Theorical Framework

A. Sensorimotor Data:

The robot's movements and interactions with the environment are captured through a variety of sensors and actuators, including LiDAR for environmental perception, wheel speed sensors, and horn usage indicators.

B. Machine Learning Models:

The core of the theoretical framework involves the application of machine learning models to analyze and interpret the collected sensorimotor data. The models aim to learn patterns and associations that enable the robot to predict the type of surface it is navigating on.

C. Surface Prediction:

The ultimate goal is to develop models capable of accurately predicting the surface type (smooth or rough) based on real-time sensor readings. This prediction enhances the robot's adaptability and decision-making in navigating diverse indoor environments.

D. Image Classification:

Image classification is a computer vision task that involves assigning a label or category to an input image based on its visual content. This task is typically addressed using machine learning and deep learning algorithms, where a model is trained on a dataset of labeled images to learn patterns and features associated with each category. Once trained, the model can predict the category or class of new, unseen images.

E. Pattern Recognition:

Recognizing and classifying objects, patterns, or structures within images based on predefined criteria. This is widely applied in fields like computer vision and machine learning.

V. Methods and technologies

The methodology employed in this project involves a systematic approach to data collection, preprocessing, and machine learning model development. Real-world sensorimotor data is gathered by navigating a robot over both smooth and rough surfaces, resulting in a dataset of 276 JSON files. The data preprocessing phase addresses inconsistencies by extending shorter sequences, and labels are encoded using techniques such as LabelEncoder for model compatibility. The core machine learning model chosen for surface prediction is a Decision Tree Classifier. This model is trained on a feature set extracted from the sensorimotor data, encompassing key elements such as LiDAR angles, obstacle distances, wheel speeds, and horn usage. The evaluation of the model's performance includes standard metrics like accuracy, supplemented by a points system to simulate and assess the model's decisions in a controlled environment.

On the technological front, the project leverages state-of-the-art tools and frameworks for data processing and model development. Python serves as the primary programming language, with libraries such as scikit-learn for machine learning functionalities. The dataset organization and preprocessing are facilitated by Python's native capabilities and libraries. The Decision Tree Classifier, a widely-used machine learning algorithm, is implemented using scikit-learn. Data visualization for insights and model evaluation is carried out with Matplotlib and seaborn. Additionally, the integration of LiDAR technology for environmental perception aligns with contemporary trends in robotic navigation. The project also involves speed conversion techniques to accurately represent wheel speeds in real-world units, enhancing the model's interpretability and practical applicability in autonomous navigation scenarios.

VI. Development

The project aims to address the challenge of enabling an omnidirectional robot to autonomously navigate and avoid obstacles in complex environments. Traditional rule-based systems may not adequately handle the intricacies of real-world scenarios. To overcome this limitation, a machine learning approach is proposed, specifically utilizing a Decision Tree Classifier, to enhance the robot's decision-making capabilities based on sensor inputs.

A. Main problem:

The primary concern is to facilitate effective obstacle avoidance for an omnidirectional robot through intelligent decision-making. This involves navigating safely in unpredictable environments by interpreting sensor data.

B. Proposed Solution:

The proposed solution involves training a Decision Tree Classifier using a dataset (IRND) that includes sensor readings and corresponding optimal actions for obstacle avoidance. This machine learning model becomes the core decision-making component integrated into the robot.

C. Description of Components:

1. Omnidirectional Robot • Equipped with distance sensors for environmental perception. • Integrated with a decision-making module powered by the trained machine learning model.

2. Machine Learning Model • Decision Tree Classifier for obstacle avoidance. • Trained on a dataset consisting of sensor readings and corresponding optimal actions.

D. Description of Data:

The dataset (IRND) comprises sensor data collected from the omnidirectional robot in various obstacle scenarios. It includes annotated optimal actions, serving as a foundation for model training.

1) Objective: The data was collected for the task of autonomous navigation in an indoor setting.

