Universidad Zaragoza

Master Thesis

Multi-modal Place Recognition in Aliased and Low-Texture Environments

Author

Alberto García Hernández

Supervisors

Riccardo Giubilato

Klaus Strobl

Javier Civera

ESCUELA DE INGENIERÍA Y ARQUITECTURA
MSc. Robotics, Graphics and Computer Vision, 2023

TRABAJOS DE FIN DE GRADO / FIN DE MÁSTER

Escuela de Ingeniería y Arquitectura Universidad Zaragoza

DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe remitirse a seceina@unizar.es dentro del plazo de depósito)

D./Dª. Alberto García Hernández ,

en aplicación de lo dispuesto en el art. 14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de Estudios de la titulación de Máster Universitario en Robótica, Gráficos y Visión por Computad (Título del Trabajo)

Multi-modal Place Recognition in Aliased and Low-Texture Environments
Reconocimiento multimodal de lugares en entornos sin textura

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, 1 de Junio 2023

Fdo: Alberto García

ABSTRACT

In planetary environments with extreme visual aliasing, traditional place recognition systems for robots encounter difficulties in unstructured and aliased environments. Effective place recognition is essential for robust localization and mapping, which, in turn, significantly impacts the performance of Simultaneous Localization and Mapping (SLAM) systems. This research aims to enhance existing place recognition systems by utilizing both LiDAR and visual information, improving performance in extreme environments. The use of LiDAR is crucial, as it provides valuable geometric data that complements visual data, resulting in more expressive and robust 3D grounded global features.

We evaluated our methods using the Mt. Etna dataset and a synthetic dataset generated with the OAISYS tool. Our comprehensive review of state-of-the-art place recognition systems led to the development of a novel UMF (Unifying Local and Global Multimodal Features with Transformers) model, specifically designed for place recognition in environments with extreme aliasing. The UMF model integrates elements from the most advanced methods, enhancing performance in challenging environments by capturing intricate relationships between local and global context in both LiDAR and visual data.

Two variants of the UMF model were explored, offering alternative ways of processing and utilizing fine local features. Our UMF model outperforms other state-of-the-art methods in place recognition tasks, demonstrating the project's success. The improved place recognition capabilities offered by the UMF model can contribute to more accurate and robust SLAM systems, enabling robots to better navigate and explore unstructured and aliased environments.

This research highlights the importance of multi-modal fusion, particularly the integration of LiDAR and visual data, in addressing the challenges of place recognition in aliased and low-texture environments. It also opens an exciting line of research focus in unified fusion multimodal approaches for robotics, computer vision, and machine learning applications, with a direct impact on SLAM and other related fields.

Acknowledgment

Firstly, I would like to extend my deepest gratitude to Riccardo and Klaus for providing me with the opportunity to work with real-world robotics, particularly in the exciting field of place recognition for SLAM in planetary-like environments. Their constant guidance and support has been instrumental in shaping my understanding and perspective in the topic. The freedom and trust they placed in me when deciding the topic of my thesis was truly empowering.

I would also like to extend my appreciation to Javier for providing constant support, and offering valuable insights, especially during the writing of this thesis. His expert feedback and encouragement to strive for novel ideas have been a great source of inspiration for me. I couldn't have asked for better mentors and guides in this journey.

Lastly, I want to thank the entire team at DLR, with a special mention to Laura, for their unwavering support during these months. The willingness of the team to share their knowledge in robotics has greatly enriched my learning experience. The insights gained from our discussions have not only improved my thesis but also deepened my understanding and interest in the field. Thank you for being a part of this journey with me.

Index

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

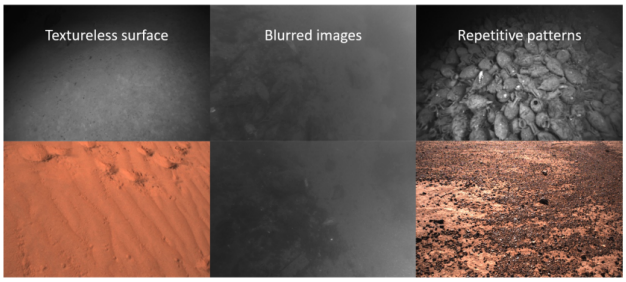
Simultaneous Localization and Mapping (SLAM) has emerged as a central technology in a multitude of industries including autonomous driving [14],[15], automated construction [16], and agriculture [17],[18]. Its development and adoption have been largely catalyzed by advances in sensor technologies, such as multi-camera setups, RGB-D sensors, and more recently, 3D Light Detection and Ranging (LiDAR) sensors. The integration of 3D LiDAR sensors into mobile robots, in particular, has facilitated the construction of large-scale and robust maps, especially in urban and artificial environments.

Yet, despite the extensive research and numerous methodologies developed for visual or LiDAR-based SLAM [19][20], the challenge of SLAM is far from being fully addressed. This becomes particularly evident in environments lacking structured features and exhibiting severe visual aliasing, which include extreme locations such as deserts or volcanic surfaces [21],[22]. In these settings, the scarcity of unique visual or structural landmarks complicates reliable place recognition, while the dearth of vertical structures poses difficulties for the robust convergence of Iterative Closest Point (ICP) algorithms in LiDAR SLAM [23].

The majority of widely-used datasets, such as KITTI [24], Oxford RobotCar [25], KAIST [26], and 4Season [27], primarily target autonomous driving in urban environments. While these datasets incorporate substantial challenges, such as dynamic objects or large seasonal variations, the highly structured scenarios and vehicle-centric perspectives simplify the tasks of odometry computation and place recognition.

Other datasets like TUM RGB-D [28], TUM VI [29] offer sequences captured using hand-held stereo or RGB-D cameras primarily in indoor environments. However, these sequences are typically short, frequently re-observe places from similar viewpoints, and

(a) Common scenes.



(b) Image and LiDAR aligned.

Figure 1.1: (a) Common scenes from Cambridge landmarks [1] and 7-Scenes [2] datasets, where discriminate features can be extracted. (b) Challenging situations from underwater Aqualoc dataset [3] and Mars-Analogue dataset [4]. Data are severely corrupted with ambiguous elements and low image quality, image extracted from [5].

have limited variations in visual appearance. To address the limitations associated with these datasets, synthetic datasets such as ICL-NUIM [30] and TartanAir [31] simulate different motion characteristics and environments. Despite the diversity they offer, unstructured natural environments still pose the most challenging conditions for the application of visual or LiDAR-based SLAM.

Over the past few decades, computer vision has evolved into a critical interdisciplinary research area, striving to emulate various aspects of human visual systems. However, even with significant progress in many areas, fully replicating the capabilities of the human visual system remains a formidable challenge. The complex interplay between our understanding of the human vision system's functionality and the limitations of computational resources significantly contributes to this challenge.

With the surge of LiDAR technology adoption in recent years, especially in autonomous vehicles and robotics, the generation of precise and high-resolution 3D maps has become crucial. These maps are integral to robust and reliable place recognition and navigation in diverse environments.

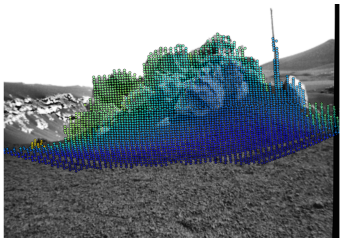
One of the most pressing challenges in computer vision is SLAM. Given the increasing interest in mobile robotics and planetary exploration, resolving the SLAM problem is essential for accurately localizing a camera within an unknown environment while simultaneously mapping it. Moreover, the fusion of multimodal data, such as visual and LiDAR information, shows great potential in enhancing SLAM performance, especially in aliased or planetary-like environments.

Recent advancements in deep learning have shown immense promise in the field of computer vision. By integrating deep learning features and attention mechanisms, we can significantly enhance the performance of visual and LiDAR-based SLAM systems.

This enhancement involves unifying local and global features, thereby optimizing the system's overall performance. The fusion of deep learning with SLAM methodologies can also facilitate better handling of challenging environments, thereby extending the range of applications of these technologies.

This thesis presents research conducted at the German Aerospace Center (DLR), specifically within the Perception and Cognition group. The primary focus of this study is the development of innovative navigation solutions using the Lightweight Rover Unit (LRU), showd in Fig. 1.2. The LRU integrates cutting-edge technologies, including stereo vision and LiDAR, paving the way for autonomous navigation in unexplored and challenging planetary-like environments.

(a) The LRU traversing a planetary-like environment.



(b) Image and LiDAR aligned.

Figure 1.2: DLR Planetary Stereo, Solid-State LiDAR, Inertial (S3LI) dataset[6], recorded on Mt. Etna, Sicily, an environment analogous to the Moon and Mars.

The potent combination of advanced SLAM methodologies and deep learning mechanisms, along with the extensive capabilities of the LRU, offers promising prospects in the field of autonomous navigation and exploration. The continuous pursuit to refine these techniques remains crucial not only for progress in planetary exploration but also for the broader field of robotics. The work carried out in this study contributes to this ongoing endeavor, proposing novel solutions and methodologies that take advantage of the most recent advancements in sensor technology and deep learning.

1.2 Problem Statement

The central focus of this thesis is to explore and develop novel approaches to visual place recognition, specifically through the integration of multimodal fusion of LiDAR and visual data. Given the inherent complexities and challenges of certain environments - notably aliased or planetary-like settings[6] - traditional methodologies often fall short. As such, our research focuses on the unification of local and global features

and the strategic employment of deep learning with attention mechanisms, all aimed at bolstering place recognition performance in these challenging environments. To provide clarity, the specific goals of this thesis are outlined as follows:

- Undertake an exhaustive review of the current landscape of place recognition methods, including those that utilize visual and LiDAR-based techniques, as well as those that leverage the capabilities of deep learning. The intent is to identify and select a handful of the most efficient and accurate approaches that can serve as a baseline for our research.

- Develop an innovative multimodal place recognition method that integrates both local and global features. This approach is designed to leverage deep learning features and attention mechanisms to offer a more robust solution for place recognition, particularly in challenging environments.

- Validate the proposed method using a novel challenging dataset, the Mt. Etna Dataset. This dataset, which was captured with Stereo and Solid-State LiDAR Inertial sensors, offers an accurate representation of the Moon-like environment of Mount Etna, providing an ideal testbed for the proposed methodology.

The overarching goal of this research is to contribute to the body of knowledge in the field of SLAM and place recognition by developing innovative methods capable of handling challenging environments. By bringing together visual and LiDAR data, and leveraging deep learning techniques, we aim to create a more versatile and reliable solution that can be used in a wide range of applications, from autonomous navigation to planetary exploration.

1.3 Methodology

To achieve the objectives of this thesis, we will follow the following methodology:

1. Literature Review: Conduct a comprehensive review of existing research and methods in the field of visual place recognition, focusing on multimodal fusion, unifying local and global features, and deep learning with attention mechanisms.

2. Data Collection amnd Processing: Gather appropriate datasets that contain LiDAR and visual information from planetary-like environments to train and evaluate the proposed approach.

3. State-of-the-art exploration: Investigate the most recent and cutting-edge techniques in visual place recognition, analyzing their performance, limitations, and potential for improvement.

4. Algorithm Development: Develop a novel visual place recognition algorithm that leverages deep learning and attention mechanisms, as well as multimodal fusion and unification of local and global features.

5. Implementation: Implement the proposed approach using the aforementioned tools, such as Pytorch and OpenCV, following best practices in software development.

6. Evaluation and Benchmarking: Assess the performance of the proposed approach and compare it with existing methods using appropriate metrics and benchmarks, to identify the strengths and weaknesses of the approach.

In the annex A a table has been included with the duration of each task, as well as a Gantt chart, which details how time has been managed during the course of the project.

1.4 Tools

The implementation, evaluation, and benchmarking of our proposed approach depend on a suite of specialized tools, encompassing both hardware and software components. This section details the main tools that facilitate our research:

1.4.1 Hardware

LiDAR: Our research utilizes the Blickfeld Cube-1 LiDAR, known for its MEMS-actuated beam deflection mirror. This feature makes it superior to traditional 360-degree LiDARs in terms of mechanical robustness, weight, and power consumption - qualities especially important for space applications. The LiDAR is configured to capture a maximum of 17400-point clouds within a field-of-view of approximately 70°H $\times$ 30°V, resulting in a scan rate of 4.7 Hz.

Stereo Cameras: Two AVT Mako cameras form a stereo setup with a 20 cm baseline. They are programmed to capture monochromatic images at a frequency of 30 Hz. Each image has a resolution of 688 $\times$ 512 pixels. Automatic exposure control helps manage the extreme lighting conditions encountered on the mountain.

Inertial Measurement Unit (IMU): The XSens MTi-G 10 unit, connected via USB, records linear acceleration and angular velocity data at a rate of 400 Hz.

Figure 1.3: Suite of sensors onboard the LRU. Solid-State LiDAR, Inertial (S3LI).

Global Positioning System (GPS): We employ a Ublox f9p GNSS receiver for accurate differential estimates in tandem with a base station. The data logs are later processed using RTKLIB, which enables the acquisition of ground truth positions at 5 Hz with centimeter-level accuracy.

Server: Our dedicated server, outfitted for training purposes, houses two 32-core AMD CPUs, 768GB RAM, and eight 3090 RTX (Ampere) GPUs, each with 24GB of memory.

1.4.2 Software

Python3: As a flexible and powerful programming language, Python is utilized for a wide array of tasks, including algorithm implementation, data processing, and visualizations.

OpenCV[32]: This open-source computer vision library assists with image processing, feature extraction, and other computer vision tasks.

PyTorch[33]: PyTorch, an open-source machine learning library for Python, supports our research with its extensive deep learning capabilities and GPU acceleration. It is particularly useful for implementing our deep learning models, attention mechanisms, and multimodal fusion techniques.

OAISYS[12]: The Outdoor Artificial Intelligent SYstems Simulator (OAISYS) is specifically designed for unstructured outdoor environments, considering the unique requirements of planetary robotics. Based on the open-source Blender engine, it generates a wide variety of outdoor scenes and rich metadata, including multi-level semantic and instance annotations. The use of OAISYS does not require expert knowledge in rendering pipelines, making it a user-friendly tool for our research.

1.5 Outline

The remainder of this thesis is structured as follows: Chapter 2 reviews the existing literature in the field of visual place recognition, placing particular emphasis on multimodal fusion, unification of local and global features, and deep learning approaches leveraging attention mechanisms.

In Chapter 3, we introduce various relevant place recognition techniques, with a special focus on the specific methods that will be evaluated within the scope of this research. Chapter 4 also presents our innovative algorithm for place recognition, UMF. Chapter 5 provides a comprehensive evaluation of the selected approaches, discussing the benchmarking suite, the domain of the evaluation, the results obtained, as well as the interpretations of these results.

Finally, Chapter 6 brings this thesis to a close with a summary of the study, underscoring the contributions made to the field of place recognition for low-texture or planetary-like environments. This chapter also explores potential avenues for future work in this area.

Chapter 2

Related Work

2.1 Place Recognition

Place recognition is a central task in several areas, including robotics, autonomous driving, and augmented reality. It involves the encoding of sensor data into global descriptors, which are then used to retrieve similar places using appropriate distance metrics in the feature space. The encoding methods are largely dependent on the type of sensor data used. This has led to the development of a wide variety of place recognition approaches, which can broadly be classified into vision-based, LiDAR-based, and visual-LiDAR fusion methodologies. The latter leverages the benefits of both vision and LiDAR data to offer more robust and reliable place recognition, particularly in aliased and low-texture environments.

2.2 Vision-Based Place Recognition

The vision-based place recognition paradigm involves the extraction and encoding of features from image data. Traditional methods in this category utilize hand-crafted local feature descriptors such as SIFT, SURF, and ORB [10, 34] to capture salient information in images. However, these handcrafted features have limitations in handling severe appearance changes. With the advent of deep learning, researchers have started using networks [35, 36, 37, 38, 39] to replace hand-crafted feature descriptors.

Specifically, NetVLAD [35], a trainable framework that combines Convolutional Neural Networks (CNNs) and Vector of Locally Aggregated Descriptors (VLAD), has inspired several subsequent works. For instance, Patch-NetVLAD [40] implements both local and global descriptors in an end-to-end manner. Other learning-based methods have incorporated attention mechanisms [37, 41, 42, 43] to enhance resistance to visual appearance changes. These attention mechanisms, implemented as shallow CNNs, can be trained either separately [44] or jointly [45] with the backbone network.

2.3 LiDAR-Based Place Recognition

LiDAR-based place recognition methods exploit the geometric structural information of the environment, which is captured as point clouds. The irregular format of point clouds presents unique challenges for applying conventional CNNs. Some solutions include discretizing the 3D space into voxel grids and applying 3D convolution [44, 13], or using PointNet [36] to process point clouds directly.

Several works such as PointNetVLAD [36] and LPD-Net [29] have been proposed based on PointNet. These methods have also explored the integration of attention mechanisms for better concentration on important features, such as in TransLoc3D [46] and OverlapTransformer [38].

In contrast, MinkLoc3D [13] uses a sparse voxelized representation. It employs a 3D convolutional architecture modeled on the Feature Pyramid Network (FPN) [11] design pattern to extract informative local features, which are then aggregated using the Generalized Mean Pooling (GeM) [47] layer into a discriminative global descriptor. MinkLoc3D has demonstrated superior performance in standard benchmarks, outperforming other point cloud-based global descriptors.

2.4 Visual-LiDAR Fusion for Place Recognition

The fusion of visual and LiDAR data for place recognition capitalizes on the strengths of different sensor modalities and can improve performance in challenging environments [44, 13, 48]. Fusion techniques include projecting segments of point clouds onto images and using 2D and 3D convolution to extract features [13, 44], concatenating image and point cloud features directly [13], and integrating attention modules to enhance feature representation [44]. This approach leverages the complementary strengths of both RGB images and 3D point cloud data, thus proving beneficial for applications like SLAM where robust and accurate recognition is crucial for loop closure and relocalization.

2.5 Transformers in Multimodal Place Recognition

Transformers [49], originally designed for natural language processing, have been progressively adopted in various vision tasks [50, 45]. Their self-attention mechanisms have proven valuable in multiple areas, including image retrieval and visual place recognition [42, 43, 51], lidar-based recognition [46], and fusion of modalities [44, 48]. Transformers are particularly adept at managing long-range dependencies and adaptively identifying relevant regions in complex environments. Their ability to offer

a global perception field for each output token enables the capture of semantically meaningful structures, thereby showing promise across various domains in recent years.

There are several reasons to use transformers for multimodal place recognition:

Flexible data fusion: Transformers can adaptively learn to attend to different modalities, allowing the model to weigh the contributions of each modality depending on their relevance to the task [44].

