Project Report

Analyzing 3D Point Cloud Map Data for Exit Detection

Prepared by:

Elias Daw – 314934092

Kareen Ghattas – 207478728

Course: Data learning in real time system
Date: 31/08/2023

DataLearningInRealTimeSystems - Github

Table of Contents

Introduction:

The project provides a systematic approach to detecting potential exits in 3D point cloud data. Starting from the raw input, the data is loaded from XYZ files, a prevalent format for such spatial datasets. Once loaded, the data undergoes preprocessing where it's centered around its mean, ensuring uniformity. Extreme values that could skew results are also trimmed.

The Convex Hull technique is then applied on the 2D projection of the dataset. Recognizing that the initial convex hull might capture extraneous details, an iterative refinement strategy is used to adjust the hull vertices, aiming to maintain a 90-degree angle between them. This involves shrinking the original hull towards its center, and the points residing between the original and shrunk hull are assumed to potentially signify exits.

For the identification of these exits, hierarchical clustering techniques come into play. In the 2D context, the Agglomerative Hierarchical Clustering technique is deployed, focusing on the variance of distances from cluster centroids combined with cluster density to pinpoint the most probable exit cluster. This methodology is then extended into 3D, applying the same hierarchical clustering logic but adapted for three-dimensional data.

Visualization forms a pivotal component of this project. The tool offers both 2D and 3D visual representations. In 2D, users can view the raw data points juxtaposed with the convex hull, clusters, and the identified exits. The 3D visualization, on the other hand, immerses the user into the full depth of the dataset, with the potential exit emphasized using a bounding box.

Descriptive summary:

1. Data Collection and Manipulation:

- The XYZ format, a standard in 3D mapping, is used to import data.
- Post-import, data undergoes processing: it is centered around its mean and extreme values are removed. This not only ensures consistency but also enhances accuracy for subsequent steps.

2. Convex Hull Computation:

- The code employs the Convex Hull method to derive the principal shape of the 2D projection of the data.
- To refine the hull's shape, an iterative refinement method is applied, adjusting vertices to reduce deviations from a 90-degree angle.
- A secondary, shrunk convex hull is computed. The points lying between this shrunk hull and the original hull are of primary interest, as they potentially indicate exits.

3. Hierarchical Clustering:

- For 2D analysis, the Agglomerative Hierarchical Clustering technique is used. By considering the variance of distances from cluster centroids and cluster density, the most likely exit cluster is identified.
- A similar approach is extended to 3D, refining exit detection further.

4. Visualization:

- Visualization is an integral part of the code, allowing users to see the raw data points, convex hull, clusters, and potential exits.
- 2D visualizations provided, highlighting clusters, the shortest path connecting cluster centroids, and potential exits.
- 3D visualizations are more immersive, with the entire point cloud displayed. The potential exit is accentuated using a bounding box.

5. GUI Integration:

- A user-friendly GUI is created to facilitate map selection.

6. Scalability:

- The code is designed to handle various map configurations.

7. Performance:

- Time metrics are integrated into key functions, like Convex Hull computation and clustering, providing insights into the code's efficiency.

Detailed Function Summaries:

1. **load_xyz(filename)**:
    - **Description**: Reads the point cloud data from an XYZ file. Each row in the file represents a 3D point with its X, Y, and Z coordinates.
    - **Inner Workings**: Uses **numpy** to load data from the file and then uses Open3D to convert this data into a point cloud format.
    - Input: Filename of the XYZ file.
    - Output: Point cloud data in Open3D format.
2. **create_gui()**:
    - **Description**: Constructs a simple GUI for map file selection.
    - **Inner Workings**: Uses the **tkinter** library. For each map configuration, a button is created. Clicking a button loads and runs the corresponding map.
    - Input: None.
    - Output: GUI with map selection options.
3. **retrieve_3d_points_from_2d(gap_points_2d, original_points_3d)**:
    - **Description**: Matches 2D gap points with their corresponding 3D points.
    - **Inner Workings**: Iterates over each 2D point and finds matching points in the 3D dataset based on X and Y coordinates. This is done using a conditional numpy selection.
    - Input: 2D gap points, Original 3D points.
    - Output: Corresponding 3D points.
4. **trim_extreme_values(points, percentage=5.0)**:
    - **Description**: Excludes extreme values based on percentiles.
    - **Inner Workings**: Calculates the specified percentile thresholds for X and Y values separately. Points that fall outside these thresholds are excluded.
    - Input: Data points, percentage to trim.
    - Output: Trimmed data points.
5. **computeConvexHull(points)**:

- **Description**: Generates the convex hull for a set of points.
- **Inner Workings**: Uses the **ConvexHull** function from the **scipy.spatial** module to compute the convex hull and returns the vertices that make up the hull.
- Input: Data points.
- Output: Indices of points forming the convex hull.