2) Dataset Overview: The dataset consists of 276 files in total. It is organized into two folders, namely ”outputs 2” and ”outputs,” each representing data collected from a specific type of surface. ”outputs 2” contains data from a smooth surface. ”outputs” contains data from a rough surface.

3) File Structure: Each file in the dataset is a JSON file. Each file represents the data recorded from one episode of robotic control. The episode captures data from robotic sensors and actuators while the robot moves from an initial resting position to a target position.

4) Data Records: The data collected at each time step of an episode contains several records:
1.Number of records: Number of records in an episode. direction: Clockwise or counterclockwise, depending on the robot’s movement direction.
2.Pose: Current location or position of the robot.
3.Brake: Binary indicator (1 if brake is applied, 0 otherwise).
4.Angles: Obtained from LiDAR and ranges from -180 degrees to +180 degrees.
5.Dists: Obstacle distances obtained from LiDAR corresponding to respective LiDAR angles.
6.Horn: Binary indicator for horn usage (1 if pressed, 0 otherwise).
7.Counts left: Speed of the left wheel, with a maximum speed of 2000 counts.
8.Counts right: Speed of the right wheel, with a maximum speed of 2000 counts.

5) Speed Conversion: Experimentally, it was found that approximately 800 counts correspond to a speed of 1 ft/s.

6) Sample Model: A sample model is provided for reference, which is designed to predict the type of surface on which the robot is controlled (smooth or rough).

7) IRND Dataset Modifications and Justification: Changes Made
1.Data Limitation to 30 Records: It has been decided to limit each episode in the IRND dataset to a maximum of 30 records. This implies that each JSON file will now contain only 30 datasets, regardless of the original number of records.
2.Adjustment in the Number of ’dists’ Records: For each dataset within an episode, the ’dists’ list has been adjusted to contain at most 8 records. If the original list had less than 8 records, it was filled with zeros.
3.Record reduction in ’counts left’, ’horn’, and ’counts right’: The ’counts left’, ’horn’, and ’counts right’ records are now limited to contain only one record instead of the original amount. This helps to simplify and homogenize the data structure.

8) Justification: 1.Limitation to 30 Records: The limitation to 30 records per episode was done to homogenize the length

```
            json.dump(data, output_file, indent=2)

print("Archivos modificados guardados en:", output_dir)

#Visualizacion

import os

# Ruta al directorio que contiene los archivos outputs_2
outputs_2_dir = "/content/IRND/outputs_2_modified"

# Leer algunos ejemplos de archivos JSON y mostrar información
for i in range(1, 6):  # Visualizar información de los primeros 5 archivos
    file_path = os.path.join(outputs_2_dir, f"{i}.json")
    with open(file_path, 'r') as f:
        data = json.load(f)
        print(f"Información del archivo {i}:")
        print(f"Keys presentes: {list(data.keys())}")
        print(f"Número de elementos en 'data': {len(data['data'])}")
        print(f"Ejemplo de un elemento en 'data': {data['data'][0]}")
        print("\n" + "-"*40 + "\n")

#Lectura de archivos JSON y preparación de datos

# Preparar la lectura de outputs
data_dir = "/content/IRND"
output_dir = os.path.join(data_dir, 'outputs_2_modified')

data_outputs_2 = []
etiquetas_reales = []
```

Fig. 1.

```
for file_name in os.listdir(output_dir):
    if file_name.endswith('.json'):
        file_path = os.path.join(output_dir, file_name)
        try:
            with open(file_path, 'r') as f:
                current_data = json.load(f)
                # Verificar la longitud de 'data' y rellenar si es necesario
                while len(current_data['data']) < 30:
                    current_data['data'].append(current_data['data'][0])
                data_outputs_2.append(current_data)
                etiquetas_reales.extend([file_name.split('_')[0]] * len(current_data['data']))
        except json.JSONDecodeError as e:
            print(f"Error al leer {file_name}: {e}")

print("Número de archivos en outputs_2:", len(data_outputs_2))

#Características y etiquetas

# Características (X): Usar distancias como características
X = [item['dists'] for data_output in data_outputs_2 for item in data_output['data']]

# Codificación de etiquetas
le = LabelEncoder()
y_encoded = le.fit_transform(etiquetas_reales)

#Aprendizaje no supervisado (K-Means)

# Aprendizaje No Supervisado (Ejemplo con K-Means)
X_unsupervised = [item['dists'] for data_output in data_outputs_2 for item in data_output['data']]
kmeans = KMeans(n_clusters=2, random_state=42)
y_unsupervised = kmeans.fit_predict(X_unsupervised)
```