Improved performance: Transformers have demonstrated state-of-the-art performance in various tasks, including image recognition and 3D point cloud analysis [46, 51], [51] suggesting their potential to excel in multimodal place recognition.

2.5.1 Attention Mechanisms

The application of attention mechanisms in place recognition has shown significant potential. They enhance the models' ability to adaptively identify task-relevant regions in complex scene images. Attention maps can be utilized in multiple ways: as patch descriptor filters [45, 37] that highlight the significant parts of the scene, or as weight maps that modulate the CNN feature maps to generate global features [51]. This enables models to focus on salient features and ignore irrelevant information, thereby improving the robustness of place recognition. In the context of transformer-based models, attention modules can be implemented with simplicity and efficiency, often as a linear layer that decodes the attention information from transformer tokens. This exemplifies the flexibility and power of transformer-based models in multimodal place recognition tasks, opening up exciting avenues for future research and development.

Receptive field:

As seen in Fig. 2.1, if we want the model to learn to establish a connection between the L and R elements to extract their joint feature representation. Due to the local connectivity of convolutions, many convolution layers need to be stacked together in order to achieve this connection. The global receptive field of Transformers enables this connection to be established through only one attention layer

2.6 Unification of Local and Global Features

Local descriptors, such as SIFT or ORB [52], encode an image I_i with a set $D_i = d_k|k = 1, ..., K$ of vectors $d_k \in R^d$ at K regions of interest. They often provide better performance than holistic descriptors, but require computationally expensive methods for local feature matching like homography estimation, computation of the epipolar constraint, or deep-learning matching techniques, e.g., SuperGlue[53] or LoFTR [7].

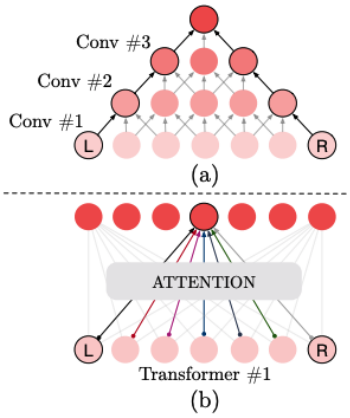


Figure 2.1: Illustration of the receptive field of (a) Convolutions and (b) Transformers[7]. .

On the other hand, global descriptors represent an image $I_i \in DB, Q$ with a single vector $d_i \in R^d$. This allows for efficient pairwise descriptor comparisons with low runtimes.

Figure 2.2: Example of local and global descriptors for place recognition (Source: [8])

The fusion of local and global features can result in more robust and informative representations. Patch-NetVLAD [40], for instance, demonstrates the effectiveness of combining local and global features in an end-to-end manner. It uses a global descriptor technique, NetVLAD [35], to extract descriptors from predefined image patches. This fusion approach capitalizes on the strengths of both local and global features, providing a comprehensive representation of the environment.

A similar concept was explored by Zheng et al. [5], albeit in the context of 6D camera relocalization. Their work is particularly relevant because it focuses on the aliasing problem using the MADMAX dataset[4], which includes stereo data but no LiDAR.

Zheng et al. trained their model on images from Mars analog sites in the Moroccan desert. These scenes are characterized by desolate landscapes like sands and rocks, where textureless places and repetitive contents amplify ambiguity in localization, simmilar to the Etna dataset. They employed pretrained SuperPoint

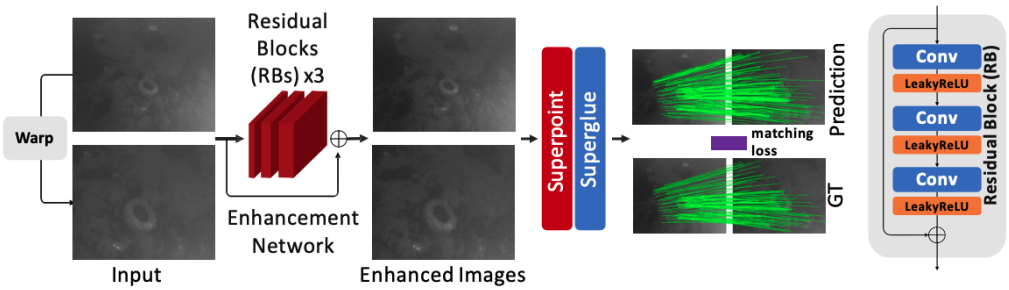


Figure 2.3: Environment-aware image enhancement method proposed by Zheng et al. [5]. The method employs a CNN to predict the residual of the input image under a self-supervised framework by minimizing the keypoint matching assignment loss. The robust pretrained keypoint detectors and matchers, SuperPoint and Superglue, are used.

and Superglue to extract and track sparse features for efficient Structure from Motion (SfM) reconstruction. For global descriptors, they referred to the state-of-the-art image retrieval method SFRS. During inference, they performed temporal matching within a 30-frame window for 10 iterations, and empirically defined an inlier threshold. They also trained a lightweight network for image enhancement, which improved the expected performance of these methods.

Despite the challenging desert environment, their system managed to generate a coherent trajectory. In comparison, other state-of-the-art methods such as DSAC++/DSAC*[54, 55] struggled due to the ambiguity. While the majority of the time was consumed by the 2D keypoints matching step in both global and temporal matching, they noted that this could be significantly reduced in a parallel manner. Their work provides a robust and efficient solution for feature extraction in ambiguous scenarios, demonstrating the promise of unifying local and global features.

2.7 Self-Supervised Pretraining

Self-supervised pretraining is a prominent methodology that has shown remarkable results in a variety of machine learning tasks. It enables a model to learn useful features from the data without the need for explicit labeling, which can drastically reduce the cost and effort required for manual annotation. This approach is often used to initialize models before fine-tuning them on a smaller, task-specific labeled dataset, thereby enhancing their robustness and data efficiency.

2.7.1 Contrastive Learning Methods

Contrastive methods form a critical part of self-supervised learning, focusing on learning representations such that similar instances are brought closer together in the latent space while dissimilar ones are pushed apart. Two key examples of this approach are SimCLR [56] and BYOL [57].

SimCLR (Simple Contrastive Learning of Visual Representations) [56, 58] leverages data augmentations to generate positive pairs, and then optimizes the representations using a contrastive loss function. The key idea is that different augmentations of the same image should be similar to each other and dissimilar to other images in the latent space.

BYOL (Bootstrap Your Own Latent) [57] also aims to bring the representations of augmentations of the same image closer together, but uniquely, it does not rely on negative pairs. It uses two networks, a target network, and an online network, and minimizes the mean squared error between the normalized predictions of the two networks.

These methods have proven effective for pretraining on large unlabeled datasets and have shown to improve performance on downstream tasks, particularly in the field of computer vision.

2.7.2 Generative Methods: Masked Autoencoders

Recently, a shift towards generative self-supervised methods, such as masked autoencoders, has been observed, as they have demonstrated superior performance in many applications.

Masked autoencoders work by learning to reconstruct an input after part of it has been masked out or corrupted. This way, the model learns to capture the underlying structure and dependencies in the data. By learning to fill in the missing parts, the model implicitly learns a rich, useful representation of the data.

Self-Supervised Learning for Images

Two recent developments in masked autoencoders are ConvNeXt [59] and Spark [60]. ConvNeXt employs 3D sparse convolutions (fig. 2.4), which allows for the application of CNNs to point clouds, a departure from the common use of transformers. Spark also uses sparse convolutions and is particularly designed for pretraining on point clouds.

These methods present a significant advantage as they are not dependent on carefully designed data augmentations or pair constructions, unlike contrastive methods. Moreover, they are particularly suited for modalities such as point clouds,

where defining appropriate data augmentations can be challenging.

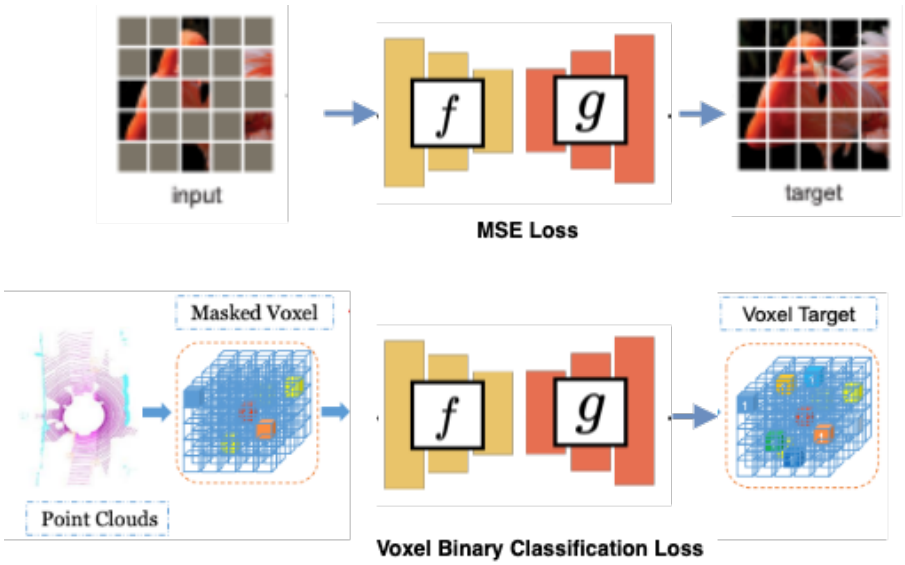


Figure 2.4: Pretraining pipeline using masked autoencoders for visual and point cloud data. The masked input is fed into the encoder, and the decoder reconstructs the original input. During pretraining, the masked regions are used as a self-supervised signal. Source: [9].

In this work, we employ generative self-supervised pretraining methods due to their promising results, adaptability to various data modalities, and lack of dependence on intricate data augmentations. The masked autoencoder approach provides a robust basis for learning powerful representations from our multi-modal data, contributing to the overall performance of our place recognition system.

2.7.3 Self-Supervised Learning for LiDAR Data

LiDAR (Light Detection and Ranging) data, often represented as point clouds, presents unique challenges for self-supervised learning. Point clouds are inherently sparse and unordered, and contain rich geometric and spatial information that needs to be adequately captured. Despite the successes of self-supervised learning in other domains like images and language, its application to large-scale point clouds has remained relatively unexplored.

Recently, a novel self-supervised pretraining approach specifically designed for large-scale point clouds has been proposed: Voxel-MAE (Masked Voxel Autoencoder) [61]. It combines the advantages of voxelization and mask-based pretraining. Point clouds are transformed into voxel representations, and a portion of these voxels are randomly masked out during training. The task of the network is to predict whether

a voxel contains points, thus making the model voxel-aware of the object shape. This approach effectively utilizes the redundancy in large-scale point clouds and learns representative features, even with a high masking ratio of up to 70% and outperforms training from scratch on various downstream tasks, including 3D object detection, semantic segmentation.

In this work, we focus on the use of the Voxel-MAE method due to its ability to handle large-scale point clouds and its proven effectiveness in various downstream tasks. The unique design of Voxel-MAE, with its voxel-aware approach and high masking ratio, provides a robust and efficient way to learn representative features from large-scale LiDAR point clouds, thus enhancing the perception capabilities of autonomous vehicles.

Chapter 3

Multi-modal Place Recognition

Place recognition plays a crucial role in autonomous navigation and mapping technologies. The ability of an autonomous system to recognize its location within a pre-mapped environment is pivotal for tasks such as relocalization, loop closure detection, and waypoint navigation. The place recognition models used for this project are categorized into unimodal and multimodal approaches.

3.1 Unimodal Approaches

Unimodal approaches leverage a single modality of data, either visual or LiDAR, for place recognition. Visual models primarily depend on image data, extracting features and descriptors that capture the appearance of a place. However, these models may struggle in low-light conditions or when the appearance of a place changes over time due to factors like seasonal changes, weather variations, and human activity.

On the other hand, LiDAR models primarily utilize three-dimensional point cloud data generated by LiDAR sensors. They exploit geometric properties of the environment, which are usually invariant to lighting conditions and less affected by appearance changes. Nevertheless, LiDAR models may face challenges in sparse or texture-less environments, where geometric features are scarce or repetitive.

Despite their individual limitations, unimodal models are often straightforward and computationally efficient, making them suitable for real-time applications.

3.2 Multimodal Approaches

Multimodal approaches, as the name implies, combine multiple modalities of data, typically visual and LiDAR, for place recognition. These models exploit the complementary nature of visual and LiDAR data, leveraging the strengths of each modality to enhance the robustness and performance of the place recognition system.

For instance, visual data provides rich texture and color information, while LiDAR data offers accurate geometric structure.

By fusing these modalities, multimodal models can handle a wider variety of environmental conditions and often outperform their unimodal counterparts. They can also mitigate some of the challenges faced by unimodal models, such as lighting variations for visual models and sparse environments for LiDAR models.

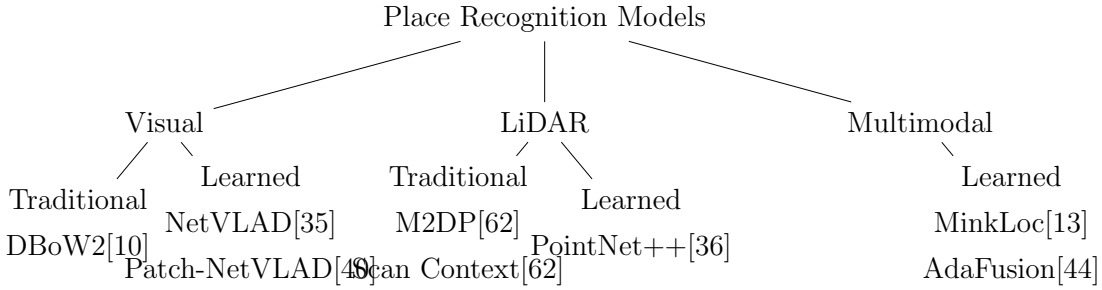
3.3 Taxonomy of Visual and LiDAR Modalities

Visual and LiDAR modalities can be further categorized into traditional and learned approaches. Traditional approaches typically involve handcrafted feature descriptors, which are manually designed to capture specific properties of the data. These methods have been widely used in computer vision and robotics due to their interpretability and computational efficiency.

However, traditional approaches often struggle to handle complex and diverse data as they rely on predefined feature representations that might not generalize well to different environments or conditions. To overcome these limitations, learned approaches employ machine learning algorithms, particularly deep learning, to automatically learn feature representations from data.

Learned approaches have shown superior performance in various tasks, including place recognition, due to their ability to learn complex and high-dimensional feature representations. Nevertheless, these models often require large amounts of data for training and are more computationally demanding than traditional methods.

The taxonomy of the place recognition models used in this project can be illustrated as follows:



3.4 Selected Methods

The choice of methods for this project hinged on their ability to effectively handle the respective modalities, their compatibility with the overarching architecture of the place recognition system, and their proven performance in previous research. Furthermore,

the chosen models offer a balanced representation of both traditional and learned approaches, as well as unimodal and multimodal methods.

3.5 DBoW2

DBoW2[10], or Dynamic Bag of Words, is a prominent model for visual place recognition. It leverages ORB[52] descriptors, a type of feature descriptor that is binary and thus efficient to compute and match. By using binary descriptors, DBoW2 reduces the computational complexity of matching operations, which is crucial for real-time applications.

The process for constructing the vocabulary begins by obtaining ORB keypoints from thousands of generic images. In the context of our work, we have elected to use a database of planetary-like images in order to construct a more relevant vocabulary. Once these keypoints are gathered, a 'bag of words' is created. This technique utilizes a visual vocabulary (fig. **??**3.1 3.1onvert an image into a sparse numeric vector, allowing us to handle large image sets. The visual vocabulary is created offline by discretizing the descriptor space into W visual words.

In the case of a hierarchical bag of words, the vocabulary is structured as a tree. To build it, a large set of features from training images, independent from those processed online later, are extracted. These extracted descriptors are first discretized into kw clusters by performing k-medians clustering with a k-means++ [63] seed. Any resulting medians that are non-binary are truncated to zero. These clusters form the first level of nodes in the vocabulary tree. Subsequent levels are created by repeating this operation with the descriptors associated with each node, up to Lw times. Finally, a tree with W leaves is obtained, which are the words of the vocabulary.

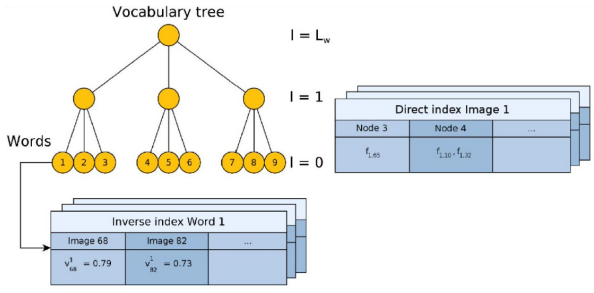


Figure 3.1: Diagram of DBoW vocabulary, extracted from [10].

The importance of each word is assigned a weight according to its relevance in the training image set. Words that are very frequent, and therefore less discriminatory, have their weights reduced. This is achieved using the term frequency-inverse document

frequency (tf-idf) [64].

When converting an image It, taken at time t, into a bag of words vector $v_t \in RW$, the descriptors of the detected features traverse the tree from the root to the leaves, selecting intermediate nodes at each level that minimize Hamming distance.

To calculate the similarity between two bag-of-words vectors v1 and v2, the L1 score $s(v1, v2)$, which falls in $[0, 1]$, is computed:

$$s(v1, v2) = 1 - \frac{||v1 - v2||_1}{2|v1||v2|}$$

Along with the bag of words, an inverted index is maintained. This structure stores, for each word in the vocabulary, a list of images in which it is present. This is beneficial when querying the database, as it enables comparisons only with those images that share some words with the query image. The inverted index is updated when a new image is added to the database.

Additionally, a direct index is used to store the features of each image. The nodes of the vocabulary are separated according to their level l in the tree, starting with the leaves (level l = 0) and ending at the root ($l = L_w$). For each image It, the direct index stores the nodes of level l that are ancestors (higher level, closer to the root) of the words present in It, along with the list of features ft_j associated with each node.

DBoW2 uses the direct index to expedite the computation of correspondences between two sets of ORB features. It can limit brute force pairings to only those features that belong to the same vocabulary tree node at a certain level. This trick is used when searching for matches to triangulate new points, and in loop closure and relocalization.

3.5.1 Implementation details

In terms of implementation details, we follow the same configuration used for ORB-SLAM2[52]. This includes a vocabulary with a branching factor of 10 and depth levels of 6, creating a dictionary of 1 million words.

We extract ORB features as seen in fig. 3.2 and create a custom vocabulary using images from Mt Etna and similar environments.

Despite its efficiency and simplicity, DBoW2 can struggle with changes in viewpoint and scale, as well as appearance changes over time. Its reliance on visual data also makes it sensitive to lighting conditions and occlusions.