6. **kmeans_clustering_and_exit_detection(points, n_clusters=70)**:
   - **Description**: Clusters the data and identifies the potential exit.
   - **Inner Workings**: Utilizes the KMeans algorithm to cluster the data points. After clustering, it calculates the density for each cluster. The cluster with the lowest density (indicative of sparse data or an exit) is flagged as the potential exit.
   - Input: Data points, number of clusters.
   - Output: Cluster labels and exit cluster.
7. **hierarchical_clustering_and_exit_detection(points, n_clusters=5)**:
   - **Description**: Applies hierarchical clustering on 2D points and determines the exit based on a combination of variance and density.
   - **Inner Workings**: Uses AgglomerativeClustering to segment the points. For each cluster, the variance of distances from the centroid and the density (based on the convex hull's area) are computed. A metric combining variance and inverse density identifies the potential exit.
   - Input: Data points, number of clusters.
   - Output: Cluster labels and exit cluster.
8. **hierarchical_exit_detection_3d(points, n_clusters=5)**:
   - **Description**: Similar to the 2D hierarchical clustering but operates in 3D space.
   - **Inner Workings**: Works similarly to its 2D counterpart but considers 3D points and their volume in the density calculation. The potential exit is determined based on the cluster with a combination of high variance and high mean distance from the centroid.
   - Input: Data points, number of clusters.
   - Output: Cluster labels and exit cluster.
9. **visualize_clusters_and_exit_3D(points, labels, exit_cluster)**:
   - **Description**: Renders a 3D visualization of the clusters and highlights the detected exit.

- **Inner Workings**: Utilizes the Open3D library to visualize the 3D point cloud. Each cluster is assigned a unique color. The exit cluster, if present, is highlighted with a bounding box.
- Input: Data points, cluster labels, and exit cluster.
- Output: 3D visualization.

Usage Guide:

1. Running the Project:

To execute the project, simply run the main script. This will launch the GUI, allowing you to select a map for processing.

2. Adding New Maps:

- Place the XYZ file in the same directory as the main script.
- Update the **map_configurations** dictionary in the script with the new map's filename as the key and its associated 2D and 3D cluster counts as values.

3. Adjusting Parameters:

Tweak function parameters like percentage for trimming and cluster count for clustering.

4. Visualization:

Call visualization functions to see 2D and 3D visual representations of data, clusters, and exits.

5. Extending the Code:

- Add new functions and integrate them into the script.
- Import new dependencies and libraries as needed.

6. Dependencies:

Ensure required libraries like **numpy, scipy, matplotlib, tkinter, open3d,** and **sklearn** are installed. Additionally, Python 3.9 is necessary for open3d.

Challenges Encountered:

Throughout the development of this project, we encountered several challenges that spanned across three main segments: cleaning the data, determining the room shape, and pinpointing the exit. These challenges required iterative refinements and the exploration of various methodologies.

1. Cleaning the Data:

The early stages of data cleaning were met with challenges stemming from the quality of our drone-scanned maps. These maps often had a plethora of noisy points due to reflective elements and walls, which distorted the true representation of the room. To address this, we initially considered using a fitting line for the 2D representation of the data points, categorizing them based on squared distances from the center using the phi and radius coordinates. While conceptually sound, this approach had its limitations in truly capturing the intricacies of our dataset.

In light of this, we turned to clustering algorithms, hoping they would offer a more nuanced filtering mechanism. The concept of accurate coresets was particularly intriguing. However, after diving into the research by Ibrahem Jubran and evaluating the accurate coreset code, it was clear this method wasn't optimal for our dataset due to its drastic reduction in points.

