ROBOT LEARNING FOR LOOP CLOSURE DETECTION AND SLAM

by

Zachary S. Nahman

A thesis submitted to the Faculty and the Board of Trustees of the Colorado School of Mines in partial fulfillment of the requirements for the degree of Master of Science (Computer Science).

Golden, Colorado

Date _____

Signed: _____
Zachary S. Nahman

Signed: _____
Dr. Hao Zhang
Thesis Advisor

Golden, Colorado

Date _____

Signed: _____
Dr. Tracy Camp
Department Head
Department of Computer Science

ABSTRACT

Robotics and autonomy continues to be a key research and development focus around the world. Robots are increasingly prevalent in everyday life. From manufacturing, home cleaning, to self-driving vehicles, robots are an ever-present reality with demonstrated capability to increase quality of life for humans. As more and more robots exist surrounding humans, it becomes increasingly critical that robots can accurately sense and reason about the environment. The functionality of a robot building a map of its environment and locating itself constantly within the map is known as Simultaneous Localization and Mapping (SLAM). SLAM is a difficult problem, and can be especially challenging when environmental appearance changers occur or when a GPS signal is not available. However, it's within these challenging environments where the use of robots is critical. Consider a partially collapsed underground mine environment. If the environment is potentially dangerous, it doesn't make sense to risk human life to enter the mine to perform search and rescue. If robots can be enabled to operate in challenging environments such as collapsed mines, human life can be saved. This Master's thesis addresses the problem of increasing the effectiveness of SLAM in these challenging environments. First, I describe a data structure capable of capturing environmental metadata for semantic description overlay to augment mapping capability. Secondly, I introduce a novel loop closure detection technique that utilizes robot learning to understand complex environments. These efforts combined contribute to increasing the effectiveness of SLAM in GPS-denied environments or environments with varying lighting conditions.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS

a matrix $\mathbf{U}$ . . . . . $\mathbf{U} = \{u_{ij}\} \in \Re^{m \times n}$

the i-th row of matrix $\mathbf{U}$ . . . . . $\mathbf{u}^i$

the j-th column of matrix $\mathbf{U}$ . . . . . $\mathbf{u}_j$

the Frobenius norm of matrix $\mathbf{U}$ . . . . . $\|\mathbf{U}\|_F = \sqrt{\Sigma_{i=1}^m \Sigma_{j=1}^n u_{ij}^2}$

a vector u . . . . . $\mathbf{u} \in \Re^n$

ℓ_2-norm of u . . . . . $\|\mathbf{u}\|_2 = \sqrt{\mathbf{u}^\top \mathbf{u}}$

ℓ_1-norm of u . . . . . $\|\mathbf{u}\|_1 = \Sigma_{i=1}^n |u_i|$

LIST OF ABBREVIATIONS

Simultaneous Localization and Mapping . . . . . SLAM

Global Positioning Satellite . . . . . GPS

Voxel Based Representation Learning . . . . . VBRL

Light detection and ranging . . . . . LiDAR

Lidar Odometry and Mapping . . . . . LOAM

Scale-Invariant Feature Transform . . . . . SIFT

Bag of Words . . . . . BoW

Binary Robust Independent Elementary Features . . . . . BRIEF

Features from Accelerated Segment Test . . . . . FAST

Oriented FAST and rotated BRIEF . . . . . ORB

Histogram of Oriented Gradients . . . . . HOG

Long Term Appearance Change . . . . . LAC

Shared Representation Appearance Learning . . . . . SRAL

ACKNOWLEDGMENTS

I want to thank my adviser, Dr. Hao Zhang, for his endless support and encouragement. Committee members Dr. Andrew Petruska and Dr. Bo Wu were also integral to my success; thank you!

Being a part of the Human-Centered Robotics lab at Mines has been a wonderful and enriching experience. I've enjoyed the research I was a part of, and I'm interested to see what the lab accomplishes in the future. Thank you to all of the members of the lab, and in particular, thank you to Sriram Siva for truly helping me understand difficult concepts and co-authoring the VBRL paper.

I'd also like to thank my fiancée, Taylor Madden, for always pushing me to be my best. Of course, thanks is also due to my parents, Scott and Karen, and my brother Steve. Without the help of my family I would not be the person I am today. Thank you.

Dedicated to my late Grandfather Norris Nahman (GPN)

Thank you for being a constant inspiration, an awesome grandfather, and the best engineer.

CHAPTER 1
INTRODUCTION

This Master's thesis addresses challenges associated with simultaneous localization and mapping (SLAM) for mobile robots in difficult environments. The focus of the thesis is to improve SLAM techniques in challenging robot environments such as varying lighting conditions and/or GPS-denied environments (locations where a GPS signal is unavailable). A particular focus is placed on utilizing robot learning (or machine learning) techniques in order to enable robots to understand and reason within complex environments. As robots are utilized in more challenging environments (such as search and rescue, industrial inspection and repair, and underground mapping) it is critical for SLAM algorithms to adapt to perform accurately. A robot's ability to navigate in a challenging environment is only as effective as its ability to perceive the environment. To aid in this robot perception, robot learning techniques are explored as well as semantic multi-layer data structures enabling high-quality map representation.

1.1 Background

Background information is included in this section including a brief overview of *SLAM*, *loop closure detection*, and *semantic multi-layer mapping*.

1.1.1 Simultaneous Localization and Mapping (SLAM)

Simultaneous localization and mapping (SLAM) describes a robot's ability to create a map of the environment and localize itself within the created map at the same time. The problem is particularly difficult because building an accurate map requires pinpoint localization, and accurate localization depends directly on quality of map accuracy. Because of this dual-dependency, SLAM is often described as a "chicken and egg" problem.

A common architecture of SLAM solutions is the separation of a front end and a back end. The front end of the SLAM algorithm is responsible for perception and data fusion. The back end is responsible for generating a map and keeping the map updated based on data collected by the front end, and underlying mapping logic.

Because sensor systems are not perfect, mobile robots accumulate incremental error as they travel through an environment. A primary example of back end mapping logic is being able to detect a *loop closure*. A loop closure occurs when the robot recognizes that it's within an area that it has previously encountered. This key recognition allows the back end SLAM algorithms to update the map and reduce accumulated incremental pose drift accrued during navigation.

Currently, SLAM is particularly challenging within environments that have changing lighting conditions or when GPS signal is not available. Utilizing the techniques described in this thesis can help SLAM algorithms operate more effectively given these challenges.

1.1.2 Loop Closure Detection

Loop closure detection (or *place recognition*) enables robots to recognize previously visited locations and correct incremental pose drift accumulated during navigation. Much research has been conducted by comparing similarity in 2D images gathered by robots to enable loop closure detection. However, loop closure detection utilizing only 3 dimensional data such as point clouds is not yet well addressed. 2D cameras can have difficulties in challenging environments. For instance, when underground, 2D cameras may not work if there is low or no light. LiDAR sensors, conversely, can continue to operate in the dark or low lighting conditions and can provide a means by which to perform loop closure detection.

In particular, the loop closure detection technique described in Chapter 3 utilizes a learning method performed on only 3D LiDAR data to address these challenges.

1.2 Semantic Multi-Layer Mapping

Semantic multi-layer mapping describes generating SLAM maps with metadata that can help distinguish unique environments, or track key information recognized while navigating. Consider the case of a robot navigating in an indoor hallway environment such as a school. Hallways are generally void of unique features. Enabling a robot to collect and store specific environmental information can help make up for the lack of unique features. Also, consider the case of a robot designed for industrial inspection and repair. Storing surface defect information in the same data structure as the map can enable the robot to return to previously recognized defects for further repair or inspection.

1.2.1 Objective

The research objective of this Master's thesis is to increase the effectiveness of SLAM in challenging environments by implementing robot learning based methods utilizing 3D data and creating data structures to better represent mapping information and metadata.

1.3 Main Contributions

The contributions I have made towards addressing SLAM in challenging environments are as follows:

1.3.1 Semantic Multi-Layer Mapping

I have implemented a multi-layer data structure capable of storing metadata along with robot pose information obtained during robot navigation. This metadata can be customizeable to be useful in any application ranging from navigation in featureless environments to industrial inspection and repair. (Chapter 2)

1.3.2 Voxel Based Representation Learning

I proposed and implemented a novel loop closure detection technique called Voxel Based Representation Learning (VBRL) that utilizes only 3D LiDAR data and learns a shared

representation of the environment. The shared representation is capable of reasoning about which voxels are most important and which feature extraction modality is most important to solving the place recognition problem for mobile robots. (Chapter 3)

1.4 Guide to the Thesis

This Master's thesis is structured as follows. Chapter 1 provides an introduction and overview of the challenges and content of the thesis. Chapter 2 discusses work enabling semantic multi-layer mapping for storing map metadata alongside pose robot pose information. Chapter 3 details the Voxel-Based Representation Learning (VBRL) method enabling loop closure detection utilizing only 3D LiDAR data. The thesis is concluded in Chapter 4.

CHAPTER 2

SEMANTIC MULTI-LAYER MAPPING

2.1 Introduction

The term semantic mapping can Storing semantic data alongside pose data can be critical to loop closure detection and navigation. For instance, if two point clouds are very similar, perhaps a reference to the metadata can help determine if a loop closure detection is a real match. Perhaps they are very similar in 3D space, but have drastically different metadata; this extended data can help to refute false positives. The storage of this metadata has many crucial applications. Some environments are extremely repetitive and devoid of descriptive landmarks such as a hallway in a school. In this case, being able to recognize landmarks and "tag" them to the map can help to describe an individual scene.

2.2 Related Work

In this section, I describe work related to both *Semantic Labeling/Understanding* and *Mapping*.