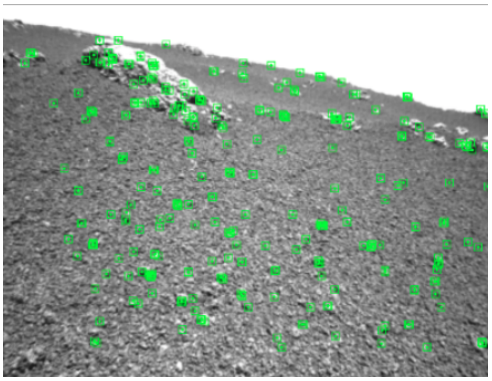


Figure 3.2: Extracted ORB keypoints using an image from the Etna datset.

3.6 PointNet++

PointNet++[36] is a significant deep learning model designed to process point cloud data, which is particularly applicable to LiDAR-based place recognition. This model's significance lies in its ability to capture local structure from raw point clouds, compute point features, and aggregate these features to classify or segment the input. Unlike traditional methods that require preprocessing steps like voxelization or meshing, PointNet++ operates directly on raw point clouds, thus preserving the original geometric structure of the data. Its hierarchical structure enables the processing of a large number of points, making it ideal for handling large-scale LiDAR data. However, despite its robustness to various transformations, it might struggle in cluttered environments or when point clouds are sparse, as it relies on the local geometric structure of the data [36].

The mathematical basis of PointNet++ lies in its consideration of a discrete metric space $\mathcal{X} = (M, d)$, where the metric is inherited from a Euclidean space $\mathbb{R}^n$, $M \subseteq \mathbb{R}^n$ is the set of points, and d is the distance metric. In scenarios where the density of M in the ambient Euclidean space is not uniform, PointNet++ learns set functions f that take $\mathcal{X}$ as input and produce information of semantic interest regarding $\mathcal{X}$.

PointNet++ introduces a hierarchical grouping of points and progressively abstracts larger and larger local regions along the hierarchy to capture local context at different scales, as seen in Fig. 3.3. This hierarchical structure is composed of several set abstraction levels. At each level, a set of points is processed and abstracted to produce a new set with fewer elements. The set abstraction level comprises three key layers: Sampling layer, Grouping layer, and PointNet layer.

The Sampling layer employs an iterative farthest point sampling (FPS) approach to select a subset of points, ensuring better coverage of the entire point set. In the Grouping layer, local region sets are constructed by finding "neighboring" points

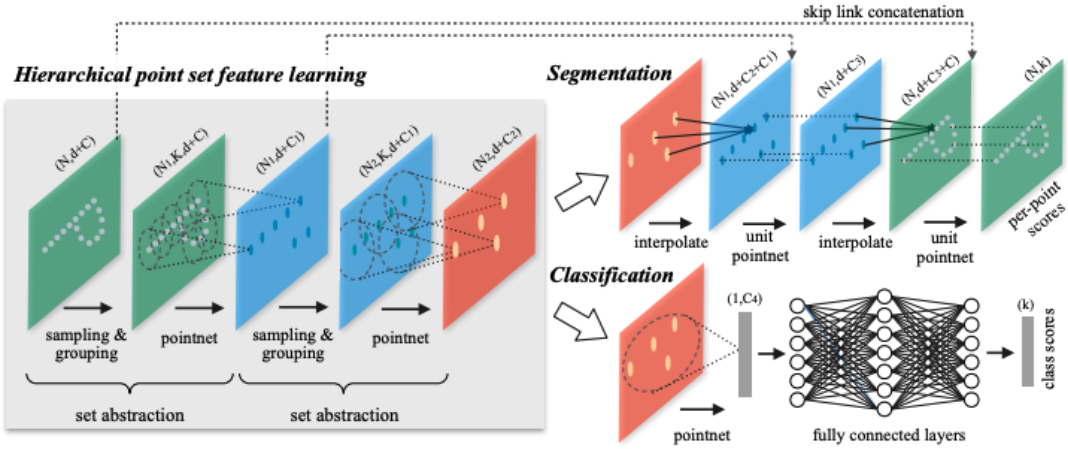


Figure 3.3: Pointnet++ architecture. Illustration of our hierarchical feature learning architecture and its application for set segmentation and classification using points in 2D Euclidean space as an example

around the centroids. This is done using a kNN search.

For tasks such as semantic point labeling, which require detailed information for all points in the data set, the subsampling approach inherent to PointNet++ might not provide sufficient granularity. Hence, to obtain features for all original points without the high computational cost of treating all points as centroids, the authors propose a hierarchical propagation strategy.

This strategy involves distance-based interpolation and the integration of skip links across levels. In this context, a feature propagation level is defined as the process of propagating point features from a set of $N_l \times (d + C)$ points to a larger set of N_{l-1} points. Here, N_l and N_{l-1} denote the point set size at the input and output of set abstraction level l, with $N_l \leq N_{l-1}$.

The propagation of point features is realized by interpolating feature values f from the N_l points at the coordinates of the N_{l-1} points. Among the various possible interpolation methods, we employ inverse distance weighted average based on k nearest neighbors. The interpolation is given by the formula:

$$f^{(j)}(x) = \frac{\sum_{i=1}^{k} w_i(x) f_i^{(j)}}{\sum_{i=1}^{k} w_i(x)} \quad \text{where} \quad w_i(x) = \frac{1}{d\left(x, x_i\right)^p}, j = 1, \ldots, C$$

The interpolated features on N_{l-1} points are then combined with the skip-linked point features from the set abstraction level. This fusion of features is subsequently passed through a 'unit pointnet', which can be compared to a one-by-one convolution in convolutional neural networks (CNNs).

Finally, to update each point's feature vector, we apply several shared fully

connected layers with ReLU activation. This process is repeated until features have been propagated to the entire original set of points, providing comprehensive, point-wise feature information necessary for tasks like semantic point labeling.

3.7 NetVLAD

NetVLAD[35] combines the robust feature extraction capabilities of Convolutional Neural Networks (CNNs) with the Vector of Locally Aggregated Descriptors (VLAD) encoding, a widely used method in image retrieval tasks. This integration results in a potent model for visual place recognition that leverages the strengths of deep learning and traditional image retrieval techniques.

The design of NetVLAD is inspired by the standard pipeline in the image retrieval community, which includes extracting local descriptors and subsequently pooling them in an orderless manner. To learn the representation end-to-end, NetVLAD mirrors this pipeline using differentiable modules within a CNN architecture.

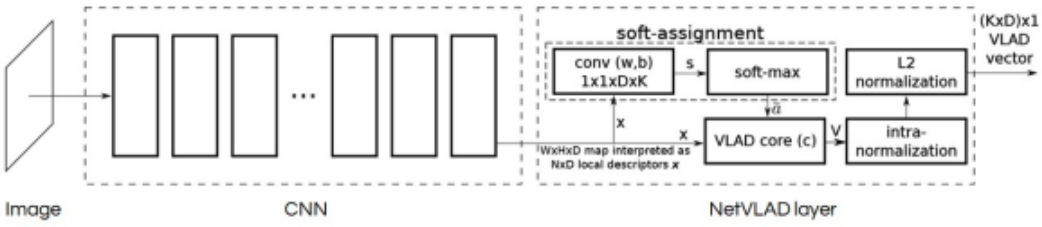


Figure 3.4: NetVLAD architecture.

NetVLAD employs a CNN as a dense descriptor extractor by truncating the network at the last convolutional layer. This practice is effective for instance retrieval and texture recognition, transforming the output into a $H \times W \times D$ map. This map represents a set of D-dimensional descriptors extracted at $H \times W$ spatial locations.

A VLAD layer, inspired by the VLAD encoding method, is integrated into the architecture for pooling the extracted descriptors into a fixed image representation. This VLAD layer contains learnable parameters optimized via back-propagation. It facilitates the generation of a fixed-size vector $f\left(I_i\right)$ for each image, a crucial component for place recognition tasks.

ResNet50, a well-established CNN architecture, serves as the backbone of NetVLAD. The backbone is truncated at the last convolutional layer (conv5), before the ReLU activation. The network is further extended with Max pooling (fmax) and NetVLAD (f$VLAD$). For the VLAD layer, we set the cluster size K = 64, resulting in 16k and 32k-D image representations.

The training parameters θ of the representation f_θ are optimized using Stochastic Gradient Descent (SGD).

NetVLAD has demonstrated superior performance in various place recognition benchmarks. It exhibits robustness to variations in viewpoint, scale, and lighting conditions, attesting to the capability of CNNs in learning invariant features from complex visual data and the power of VLAD encoding in aggregating these features into a single, robust descriptor.

3.8 MinkLocMultimodal

MinkLocMultimodal[13] is a multimodal place recognition model that fuses visual and LiDAR data. Utilizing 3D convolution on sparse voxelized point clouds, it extracts local features that are then aggregated into a global descriptor. The model is designed to harness the complementary strengths of visual and LiDAR data to achieve robust place recognition under various environmental conditions [13].

The architecture of MinkLocMultimodal consists of two branches, a sparse convolutional neural network (Sparse CNN) for LiDAR processing and a traditional CNN for visual processing. The Sparse CNN operates on a voxelized representation of the point cloud, allowing for effective processing of sparse 3D data. Meanwhile, the CNN processes visual data to extract image features encapsulating the appearance of a place [13].

However, despite its strengths, MinkLocMultimodal has certain limitations. The fusion strategy employed by MinkLocMultimodal is static and does not account for the varying importance of each modality in different situations. This leads to the "dominating modality" problem during training. This issue arises when the network overly focuses on a modality that overfits to the training data, driving the loss down during training but leading to suboptimal performance on the evaluation set. This

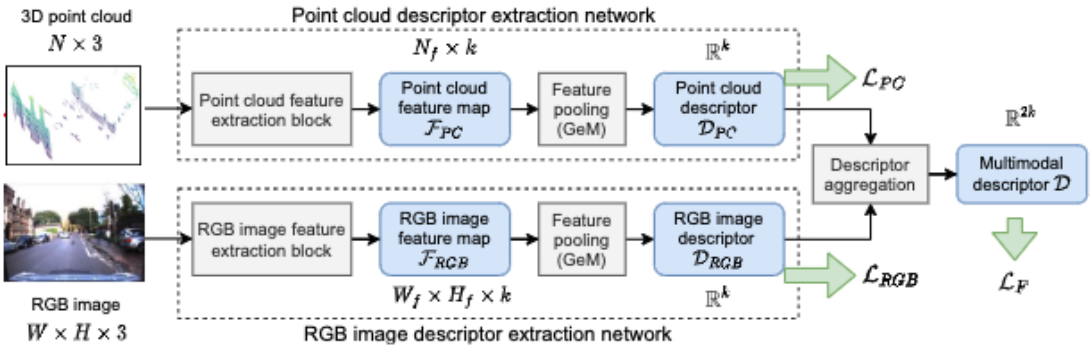


Figure 3.5: MinkLocMultimodal architecture.

research aims to address this problem by proposing an adaptive fusion strategy that takes into account the importance of each modality based on the environmental context.

3.8.1 Implementation details

The loss function $\mathcal{L}$, defined by Eq. 3.1, is a sum of three terms: the first based on the fused multimodal descriptor $\mathcal{D}$, the second on the point cloud descriptor $\mathcal{D}_{PC}$, and the third on the RGB image descriptor $\mathcal{D}_{RGB}$:

$$\mathcal{L} = (1 - \alpha - \beta)\mathcal{L}_F + \alpha\mathcal{L}_{PC} + \beta\mathcal{L}_{RGB}, \tag{3.1}$$

where α, β are experimentally chosen weights, and each component $\mathcal{L}_F, \mathcal{L}_{PC}, \mathcal{L}_{RGB}$ is a triplet margin loss function [65] defined as:

$$\mathcal{L}_{\square}\left(a_i, p_i, n_i\right) = \max\left\{d\left(a_i, p_i\right) - d\left(a_i, n_i\right) + m, 0\right\}, \tag{3.2}$$

where $d(x, y) = \|x - y\|_2$ is the Euclidean distance between descriptors x and y; a_i, p_i, n_i are descriptors of an anchor, a positive, and a negative element in i-th triplet and m is a margin.

For each batch, $\mathcal{L}_F$ is computed from triplets constructed from multimodal descriptors $\mathcal{D}$; $\mathcal{L}_{PC}$ from triplets constructed from point cloud descriptors $\mathcal{D}_{PC}$; and $\mathcal{L}_{RGB}$ from triplets constructed from RGB image descriptors $\mathcal{D}_{RGB}$. This loss function is designed to encourage the network to learn useful representations from both modalities and to fuse these representations effectively.

3.8.2 Parameters and Hyperparameters

3D point coordinates are normalized to [-1,1] and quantized with a 0.01 quantization step to ensure computational efficiency. The initial learning rate for network parameters in the RGB image feature extraction block is set to 10^{-4} and for all other parameters to 10^{-3}. The network is trained for 50 epochs, with the learning rate reduced by a factor of 10 at the end of the 30th epoch. This learning rate schedule is intended to allow the network to make large adjustments in the early stages of training and more refined adjustments in the later stages.

The dimensionality of point cloud and RGB image descriptors is set to $k = 128$, and the multimodal descriptor has $2k = 256$ dimensions. This dimensionality setting provides a balance between computational efficiency and the ability to capture sufficient information in the descriptors.

To prevent embedding collapse in early epochs of training, the author propose a dynamic batch sizing strategy. The initial batch size is set to 8. When the number

of active triplets falls below 70% of the current batch size, the batch size is increased by 40% until the maximum size of 160 elements is reached. This strategy allows the model to adapt the batch size based on the complexity of the learning task at each stage of training.

To mitigate overfitting, we can employ L2 weight regularization with a coefficient of $\lambda = 10^{-3}$. The coefficients of the loss terms in Eq. 3.1 are $\alpha = 0.5, \beta = 0$. These values were chosen to balance the contributions of the multimodal, point cloud, and RGB image descriptors to the total loss.

In the next chapter, we will propose modifications and improvements to this model to address the limitations identified in this chapter, particularly the dominating modality problem and the static fusion strategy.

3.9 AdaFusion

Developed by Lai et al.[44], AdaFusion is a multimodal place recognition model which fuses visual and LiDAR data using a trainable fusion module. This model allows for adaptive combination of features based on their relative importance, thereby, increasing the robustness of place recognition, particularly in challenging environmental conditions.

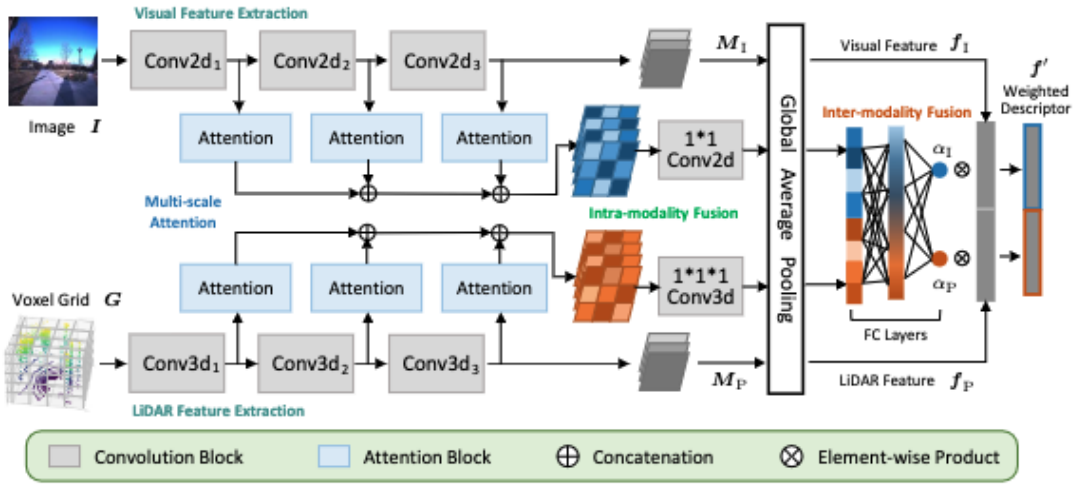


Figure 3.6: AdaFusion architecture.

The AdaFusion model consists of separate branches for visual and LiDAR data processing, similar to MinkLocMultimodal[13]. However, in contrast to merely concatenating the global descriptors, AdaFusion utilizes a fusion module that adaptively weighs each modality's contribution. This approach allows the model to

emphasize the most reliable modality in different situations, thereby improving place recognition performance.

AdaFusion has demonstrated promising results in various place recognition benchmarks, indicating its ability to handle diverse and challenging conditions. Nevertheless, the model's complexity and computational demand are higher than traditional methods and some learned approaches due to the adaptive fusion module.

3.9.1 Implementation Details

In AdaFusion, the convolution blocks ConvXd $\mathrm{d}_i, X \in \{2,3\}, i = 1,2,3$ are constituted by basic convolution blocks (denoted as $\mathbb{C}$). The structure of $\mathbb{C}$ is outlined in Fig. 3, where a convolution with kernel = 3, stride = 1 and padding = 1 is applied, followed by a ReLU activation function, repeated twice. As shown in Fig. 4, we add a batch normalization (BN) layer [66] to the last blocks ConvX d_3 to reduce the internal covariate shift before obtaining the local visual feature map $\boldsymbol{M}_{\mathrm{I}} = \left(m^{\mathrm{I}}_{c,h,w}\right) \in \mathbb{R}^{C_1 \times H_1 \times W_1}$ and the local LiDAR feature map $\boldsymbol{M}_{\mathrm{P}} = \left(m^{\mathrm{P}}_{c,x,y,z}\right) \in \mathbb{R}^{C_1 \times X_1 \times Y_1 \times Z_1}$.

The use of BN in our structure significantly improves the feature extraction ability. Our observations reveal that when directly using a single visual or LiDAR feature as a global descriptor for searching nearest neighbors, a significant improvement of about 10% can be achieved with the BN layer compared to those without.

Global features are produced by applying a Global Average Pooling (GAP) [67] to M_{I} and M_P. GAP is a special case of generalized-mean (GeM) pooling [67]. These global pooling methods, compared to FC layers, have fewer parameters and are more robust to resolution changes of the input data.

Adaptive Weights are proposed, utilizing a two-stage fusion strategy to combine the attention information of images and point clouds. These weights, $\alpha = [\alpha_1, \alpha_{\mathrm{P}}]^\top$, are derived through a weight generation branch that employs a two-stage fusion strategy to effectively fuses the attention information of images and point clouds. Unlike traditional attention-augmented place recognition methods, the attention mechanism within this network doesn't function as salient region masks. Instead, it modifies the contribution of visual and LiDAR features, effectively acting as a dynamic weighting function.