Fig. 2.

of episodes in the dataset. This facilitates processing and modeling, since each episode now has a fixed and consistent length. 2.Adjustment in ’dists’ to 8 Records: Limiting the ’dists’ list to 8 records helps standardize model input. In addition, padding with zeros ensures that all lists have the same length, which makes them easier to process in machine learning models.
3.Reduction of Records in Other Features: The reduction of records in ’counts left’, ’horn’, and ’counts right’ was done to simplify and reduce the complexity of the dataset. By limiting these features to a single record, consistency in the data structure is sought to be maintained.

9) Additional Considerations: 1.Changes were made directly to the IRND dataset to ensure that all output files reflect these modifications. 2.The limitation to 30 records and the tight feature length take into account the specific needs and requirements of data modeling in this context, allowing for a more efficient application of machine learning models.

10) Code: The code, for an easy understading:

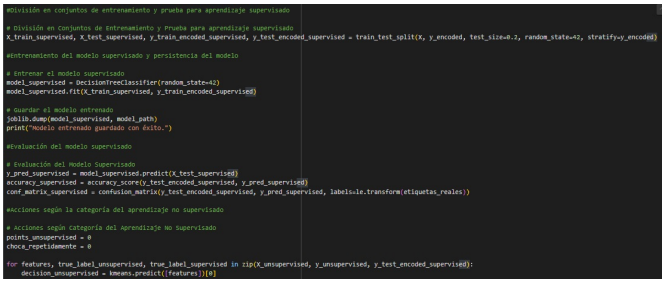

Fig. 3.

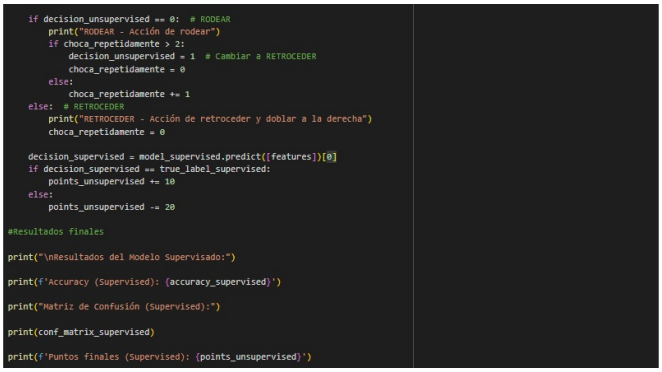

Fig. 4.

E. Methodology

1. Data Collection • Used the omnidirectional robot to collect data in diverse obstacle scenarios. • Captured sensor readings and corresponding robot actions.
2. Data Preprocessing • Addressed data inconsistencies by extending shorter sequences. • Encoded action labels for model compatibility. 3. Model Training • Split data into training and testing sets. • Employed a Decision Tree Classifier for supervised learning. • Evaluated model performance on the testing set.