2.2.1 Semantic Labeling/Understanding

Storing additional data along with mapping information has been previously studied, and can be broadly categorized into two separate categories. The first category is recognition of semantic labels (often called semantic segmentatation) that are automatically extracted via segmentation algorithms. [1] autonomously labels furniture, drawers, and doors from indoor environments. Multiple semantic segmentation methods utilize deep learning approaches: [2], [3], [4]. While semantic segmentation in point clouds is well studied, my goal within this work is to develop the data structure for capturing the information. These methods can be implemented to enable autonomous detection.

The second category is semantic understanding from human-robot interaction. [5] aims to enable complex object manipulation from human utterances. [6] enables robots to understand directions provided in natural language. Further, [7] implements a voice-controlled forklift that works alongside humans. [8] enables robots to not only recognize objects, but also be able to reason about relevant objects based on human utterances. Human-robot interaction is also an ideal candidate for autonomously detecting semantic information, and these methods are key for implementation.

2.2.2 Mapping

A hallmark of mapping algorithms is the pose graph data structure in which each node in the graph represents a robot pose measurement. Graph based SLAM techniques and optimization are also well studied [9], [10], [11]. The latest release of MATLAB (R2019B) [12] includes a navigation toolbox that implements a pose graph data structure providing a perfect test bed for SLAM research.

2.3 SLAM Package

SLAM solution software is usually divided into a front end and a back end. The front end is responsible for reading sensor data and stitching it together to form a map. The back end is responsible for updating the map based on loop closure detection and optimization, and is often graph-based. To demonstrate the SLAM solution, I construct a SLAM algorithm within MATLAB.

The front end uses "Point-To-Plane" iterative closest point (ICP) which is an iterative point cloud registration algorithm that calculates relative pose by minimizing point to plane error. A scan is considered "accepted" and incorporated into the pose graph data structure if the relative pose is greater than 0.6 meters from the previous. Therefore, the trajectory becomes a series of nodes that are approximately 0.6 meters apart.

The back end has VBRL (discussed in Chapter 3) loop closure detection. When the VBRL algorithm returns a scene match score greater than 0.97 (or 97% similarity), a valid loop

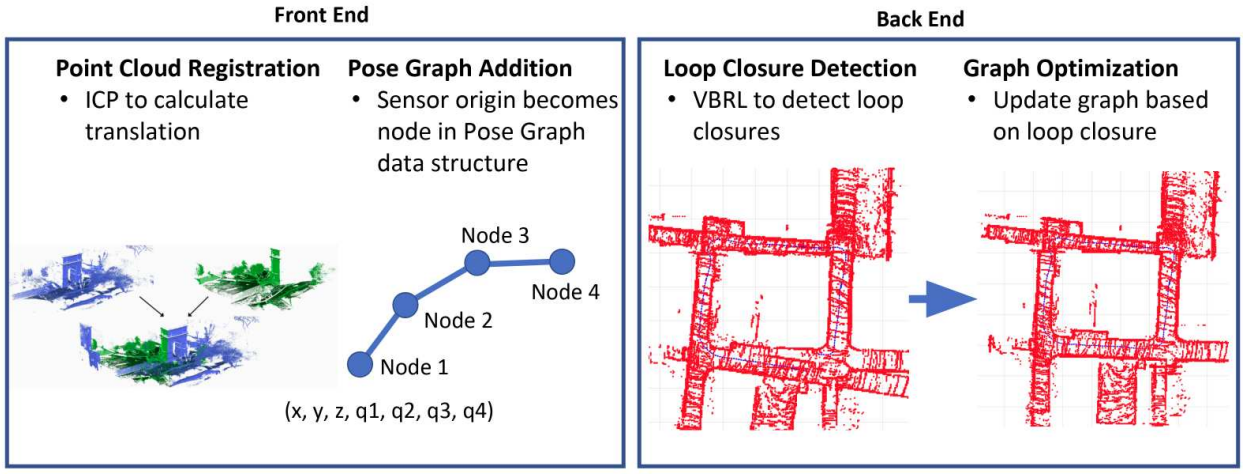


Figure 2.1: Overall SLAM structure. The front end consists of point cloud registration and Pose Graph node addition. The back end consists of VBRL to detect loop closures and Pose Graph Optimization to maintain an updated map.

closure is detected. This detection then triggers pose graph optimization utilizing MATLAB's pose graph optimization function. The pose graph data structure itself is augmented to maintain a field with semantic data. Figure 2.1 shows the overall SLAM pipeline.

2.4 Metadata Representation

A robot *pose* is a vector of size 7 of the form (x, y, z, q1, q2, q3, q4) where (x, y, z) are Cartesian coordinate locations in 3D space and (q1, q2, q3, q4) are quaternion rotations. The pose graph is made up of nodes in which each node represents the LiDAR sensor origin (effectively plotting the robot trajectory through 3D space). The edges between nodes represent the incremental change in trajectory the robot experiences. In the case of point clouds, this incremental change can be calculated with one of many point cloud registration algorithms (such as iterative closest point or normal distribution transform). The pose graph also supports loop closure detection. A node entry in the graph can be specified a loop closure to a matching node. There are built-in functions for optimizing the pose graph to update each and every point in the graph to accurately reflect the loop closure detection. I created a copy of the data structure that keeps track of the pose, and augmented it with an

8th column. The 8th column expects a string input that describes any metadata that it is effective to keep track of. Because it's a string, the field could contain "doorway" in a hallway navigation example or "surface defect" in a industrial setting. The string is customizeable to contain whatever data is important.

2.5 Experimental Results

Performance is evaluated using a data set of LiDAR scans captured from a mobile robot exploring Colorado School of Mines' Brown Hall 2nd floor in a big loop. The data set consists of 3400 LiDAR scans gathered by a robot navigating in the corridor.

2.5.1 Loop Closure Performance

The experiment on Brown Hall 2nd floor confirms that VBRL loop closure detection is a viable algorithm for detecting loop closure, and that the framework and data structure in the MATLAB navigation toolbox work correctly. The left side of Figure 2.2 shows the map when a loop closure is detected. Notice how the map overlaps incorrectly due to the error accumulated while the robot was navigating. Because the loop closure has a score greater than the threshold specified (0.97) the loop closure is accepted as valid. The right side of the image shows the map after it is optimized based on closing the loop. The end result is a more accurate map of Brown Hall 2nd floor.

2.5.2 Semantic Examples

There are three locations in which it is effective to store metadata. 1.) When the robot reaches a doorway. 2.) A designation that a particular room encountered is unexplored. 3.) A designation that a particular hallway continuation is unexplored. The detection of these metadata markers is not programmatic; I merely tell the program exactly when they occur in the data, but it is feasible to imagine that the detection of these metadata points could be implemented automatically relatively easily using object detection algorithms ([4][13][14][15]) on the point cloud data.

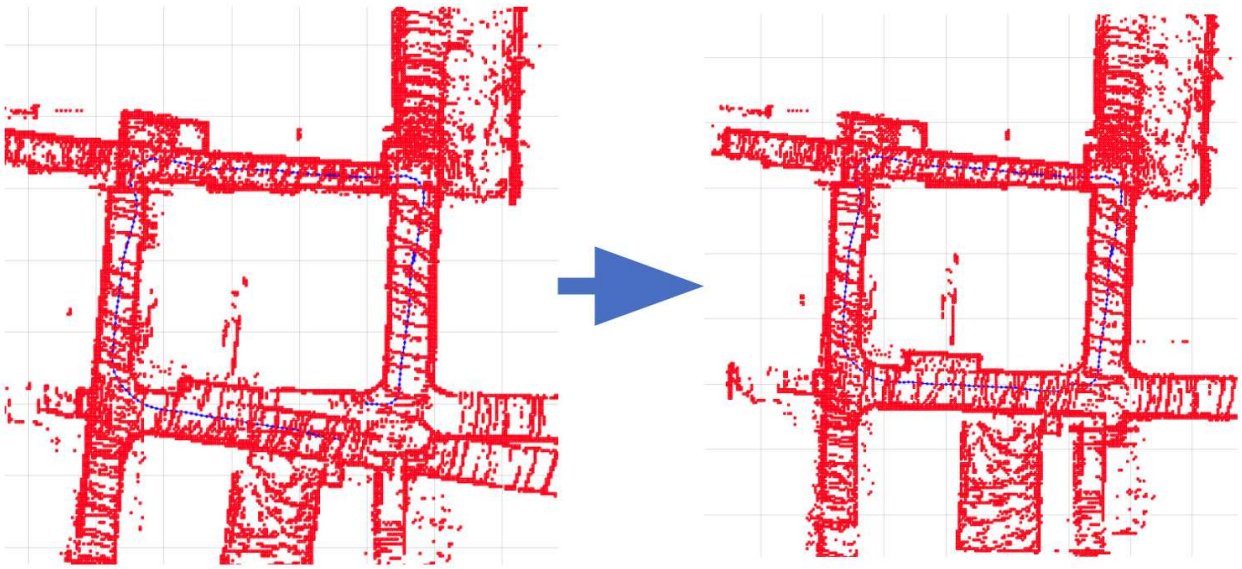


Figure 2.2: Loop closure results using VBRL on data from CSM Brown Hall 2nd floor. The left side shows the map when a loop closure is detected. The right side shows the map after it is updated based on the loop closure.

Figure 2.3 showcases the idea with the left image showing the raw occupancy map generated and the right showing the second layer of information contained within the metadata. There is a bounding box surrounding "doorway", "unexplored room", and "unexplored hallway". Note that just like the detections are not programmatic, the determination of the size of the bounding boxes is also not programmatic. I manually program the size of the bounding boxes for the demonstration as well.

2.6 Conclusion

The data structure exists and is implemented within MATLAB with an 8th column of the pose graph node structure that can be used for storing multi-layer string metadata. This data structure can be combined with other loop closure detection algorithms to further encode the observed robot environment. While the demonstration here does not programatically detect important features or their bounding boxes, these can be feasibily implemented by incorporating state-of-the-art 3D object detection algorithms.