The two-stage fusion strategy begins with multi-scale attention, as seen in [68], which is utilized to fully leverage the information available in different network layers. As depicted in Fig. 2 and Fig. 3, for each modality, spatial attention and channel attention are computed from the feature extraction branch following the methodology presented in [69]. However, the implementation in [69] and other related works, such as

[70], [71], focus only on 2D images. We extend this to encompass 3D voxel grid form, given the query map $\boldsymbol{Q}$, the key map $\boldsymbol{K}$, and the value map V of shape $C_2 \times X_2 \times Y_2 \times Z_2$.

The attention maps are derived as follows:

$$\boldsymbol{S}_x = \text{ Softmax } \left(\boldsymbol{K}^\top \boldsymbol{Q}\right), \quad \boldsymbol{S}_c = \text{ Softmax } \left(\boldsymbol{Q}\boldsymbol{K}^\top\right),$$

where the softmax operation is performed along the row (second dimension) of the matrix. The spatial attention and the channel attention can then be obtained by reshaping $\boldsymbol{V}\boldsymbol{S}_x^\top$ and $\boldsymbol{S}_c\boldsymbol{V}$ back to shape $C_2 \times X_2 \times Y_2 \times Z_2$.

To reduce computational cost and retain useful information, a convolution with a kernel of 1 is applied to linearly fuse the attention. This is performed before outputting from the attention block ($\mathbb{A}$), where downsample with nearest interpolation is performed to ensure all the results have the same size.

Intra-modality fusion involves concatenating the multi-scale attention of the same modality along the channel dimension. As mentioned previously, attention from different layers of the network concentrates on varying contents. To fuse and merge these, a convolutional layer with a kernel size of 1 is applied to the attention of the same modality. This operation acts as a transformation of the number of channels, balancing the attention across layers.

For inter-modality fusion, an issue that arises when fusing attention of different modalities is the inconsistency of data structures. For instance, in this case, the visual attention A_1 is of 3D shape $(C_3 \times H_3 \times W_3)$ while the LiDAR attention A_{P} is of 4D shape $(C_3 \times X_3 \times Y_3 \times Z_3)$. To address this, GAP is applied to A_I and A_P respectively, yielding two 1D vectors $a_1 \in \mathbb{R}^{C_a}$ and $a_{\text{P}} \in \mathbb{R}^{C_3}$ that have the same shape and dimension. The length of these vectors can be easily controlled in the intra-modality fusion stage, where a pre-determined size that represents the number of channels after fusion within each modality. The key is to set C_3 such that it captures enough information from each modality but doesn't overly complicate the fusion process or increase computational burden. For example, C_3 might be set to 128 or 256 in typical applications.

These two vectors, a_1 and a_{P}, are then concatenated and passed through a fully connected layer, and a softmax activation function is applied to obtain the weights $\alpha = [\alpha_1, \alpha_{\text{P}}]^\top$:

$$\alpha = \text{Softmax}\left(\text{FC}\left(\text{Concat}\left(a_1, a_{\text{P}}\right)\right)\right)$$

The softmax function ensures that these weights sum up to 1 and are within the range of 0 to 1, thus they can serve as a valid weighting scheme for the fusion of features

from the two modalities.

These weights $\alpha = [\alpha_1, \alpha_{\mathrm{P}}]^\top$ are then used in the fusion of features from the two modalities. The final feature representation F is a weighted sum of the image and point cloud features, F_1 and F_{P}, respectively:

$$F = \alpha_1 \cdot F_1 + \alpha_{\mathrm{P}} \cdot F_{\mathrm{P}}$$

This strategy allows the network to adaptively adjust the contribution of each modality based on the attention information, which could be especially useful in challenging scenarios, such as low light conditions, where one modality may provide more reliable information than the other. This enhances the robustness of the place recognition task and could potentially lead to improved performance.

Additionally, the adaptive weighting scheme allows the network to be more versatile in different scenarios. For instance, in a low-light or obscured vision scenario, the network may assign a higher weight to the point cloud information, whereas in scenarios with clear vision but cluttered or unstructured point cloud data, the network might assign a higher weight to the image information.

In conclusion, this fusion strategy allows the system to better handle the heterogeneity and varying reliability of different sensor modalities, and therefore enhances its robustness. However, to ensure the effectiveness of this approach, it is important to train the network with diverse data covering a wide range of scenarios.

Chapter 4

Proposed Method: UMF

In this chapter, we introduce our novel approach named Unifying Local and Global Multimodal Features (UMF) - an architecture explicitly designed the aforementioned challenges in extreme environments. The UMF architecture is uniquely engineered to overcome the identified challenges encountered in aliased and low-texture environments, representing a step-change in place recognition methodologies.

UMF distinguishes itself from conventional methods through its sophisticated fusion of local and global multimodal features via transformer-based mechanisms. The principal contributions of this method revolve around exploiting multimodal inputs integrated via transformers, inspired by the performance and versatility of Adafusion, strengthen by robust pre-training. Additionally, it capitalizes on both local and global features for initial matching and subsequent re-ranking of top K candidates, manifesting a formidable strategy to excel in challenging environments while minimizing the matching overhead.

The UMF framework processes LiDAR and visual data inputs, each being encoded through distinct branches. The intermediary outputs of these branches are coalesced through a transformer-based attention mechanism, forming a compact feature representation that is employed for candidate retrieval.

The core of the multimodal fusion process employed by UMF lies in the extraction of features from the visual and LiDAR data, accomplished via a Feature Pyramid Network (FPN) as shown in fig. 4.1. The FPN is specifically chosen for its proficiency in extracting multi-scale features, thereby capturing both the local fine detail and the global context. Using a ResNet50 backbone, the FPN within UMF is endowed with the capability to learn high-level features while displaying resilience against environmental variations.

The coarse intermediate features retrieved from the FPN are fused employing a series of attention layers, balancing the contribution from each modality, taking into account the relative importance of each type of data.

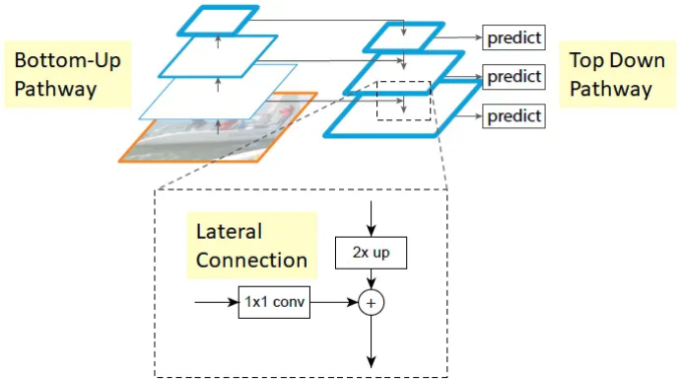


Figure 4.1: Diagram of the Feature Pyramid Network (FPN) used in the UMF architecture. Source: [11]

UMF is deployed in two distinctive versions, each uniquely addressing the incorporation of fine local features:

– Superfeatures, elucidated further in section 4.3

– RANSAC, a method explained in detail in section 4.4

The unique approach for feature extraction, fusion, and attention mechanism in the UMF architecture enhances its ability to adapt and perform reliably in the challenging environments of low-texture and aliasing, improving previously evaluated baselines.

4.1 Self-Attention and Cross-Attention

In the UMF model, attention mechanisms are employed to enhance the ability of the model to dynamically focus on different parts of the input data. UMF utilizes both self-attention and cross-attention mechanisms, as established by Vaswani et al. [49] in the original Transformer model.

In self-attention layers, the model assigns different weights of importance to the features within a single modality (either F_A or F_B), thereby capturing intricate relationships within local and global contexts. This allows the model to identify distinctive patterns within each modality and enhances its ability to recognize places based on a single modality.

Cross-attention layers, on the other hand, take as inputs features from both modalities (either F_A and F_B or F_B and F_A). By interleaving self and cross-attention layers in the UMF module, the model is able to capture the intricate relationships between the two modalities, and this helps improve the robustness of place recognition.

Visualizations of attention weights from self attention layers in UMF are shown in sec. 5.4.2.

The proposed UMF model effectively combines the strengths of the FPN, positional encoding, and transformer-based attention mechanisms, offering an innovative and effective solution for multimodal place recognition in aliased and low-texture environments. Through extensive experiments and evaluations, we demonstrate the superior performance of the UMF model, particularly in challenging scenarios where traditional place recognition methods struggle.

4.2 Integration of Local and Global Features

UMF model incorporates both local and global features to enhance the representation of the environment, thereby reducing perceptual aliasing. This methodology integrates fine-grained details along with the global spatial embedding, refining the model's differentiation capabilities between visually similar locations. The integration of these features builds upon existing methodologies [72] with further advancements in merging these features for multimodal scenarios.

The model utilizes transformers with positional encoding for coarse-level fusion, employing both self and cross-attention. The use of positional encoding is a standard technique used in transformers [49]. We implement the same approach introduced in DETR [73], which ensuring that each element in the feature maps F_A and F_B has unique position information and the transformed features become position-dependent, which enables the model to improve its spatial awareness and the inter and itra modality relationships between features in the fusion branch.

Additionally, it employs transformers along with self-attention for processing fine-grained local features specific to each modality as seen in Fig. 5.5 5.6.

4.2.1 Reranking Mechanism for Ambiguity Resolution

The UMF model incorporates a reranking mechanism to refine the initial ranking of place recognition candidates, taking into consideration the relationships amongst top-ranked candidates as illustrated in Figure 4.2. This mechanism aids in ambiguity resolution, thereby improving place recognition accuracy, particularly in visually similar environments. This unique attribute enhances the post hoc output of the place recognition model.

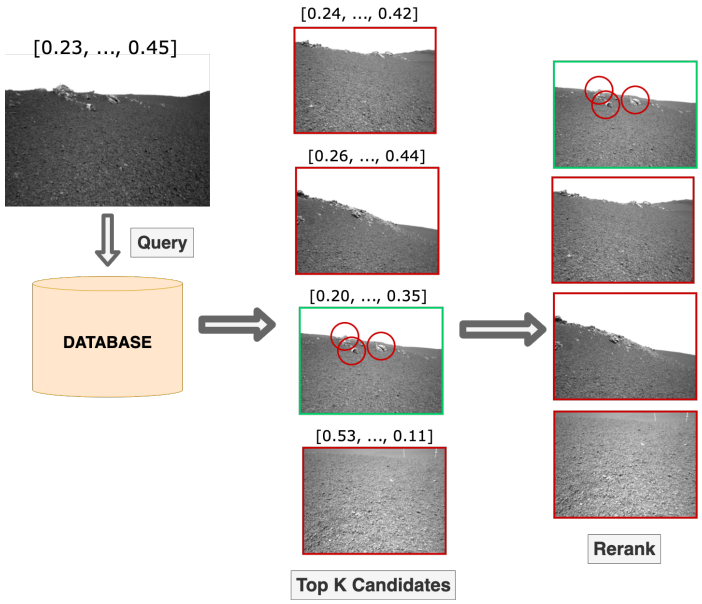


Figure 4.2: UMF's multimodal place recognition reranking pipeline.

4.3 SuperFeatures

Several recent studies have reported excellent performance by methods that merge local and global features in demanding deep image retrieval benchmarks. However, the utilization of local features presents two primary concerns. Firstly, they often amount to localized map activations of a neural network and can therefore be extraordinarily redundant. Secondly, they are typically trained with a global loss acting on top of an aggregated set of local features. Testing, however, is based on local feature matching, leading to a discrepancy between training and testing stages [42].

To address these issues, Weinzaepfel et al.[42] presented a novel Local Iterative Transformer (LIT) module trained via contrastive learning, deviating from traditional methods such as Deep Local and Global features (DELG)[72]. This approach introduces a contrastive loss that directly operates on Super-features, requiring only image-level labels for training.

Super-features, serving as high-level representations encapsulating the most pertinent information for place recognition, are produced by passing local features through a transformer layer. This process results in an ordered set of Super-features of dimensions $[N, F]$, yielding a compact and expressive representation of the local features.

The construction of Super-features involves an iterative attention module, generating an ordered set where each element focuses on a localized and discriminative image pattern. We took the same superfeature principle and extended it to 3D

scenarios, extracting Superfeatures from the voxelized point cloud as shown inf Fig. 4.3.

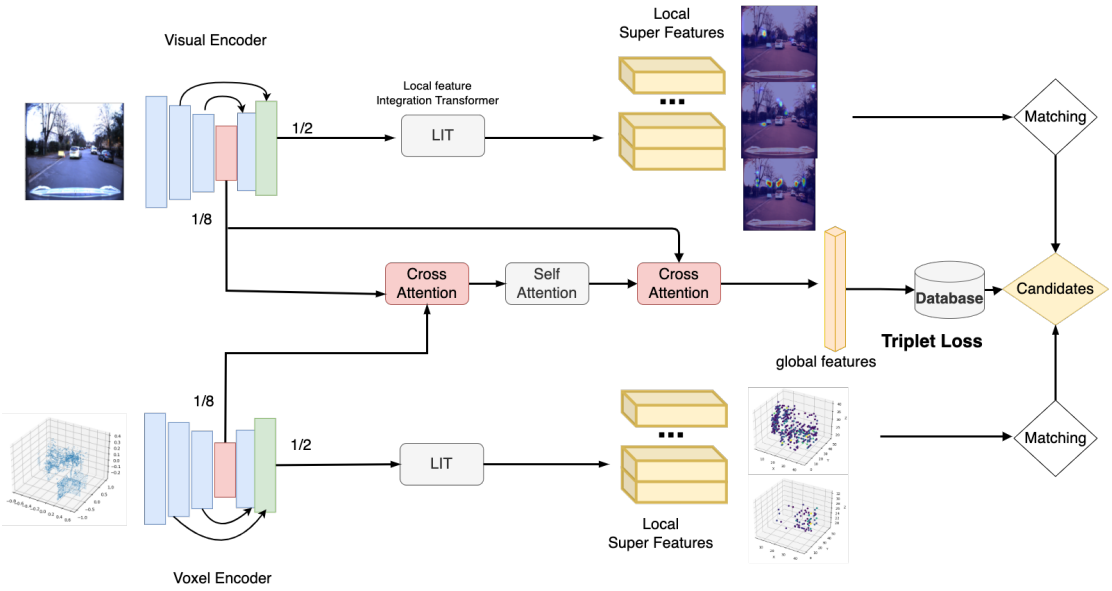


Figure 4.3: Diagram of the proposed UMF model with Superfeature learning. Each branch separately encodes the inputs, the fusion transformer merges the local and global features, and the Superfeature representation is utilized for candidate retrieval and reranking during inference.

To construct a set of eligible Super-feature pairs, the selected Super-features are subjected to a contrastive margin loss. This loss function minimizes the pairwise distance between matching Super-features, while simultaneously reducing the spatial redundancy of Super-features within an image. This process generates a diverse set of Super-features that attend to different local features or different image locations.

Contrastive loss on Super-features:

$$\mathcal{L}_{super} = \sum_{(s,s+)\in\mathscr{P}} \left[\left\| s - s^{+} \right\|_2^2 + \sum_{n\in n(i(s))} \left[\mu' - \|s - n\|_2^2 \right]^{+} \right]$$

where μ is a margin hyper-parameter and the negatives for each s are the Super-features from all n negative images of the training tuple with Super-feature ID equal to $i(s)$.

Reducing spatial correlation between attention maps:

To create as complementary Super-features as possible, they are encouraged to attend to different local features, i.e., different image locations. The cosine similarity between the attention maps of all Super-features of every image is minimized. Let matrix $\boldsymbol{\alpha} = [\tilde{\boldsymbol{\alpha}}_1, \ldots, \tilde{\boldsymbol{\alpha}}_N]$ denote the N attention maps after the last iteration of LIT. The attention decorrelation loss is given by:

$$\mathcal{L}_{attn}(\boldsymbol{x}) = \frac{1}{N(N-1)} \sum_{i \neq j} \frac{\tilde{\boldsymbol{\alpha}}_i^\top \cdot \tilde{\boldsymbol{\alpha}}_j}{\|\tilde{\boldsymbol{\alpha}}_i\|_2 \left\|\tilde{\boldsymbol{\alpha}}_j\right\|_2}, \quad i, j \in \{1, \ldots, N\}$$

In other words, this loss minimizes the off-diagonal elements of the NN self-correlation matrix of α. We ablate the benefit of this loss and others components presented in this section in Section 5.5

4.3.1 Reranking

One of the key challenges in our task is the determination of correspondences at the Super-feature level, especially considering that we only have access to pairs of matching images, i.e., image-level labels. To address this, we propose a simple yet effective reranking mechanism that relies on nearest-neighbor-based constraints.

For any Super-feature $s \in \mathcal{S}$, we define a function $i(s)$ that returns the Super-feature ID, i.e., $i\left(s_i\right) = i, \forall s_i \in \mathcal{S}$. We also define a function $n(s, \delta) = \arg\min_{s_i \in \delta} |s - s_i|_2$ that returns the nearest neighbor of s from the set δ.

Given a positive pair of images $\boldsymbol{x}, \boldsymbol{x}^+$, and two Super-features $s \in \delta, s' \in \delta'$ from their respective Super-feature sets δ, δ', we impose several criteria to consider the Super-feature pair (s, s') eligible:

1. Reciprocal nearest neighbors: $s = n(s', \delta)$ and $s' = n(s, \mathcal{S}')$.
2. Pass Lowe's first-to-second neighbor ratio test: $|\boldsymbol{s} - \boldsymbol{s}'|_2 \,/\, \left|\boldsymbol{s}' - n\left(\boldsymbol{s}', \delta \backslash \boldsymbol{s}\right)\right|_2 \geqslant \tau$.
3. Have the same Super-feature ID: $i(s) = i\left(s'\right)$.

Formally, these conditions can be expressed as:

$$\left(\boldsymbol{s}, \boldsymbol{s}'\right) \in \mathscr{P} \Longleftrightarrow \left\{ \begin{array}{l} s = n(s', \delta) \\ s' = n(s, \mathcal{S}') \end{array} \right\} \quad \text{and} \quad \left\{ \begin{array}{l} i(s) = i\left(s'\right) \\ |\boldsymbol{s} - \boldsymbol{s}'|_2 \,/\, \left|\boldsymbol{s}' - n\left(\boldsymbol{s}', \delta \backslash \boldsymbol{s}\right)\right|_2 \geqslant \tau \end{array} \right\} \tag{4.1}$$

where $\mathscr{P}$ is the set of eligible pairs, and τ is a hyperparameter controlling the Lowe's ratio test. We empirically set $\tau = 0.9$.

Overall, the Super-feature concept and its usage in reranking offer flexibility and significant performance improvement, whilst still maintaining computational and memory efficiency.