Despite these setbacks, our determination remained unwavering. We further experimented with filtering techniques such as statistical outlier removal and radius outlier removal, each presenting its unique challenges and insights.

2. Finding the Room Shape:

When it came to shaping the room, the noisy data from our maps presented a significant hurdle. Our initial strategy was to use a bounding rectangle to fit our filtered data. Although functional, we desired a more innovative approach to truly capture the room's shape.

This quest led us to experiment with a plethora of techniques. From the alpha shape to multiple variations of the convex hull, each method had its own set of challenges. The convex hull, in particular, was of great interest. We explored shrinking it as a potential filter and even ventured into 3D techniques in hopes of a more accurate wall detection, aiming for a holistic representation of the room.

3. Finding the Exit:

Identifying the exit proved to be one of the most intricate challenges. The noisy points in our maps, a result of reflective elements and walls, added layers of complexities. With these imperfections in mind, our approach was multifaceted. We delved into k-means clustering with density-based exit detection and explored advanced clustering techniques like DBSCAN, HDBSCAN, and OPTICS Clustering.

Further intricacies emerged as we delved into grid-based density calculations on the established boundaries. We also considered identifying the largest gap by focusing on the depth of each wall's z-coordinate. Determining the optimal number of clusters became an essential aspect of our research, leading us to various metrics such as the Davies-Bouldin Index (DBI), Silhouette, and Elbow methods.

Future Research: Real-time Shape Approximation in Dynamic Environments using Convex Hull and Shrinking Methods

Navigating the vast realm of spatial data analysis, our project has spotlighted the potential of the convex hull computation, notably when synergized with the shrinking method. This combination lays the foundation for some compelling areas of future exploration:

1. **Real-time Processing and Analysis**: The agility of the convex hull coupled with the shrinking method positions it uniquely for real-time spatial data interpretations. Such rapid feedback mechanisms can be transformative, especially in fluctuating environments where spatial layouts are ever-evolving. This speed and efficiency make it particularly valuable for applications in robotics, drone navigation, and immersive augmented reality experiences.

2. **Noise and Outlier Management**: The challenge of noisy data, exacerbated by reflective elements in scanned environments, underscores the necessity for sturdy noise and outlier resolution strategies. By dovetailing our convex hull-shrinking approach with sophisticated noise filtration mechanisms, we envision a methodology that delivers cleaner, more precise shape approximations.

Building on these insights, our aim is to push the boundaries of real-time spatial data processing, ensuring both speed and precision, thus catering to the pressing demands of modern dynamic systems.

Conclusion: *Reflecting on Our Efforts and Dedication*

Throughout the duration of this project, our commitment never wavered. We devoted countless hours to the project, with a significant portion of that time spent working diligently in the lab, which became a nexus for our creativity and collaboration.

Our project was a blend of individual initiative and collective input. While we leaned heavily into our own problem-solving abilities, the environment of the university and the lab provided us a unique advantage. We could tap into the collective expertise of our peers: some who were actively collaborating with us on the project, while others, even if working on their separate endeavors, generously offered their insights when approached. This fostered a community of shared knowledge.

Furthermore, our discipline was evident in our consistent communication. Each week, we meticulously compiled a weekly report, ensuring that our progress was transparent and our code was regularly updated.

In essence, our journey was a testament to relentless effort, the power of collaboration, and a pursuit of excellence. Our dedication to these values, alongside the tangible results we have achieved, serves as evidence of our unwavering commitment and the depth of our understanding. We believe these factors combined make a compelling case for the recognition of our efforts.

Throughout the course, our journey was enriched by diverse guidance and insights. On two separate occasions, we had the opportunity to consult with Tsok in the lab. Additionally, today's fruitful interactions with Loay Moalem and Murad Tukan provided us valuable advice, particularly in understanding clusters and refining our density calculations. Our discussions with Prof. Dan Feldman further broadened our perspective. Online research played a pivotal role in our learning curve; it was there that we stumbled upon the concept of trimming filtering. Though inspired, we adapted and reshaped the original idea to fit seamlessly into our program. Our foundational knowledge was amplified by the resources available on the lab's website, which acted as a beacon, guiding us to pertinent academic papers and enlightening sections of books.