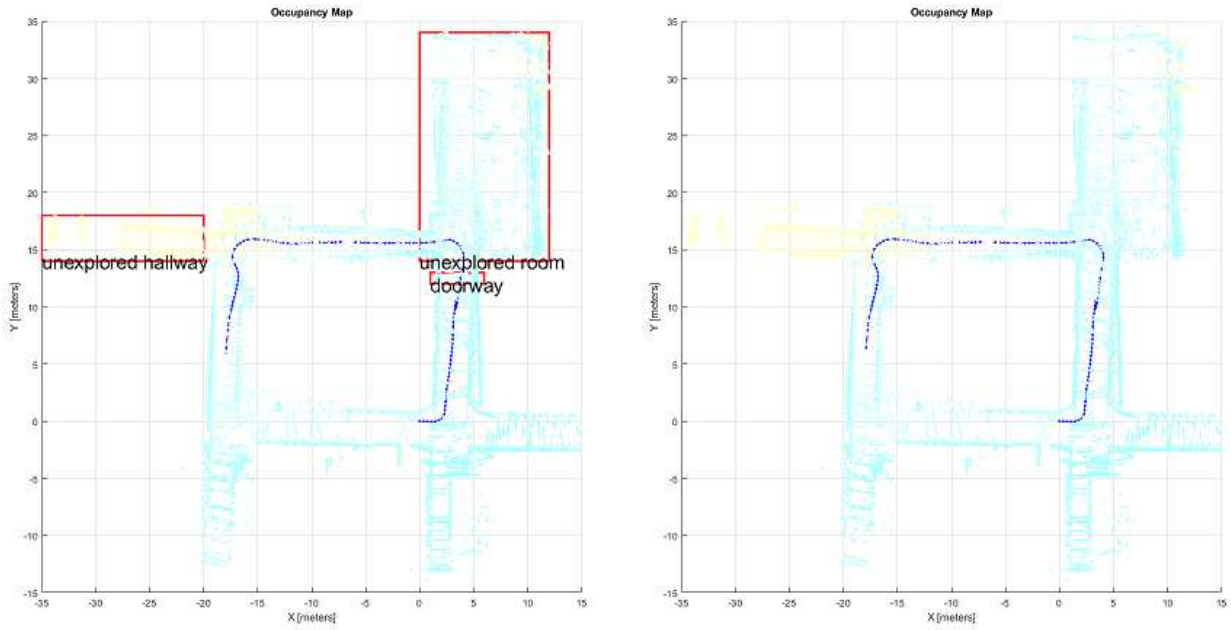


Figure 2.3: Example of semantic data detected and labeled on the point cloud map. The left image has the semantic overlay with "doorway", "unexplored room", and "unexplored hallway" labeled with bounding boxes. The right image does not have the semantic overlay. Robot trajectory is shown in dark blue.

CHAPTER 3

VOXEL BASED REPRESENTATION LEARNING

This work was submitted to IEEE conference: *International Conference on Robotics and Automation* (ICRA) 2020. Included here with permission from authors: Zachary Nahman[1], Sriram Siva[2], Hao Zhang[3]

3.1 Abstract

A critical component of Simultaneous Localization and Mapping (SLAM) is place recognition, which is defined as the capability for a robot to recognize a previously visited place. Most existing approaches are based on visual images, and place recognition using 3D point clouds, especially based on the voxel representation, has not been well addressed yet. In this paper, we introduce the novel approach of voxel-based representation learning (VBRL) that uses 3D point clouds to recognize places with long-term environment variations, such as in weather and vegetation. VBRL divides a 3D point cloud input into voxels and uses multi-modal features extracted from the voxels to perform place recognition. VBRL also uses structured sparsity-inducing norms to learn representative voxels and feature modalities that are important to match places under long-term changes. Both place recognition and voxel and feature learning are integrated into a unified regularized optimization formulation. Because the introduced sparsity-inducing norms are not smooth, it is hard to solve the formulated optimization problem. Thus, we design an iterative algorithm to solve the optimization problem, which has a theoretical convergence guarantee. To evaluate VBRL, we perform experiments based on the AirSim self-driving simulator and the NCLT benchmark dataset. Experimental results have demonstrated that VBRL performs place recognition well

[1]Primary researcher and author, Graduate Student, Department of Computer Science, Colorado School of Mines

[2]Secondary researcher and author, Graduate Student, Department of Computer Science, Colorado School of Mines

[3]Assistant Professor, Department of Computer Science, Colorado School of Mines

using 3D point cloud data and is capable of learning the importance of voxels and feature modalities.

3.2 Introduction

For decades, one of the core robotics challenges has been Simultaneous Localization and Mapping (SLAM). A critical component of SLAM is place recognition (also referred to as loop closure detection). Place recognition is the capability for a robot to recognize a previously visited location. It enables the robot to more accurately localize itself within its global map through correcting incremental pose drifts accumulated during exploration. Place recognition, together with SLAM, has been applied in a wide variety of real-world applications, including assistive robotics [16][17][18], environment exploration [19][20][21], and autonomous driving [22][23][24].

Most previous research utilizes environmental images obtained from a visual camera installed on a robot to perform visual place recognition [25]. Long-term visual place recognition has received extensive attention during the past few years [26]. It addresses the challenge that places are dynamic environments that change over time. For example, indoor places can experience environment changes in human activity, lighting, and arrangement on a daily basis. Outdoor environments can look drastically different from time to time, e.g. the same environment appears quite different in summer versus winter and at morning versus night.

Given the advances in visual place recognition and SLAM, the use of cameras in some environments is difficult or inappropriate. For instance, in the low light condition or complete darkness (e.g., in subterranean environments), utilizing visual images for place recognition would necessitate bringing light sources to illuminate the environment, which may not be feasible all the time. LiDAR sensors can offer a solution to accurately perceive the environment independent of lighting conditions. By actively projecting laser light and measuring the reflected light, LiDAR measures distance to a target and provides a 3D point cloud representation of the environment, which can be used by a robot operating in the dark.Figure 3.1

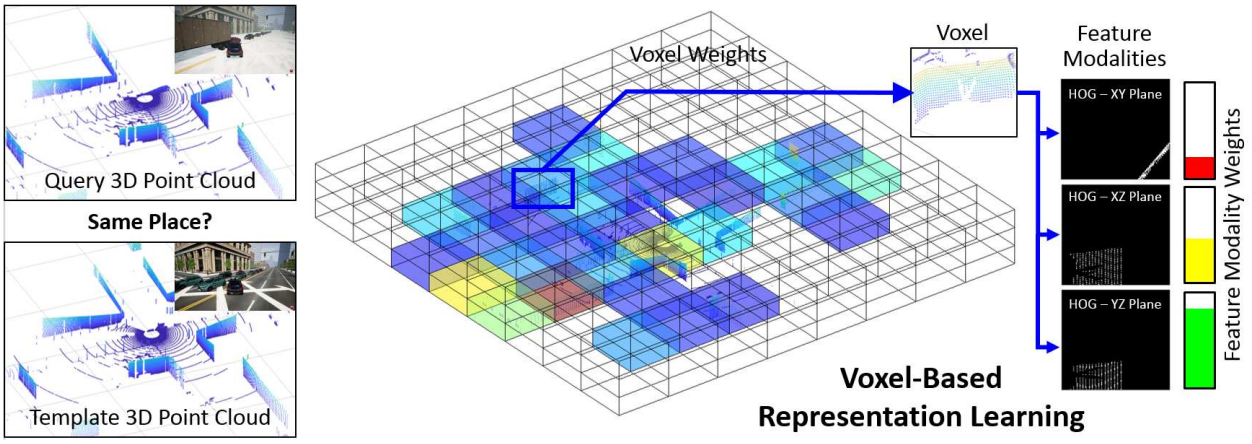


Figure 3.1: Illustration of the proposed VBRL method for place recognition on 3D point cloud data. VBRL divides each 3D point cloud into multiple voxels in the 3D space and extracts multi-modal features from each voxel. Then, VBRL performs joint learning of representative voxels and feature modalities to represent places and integrates the representation for place recognition in a unified regularized optimization formulation.

Although visual place recognition was extensively studied, long-term place recognition based on 3D point clouds (e.g., obtained from LiDAR sensors) has received limited attention. Direct matching of a query scan with a database of previous scans was implemented to determine the best match of places [27]. Keypoint voting [28] and histogram-based matching (e.g., based on normal distributions transform [29]) were also used for LiDAR-based place recognition. However, these previous methods generally rely on manually defined features without using learning. In addition, place recognition methods using 3D point clouds cannot well address long-term environment changes. Furthermore, while voxel-based 3D representations have been commonly utilized in 3D SLAM and robot navigation [30–32], 3D place recognition using voxel-based representations has not been well addressed.

In order to address these challenges, we introduce a novel *Voxel Based Representation Learning* (VBRL) approach for place recognition using 3D point clouds acquired by LiDAR sensors. The 3D data is obtained from a 360 degree field of view by a LiDAR sensor, and our approach divides each 3D point cloud into multiple voxels in the 3D space. Multiple types of features are then extracted from each of the voxels. Given the multiple types

(modalities) of features from all 3D voxels, our proposed VBRL method automatically learns the importance of voxels and feature modalities, and integrates all features in a unified regularized optimization formulation in order to best represent places. For learning the importance of voxels, our approach is inspired by the insight that a subset of voxels are typically more representative to encode a place. For example, voxels closer to the sensor can be more useful, since it can include 3D points that describe the environment with more details as objects are closer to the sensor. Learning voxel importance is achieved by VBRL through introducing structured sparsity-inducing norms as regularizations into the optimization formulation. Similarly, the VBRL approach is able to learn the importance of feature modalities in the same regularized optimization formulation.

There are two main contributions of this paper:

- We propose a novel formulation and the VBRL method to perform simultaneous learning of representative voxels and feature modalities to represent places for place recognition from 3D point cloud data.

- An optimization algorithm is implemented to solve the regularized optimization problem that has a theoretical guarantee to converge to the global optimal solution.