4.4 RANSAC Variant

The RANSAC (Random Sample Consensus) variant of the UMF model emphasizes salient feature selection and geometric verification. This variant operates by initially

applying a transformer layer to generate attention maps. These maps then filter and pinpoint salient local features filtering using a hyper parameter δ, retaining only those that carry significant information for place recognition.

The local features are processed using stacked layers of transformer blocks, we use the attention maps to filter the salient features $attn\_score > \delta$ for both modalitites.

Similarly to the super-features, we project the local features to a one dimensional embedding and use a constrstive loss on each local modality during training.

4.4.1 Reranking

Subsequently, the RANSAC algorithm is employed to estimate the geometric transformation between the current observation and candidate locations. This estimation process provides a robust approach to matching local features, accommodating the presence of noise and outliers effectively.

The spatial consistency score is given by the number of inliers returned when fitting a homography between the two images or voxel grids, using corresponding keypoints computed using nearest neighbor matching.

While this approach has the potential to deliver higher accuracy owing to the geometric consistency between matched features, it can also be computationally expensive due to the iterative nature of the RANSAC.

4.5 Training Pipeline

4.5.1 Self-supervised Pretraining

The UMF model leverages unlabeled data from similar domains, such as Mars-like environments in Morocco, for pretraining. This self-supervised learning approach makes the encoder robust to environmental variations, thus minimizing the dependency on labeled data and accelerating model convergence during downstream task fine-tuning.

Our Unifying Local and Global Multimodal Features with Transformers (UMF) model takes advantage of self-supervised pretraining methodologies as discussed in section 2.7. Each part involves the use of different pretraining strategies tailored to the unique characteristics of the data.

This pretraining phase leverages the masked autoencoders for both visual and LiDAR modalities, inspired by Spark [60] and Voxel-MAE [61].

The main objective of the self-supervised pretraining is to generate robust and informative representations of the input data, which are then transferred to downstream

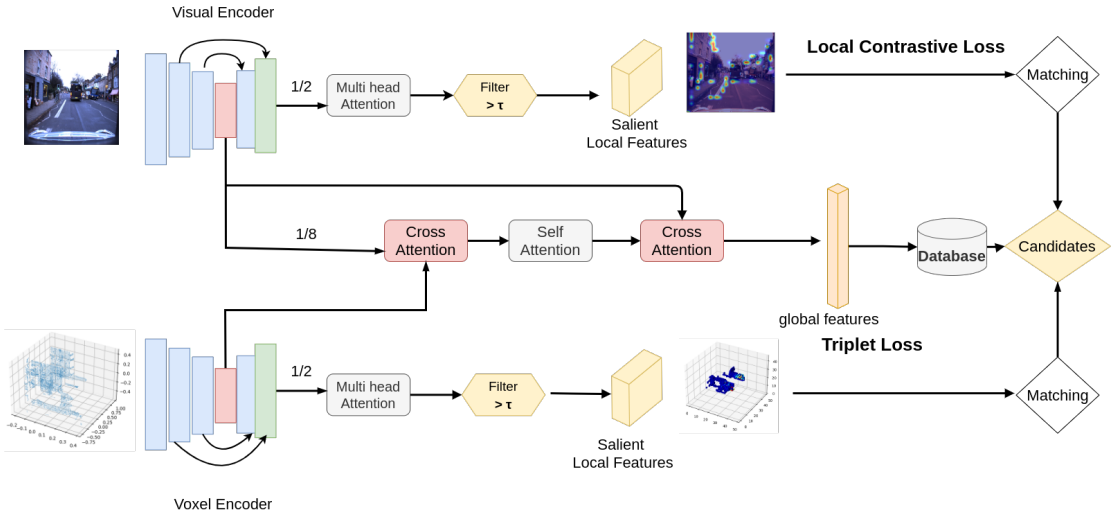


Figure 4.4: Schematic of the proposed UMF RANSAC variant. Each branch independently encodes inputs that contribute to the fusion module and the local feature extraction process. During inference, the top K candidates are retrieved and reranked utilizing the salient local features, where attention $>\tau$.

tasks.

Visual Pretraining

As one of the crucial components of the multimodal pipeline, the visual input undergoes a pretraining strategy focusing on enhancing the spatial consistency within the data. A contrastive learning approach, as depicted in Figure 4.6, is adopted on a vast unlabeled dataset. This approach is designed to discern visually similar yet distinct locations, thereby improving the model's ability to tackle visual aliasing issues.

The pretraining begins with the patch-wise masking strategy that is commonly employed in masked image modeling. An image is segmented into multiple non-overlapping square patches, each subjected to independent masking according to a predetermined mask ratio. The main challenge lies in obscuring the pixel information from these masked patches without disturbing the data distribution of pixel values, preventing the loss of mask patterns through successive convolution operations, and eliminating unnecessary computations on masked regions.

To address these issues, the authors propose to assemble all unmasked patches into a sparse image, as shown in Fig. 4.5, which is then encoded using sparse convolutions. This approach ensures no information leakage, maintains compatibility with any convolutional neural network (convnet) without the need for backbone modifications, and effectively manages the issues of "pixel distribution shift" and "mask pattern vanishing." Moreover, sparse convolution only computes at visible places, leading to

a more efficient process. When fine-tuning, all sparse convolutional layers naturally transform into ordinary dense ones, as dense images can be considered a specific case of sparse images without "holes".

The encoding process is hierarchical; the encoder generates a set of feature maps of varying resolutions or scales. For instance, the selcted ResNet model produces four scales of feature maps, each with different resolutions, which are then used for decoding.

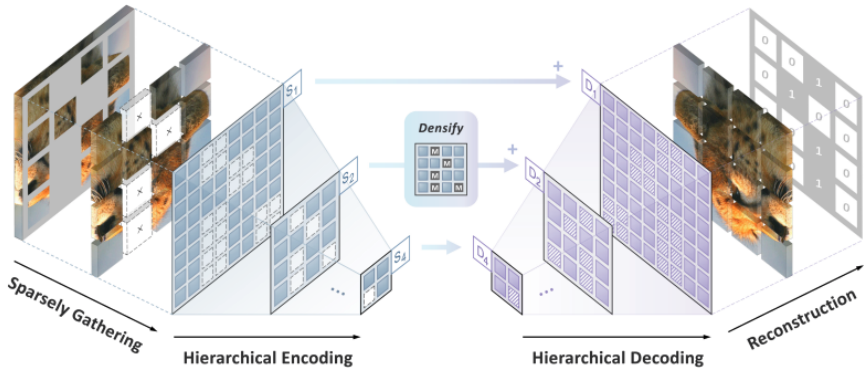


Figure 4.5: Sparse masked modeling with hierarchy. To adapt convolution to irregular masked input, visible patches are gathered into a sparse image and encoded by sparse convolution. To pre-train a hierarchical encoder, we employ a UNet-style architecture to decode multi-scale sparse feature maps, where all empty positions are filled with mask embedding. This "densifying" is necessary to reconstruct a dense image. Only the regression loss on masked patches will be optimized. After pre-training, only the encoder is used for downstream tasks.

The decoder mirrors the design of UNet and includes three successive blocks, $[\mathcal{B}_3, \mathcal{B}_2, \mathcal{B}_1]$, with upsampling layers. Prior to the reconstruction of a dense image, it is necessary to densify all the empty positions on sparse feature maps. This process, termed "densifying", involves the use of mask embeddings $[\mathrm{M}_4]$ to get a dense feature and projection layers ϕ_4, in case encoder and decoder have different network widths.

The optimization target is to reconstruct an image, as shown in fig. 4.6, from D_1 using a head module h that should include two more upsampling layers to reach the original resolution of the input. The authors chose per-patch normalized pixels as targets with an L^2-loss and calculated errors only on masked positions. These decisions are based on previous findings indicating that such designs enable models to learn more informative features. Following pretraining, the decoder is discarded, and only the encoder is used for downstream tasks. When fine-tuning, the pre-trained sparse encoder can be directly generalized to dense images without any tuning.

The derived representations exhibit robustness against out-of-distribution data and surpass the classification accuracy of fully supervised counterparts on diverse

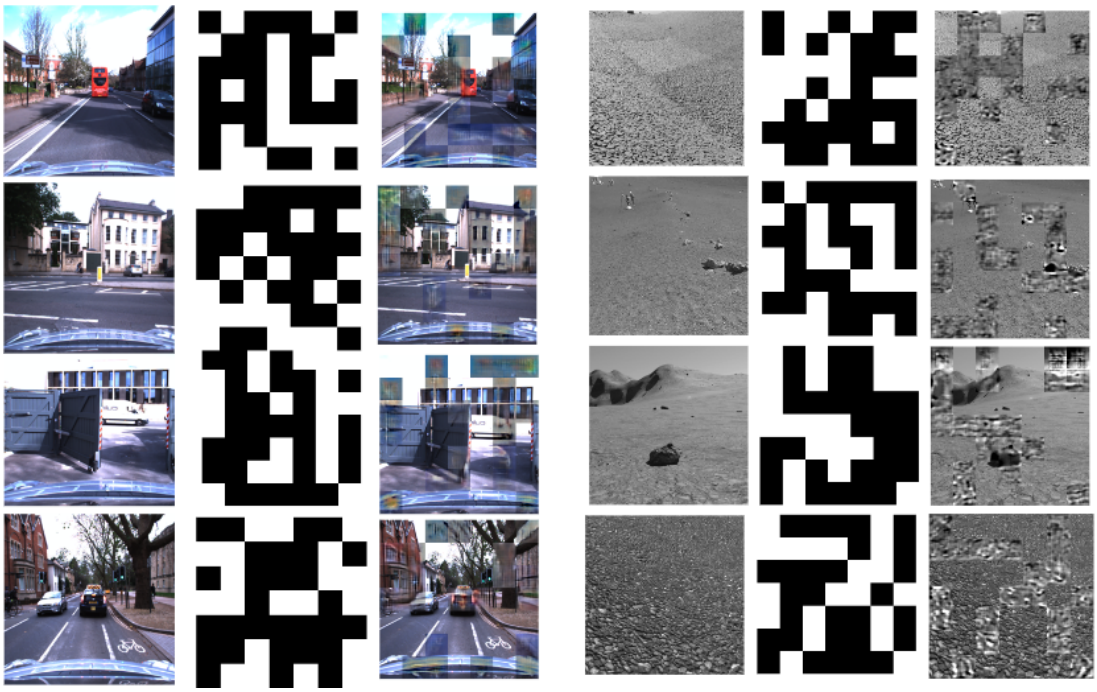


Figure 4.6: Visualization of the self-supervised pretraining phase for the Robotcar dataset, masking ratio 0.6%.

labels. Importantly, the loss computation is restricted to masked patches, thereby circumventing self-reconstruction that may dominate the learning process and obstruct knowledge assimilation.

LiDAR Pretraining

The pretraining strategy for LiDAR data involves reconstructing the original surface from which the 3D points were derived. This self-supervised approach generates latent vectors that serve as inputs for the reconstruction head, as depicted in Figure 4.7 and 4.8. The underlying assumption is that a network adept at reconstructing the scene surface from sparse input points will also capture essential semantic information, which is invaluable for perception tasks. The simplicity of this formulation ensures a straightforward implementation and broad compatibility with various 3D sensors and appications such as place recognition, semantic segmentation or object detection.

The training loss for this process is calculated using Binary Cross Entropy, specifically for the reconstruction of voxel occupancy.

To handle the spatial structure of the point cloud data, a voxel-based approach is employed. This involves partitioning the point clouds into equally spaced voxels, a strategy frequently employed in 3D perception models. Given a point cloud with dimensions $WxHxD$ along the $XxYxZ$ axes, each voxel is of size $v_W x v_H x v_D$, resulting

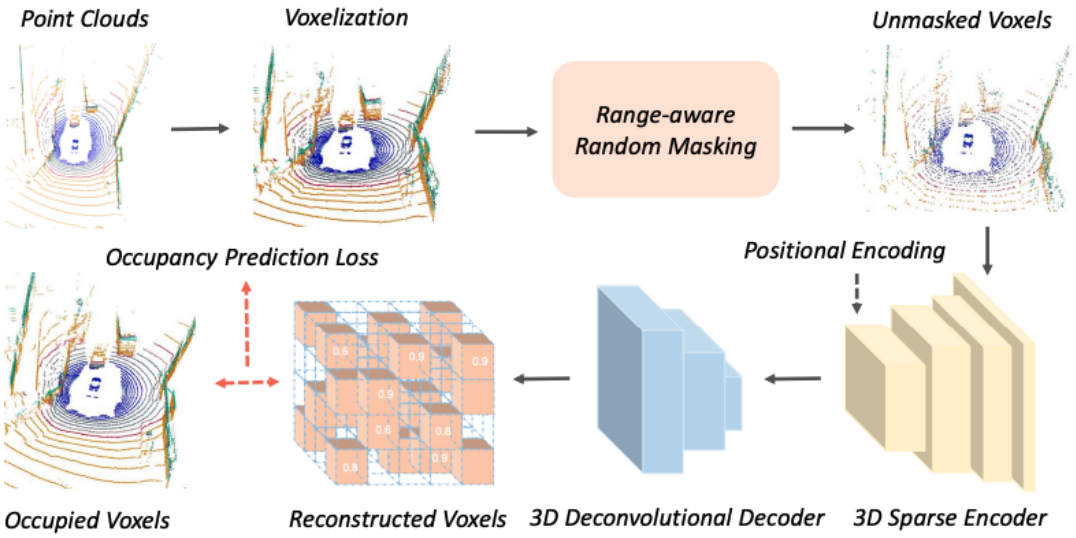


Figure 4.7: Architecture of Voxely-MAE. First transform the large-scale irregular LiDAR point clouds into volumetric representations, randomly mask the voxels according to their distance from the LiDAR sensor (i.e., range-aware masking strategy), then reconstruct the occupancy values of voxels with an asymmetric autoencoder network. The encoder is formed by a set of Spatially Sparse Convolutions with positional encoding. We apply binary occupancy classification as the pretext task to distinguish whether the voxel contains points. After pre-training, the lightweight decoder is discarded, and the encoder is used to warm up the backbones of downstream tasks.

in a total of n_l voxels, where n_v voxels contain points. This voxel-based approach enhances computational efficiency compared to point-based methods.

LiDAR point clouds are unique due to their sparsity levels being directly associated with the distance from the LiDAR sensor. The points closer to the sensor are densely packed, whereas those further away are notably sparse. As such, a standard masking strategy cannot be applied uniformly across both near-range and far-range points. To address this, we employ a range-aware random masking strategy, proposed by Min et al.[61], which takes into consideration the distance information. This strategy separates the occupied voxels into three groups based on their distance from the LiDAR sensor: 0-15 meters, 15-30 meters, and ¿30 meters. The masking ratio decreases with increasing distance, applying a distinct random masking strategy to each group.

In contrast to other masked autoencoding works that primarily aim to reconstruct the masked parts through a regression task, the pretraining for LiDAR data in this work focuses on predicting the occupancy of the 3D scene. This task is crucial in 3D perception, where the occupancy structure of the 3D scene plays a vital role in perception models. Motivated by this, the pretraining aims to encourage the network

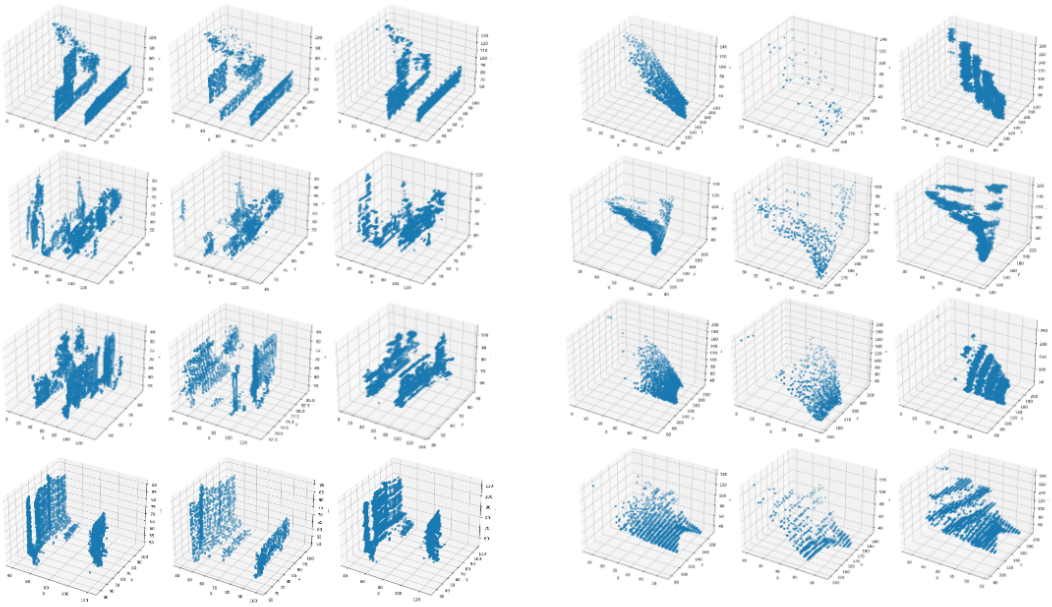


Figure 4.8: Visualization of the self-supervised pretraining phase for the Robotcar on the left and Etna datatset on the right, the first shows the original point cloud, then the masked and final reconstructed one, masking ratio 0.6%.

to reason about high-level semantics to recover the masked occupancy distribution of the 3D scene from a limited number of visible voxels. To this end, a binary occupancy classification loss is calculated using cross-entropy between the predicted occupied voxels $\mathbf{P}$ and the ground truth occupied voxels $\mathbf{T}$:

$$\text{loss} = -\frac{1}{\text{batch}} \sum_{i=1}^{\text{batch}} \sum_{j=1}^{n_l} \mathbf{T}_j^i \log \mathbf{P}_j^i,$$

where $\mathbf{P}_j^i$ represents the predicted occupancy probability of voxel j for the i-th training sample, and $\mathbf{T}_j^i$ corresponds to the ground truth indicating whether the voxel contains point clouds.

4.5.2 Downstream task: Place recognition

Upon completing the self-supervised pretraining phase, we fine-tune the UMF model on place recognition tasks using a triplet margin loss with batch hard negative mining strategy, inspired by MinkLoc [13]. Each triplet consists of an anchor, a positive, and a negative example. We define similarity based on spatial proximity, with a radius of 12 meters for similar locations and a distance of more than 60 meters for dissimilar locations, thereby introducing a neutral zone.

Our training strategy employs batch-hard negative mining to construct informative triplets and disregard less informative ones. Specifically, we focus on active triplets, where the loss exceeds the margin, as they provide valuable insights for model refinement.