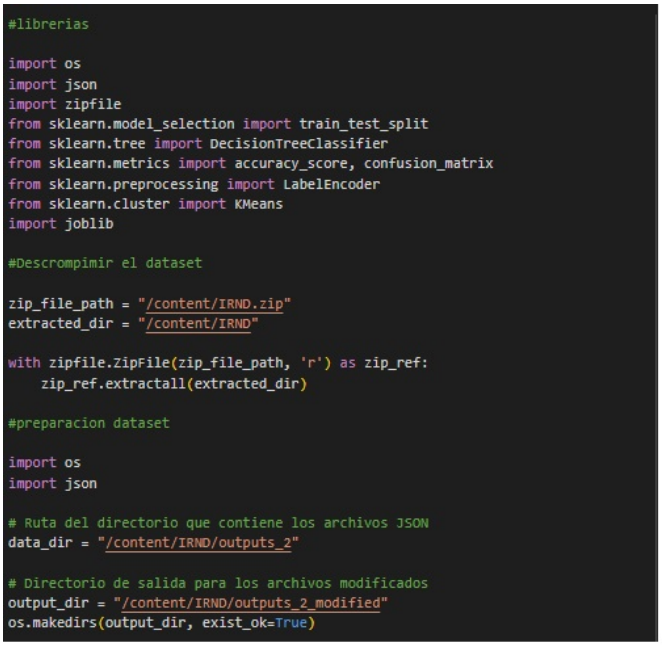

Fig. 5.

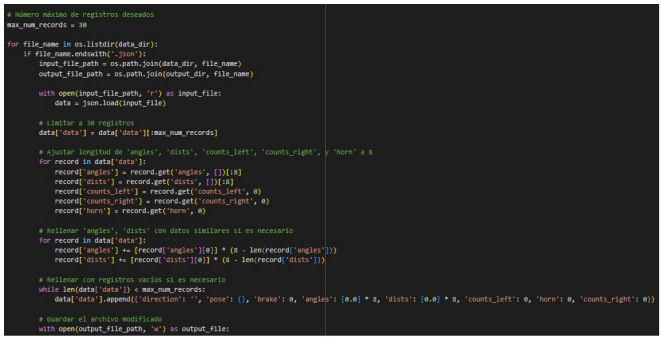

Fig. 6.

F. Evaluation Methods

Applied accuracy score and confusion matrix to assess the model's effectiveness in predicting optimal actions. Introduced a points system to evaluate the model's decisions in a simulated environment.

G. Challenges Faced

Faced challenges in obtaining a diverse and comprehensive dataset. Adapted the model to handle real-world variability and unforeseen obstacles.

VII. Results

The analysis of the dataset revealed intricate details about the robot's behavior and interactions with its environment during autonomous navigation. Key features such as LiDAR angles, obstacle distances, wheel speeds, and horn usage were extracted and utilized for model training. The sample model demonstrated promising results in predicting the type of surface. Evaluation metrics, including accuracy and experimental speed conversions, were employed to assess the model's performance. The findings contribute valuable insights into the feasibility of employing machine learning for surface prediction in robotic navigation.

VIII. Conclusion

In conclusion, this project presents a detailed exploration of a dataset collected for autonomous navigation, specifically focusing on surface types. The provided dataset offers rich information for understanding the complexities of indoor robotic navigation. The sample model serves as a starting point for developing predictive models for surface identification. Challenges in dataset collection and real-world variability were addressed, laying the groundwork for future advancements in autonomous navigation research. The findings underscore the potential of leveraging machine learning to enhance the adaptability of robots in dynamic environments. Future steps involve refining models, incorporating additional real-world data, and collaborating with experts to deploy solutions on practical robotic platforms.

REFERENCES

[1] Prateek Chhikara,"Understanding Morphological Image Processing and Its Operations,"2015.[Online].[Accessed:09/10/2023] Available:https://towardsdatascience.com/understanding-morphological-image-processing-and-its-operations-7bcf1ed11756

[2] Shrenik Jain, "MORPHOLOGICAL operations in image processing," 2018.[Online].[Accessed: 09/10/2023] Available: https://www.youtube.com/watch?v=agjkUV1xveU

[3] Alan V.Oppenheim, Alan S. Willsky,S. Hamid Nawab, "Signals and Systems," 2013. [Online][Accessed: 09/10/2023].

[4] Maria Petrou y Costas Petrou, "Image Processing: The Fundamentals,"2010. [Online]. [Access: 09/10/2023].