The remainder of this paper is organized as follows. We review related methods for place recognition based on visual images and 3D point clouds in Section 3.3. The VBRL method is introduced in Section 3.4. Experimental results are reported in Section 3.6. Finally, this paper is concluded in Section 3.9.

3.3 Related Work

In this section, we first present the state-of-the-art in long-term visual place recognition, followed by an overview of the latest place recognition techniques utilizing 3D data.

3.3.1 Long-Term Visual Place Recognition

Most of the present state of the art visual place recognition techniques can be broadly classified as methods based on local features, global features or representation learning.

Local features apply a detector to detect points of interest in an image and thus encode the local information. The Scale-Invariant Feature Transform (SIFT) local features were used in combination with a bag-of-words (BoW) approach to detect previously visited places in a 2D image [33]. In [34], binary BRIEF and FAST features are applied to get a BoW representation and perform loop closure detection. ORB features have also been successfully used to perfrom place recognition [35].

Global features portray the holistic representation of the scene. For example, HOG [36] features use unsigned gradient changes within each of the pixels in a grid and stores it in a histogram. GIST [37] features, constructed steerable Gabor filters at different orientation and scales to perform place recognition [38]. Local Binary Pattern are used to represent the whole image using intensity and gradient differnences within the image to calculate a binary string [39]. It has been observed that, while performing long-term place recognition, global features outperform local features [40], [41], [25].

The third category is based upon representation learning. Several techniques utilize convolutional neural networks in order to learn representative features from visual images [42, 43]. In Shared Representation Appearence Learning (SRAL) [26], the authors address the challenges of the LAC problem by using a learning method to learn which visual feature extraction modalities are shared between different scene scenarios. Rather than simple concatenation, the SRAL approach learns which modalities are shared between different instances, such as, which modalities are shared of an environment in both summer and winter.

Although visual place recognition showed promising performance in good lighting conditions, in harsh environments with low lighting, high quality images can be hard to obtain. Place recognition approaches based on LiDAR data becomes necessary in such conditions.

3.3.2 3D Place Recognition

3D place recognition methods fall in four broad categories: global scene descriptors, histogram feature binning, keypoint voting, and learning approaches. Constructing a global scene descriptor and comparing to a database of previously seen locations has been well studied. In [27] and [44], range image similarity is calculated to recognize places. [45] constructs a feature from sparse triangulated landmarks. Other techniques utilize histograms of point cloud features in order to perform place recognition. In [29], histograms of normal distribution transform are constructed based on surface orientation and smoothness. In [46, 47], histograms are constructed of simple global features extracted from LiDAR scans. In [28, 48], keypoints are 3D descriptors that are calculated from a subset of 3D data. The keypoint downsamping and extraction allows matching to be performed quickly. Other approaches utilize local features called segments and implement a segment matching algorithm [49]. There has also been research into extracting learned deep features from point cloud data ([50], [51], [52]). The PointNetVlad approach extracts deep features from the point cloud and also uses a separate deep network for place recognition [53].

Various mapping methods exist including LiDAR odometry and mapping (LOAM) [54]. LOAM uses point to plane techniques to build up a map of the environment, but does not have a place recognition back-end built in. Other LiDAR SLAM methods use the popular extended Kalman filter [55].

The previous methods for 3D place recognition are generally based on manually defined features without representation learning. Long-term variations have also not been explicitly addressed by the methods using 3D point clouds. In addition, while voxel-based 3D representations have been commonly used for 3D SLAM, 3D place recognition using voxel-based representations has not been well addressed.

3.4 The VBRL Approach for Place Recognition from 3D Point Clouds

In this section, we describe the proposed VBRL approach for place recognition from 3D point clouds. In addition, we present our algorithm to solve the formulated regularization optimization problem.

Notations: Matrices are written as boldface capital letters, and vectors are represented using boldface lowercase letters. Given a matrix $\mathbf{U} = \{u_{ij}\} \in \Re^{m \times n}$, we refer to its i-th row and j-th column as $\mathbf{u}^i$ and $\mathbf{u}_j$, respectively. The Frobenius norm of $\mathbf{U}$ is computed by $\|\mathbf{U}\|_F = \sqrt{\Sigma_{i=1}^{m}\Sigma_{j=1}^{n}u_{ij}^2}$. Given a vector $\mathbf{u} \in \Re^n$, its ℓ_2-norm is defined as $\|\mathbf{u}\|_2 = \sqrt{\mathbf{u}^\top \mathbf{u}}$, and its ℓ_1-norm is computed by $\|\mathbf{u}\|_1 = \Sigma_{i=1}^{n}|u_i|$.

3.4.1 Problem Formulation

Given a set of point cloud instances aquired during long-term LiDAR based navigation over different scenarios (e.g., different times of the day, months, and seasons), each point cloud is divided into a set of voxels. Then, multiple types of features are extracted from each of these voxels, where a modality is defined as the features computed from a specific feature descriptor. The multi-modal features extracted from all voxels are denoted as $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \Re^{d \times n}$. $\mathbf{x}_i \in \Re^d$ is the feature vector extracted from all the voxels of the i-th 3D point cloud, which is a concatenation of features from all m modalities, such that $d = \Sigma_{i=1}^{m}\Sigma_{j=1}^{v}d_{ij}$, where d_{ij} is the dimensionality of the i-th feature modality in the j-th voxel, and v is the total number of voxels. The corresponding long-term scenarios (e.g., summer and winter) are represented as $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_n] \in \Re^{n \times c}$, where c denotes the number of scenarios and $\mathbf{y}_i$ is the scenario indicating vector, with each element $y_{ij} \in \{0, 1\}$ denoting that the i-th 3D point cloud is collected from j-th scenario.

Then, we formulate place recognition based on 3D point clouds as a regularized optimization problem:

$$\min_{\mathbf{W}} \mathcal{L}(\mathbf{X}, \mathbf{Y}; \mathbf{W}) + \lambda \mathcal{R}(\mathbf{W}) \tag{3.1}$$

where $\mathcal{L}(.)$ is the loss function, $\mathcal{R}(.)$ is the sparsity-inducing regularization term, and $\lambda \geq 0$ is a trade-off hyperparameter to balance the loss function and the regularization term. The model parameter $\mathbf{W}$ is a weight matrix, which represents the importance of the features in $\mathbf{X}$ to represent the scenarios $\mathbf{Y}$ in general. The loss function is designed to encode the error of using the learned model to represent the scenarios, which can be defined as $\mathcal{L}(\mathbf{X}, \mathbf{Y}; \mathbf{W}) = \min_{\mathbf{W}} \|\mathbf{X}^\top \mathbf{W} - \mathbf{Y}\|_F^2$.

The solution to the optimization problem defined in Eq. (3.1) is $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_c] \in R^{d \times c}$, where $\mathbf{w}_i \in \Re^d$ denotes the weights of features from all views and modalities to represent the i-th scenario. Since $\mathbf{w}_i$ contains the weights of features from m-modalities in all voxels, it can be further denoted as $\mathbf{w}_i = [\mathbf{w}_i^1, \ldots, \mathbf{w}_i^m]^\top$. In addition, since each $\mathbf{w}_i^j$ includes the weights of features (extracted from the j-th modality with respect to the i-th scenario) from all voxels, it can be further divided into v parts as $\mathbf{w}_i^j = [\mathbf{w}_i^{j^1}, \ldots, \mathbf{w}_i^{j^v}] \in \Re^{d_{ij}}$, where $\mathbf{w}_i^{j^k}$ demotes the weights of features extracted from the k-th voxel and j-th modality with respect to the i-th scenario.

3.4.2 Learning Representative Voxels

Our VBRL approach divides a 3D point cloud into voxels to encode the 3D environment. When performing long-term place recognition, we hypothesize that some voxels within the 3D point cloud are more representative than others. For example, voxels that are closer to the LiDAR sensor can be more representative, since it can include more 3D points that describe the place with more details as objects are closer to the sensor. In order to identify representative voxels for place recognition, we introduce a regularization term called a voxel norm. Formally, this norm is a sparsity-inducing norm that can be mathematically defined as $\mathcal{R}_V(\mathbf{W}) = \Sigma_{i=1}^{v} \|\mathbf{W}^i\|_F$. This voxel norm $\mathcal{R}_V$ is used as a regularization term in our optimization formulation to enforce the grouping effect of the multimodal features shared among different scenarios and promote sparsity among different voxels. Then, our problem

formulation becomes:

$$\min_{\mathbf{W}} \mathcal{L}(\mathbf{X}, \mathbf{Y}; \mathbf{W}) + \lambda \mathcal{R}_V(\mathbf{W}) \tag{3.2}$$

3.4.3 Learning Discriminative Feature Modalities

Different feature modalities usually capture different characteristics of a place. Some feature modalities can be more representative to describe a place than others. Thus, it is also beneficial to identify the importance of feature modalities to improve long-term place recognition performance. Accordingly, we also propose a regularization term to identify representative feature modalities under the unified regularized optimization framework, which is named modality norm. It is mathematically defined as

$$\mathcal{R}_M(\mathbf{W}) = \sum_{i=1}^{m} \|\mathbf{W}^i\|_F + \sum_{i=1}^{d} \|\mathbf{w}^i\|_2 \tag{3.3}$$

which is a combination of two structured sparsity-inducing norms. The first term applies the Frobenius norm within each modality and the group ℓ_1-norm across different modalities. By enforcing sparsity among modalities, this term allows the VBRL method to identify representative modalities that have larger weights, and to make the weights of non-representative features to have a value close to 0. The second term in Eq. (3.3) denotes the $\ell_{2,1}$-norm used to enforce the sparsity of the rows of $\mathbf{W}$ and grouping effect of the weights in each row. By enforcing sparsity of individual features, this norm helps recognize representative individual features and assign a zero value to the weights of non-representative features (e.g., from noise).