Lastly, we have to balance and adjust the global triplet loss to the individual local modality losses. This balance aids in fostering harmonious interaction between the different branches, which subsequently boosts overall performance. Further research is required to optimally calibrate this interplay, potentially revealing more sophisticated ways of integrating multi-modal data within the UMF framework.

4.6 Implementation Details

UMF, built on the PyTorch deep learning library, is trained on a multi-GPU server. We employ the Adam optimizer for weight updates, with an initial high learning rate for quick convergence, which is gradually decreased via a learning rate scheduler for fine-tuning the model.

The key advantage of this 2 step training strategy is the option to freeze the encoder during the fine-tuning phase. This action dramatically decreases computational requirements and helps prevent the model from overfitting. The encoder, post its pre-training phase, is already adept at recognizing necessary features. Therefore, any further fine-tuning on a narrower dataset might lead to the model fitting too closely to this specific dataset, reducing its general applicability.

The model's performance is assessed on two datasets: the common benchmark RobotCar and the Etna dataset. An in-depth comparison with state-of-the-art methods is covered in Chapter 5.4.

Chapter 5

Evaluation

In this chapter, we present a comprehensive evaluation of our proposed Unified Multimodal Fusion (UMF) model's performance is assessed primarily on the Mt. Etna Dataset, a novel dataset obtained from Stereo and Solid-State LiDAR Inertial sensors in the lunar-like environment of Mount Etna. This dataset provides a rigorous testbed for evaluating the robustness and precision of the model in challenging scenarios. Additionally, we also deploy the renowned Oxford RobotCar dataset [74] for further validation and comparison against state-of-the-art methods.

5.1 Datasets and Preprocessing

5.1.1 Oxford RobotCar

The Oxford RobotCar dataset [74] is a cornerstone dataset in the autonomous driving research domain, offering an extensive variety of driving scenarios across distinct weather conditions and times of day. This dataset, featuring a suite of sensors (RGB cameras, LiDAR sensors, GPS/INS) mounted on a car that repeatedly traverses the same route in the city of Oxford at different times of day and year, offers a rich and diverse data source for training and testing our model.

Point clouds are generated from consecutive 2D LiDAR scans during a 20-meter drive, with the ground plane removed and the point clouds downsampled to 4096 points. Corresponding RGB images with the closest timestamps are retrieved from the original RobotCar dataset for each point cloud, with each image downsampled from 1280x960 to 320x200 resolution.

To enhance data diversity and limit overfitting, we randomly sample from 15 closest RGB images during training, while only one RGB image with the closest timestamp is used during evaluation. Similarity between elements is defined based on their spatial proximity: elements within 10m are deemed similar, while those separated by at least 50m are considered dissimilar. Those falling between 10 and 50m are treated as neutral.

The dataset is split into disjoint training (21.7k elements) and test (3k elements) areas based on UTM coordinates, following the evaluation protocol and train/test split (Baseline scenario) introduced in [13].

5.1.2 Mt. Etna Dataset

The Mt. Etna dataset [6], collected in the volcanic environment of Mt. Etna, Sicily, offers a unique challenge due to its resemblance to lunar and Martian landscapes. The dataset consists of 7 sequences recorded at an altitude of 2650 meters near the Cisternazza crater, characterized by a surface of smooth dark lava ash, extreme visual contrast, and a scarcity of unique geological features. This challenging environment for localization algorithms provides a realistic multisensory dataset for testing UMF in planetary exploration-like scenarios.

Sensor Suite

The Lightweight Rover Unit (LRU) sensor suite, detailed in Section 1.4.1, was employed for collecting the Mt. Etna Dataset. It comprises the Blickfeld Cube-1 LiDAR, two AVT Mako stereo cameras, an XSens MTi-G 10 IMU, and a Ublox f9p GNSS receiver. The data was captured as the LRU navigated diverse terrain and harsh lighting conditions on Mount Etna, providing a realistic and challenging dataset for our research.

Data Preprocessing

To ensure that the data is suitable for training and testing, it undergoes a series of preprocessing steps. These include filtering and downsampling the LiDAR data, stereo rectification, and disparity computation for the stereo images. The data from different sensors is synchronized based on timestamps to ensure consistency. To avoid overlap between samples, the images are subsampled and aligned using their timestamps. The estimated pose is used as ground truth for evaluation.

Ground Truth Generation

For the Mt. Etna dataset, ground truth positions of the LRU's trajectory are derived from the processed GPS and IMU data, using VINS-Fusion [75, 76, 77]. These ground truth positions serve a dual purpose: they provide a benchmark for assessing the accuracy of our proposed methods and they provide camera poses for training our models. The estimation of accurate ground truth is a crucial aspect of our evaluation as it provides a reliable reference for assessing the model's performance.

For this research, we consider points as positives if they are less than 12 meters away and negatives if they are more than 60 meters away or the orientation diverges for more than 30ª in either direction.

Sequences

The Mt. Etna dataset is partitioned into unique traverses, each encapsulating a distinct set of terrain types, lighting conditions, and geological features. This dataset's comprehensive nature provides an exhaustive and challenging testing ground for our proposed UMF model. The full trajectories can be seen in Fig. 5.1 alongside a detailed breakdown in table 5.3.

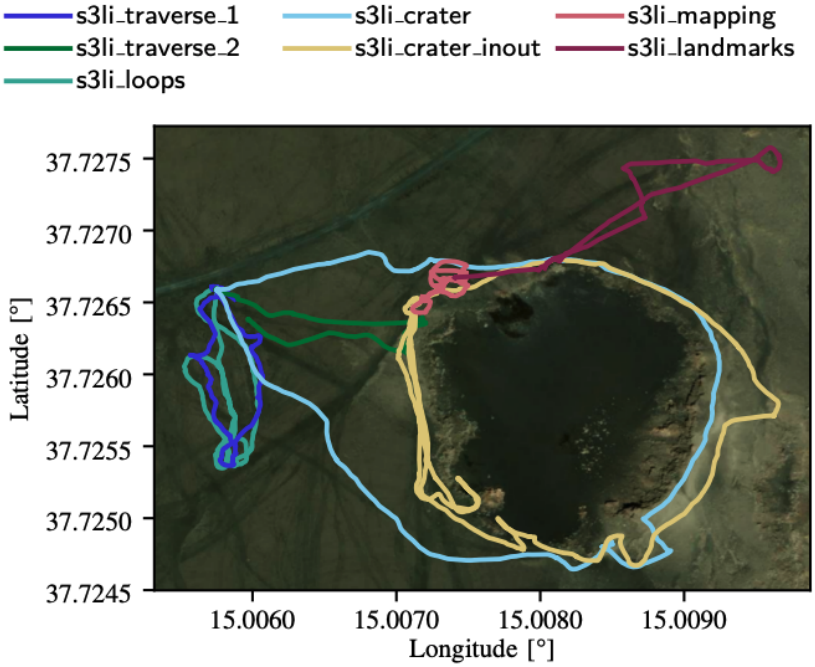


Figure 5.1: Bird's-eye view of the dataset recording site with the trajectories for each sequence overlaid. The Cisternazza crater is visible on the right. Source: [6]

Sequence	Length
s3li_traverse_1	726
s3li_traverse_2	642
s3li_loops	858
s3li_crater	1148
s3li_crater_inout	1590
s3li_mapping	696
s3li_landmarks	960

Table 5.1: Mt Etna place recognition dataset composition, each sample has stereo, lidar, and ground truth pose.

We divide the Mt. Etna Dataset into training and validation sets. All sequences are used for training except for *s3li_loops* and $'s3li\_traverse\_1'$, as these sequences have more overlapping sections. This partitioning strategy ensures the robustness and reliability of our evaluation by preventing model bias towards the training set.

5.1.3 Additional Datasets for Pretraining

To further bolster the model's robustness and improve its final performance, we incorporate additional datasets with images from planetary-like environments. These datasets lack LiDAR or use a different sensor and are utilized during the pretraining phase.

The additional datasets include images captured from the Australian team present in the last Arches mission from the University of Technology Sydney (UTS) in Mt. Etna, other missions from the DLR team such as the MADMAX dataset from Morocco, and Mt. Etna from previous years. The details of these datasets are as follows:

Dataset	Notes
Merzouga[78]	Stereo + HDL64 or HDL32
MADMAX	Morocco, Stereo only
Etna_2018	Stereo only
Etna_UTS	Stereo LiDAR, no ground truth

Table 5.2: Additional datasets used for the pretraining phase. These datasets do not require LiDAR information or ground truth poses.

5.1.4 Synthetic Dataset

To overcome the restrictions imposed by limited access to real-world planetary like environments, we create a new synthetic dataset. This approach aims to supplement our training data, boosting the model's robustness and generalization capabilities to a broader range of situations. To achieve this goal, we use OAISYS [12], a state-of-the-art, photorealistic terrain simulation tool explicitly designed for robotics research, built upon the foundations of BlenderProc [79]. This tool offers the ability to simulate a broad range of terrains, lighting conditions, textures, and rock formations, shown in Figs. 5.3 5.4, thereby enriching our training data with scenarios that might be absent in our real-world Mt. Etna dataset.

As demonstrated in Fig. 5.2, it can produce instance, semantic segmentation, depth, and point cloud data.

The generation of synthetic datasets begins with the random sampling of the initial location and trajectory of the robot. Subsequent trajectory generation involves

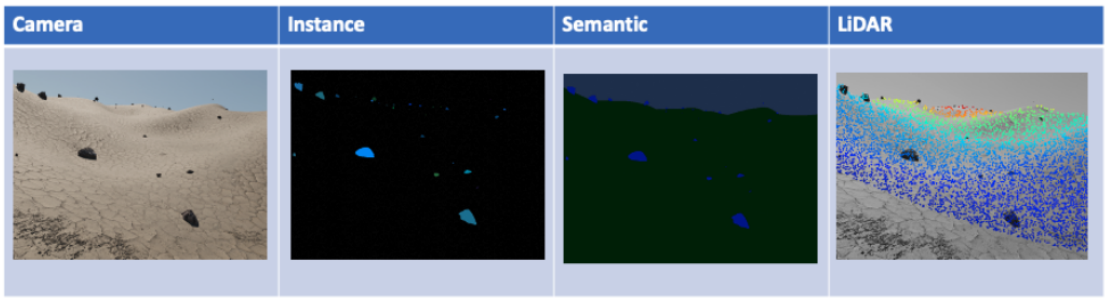


Figure 5.2: Sample from the generated synthetic dataset: (a) Stereo camera view, (b) Instance segmentation, (c) Semantic segmentation, and (d) Point cloud.

consecutive incremental steps that navigate the simulated environment. These steps mimic the natural movement of a robotic system, providing an authentic learning setting for the UMF model.

We used a sensor module that sends viewpoint poses to OAISYS, waits for the rendering to finish, and then saves the RGB and depth data alongside the labels. The point clouds were generated from depth renderings using a pinhole camera model, optinally some noise can be added to imitate natural ocurring noise from the LiDAR sensor.

The synthetic dataset comprises various sequences as detailed in Table 5.3, each offering different environmental characteristics. All samples include stereo, lidar, instance, and semantic segmentation maps, providing a rich dataset for training and evaluation.

Sequence	**Length**
$OAYSIS\_random\_1, 2, 3$	500, 500, 500
$OAYSIS\_canyon\_1, 2$	500, 500
$OAYSIS\_terrain\_1, 2$	500, 500
$OAYSIS\_dunes\_1, 2$	500, 500
$OAYSIS\_mesa\_1, 2$	500, 500
$OAYSIS\_mounds\_1, 2$	500, 500

Table 5.3: Synthetic dataset composition, each sample has stereo, lidar, instance and semantic segmentation maps.

5.1.5 Data Augmentation

Data augmentation techniques are employed to prevent overfitting and to increase model robustness. These techniques include random cropping, flipping, and rotation of the images, as well as random scaling and rotation of the LiDAR point clouds. By introducing variability and complexity to the data, these techniques help our models

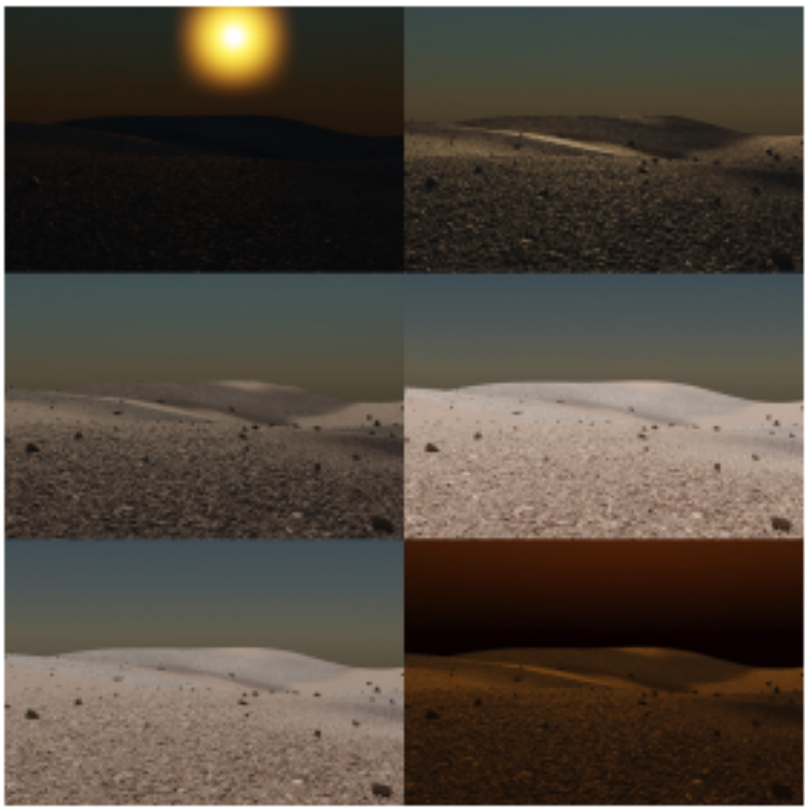


Figure 5.3: Examples of various lighting and atmospheric conditions extracted from [12].

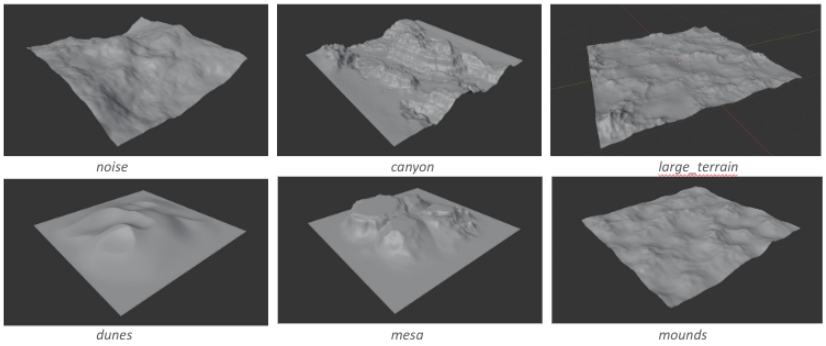


Figure 5.4: Examples of various types of terrains simulated in the synthetic dataset.

learn more generalized and robust representations.

For images, we employ RandomErasing, ColorJitter, and Normalization. For point clouds, we use RandomRotation, RandomFlip, and RandomErasing. These augmentations help generalize and prevent overfitting, especially to the visual modality.

5.1.6 Hyperparameters

This information is crucial for understanding the behavior of our models and for reproducing our results.

The specific values of the parameters and hyperparameters used in our implementation, such as the learning rate, batch size, and number of training epochs, are detailed in the appendix B.1.

5.2 Metrics

In order to evaluate the performance of our model, we employ a set of standard metrics commonly utilized in place recognition literature [44, 13]. These include recall@K and Area Under the Curve-Precision Recall (AUC-PR).

Precision and Recall constitute the core components of these metrics. Precision represents the fraction of correctly identified loop closures amongst all detected closures, whereas Recall or Sensitivity reflects the ratio of true loop closures correctly identified by the model. These metrics are formally defined as:

$$Precision = \frac{TP}{TP + FP} \tag{5.1}$$

$$Recall(Sensitivity) = \frac{TP}{TP + FN} \tag{5.2}$$

Within the field of place recognition for Simultaneous Localization and Mapping (SLAM), TP (true-positive) is associated with correctly recognized loop closures, FP (false-positive) with incorrectly recognized loop closures, and FN (false-negative) with genuine loop closures that remained unrecognized by the model.

To further understand the model's performance, we consider the following specific metrics:

- **Recall@N:** This metric gauges the model's competency in identifying the correct match within the top N results. A superior recall@N score signifies that the accurate match is more probable to be found within the top N matches delivered by the model.

- **Recall@1:** This is a special case of recall@N with N=1. It evaluates the model's precision in pinpointing the correct match as the first result.

- **Recall top** 1%**:** This metric assesses the model's capacity to discover the correct match within the top 1% of the retrieved results.

Through these comprehensive metrics, we aim to evaluate and understand the overall performance of our UMF model in place recognition tasks.

5.2.1 Similarity Measure

We employ a cosine similarity measure to determine the similarity between query and database images, defined as:

$$Similarity = \frac{A \cdot B}{||A|| \times ||B||} \tag{5.3}$$

Here, A and B are the feature vectors of the query and database images, respectively. The cosine similarity measures the cosine of the angle between two vectors. This measure is insensitive to the scale of the features and instead focuses on the angle between the feature vectors, making it more robust to variations in feature magnitude introduced by changes in lighting, viewpoint, or other environmental factors.

The cosine similarity is often used because it provides a normalized measure that is robust to changes in the magnitude of the feature vectors. It is more interested in the direction of the feature vectors, which is crucial when comparing high-dimensional features, as we do in our work.

5.3 Experiments

5.3.1 Baseline Methods

First we evaluate the selected baselines in the Etna dataset. Then, we analyze the results and compare them to our proposed UMF method.

5.3.2 Implementation Details

All models were implemented using PyTorch [33] and trained on a system equipped with an NVIDIA RTX 3090 GPU. The learning rate was set to $1e-4$ and reduced by a factor of 0.1 upon plateauing. The models were trained for a total of 100 epochs using the Adam optimizer [80] with a batch size of 64. The input image size was set to 224×224. When re-ranking global feature retrieval results with local feature-based matching, the top 25 ranked images from the first stage are considered, each version of UMF uses a different reranking strategy and if there is a tie we use the global feature distance.

Superfeatures The strength of each superfeature was determined using *L2Attention*. The attention maps were generated for image sizes of 56×56 and voxel sizes of $40 \times 40 \times 40$. The superfeature was represented as a tensor of size $[N, F]$, where

N is the number of super-features and F is the dimensions of each one, the dimensions for the visual and point cloud features is set to 128 and 32 respectively. The final ranking is based on the number of matching features that satisfy the criteria described in sec. 4.3.

RANSAC We used a total of 2 transformer layers in our model. The model returns the average of all attention maps and selects an optimal threshold to identify keypoints.

The output consists of attention maps for the image and voxel, each of size $[N, 56, 56]$ and $[N, 40, 40, 40]$, respectively, where N is the number of keypoints. The feature maps for the image and voxel have dimensions $[56, 56, 128]$ and $[40, 40, 40, 8]$, respectively. The reranking of candidates is done based on the total number of inliers, for one or both modalities: $score = [\#inliers_{pc} + \#inliers_{img}]$

5.4 Results

This section outlines the experimental results. The performance of the proposed UMF method is compared with the baseline approaches in terms of AUC-PR, and Recall@N. Extensive experiments were conducted to contrast our proposal with the baselines on the Etna datasets. Furthermore, we validated UMF using the RobotCar dataset, which allows us to compare our method's performance with the most recent state-of-the-art multimodal systems.

5.4.1 Quantitative Results

The comparative performance of the UMF model versus other methods on the Etna dataset is delineated in Table 5.7. The data is segmented according to modality—Visual, LiDAR, and Multimodal—for clarity and concise evaluation.

UMF's superior performance is evident in the Etna dataset results. This can be attributed to the method's effective fusion of local and global image features.

While the LiDAR data from the Etna dataset has a limited field of view, inherently restricting its application for place recognition, it provides valuable input under challenging conditions by diminishing uncertainty. Moreover, its delivery of accurate depth information plays a crucial role in establishing the correct positive pairs.

The geometric verification using RANSAC shows significant impact in aliasing environments, outperforming other approaches. It is noteworthy that it enhances the robustness of the final predictions. Conversely, Superfeatures struggle to consistently focus on the most salient regions when there are few landmarks, which may be a consequence of the decorrelation loss at the attention maps level. These features are compelled to investigate different areas.

Method	Recall@1	Recall@5	Top 1% recall
Visual			
DBoW2	37.44	66.1	68.12
NetVLAD	67.2	75.5	78.3
MinkLoc++ (visual)	68.8	77.3	79.2
LiDAR			
PointNet++	48.41	67.77	71.8
MinkLoc++ (LiDAR)	42.4	65.8	69.4
Multimodal			
MinkLoc++ (fusion)	71.4	80.1	85.2
AdaFusion	73.1	82.3	87.2
UMF	73.5	82.9	87.5
UMF (superfeat)	75	85.1	89.1
UMF (RANSAC)	**75.3**	**85.3**	**89.5**

Table 5.4: Performance comparison of various methods on Mt. Etna dataset, categorized by data modality. The versions of UMF with reranking for both modalities

In aliased environments, geometrical verification via RANSAC appears crucial as it outperforms other approaches. This is anticipated since it enhances the robustness of the final predictions. Additionally, superfeatures occasionally struggle to focus on the most salient regions when there are hardly any landmarks, due to the decorrelation loss at the attention map level, causing them to be compelled to look at different areas (Fig. 5.5).

We further assess the performance of UMF using the RobotCar dataset, and the results are presented in Table 5.8.

Method	Recall@1	Top 1% recall
Multimodal		
MinkLoc++ (fusion)	96.7	99.1
AdaFusion	98.1	99.2
UMF	97.9	99.1
UMF (superfeat)	98.1	99.1
UMF (RANSAC)	**98.3**	**99.3**

Table 5.5: Performance comparison of various methods on the RobotCar, the versions of UMF with reranking use both modalities.

The RobotCar dataset, which provides richer geometric information via LiDAR compared to Etna, demonstrates the robustness and adaptability of UMF across different data modalities and environments. This richness of geometric information can be leveraged by UMF to further enhance its place recognition capabilities.

It is pertinent to note that the quality of local features and the effectiveness of the reranking mechanism significantly influence UMF's overall performance. Thus, these

components necessitate meticulous optimization.

Our UMF model, specifically devised for place recognition tasks, exhibits proficiency in handling aliased and low-texture environments. It synthesizes the strengths of LiDAR and visual data through the use of transformers to coalesce local and global features. Further, the model employs an attention mechanism coupled with a reranking process to augment its performance.

The super-feature variant of the UMF model demonstrates adaptability to diverse requirements in terms of accuracy and computational efficiency. Regardless of the degree of aliasing or complexity of the scenario, both UMF model variants can effectively navigate challenging place recognition tasks. This resilience underscores the potential of UMF as a robust solution for multimodal place recognition.

5.4.2 Qualitative Results

This subsection provides an in-depth qualitative evaluation of our proposed methodology. It aims to show the potential strengths and challenges faced by the model, thereby shedding light on its overall efficacy, robustness, and scalability.

As previously explained in Section 4.3, our proposed UMF model with superfeatures facilitates the alignment of the extracted superfeatures when presented with a positive pair. This pivotal functionality can be seen in Figures 5.5 and 5.6. These figures effectively visualize the model's aptitude to accurately map similar features across different images, emphasizing the potency of our approach in identifying and aligning intricate patterns and objects.

Nevertheless, our model may struggle under certain circumstances. Specifically, when the overlap between images is sparse, the model may stumble upon challenges in aligning superfeatures, leading to some misalignment. These non-aligned superfeatures are subsequently discarded during the matching process, as elaborated in Section 4.3.1. This finding underscores the model's self-corrective mechanism, which discards misaligned features, thereby enhancing the precision of its output.

Our model also exhibits proficiency in handling point clouds, as depicted in Figure 5.7. This demonstrates its ability to incorporate supplementary information from areas that may be concealed from the camera's line of sight, thereby augmenting the model's overall efficiency and comprehensiveness.

Similarly the RANSAC variant also showcases great effectiveness in feature extraction. As seen in Figures 5.8 and 5.9, it demonstrates a remarkable ability to highlight salient and discriminative elements within each modality, providing a more precise and comprehensive mapping between different samples.

Additionally, we present the global embeddings in a 2D space, which are derived

using Principal Component Analysis (PCA) and t-distributed Stochastic Neighbor Embedding (t-SNE). As popular techniques for dimensionality reduction, both PCA and t-SNE facilitate effective visualization of high-dimensional data, offering insights into the feature distribution and potential clusters within the embeddings.

As shown in Figures 5.10 and 5.11, our UMF model performs effectively in projecting similar locations close together in the global embedding space. This successful proximity projection is emblematic of the model's ability to understand and capture the semantic similarity between different scenes, reinforcing the robustness and efficacy of our proposed approach in place recognition tasks.

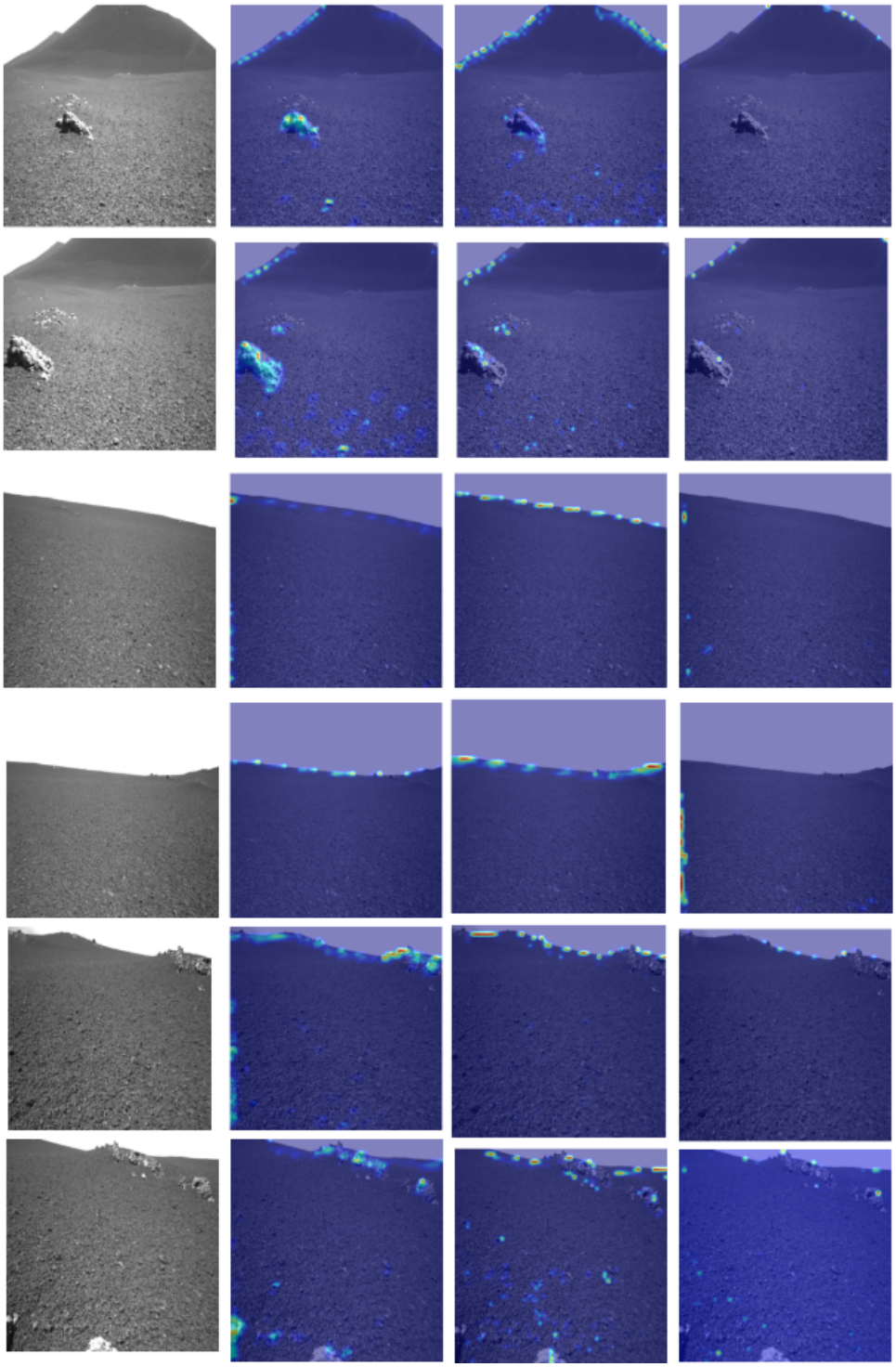


Figure 5.5: Illustration of the super-features attention maps for the image modality generated by the Learning Iterative Transformer (LIT) module for three distinct positive image pairs from the Etna dataset. The first three super-features are exhibited, showcasing the model's propensity to consistently focus on specific semantic patterns, such as varying rock formations and terrain structures.

Figure 5.6: Demonstration of the super-features attention maps for the image modality generated by the Learning Iterative Transformer (LIT) module for three image pairs from the Robotcat dataset. Each pair is a positive match, with the first three super-features depicted. These super-features exhibit a recurring focus on specific semantic patterns like windows and traffic signs while effectively discarding dynamic objects like cars or cyclists.

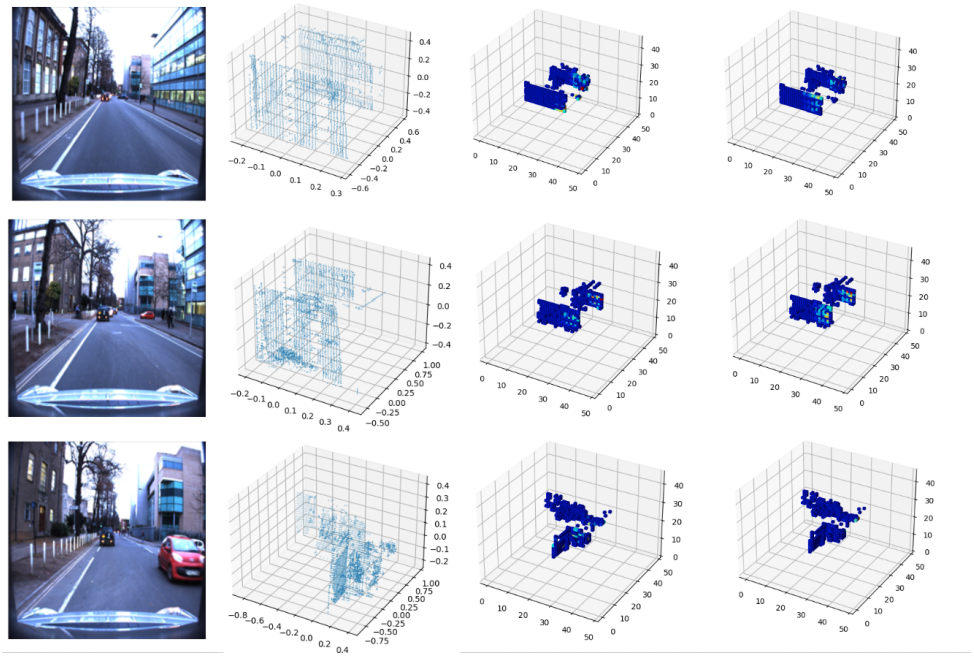


Figure 5.7: The super-features attention maps for point clouds are generated by the modified 3D Learning Iterative Transformer (LIT-3D) module for three consecutive samples from the Robotcat dataset. The first two super-features are prominently displayed, demonstrating the model's capability of processing and extracting useful features from point cloud data.

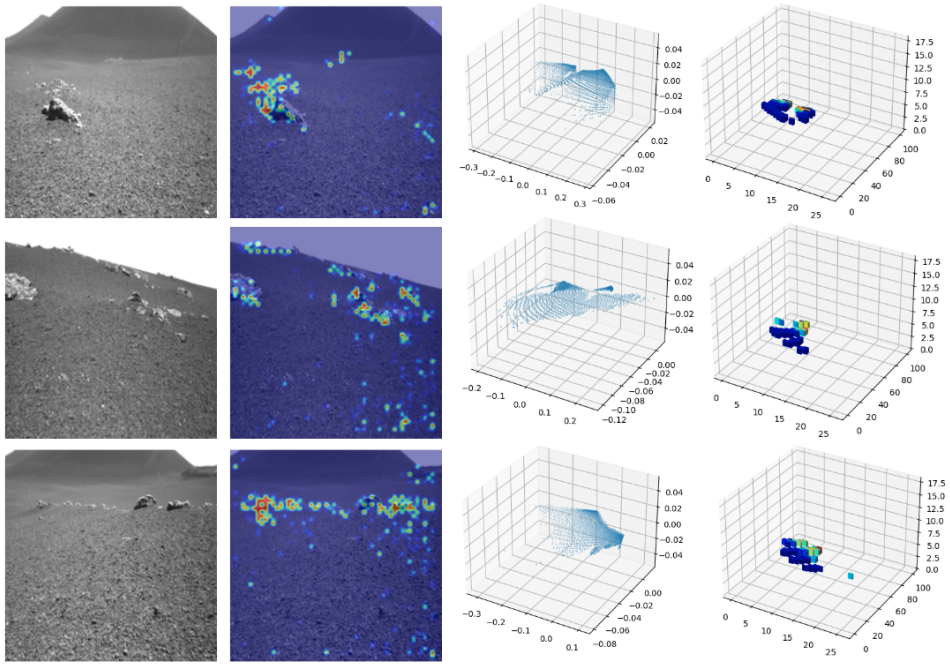


Figure 5.8: Visualization of the visual features extracted using the RANSAC variant on the Mt Etna dataset, the local features are filtered and selected to construct key points for subsequent matching.

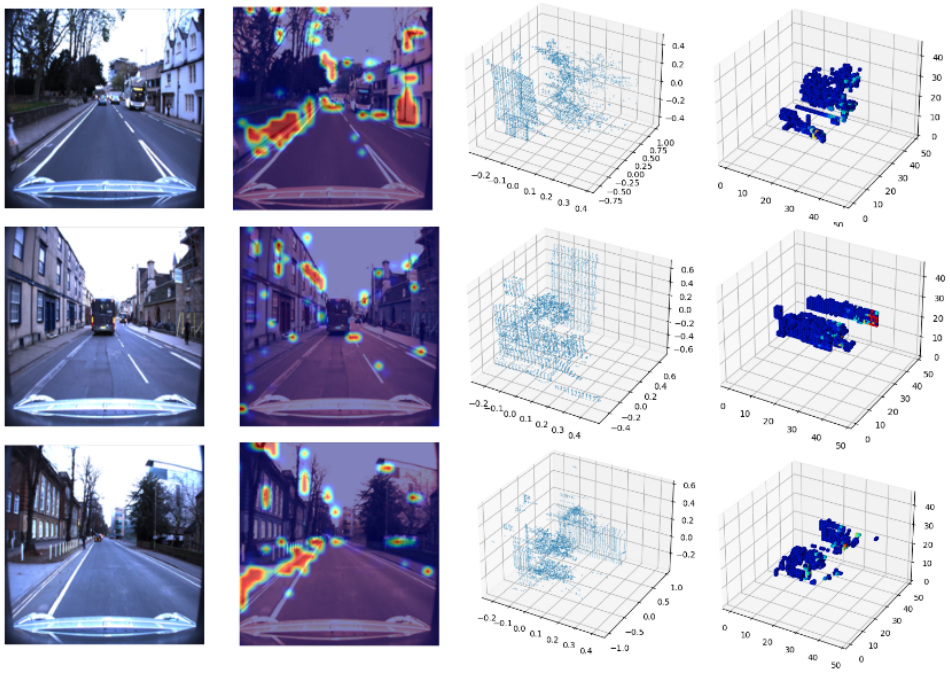


Figure 5.9: Visualization of the visual features extracted using the RANSAC variant on the Robotcar dataset, using the same process for matching.

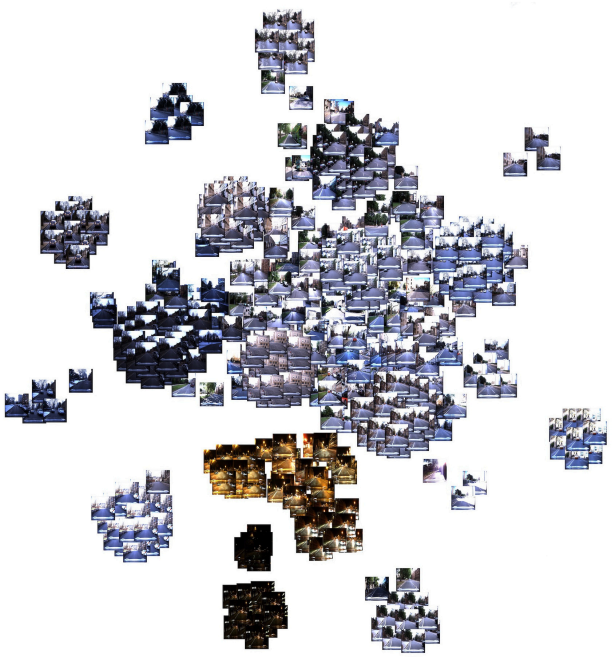


Figure 5.10: Visualization of global embeddings using PCA and t-SNE on the RobotCar dataset. Each data point on the scatter plot represents a unique place.