Incorporating both of the regularization terms to identify representative voxels and feature modalities, our final formulation of learning voxel-based multimodal representations for place recognition can be defined as the following regularized optimization problem:

$$\min_{\mathbf{W}} \mathcal{L}(\mathbf{X}, \mathbf{Y}; \mathbf{W}) + \lambda_1 \mathcal{R}_V(\mathbf{W}) + \lambda_2 \mathcal{R}_M(\mathbf{W}) \tag{3.4}$$

Input : $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_n] \in \Re^{d \times n}$ and $\mathbf{Y} = [\mathbf{y}_1, \cdots, \mathbf{y}_n]^\top \in \Re^{n \times c}$

Let $t = 1$. Initialize $\mathbf{W}(t)$ by solving $\min_{\mathbf{W}} \mathcal{L}(\mathbf{X}, \mathbf{Y}; \mathbf{W})$.

while *not converge* **do**

  Calculate the block diagonal matrix $\mathbf{D}(t+1)$, where the k-th diagonal block of $\mathbf{D}(t+1)$ is $\frac{\mathbf{I}_v}{2\|\mathbf{W}^k\|_F}$.

  Calculate the block diagonal matrix $\tilde{\mathbf{D}}(t+1)$, where each element of the matrix $\tilde{\mathbf{D}}(t+1)$, is given as $\left(\frac{\mathbf{I}_m}{2\|\mathbf{W}^i\|_F} + \frac{1}{2\|\mathbf{w}^i\|_2}\right)$.

  For each $\mathbf{w}_i (1 \leq i \leq c)$, $\mathbf{w}_i(t+1) = \big((\mathbf{X}\mathbf{X})^\top + \lambda_1 \mathbf{D}(t+1) + \lambda_2 \tilde{\mathbf{D}}(t+1)\big)^{-1}(\mathbf{X}\mathbf{y}_i)$.

  $t = t + 1$.

Output: $\mathbf{W} = \mathbf{W}(t) \in \Re^{d \times c}$

where λ_1 and λ_2 denote trade-off hyperparameters to balance the loss function and the structured sparsity-inducing norms.

3.5 Voxel-Based Multimodal Place Recognition

After the formulated regularized optimization problem in Eq. (3.4) is solved based on Algorithm 6, we obtain the optimal weight matrix $\mathbf{W}^*$. Given the feature vector $\mathbf{x} \in \Re^d$ that is extracted from all voxels and feature modalities in a query 3D point cloud, and the feature vector from a template 3D point cloud $\tilde{\mathbf{x}} \in \Re^d$, we compute a similarity score between this query and template poind clouds as follows:

$$s = \sum_{i=1}^{m} \sum_{j=1}^{v} w_M(i) * w_V(j) * (\mathbf{x}_{ij} - \tilde{\mathbf{x}}_{ij}) \tag{3.5}$$

where $\mathbf{x}_{ij}$ denotes the vector features from the i-th modality and the j-th voxel, $w_M(i)$ is sum of all weights of features in the i-th feature modality, and $w_V(j)$ is sum of all weights of features in the j-th vexel. When this score is above a user defined threshold, the query 3D point cloud is determined as a match with the template 3D point cloud.

3.5.1 Optimization Algorithm

The objective function in Eq. (3.4) comprises of three non-smooth regularization terms, which is challenging to solve in general. Accordingly, we implement an iterative algorithm

to solve the formulated regularized optimization problem.

Taking the derivative of the objective function with respect to the columns of $\mathbf{W}$ and setting it to zero gives us:

$$\mathbf{X}\mathbf{X}^\top \mathbf{w}_i - \mathbf{X}\mathbf{y}_i + \lambda_1 \mathbf{D}\mathbf{w}_i + \lambda_2 \tilde{\mathbf{D}}\mathbf{w}_i = 0 \tag{3.6}$$

where $\mathbf{D}$ is a diagonal matrix with the i-th diagonal element defined as $\frac{\mathbf{I}_v}{2\|\mathbf{W}^i\|_F}$, $\tilde{\mathbf{D}}$ denotes a diagonal matrix with the i-th diagonal element defined as $\frac{\mathbf{I}_m}{2\|\mathbf{W}^i\|_F} + \frac{1}{2\|\mathbf{w}^i\|_2}$, and $\mathbf{I}_v$ and $\mathbf{I}_m$ are identity matrices with size v and m respectively. Then, we obtain:

$$\mathbf{w}_i = \left(\mathbf{X}\mathbf{X}^\top + \lambda_1 \mathbf{D} + \lambda_2 \tilde{\mathbf{D}}\right)^{-1} (\mathbf{X}\mathbf{y}_i) \tag{3.7}$$

Since the matrices $\mathbf{D}$ and $\tilde{\mathbf{D}}$ are dependent on the weight matrix $\mathbf{W}$, we implement an iterative algorithm to solve the formulated regularized optimization problem, as described in Algorithm 6, which holds a theoretical guarantee to converge to the global optimal solution.

See supplementary materials[4].

Complexity. Since the optimization problem in Eq. (3.4) is convex, Algorithm 6 converges to the global optimal solution fast. In each of the iterations, computing Step 3 and Step 4 is trivial. Step 5 can be computed through solving a system of linear equations with a quadratic complexity.

3.6 Experimental Results

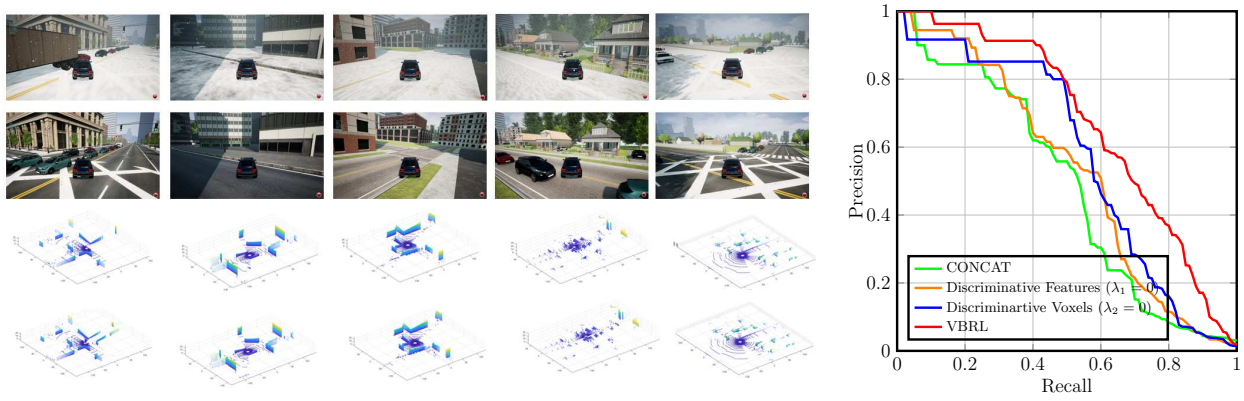
We evaluate the performance of VBRL in a self-driving simulation environment and using a real-word public benchmark dataset of point clouds for long-term place recognition.

In our implementation, each 3D point cloud from LiDAR is divided into voxels. From each voxel five different features are extracted including (1) covariance of points contained within the voxel, (2) Histogram of Oriented Gradients (HOG) features of a snapshot of the point cloud in the XY plane, (3) XZ plane, (4) YZ plane, and (5) Subvoxel Occupancy.

[4]The proof is included: in the Appendix

The subvoxel occupancy feature is obtained simply as dividing a voxel into 8 equal subvoxels. For each subvoxel that is occupied by any points, a 1 is written to the feature matrix. For each subvoxel that is unoccupied by any points, a 0 is written to the feature matrix. As opposed to concatenating these features together from each voxel, VBRL operates with the intuition that learning a shared representation of the overall scene from multiple data instances and weighting the feature matrix accordingly will fuse together the feature modalities more effectively for loop closure detection.

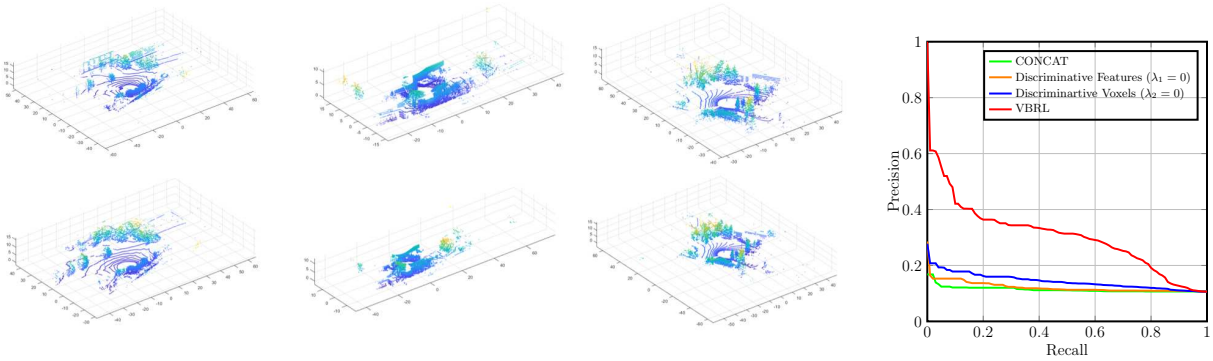
Experiments are evaluated both qualitatively and quantitatively. To showcase that VBRL learns a better representation of a LiDAR scan than feature extraction alone, we compare VBRL ($\lambda_1 = 10$ and $\lambda_2 = 0.1$) to performing loop closure detection with features concatenated together ($\lambda_1 = 0$ and $\lambda_2 = 0$), voxel learning only ($\lambda_1 = 10$ and $\lambda_2 = 0$), and modality learning only ($\lambda_1 = 0$ and $\lambda_2 = 0.1$). A sensitivity analysis for hyperparameter selection is included in the Discussion section. For all experimental results described below, the weight matrix is learned on a disjoint subset of training data. Evaluating performance on a separate (and disjoint) set of testing data ensures performance of VBRL.



(a) Matched point clouds

(b) Precision-recall curves

Figure 3.2: Qualitative and quantitative experimental results over the AirSim simulations.



(a) Matched 3D point clouds (b) Precision-recall curves

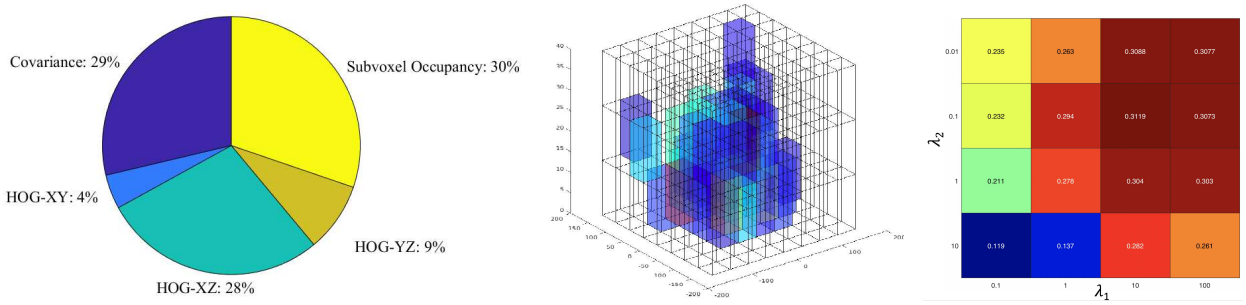
Figure 3.3: Experimental results over the NCLT dataset for long-term recognition in different seasons.

3.6.1 Results in Autonomous Driving Simulation

AirSim [56] is an autonomous driving simulator developed by Microsoft to facilitate development of self-driving vehicle methods in a virtual environment. The dataset is collected in AirSim's city environment which is a virtual cityscape with roads, skyscrapers, parks, and dynamically moving cars and pedestrians. Virtaul LiDAR scans are collected in 210 unique places within the environment. The scans are first collected with clear weather. Then, the same 210 places are visited during snow and fog. 140 point clouds are used for training and 40 are designated for testing.

The main challenges associated with this data set are the dynamic cars and pedestrians as the LiDAR scans are robust to changes in lighting conditions and are not affected by the virtual snow. Place recognition results are shown below in Figure 3.2. The qualitative point cloud scan matches are shown in Figure 3.2(a). The template point clouds from the snow scene that obtain the maximum matching score are shown in the top row, while the query scenes from the clear scene are shown in the bottom row. VBRL is able to match point clouds together despite changes in lighting and weather.

The classification problem is analyzed quantitatively using the standard precision-recall curve. Figure 3.2(b) shows the precision-recall performance of VBRL when compared with features concatenated together, discriminative voxels alone, and discriminative features alone.



(a) Modality importance for AirSim (b) Voxel weights for NCLT (c) Hyperparameter analysis

Figure 3.4: Experimental results over the NCLT dataset in different seasons. Figure 3.4(a) shows the importance of feature modalities for the AirSim simulations. Figure 3.4(b) shows the importance of voxels for long-term place recognition using the NCLT dataset, where the robot is located in the center of the point cloud at position $(0, 0)$. Figure 3.4(c) illustrates the performance variations of our VBRL approach given different hyperparameter values over the NCLT dataset.

VBRL obtains a larger area under the curve indicating it has the best performance. Therefore, by fusing multiple feature modalities together and weighting them based on importance, VBRL yields best results for loop closure detection.

The modality weights learned automatically for the AirSim dataset are shown in Figure 3.4(a). The Subvoxel Occupancy feature is the most important with a weight of 30% and the covariance feature is the second most important with a weight of 29%. The three HOG feature importances range from 4% for HOG-XY to 28% for HOG-XZ.

The learned voxel weights are shown in the color map above in Figure 3.4(b). Voxels occurring more towards the center of the workspace are learned to be weighted as more important in place recognition. This makes sense because the center voxels are most likely to be occupied because they are closest to the sensor origin.

3.7 NCLT Dataset

The North Campus Long Term (NCLT) [57] dataset is collected at the University of Michigan by a mobile robot driven around the campus. There are 27 separate sessions with varying robot routes in the dataset, which occur over the course of 15 months and span multiple times of day and seasons. The dataset contains long-term changes in lighting

condition, vegetation and weather. Two sessions are chosen because they have overlapping routes and seasonal changes: one collected in June and the other in December. A Velodyne HDL-32E LiDAR sensor was used to collect 3D point cloud data on a mobile robot. The NCLT dataset used includes 850 LiDAR scans from a travel in June and a corresponding 850 LiDAR scans from a travel in December. 700 point clouds are used for training and 150 are designated for testing.

Qualitative results are shown in Figure 3.3(a) in which query scans from the June data set are shown on the bottom row and resulting matches are shown in the top row. VBRL is able to recognize scenes from point cloud data despite seasonal, structural changes (such as geometric changes associated with leaves falling off of trees).

Figure 3.3(b) shows the qualitative precision-recall analysis of VBRL on the NCLT dataset. Once again, it is observed that VBRL yields greater area under the precision-recall curve than discriminative voxels, discriminative features, or feature concatenation. Additionally, the learned modality weights obtained are shown. The learned voxel weights are also shown in Figure 3.4(b) and results obtained are similar to the AirSim dataset in that the center voxels are learned to be of more importance than the outer voxels.

3.8 Discussion

Importance of Voxels and Feature Modalities: VBRL has the capability to automatically estimate the importance of each of the voxels and feature modility while training. The relative importance of voxels is illustrated in Figure 3.4(b). It is observed that point clouds near the center are of more important and voxels far away from the center of least importance and thus their weights are close to zero. The importance of feature modalities are illutrated in Figure 3.4(a). The pie chart here indicates the relative importance of different feature modalities towards performing voxel based place recognition. It is observed that Subvoxel occupancy, Covariance and HOG-XZ are equally important in general whereas, HOG-XY is of least importance.

Hyperparameter Selection: The hyperparameters λ_1 and λ_2 in our formulation, Eq. (3.4), are designed to control the strength of regularization norms over learning descriptive voxels and feature modalities respectively. Their optimal values can be determined using cross-validation during training. From the result in Figure 3.4(c), we observe that when $\lambda_1 = 10$ and $\lambda_2 = 0.1$, VBRL statistically obtains the best accuracy. In general, the range $\lambda_1 \in \{1, 100\}$ and $\lambda_2 \in \{0.01, 1\}$ can result in satisfactory results, which demonstrates that both regularization terms are useful to improve performance.

High-Speed Processing: Because of VBRL's ability to learn discriminative representative voxels and feature modalities in a unified approach and fuse them to perform point cloud based place recognition along with the efficiency of the proposed objective function, VBRL approach can acheive high-speed onboard processing. In order to validate this added advantage, we performed additional experiments in the *NCLT* dataset using the CPU implementation on an Intel i7-8700K 3.7Ghz computer. Without considering the time of feature extraction for the five features (Covariance, HOG-XY, YZ, XZ and Subvoxel Occupancy), a processing speed of 80 Hz was obtained. While taking feature extraction time into consideration, a processing speed of 14 Hz was observed. These results indicate the promise of real time voxel-based place recognition in several robotics applications.

3.9 Conclusion

In this paper, we study the key problem of long-term place recognition using 3D point clouds, through proposing a novel Voxel Based Representation Learning (VBRL) method. Our approach divides each 3D point cloud into multiple voxels in the 3D space and extracts multiple modalities of features from each of the voxels. Then, our VBRL approach performs joint learning of representative voxels and feature modalities to represent places and integrates the representation for place recognition in a unified regularized optimization formulation. Due to non-smooth sparsity-inducing norms, the formulated optimization problem is hard to solve. We design an iterative solver that has a convergence guarantee. Experiment results have validated that VBRL obtains promising performance on long-term place

recognition using 3D point clouds.

CHAPTER 4

CONCLUSION

Robots remain an ever-present reality, and their use in critical applications is on the rise. Successfully enabling robots to operate in challenging environments, particularly environments with varying lighting conditions or no GPS signal, can greatly expand current robot domains. There are key applications of robotics in safety-critical roles that require the ability to operate robustly, and the work in this thesis attempts to bridge the gap towards successful operating in challenging environments.

The semantic multi-layer data structure enables maintaining additional metadata along with robot pose information. This metadata can help keep track of landmark information, and also be used along with loop closure detection to provide more information about the uniqueness of a particular scene.

The Voxel Based Representation Learning method can enable loop closure detection when GPS data is unavailable or when lighting is not consistent. Enabling this critical SLAM component in GPS-denied environments can increase the effectiveness of robots in challenging environments. The effectiveness of learning methods in this application have been demonstrated to show significant promise.

The efforts described in this thesis contribute to increasing the effectiveness of simultaneous localization and mapping in challenging environments. The impact of this improvement can be monumental depending on application. For instance, if VBRL can be used by self-driving cars for place recognition and loop closure detection, it can contribute to robots that are safety-cricial. In addition, robotic underground search and rescue operations can be feasible so long as robots can better reason about challenging environments.

4.1 Future Work

While this thesis is complete, there exists future work that expands upon my efforts that the Human-Centered Robotics lab at Mines can continue. To be clear, the future work will not be completed by myself as my graduation is scheduled for December 2019 (pending successful defense of this thesis work).

Perhaps the ideal potential future work is to test the SLAM solution against more challenging data. In Chapter 2 I demonstrate the SLAM solution using data from Colorado School of Mines Brown Building 2nd floor. This dataset is not particularly challenging from a robot perception perspective. It would be much more powerful to demonstrate the efficacy of the SLAM package on challenging data (potentially, on data collected from the EDGAR experimental mine).

In addition, another ideal future work is to further explore the VBRL method. Perhaps there are additional feature representations that can be extracted from point clouds that better encode unique, discriminative features. The learning method may find that these new features are more important than the five that I have benchmarked.

Finally, an example for future work is enabling the SLAM package to work in real-time on a real-world robot. While the lab environment is great for demonstrating capability, it would be far more powerful to demonstrate a real-world robot equipped with the semantic multi-layer data structures and VBRL loop closure detection.

REFERENCES CITED

[1] Nico Blodow, Lucian Cosmin Goron, Zoltan-Csaba Marton, Dejan Pangercic, Thomas Rühr, Moritz Tenorth, and Michael Beetz. Autonomous semantic mapping for robots performing everyday manipulation tasks in kitchen environments. *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011.

[2] Yuan Wang, Tianyue Shi, Peng Yun, Lei Tai, and Ming Liu. Pointseg: Real-time semantic segmentation based on 3d lidar point cloud. In *arXiv 1807.06288*, 2018.

[3] Mohammed Abdou, Mahmoud Elkhateeb, Ibrahim Sobh, and Ahmad El Sallab. End-to-end 3d-pointcloud semantic segmentation for autonomous driving. *CoRR*, abs/1906.10964, 2019. URL `http://arxiv.org/abs/1906.10964`.

[4] Martin Engelcke, Dushyant Rao, Dominic Zeng Wang, Chi Hay Tong, and Ingmar Posner. Vote3deep: Fast object detection in 3D point clouds using efficient convolutional neural networks. In *International Conference on Robotics and Automation*, 2017.

[5] Rohan Paul, Jacob Arkin, Derya Aksaray, Nicholas Roy, and Thomas M. Howard. Efficient grounding of abstract spatial concepts for natural language interaction with robot platforms. *The International Journal of Robotics Research*, 37(10):1269–1299, 2018. doi: 10.1177/0278364918777627. URL `https://doi.org/10.1177/0278364918777627`.

[6] T. Kollar, S. Tellex, D. Roy, and N. Roy. Toward understanding natural language directions. In *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2010.

[7] Matthew R. Walter, Matthew Antone, Ekapol Chuangsuwanich, Andrew Correa, Randall Davis, Luke Fletcher, Emilio Frazzoli, Yuli Friedman, James Glass, Jonathan P. How, Jeong hwan Jeon, Sertac Karaman, Brandon Luders, Nicholas Roy, Stefanie Tellex, and Seth Teller. A situationally aware voice-commandable robotic forklift working alongside people in unstructured outdoor environments. *Journal of Field Robotics*, 32(4): 590–628, 2015. doi: 10.1002/rob.21539. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21539`.

[8] Siddharth Patki, Ethan Fahnestock, Thomas M. Howard, and Matthew R. Walter. Language-guided semantic mapping and mobile manipulation in partially observable environments. In *arXiv 1910.10034*, 2019.

[9] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard. A tutorial on graph-based slam. *IEEE Intelligent Transportation Systems Magazine*, 2(4):31–43, winter 2010. doi: 10.1109/MITS.2010.939925.

[10] Joshua G. Mangelson, Jinsun Liu, Ryan M. Eustice, and Ram Vasudevan. Guaranteed globally optimal planar pose graph and landmark SLAM via sparse-bounded sums-of-squares programming. *CoRR*, abs/1809.07744, 2018. URL `http://arxiv.org/abs/1809.07744`.

[11] Soonhac Hong and Cang Ye. A pose graph based visual slam algorithm for robot pose estimation. In *2014 World Automation Congress (WAC)*, 2014.

[12] MATLAB R2019B. *Navigation Toolbox*. The MathWorks, Inc., Natick, Massachusetts, 2019.

[13] Shuran Song and Jianxiong Xiao. Sliding shapes for 3D object detection in depth images. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 634–651, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10599-4.

[14] Dominic Zeng Wang and Ingmar Posner. Voting for voting in online point cloud object detection. In *Robotics: Science and Systems*, 2015.

[15] Shuran Song and Jianxiong Xiao. Deep sliding shapes for amodal 3D object detection in rgb-d images. pages 808–816, 06 2016. doi: 10.1109/CVPR.2016.94.

[16] Lili Meng, C. W. de Silva, and Jie Zhang. 3D visual slam for an assistive robot in indoor environments using rgb-d cameras. In *International Conference on Computer Science Education*, 2014.

[17] Matt Bailey, Andrew Chanler, Bruce Maxwell, Mark Micire, Katherine Tsui, and Holly Yanco. Development of vision-based navigation for a robotic wheelchair. In *International Conference on Rehabilitation Robotics*, 2007.

[18] Celso De la Cruz, Teodiano Freire Bastos, Fernando A Auat Cheein, and Ricardo Carelli. Slam-based robotic wheelchair navigation system designed for confined spaces. In *2010 IEEE International Symposium on Industrial Electronics*, pages 2331–2336. IEEE, 2010.

[19] S. S. Belavadi, R. Beri, and V. Malik. Frontier exploration technique for 3D autonomous slam using k-means based divisive clustering. In *2017 Asia Modelling Symposium (AMS)*, pages 95–100, Dec 2017. doi: 10.1109/AMS.2017.23.

[20] Robert Sim and Nicholas Roy. Global a-optimal robot exploration in slam. In *International Conference on Robotics and Automation*, 2005.

[21] Frederic Bourgault, Alexei A Makarenko, Stefan B Williams, Ben Grocholsky, and Hugh F Durrant-Whyte. Information based adaptive robotic exploration. In *International Conference on Intelligent Robots and Systems*, 2002.

[22] A. Singandhupe and H. La. A review of slam techniques and security in autonomous driving. In *International Conference on Robotic Computing (IRC)*, 2019.

[23] Ryan W Wolcott and Ryan M Eustice. Visual localization within lidar maps for automated urban driving. In *International Conference on Intelligent Robots and Systems*, 2014.

[24] Guillaume Trehard, Evangeline Pollard, Benazouz Bradai, and Fawzi Nashashibi. On line mapping and global positioning for autonomous driving in urban environment based on evidential slam. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 814–819. IEEE, 2015.

[25] Stephanie M. Lowry, Niko Sünderhauf, Paul Newman, John J. Leonard, David D. Cox, Peter I. Corke, and Michael Milford. Visual place recognition: A survey. *IEEE Transactions on Robotics*, 32:1–19, 2016.

[26] Fei Han, Xue Yang, Yiming Deng, Mark Rentschler, Dejun Yang, and Hao Zhang. SRAL: Shared representative appearance learning for long-term visual place recognition. *IEEE Robotics and Automation Letters*, 2017.

[27] Slawomir Grzonka, Bastian Steder, and Wolfram Burgard. 3D place recognition and object detection using a small-sized quadrotor. In *Robotics: Science and Systems*, 2011.

[28] Michael Bosse and Robert Zlot. Place recognition using keypoint voting in large 3D lidar datasets. In *International Conference on Robotics and Automation*, 2013.

[29] Martin Magnusson, Henrik Andreasson, Andreas Nuchter, and Achim J Lilienthal. Appearance-based loop detection from 3d laser data using the normal distributions transform. In *International Conference on Robotics and Automation*, 2009.

[30] J. Ryde and J. A. Delmerico. Extracting edge voxels from 3D volumetric maps to reduce map size and accelerate mapping alignment. In *Conference on Computer and Robot Vision*, 2012.

[31] Inwook Shim, Yungeun Choe, and Myung Jin Chung. 3D mapping in urban environment using geometric featured voxel. In *International Conference on Ubiquitous Robots and Ambient Intelligence*, 2011.

[32] Z. Liu, H. Chen, H. Di, Y. Tao, J. Gong, G. Xiong, and J. Qi. Real-time 6d lidar slam in large scale natural terrains for ugv. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 662–667, June 2018. doi: 10.1109/IVS.2018.8500641.

[33] Adrien Angeli, David Filliat, Stéphane Doncieux, and Jean-Arcady Meyer. A fast and incremental method for loop-closure detection using bags of visual words. *IEEE Transactions on Robotics*, pages 1027–1037, 2008.

[34] Dorian Gálvez-López and Juan D Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, 2012.

[35] Raúl Mur-Artal and Juan D Tardós. Fast relocalisation and loop closing in keyframe-based slam. In *International Conference on Robotics and Automation (ICRA)*, 2014.

[36] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. 2005.

[37] Aude Oliva and Antonio Torralba. Building the gist of a scene: The role of global image features in recognition. *Progress in brain research*, 155:23–36, 2006.

[38] Niko Sünderhauf and Peter Protzel. Brief-gist-closing the loop by simple means. In *International Conference on Intelligent Robots and Systems*, 2011.

[39] Roberto Arroyo, Pablo F Alcantarilla, Luis M Bergasa, and Eduardo Romera. Towards life-long visual localization using an efficient matching of binary sequences from images. In *International Conference on Robotics and Automation (ICRA)*, 2015.

[40] Michael J Milford and Gordon F Wyeth. Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In *International Conference on Robotics and Automation*, 2012.

[41] Tayyab Naseer, Luciano Spinello, Wolfram Burgard, and Cyrill Stachniss. Robust visual robot localization across seasons using network flows. In *Conference on Artificial Intelligence*, 2014.

[42] Niko Sünderhauf, Sareh Shirazi, Adam Jacobson, Feras Dayoub, Edward Pepperell, Ben Upcroft, and Michael Milford. Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free. In *Robotics: Science and Systems*, 2015.

[43] Zetao Chen, Obadiah Lam, Adam Jacobson, and Michael Milford. Convolutional neural network-based place recognition. *arXiv preprint arXiv:1411.1509*, 2014.

[44] Bastian Steder, Giorgio Grisetti, and Wolfram Burgard. Robust place recognition for 3D range data based on point features. pages 1400–1405, 05 2010. doi: 10.1109/ROBOT.2010.5509401.

[45] Titus Cieslewski, Elena Stumm, Abel Gawel, Mike Bosse, Simon Lynen, and Roland Siegwart. Point cloud descriptors for place recognition using sparse visual information. In *International Conference on Robotics and Automation*, 2016.

[46] Timo Rhling, Jennifer Mack, and Dirk Schulz. A fast histogram-based similarity measure for detecting loop closures in 3-d lidar data. 2015.

[47] Ehsan Fazl-Ersi and John K Tsotsos. Histogram of oriented uniform patterns for robust place recognition and categorization. 31(4):468–483, 2012.

[48] Robert Zlot and Michael Bosse. Place recognition using keypoint similarities in 2d lidar maps. In *Experimental Robotics*, pages 363–372. Springer, 2009.

[49] Renaud Dube, Daniel Dugas, Elena Stumm, Juan Nieto, Roland Siegwart, and Cesar Cadena. Segmatch: Segment based place recognition in 3D point clouds. In *International Conference on Robotics and Automation*, 2017.

[50] Huan Yin, Xiaqing Ding, Li Tang, Yue Wang, and Rong Xiong. Efficient 3D lidar based loop closing using deep neural network. In *International Conference on Robotics and Biomimetics*, 2017.

[51] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3D classification and segmentation. In *Conference on Computer Vision and Pattern Recognition*, 2017.

[52] Gil Elbaz, Tamar Avraham, and Anath Fischer. 3D point cloud registration for localization using a deep neural network auto-encoder. In *Conference on Computer Vision and Pattern Recognition*, 2017.

[53] Mikaela Angelina Uy and Gim Hee Lee. Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition. *CoRR*, abs/1804.03492, 2018.

[54] Ji Zhang and Sanjiv Singh. Loam: Lidar odometry and mapping in real-time. In *Robotics: Science and Systems*, 2014.

[55] Sebastian Hening, Corey A Ippolito, Kalmanje S Krishnakumar, Vahram Stepanyan, and Mircea Teodorescu. 3D lidar slam integration with gps/ins for uavs in urban gps-degraded environments. In *AIAA Information Systems-AIAA Infotech@ Aerospace*, page 0448. 2017.

[56] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, 2017.

[57] Nicholas Carlevaris-Bianco, Arash K. Ushani, and Ryan M. Eustice. University of Michigan North Campus long-term vision and lidar dataset. *International Journal of Robotics Research*, 35(9):1023–1035, 2015.

APPENDIX

VOXEL BASED REPRESENTATION LEARNING - SUPPLEMENTARY MATERIAL

A.1 Proof of Theorem 1

In this section, we prove that Algorithm 1 (in Chapter 3) decreases the value of the objective function with each iteration and converges to the global optimal value. But first, we present a lemma:

For any two given vectors $\mathbf{u}$ and $\tilde{\mathbf{u}}$, the following inequality relation holds: $\|\tilde{\mathbf{u}}\|_2 - \frac{\|\tilde{\mathbf{u}}\|_2^2}{2\|\mathbf{u}\|_2} \leq \|\mathbf{u}\|_2 - \frac{\|\mathbf{u}\|_2^2}{2\|\mathbf{u}\|_2}$.

We have:

$$(\|\widetilde{\mathbf{u}}\|_2 - \|\mathbf{u}\|_2)^2 \leq 0 \tag{A.1}$$

$$-\|\widetilde{\mathbf{u}}\|_2^2 - \|\mathbf{u}\|_2^2 + 2\|\widetilde{\mathbf{u}}\|_2\|\mathbf{u}\|_2 \leq 0 \tag{A.2}$$

$$2\|\widetilde{\mathbf{u}}\|_2\|\mathbf{u}\|_2 - \|\widetilde{\mathbf{u}}\|_2^2 \leq \|\mathbf{u}\|_2^2 \tag{A.3}$$

$$\|\widetilde{\mathbf{u}}\|_2 - \frac{\|\widetilde{\mathbf{u}}\|_2^2}{2\|\mathbf{u}\|_2} \leq \|\mathbf{u}\|_2 - \frac{\|\mathbf{u}\|_2^2}{2\|\mathbf{u}\|_2} \tag{A.4}$$

From Lemma 1, we can derive the following corollary:

For any two given matrices $\mathbf{U}$ and $\tilde{\mathbf{U}}$, the following inequality relation holds: $\|\tilde{\mathbf{U}}\|_F - \frac{\|\tilde{\mathbf{U}}\|_F^2}{2\|\mathbf{U}\|_F} \leq \|\mathbf{U}\|_F - \frac{\|\mathbf{U}\|_F^2}{2\|\mathbf{U}\|_F}$.

Algorithm 1 converges to the optimal solution to the optimization problem in Eq. (4) of the main paper

From Algorithm 1,

$$\begin{aligned}\mathbf{W}(t+1) = \min_{\mathbf{W}} \|\mathbf{X}^\top \mathbf{W} - \mathbf{Y}\|_F^2 \\ +\lambda_1 Tr \mathbf{W}^\top \mathbf{D}(t+1)\mathbf{W} + \lambda_2 Tr \mathbf{W}^\top \tilde{\mathbf{D}}(t+1)\mathbf{W}.\end{aligned} \tag{A.5}$$

Then, we can derive that

$$\begin{aligned} & \mathcal{J}(t+1) + \lambda_1 Tr \mathbf{W}^\top(t+1)\mathbf{D}(t+1)\mathbf{W}(t+1) \\ & +\lambda_2 Tr \mathbf{W}^\top(t+1)\tilde{\mathbf{D}}(t+1)\mathbf{W}(t+1) \\ \leq\ & \mathcal{J}(t) + \lambda_1 Tr \mathbf{W}^\top(t)\mathbf{D}(t+1)\mathbf{W}(t) \\ & +\lambda_2 Tr \mathbf{W}^\top(t)\tilde{\mathbf{D}}(t+1)\mathbf{W}(t), \end{aligned}$$

where $\mathcal{J}(t) = \|\mathbf{X}^\top \mathbf{W}(t) - \mathbf{Y}\|_F^2$.

After substituting the definition of $\mathbf{D}$ and $\tilde{\mathbf{D}}$, we obtain

$$\begin{aligned} & \mathcal{J}(t+1) + \lambda_1 \sum_{i=1}^{v} \frac{\|\mathbf{W}^i(t+1)\|_F^2}{2\|\mathbf{W}^i(t)\|_F} \\ & +\lambda_2 \Big(\sum_{i=1}^{m} \frac{\|\mathbf{W}^i(t+1)\|_F^2}{2\|\mathbf{W}^i(t)\|_F} + \sum_{i=1}^{d} \frac{\|\mathbf{w}^i(t+1)\|_2^2}{2\|\mathbf{w}^i(t)\|_2} \Big) \end{aligned} \tag{A.6}$$

$$\begin{aligned} \leq\ & \mathcal{J}(t) + \lambda_1 \sum_{i=1}^{v} \frac{\|\mathbf{W}^i(t)\|_F^2}{2\|\mathbf{W}^i(t)\|_F} \\ & +\lambda_2 \Big(\sum_{i=1}^{m} \frac{\|\mathbf{W}^i(t)\|_F^2}{2\|\mathbf{W}^i(t)\|_F} + \sum_{i=1}^{d} \frac{\|\mathbf{w}^i(t)\|_2^2}{2\|\mathbf{w}^i(t)\|_2} \Big) \end{aligned} \tag{A.7}$$

From Lemma 1 and Corollary 1, we can derive that

$$\begin{aligned} & \sum_{i=1}^{v} \|\mathbf{W}^i(t+1)\|_F - \sum_{i=1}^{v} \frac{\|\mathbf{W}^i(t+1)\|_F^2}{2\|\mathbf{W}^i(t)\|_F} \leq \\ & \sum_{i=1}^{v} \|\mathbf{W}^i(t)\|_F - \sum_{i=1}^{v} \frac{\|\mathbf{W}^i(t)\|_F^2}{2\|\mathbf{W}^i(t)\|_F}. \end{aligned} \tag{A.8}$$

,

$$\sum_{i=1}^{m} \|\mathbf{W}^i(t+1)\|_F - \sum_{i=1}^{m} \frac{\|\mathbf{W}^i(t+1)\|_F^2}{2\|\mathbf{W}^i(t)\|_F} \leq \sum_{i=1}^{m} \|\mathbf{W}^i(t)\|_F - \sum_{i=1}^{m} \frac{\|\mathbf{W}^i(t)\|_F^2}{2\|\mathbf{W}^i(t)\|_F}, \tag{A.9}$$

and,

$$\sum_{i=1}^{d} \|\mathbf{w}^i(t+1)\|_2 - \sum_{i=1}^{d} \frac{\|\mathbf{w}^i(t+1)\|_2^2}{2\|\mathbf{w}^i(t)\|_2} \leq \sum_{i=1}^{d} \|\mathbf{w}^i(t)\|_2 - \sum_{i=1}^{d} \frac{\|\mathbf{w}^i(t)\|_2^2}{2\|\mathbf{w}^i(t)\|_2}, \tag{A.10}$$

Adding Eq. (A.6)-(A.10) on both sides, we have

$$\begin{aligned} & \mathcal{J}(t+1) + \lambda_1 \sum_{i=1}^{v} \|\mathbf{W}^i(t+1)\|_F & \text{(A.11)} \\ & + \lambda_2 \Big(\sum_{i=1}^{m} \|\mathbf{W}^i(t+1)\|_F + \sum_{i=1}^{d} \|\mathbf{w}^i(t+1)\|_2 \Big) \\ \leq \; & \mathcal{J}(t) + \lambda_1 \sum_{i=1}^{v} \|\mathbf{W}^i(t)\|_F & \text{(A.12)} \\ & + \lambda_2 \Big(\sum_{i=1}^{m} \|\mathbf{W}^i(t)\|_F + \sum_{i=1}^{d} \|\mathbf{w}^i(t)\|_2 \Big) \end{aligned}$$

Eq. (A.11) decreases the value of the objective function with each iteration. As our objective function is convex, Algorithm 1 converges to the optimal value. Therefore, Algorithm 1 converges to the optimal solution to the optimization problem in Eq. (4) of the main paper.