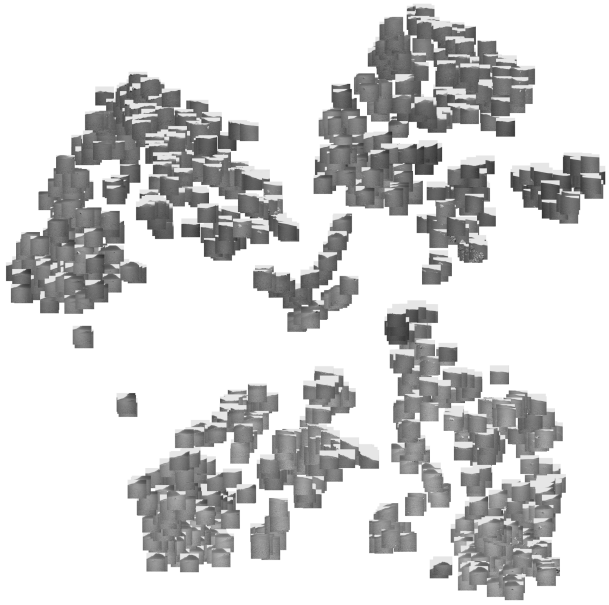


Figure 5.11: Visualization of global embeddings using PCA and t-SNE on the Etna dataset, showcasing well-separated clusters that correspond to distinct locations.

5.5 Ablations

In this section, we conduct a detailed examination of our proposed UMF model, with a focus on understanding the individual contributions of each component. This in-depth analysis of each element will shed light on its relative importance to the overall model's performance, enabling further improvements and optimizations.

5.5.1 Impact of Pre-training

To assess the impact of pre-training on our model's performance, we experiment under two distinct scenarios: 1) training the model from scratch without any pre-existing knowledge, and 2) initializing the model with pre-trained weights. This comparative study aims to emphasize the significance and contribution of pre-training to our method's overall effectiveness.

Method	**Recall@1**	**Recall@5**	**Top 1% recall**
UMF (with pre-training)	73.5	82.9	87.5
UMF (without pre-training)	70.4	81.2	85.9

Table 5.6: Influence of pre-training on UMF model's performance in a place recognition task. The comparison is between the UMF model initialized randomly and the one using pre-trained weights.

Table. 5.6 demonstrates the crucial role of pre-training in ensuring optimal model performance, especially as the complexity of the model increases. The pre-training phase also reinforces the model's robustness and generalization capabilities by reducing the propensity of the visual modality to overfit.

5.5.2 Reranking

We conducted a series of experiments to analyze the role and importance of the reranking module in our approach. This involved variations in the reranking process such as implementing with and without reranking for one or all modalities, varying the number of candidates and altering the number of super features.

The post-reranking improvements in the model's recall rates substantiate the importance of the additional reranking step in our approach. Notably, we observe that the visual modality outperforms the others, providing a significant boost in the recall rates. However, the reranking step using LiDAR data only shows marginal improvements, as the visual modality proves to be dominant. Fusion of both modalities aids in overcoming visual domain challenges such as poor lighting conditions or aliasing.

Method	Recall@1	Recall@5	Top 1% recall
Visual			
UMF (superfeat visual)	74.5	84.3	89.
UMF (RANSAC visual)	75.1	84.9	89.3
LiDAR			
UMF (superfeat pc)	73.7	83.4	87.5
UMF (RANSAC pc)	73.9	83.8	87.8
Multimodal			
UMF	73.5	82.9	87.5
UMF (superfeat all)	75.	85.1	89.1
UMF (RANSAC all)	**75.3**	**85.3**	**89.5**

Table 5.7: Comparison of various reranking methods applied to the UMF model on the Mt. Etna dataset, separated by data modality.

Method	Recall@1	Recall@5	Top 1% recall
Base			
UMF	97.9	98.3	99.1
Visual			
UMF (superfeat visual)	98.	98.4	99.1
UMF (RANSAC visual)	98.1	98.5	99.2
LiDAR			
UMF (superfeat pc)	97.8.	98.2	99.1
UMF (RANSAC pc)	98.	98.3	99.1
Multimodal			
UMF (superfeat all)	98.1	98.5	99.1
UMF (RANSAC all)	**98.3**	**98.8**	**99.3**

Table 5.8: Performance evaluation of the UMF model with various reranking methods on the real-world RobotCar dataset.

The benefits of fusing both modalities is particularly prominent in the RobotCar dataset, which possesses a rich point-cloud information set.

Notably, RANSAC emerged as a clear winner among the reranking methods, offering superior performance. However, this advantage comes at the expense of increased computational overhead, which we analyze further in sec 5.5.3.

Fig. 5.12 contains a quantitative comparison of both ranking approaches and the baseline methods where we study the impact of the number of candidates. The measurement can vary significantly depending on the dataset used, but we found taking the top 20 candidates is a reasonable trade-off for most use cases.

In our final analysis, we adjusted the normalized similarity threshold α to examine the effectiveness of each variant through the precision-recall curves shown in Fig. 5.13. Expectedly, RANSAC outperforms the superfeatures curve. Despite both approaches offering competitive performance compared to the baseline model without reranking,

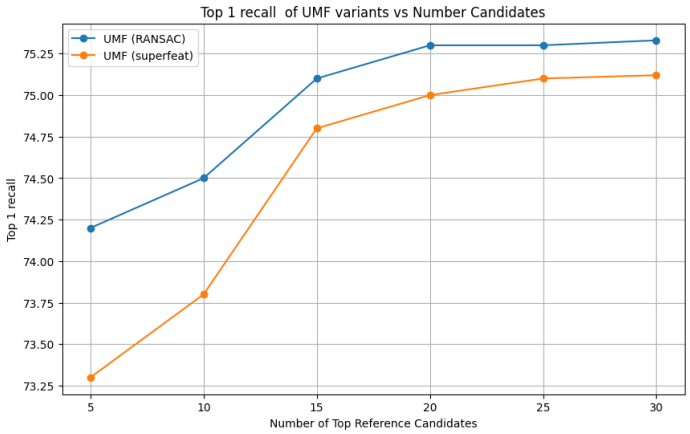


Figure 5.12: Comparison of top 1 recall of both UMF reranking approaches depending on the number of candidates used in the Etna dataset.

precision deteriorates rapidly as the Etna dataset proves challenging due to the lack of salient features and unstructured environment.

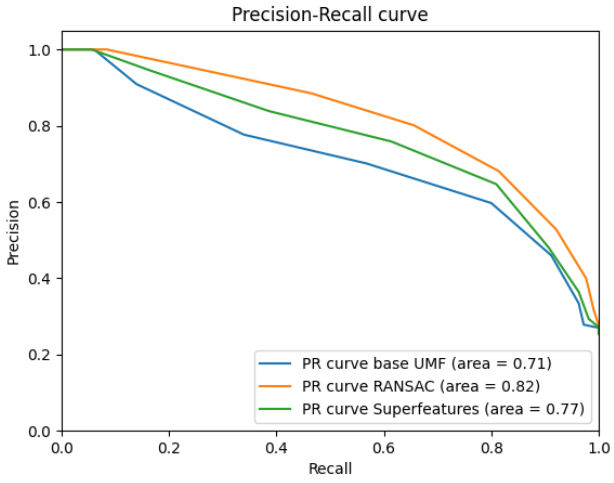


Figure 5.13: Precision-recall curves using the test set of Mt Etna, we compare the base UMF and both reranking variants.

5.5.3 Computational Complexity and Memory

In this section, we evaluate and compare the computational efficiency of our proposed UMF method and the baseline methods, focusing on two key metrics: feature extraction latency and memory usage. These metrics are crucial for understanding the practicality of these methods, especially in real-time or resource-constrained settings such as on-board processing in planetary rovers.

In Table 5.9, we present the latency and memory requirements for UMF and the baseline models. The latency is divided into two parts: feature extraction and matching. Feature extraction latency refers to the time taken to process an input and extract features, while matching latency denotes the time taken to compare these features against a database and find the best match.

The number of parameters in each model is indicative of its complexity, with a higher number of parameters generally leading to increased memory usage and longer training times.

Method	**Extraction(ms)**	**Matching(ms)**	(*#params*)
NetVLAD	32	9	24.495.904
DBoW2	3	2	–
PointNet++	26	9	671.392
MinkLoc++ (visual)	34	9	24.595.904
MinkLoc++ (LiDAR)	27	9	761.392
MinkLoc++ (fusion)	38	9	26.397.243
AdaFusion	54	9	29.358.531
UMF (super)	98	9 + 12	38.466.448
UMF (RANSAC)	98	9 + 71	39.385.229

Table 5.9: Comparison of feature extraction latency, matching time, model complexity (number of parameters), and memory requirements for different models, measured on an RTX 3090ti. the number of parameters represents the complexity of the model and we compute the inference times for a single sample using global embedding of 256 dim and/or local features. The times are measured taking the average of 10 runs. The matching is computed in a database containing 858 samples and selecting the top 20 candidates.

Table 5.9 presents a comparison of the UMF variants: the Superpoint-based approach (UMF (super)) and the RANSAC-based method (UMF (RANSAC)) in terms of feature extraction latency, matching time, and model complexity. As expected, the RANSAC method incurs a higher computational cost, but if this extra computational overhead is manageable, it can yield a more robust estimate.

It is also important to note that the reduced dimensionality of the fine features allows us to achieve a considerable speedup when performing geometrical verification, having the option to opt for more fine grained or coarse representations if necessary.

Despite optimizations, our method could still be computationally intensive for real-time applications on resource-constrained devices. Although we strive for a balance between accuracy and efficiency, the current implementation may not be suitable for all hardware configurations, will require a powerfull dedicated ML accelerator. Further work should be done in optimizing the model for such applications, via quantization and pruning.

5.6 Discussion and Limitations

Our proposed method, UMF, outperforms the baseline methods in terms of Recall@N on both the real-world and extreme planetary like environments. This improvement in performance can be attributed to the effective use of both local and global image features in our model. Moreover, the use of superfeatures enhances the model's ability to recognize places, further improving performance.

The utility of LiDAR data within the UMF framework is notable. In the case of the Etna dataset, despite its narrower field of view, LiDAR data significantly contributes to reducing uncertainty under extreme conditions and delivering authentic depth information, which is instrumental in determining accurate positive pairings. For the RobotCar dataset, LiDAR data supplies rich geometric information, substantially enhancing the place recognition prowess of our model.

The UMF model demonstrates superior performance. However, a crucial aspect that deserves further research is the scalability of our approach and how to best optimize the trade off betwwen acuraccy and computational efficiency. As alluded in previous sections 5.5.3, minimizing latency and memory usage are integral for practical deployment in real-world applications, especially in environments where computational resources are constrained, such as during extraterrestrial navigation.

Furthermore, the UMF model's reliance on both visual and LiDAR data can be a limitation in scenarios where one or both modalities may be unreliable or absent. Exploring methods to make the model more resilient to such situations, perhaps by incorporating other sensor data or employing more advanced fusion techniques, is another valuable avenue for future research.

Lastly, while the current model excels in handling aliased and low-texture environments, its performance in other challenging scenarios—such as highly dynamic environments or in the presence of severe occlusions—has yet to be assessed. Future work should aim to validate and possibly enhance the UMF model's ability to perform under such conditions.

Chapter 6

Conclusion

In this thesis, we have explored and proposed methods to address the problem of place recognition, particularly in unstructured environments characterized by aliasing and low-texture conditions, akin to the challenges presented on extraterrestrial surfaces.

In contrast to previous work that predominantly relies on global descriptor-based or keypoint-based approaches, we have taken an innovative path by considering a unified methodology, which fuses local and global features using transformers within a contrastive learning setting. Our proposed method, termed as Unified Multimodal Fusion (UMF), exhibits potential in overcoming the inherent limitations associated with traditional techniques.

The core element of our research involved a rigorous evaluation of several state-of-the-art place recognition techniques. This assessment helped us acquire a nuanced understanding of these methodologies, shedding light on their strengths and weaknesses when applied in demanding environments. Our analysis further enabled us to identify techniques that can potentially enhance the robustness and performance of multimodal methods (visual and LiDAR), especially under challenging conditions.

Our proposed UMF model was put to the test by comparing its performance against various baseline methodologies on two real-world datasets, Mt. Etna and RobotCar. These datasets, with their unique challenges and characteristics, served as suitable platforms for assessing the adaptability and robustness of our method.

The outcomes revealed that the UMF model, which optimally fuses local and global image features, outperformed the baselines in terms of Recall@N. This superior performance underscores the effectiveness of the UMF model in handling aliasing and low-texture environments, indicating its potential for place recognition tasks on planetary like environmets.

6.1 Future Work

Looking ahead, future research endeavours will primarily concentrate on addressing the identified limitations and extending the capabilities of our proposed method. Key areas of interest include:

- **Robustness Enhancement:** Further research will explore methods to further improve the model's robustness against aliasing and other extreme conditions. Such strategies may involve incorporating data augmentation techniques that diversify the training data, expand the synthetic dataset and enhance model generalizability by using some of the latest works on foundational models[81].

- **Multiple Scale Local Matching:** We plan to explore techniques to perform cross-matching of local patches at multiple scales between a query and reference image pairs. This multi-scale approach could potentially enhance the granularity of place recognition and increase the overall retrieval accuracy by producing a more comprehensive similarity score.

- **Alignment of Multimodal Local Features:** Future work will aim to ground and share representations between modalities, leveraging the complementary dynamics of different data types. Investigations will delve into methods for better alignment and fusion of multimodal local features.

- **Efficient Matching with Transformers:** We plan to explore the potential of leveraging the transformer cross-attention mechanisms, as seen in works such as [41], for forward pass matching. This approach has the potential to improve the distinctiveness and generalizability of learned features.

- **Application Expansion:** A fascinating avenue for exploration involves extending our framework to accommodate a broader range of applications, including object detection, semantic segmentation, and pose estimation, perhaps even employing a multi-task learning approach for simultaneous handling of these tasks. These investigations will not only assess the versatility of our model but also potentially unveil innovative techniques for multimodal fusion and place recognition. The ultimate goal will be to establish a unified framework capable of performing multiple tasks concurrently, leading to a more comprehensive and efficient perception system.

- **Model Deployment:** Future research will also consider the deployment of the model using smaller backbones, such as EfficientNetv2, with potential

knowledge distillation from larger models. Attention will also be paid to the memory requirements for storing local and global features, and the computational complexity of the matching step, with a view to refining these elements for practical deployment.

In summary, this work offers a new perspective and a robust solution to the problem of multimodal place recognition. Our proposed UMF model demonstrates promising results, paving the way for future research in this field, specifically for applications in extraterrestrial exploration and beyond.

Bibliography

[1] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pages 2938–2946, 2015.

[2] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2930–2937, 2013.

[3] Maxime Ferrera, Vincent Creuze, Julien Moras, and Pauline Trouvé-Peloux. Aqualoc: An underwater dataset for visual–inertial–pressure localization. *The International Journal of Robotics Research*, 38(14):1549–1559, 2019.

[4] Lukas Meyer, Michal Smíšek, Alejandro Fontan Villacampa, Laura Oliva Maza, Daniel Medina, Martin J Schuster, Florian Steidle, Mallikarjuna Vayugundla, Marcus G Müller, Bernhard Rebele, et al. The madmax data set for visual-inertial rover navigation on mars. *Journal of Field Robotics*, 38(6):833–853, 2021.

[5] Yang Zheng, Tolga Birdal, Fei Xia, Yanchao Yang, Yueqi Duan, and Leonidas J Guibas. 6d camera relocalization in visually ambiguous extreme environments. *arXiv preprint arXiv:2207.06333*, 2022.

[6] Riccardo Giubilato, Wolfgang Stürzl, Armin Wedler, and Rudolph Triebel. Challenges of slam in extremely unstructured environments: The dlr planetary stereo, solid-state lidar, inertial dataset. *IEEE Robotics and Automation Letters*, 7(4):8721–8728, 2022.

[7] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. pages 8922–8931, April 2021.

[8] S. Lowry, N. Sünderhauf, P. Newman, et al. Visual place recognition: A survey. *IEEE Transactions on Robotics*, 32(1):1–19, 2016.

[9] Chaoning Zhang, Chenshuang Zhang, Junha Song, John Seon Keun Yi, Kang Zhang, and In So Kweon. A survey on masked autoencoder for self-supervised learning in vision and beyond. July 2022.

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

[11] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[12] Marcus G. Müller, Maximilian Durner, Abel Gawel, Wolfgang Stürzl, Rudolph Triebel, and Roland Siegwart. A Photorealistic Terrain Simulation Pipeline for Unstructured Outdoor Environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021.

[13] Jacek Komorowski, Monika Wysoczańska, and Tomasz Trzcinski. MinkLoc++: Lidar and monocular image fusion for place recognition. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, July 2021.

[14] G. Bresson et al. Simultaneous localization and mapping: A survey of current trends in autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 2(3):194–220, 2017.

[15] M. Burki et al. Vizard: Reliable visual localization for autonomous vehicles in urban outdoor environments. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 1124–1130, 2019.

[16] S. Mascaro et al. Towards automating construction tasks: Large-scale object mapping, segmentation, and manipulation. *Journal of Field Robotics*, 38(5):684–699, 2021.

[17] X. Shu et al. Slam in the field: An evaluation of monocular mapping and localization on challenging dynamic agricultural environment. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1761–1771, 2021.

[18] R. Oliveira et al. Advances in agriculture robotics: A state-of-the-art review and challenges ahead. *Robotics*, 10(2):52, 2021.

[19] I. Sharafutdinov et al. Comparison of modern open-source visual slam approaches. *arXiv preprint arXiv:2108.01654*, 2021.

[20] T. Shan and B. Englot. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4758–4765, 2018.

[21] R. Giubilato et al. Gpgm-slam: a robust slam system for unstructured planetary environments with gaussian process gradient maps. *CoRR*, abs/2109.06596, 2021.

[22] C. Le Gentil et al. Gaussian process gradient maps for loop-closure detection in unstructured planetary environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1895–1902, 2020.

[23] N. Gelfand, L. Ikemoto, S. Rusinkiewicz, and M. Levoy. Geometrically stable sampling for the icp algorithm. In *Fourth International Conference on 3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings.*, pages 260–267, 2003.

[24] N. Gelfand et al. Geometrically stable sampling for the icp algorithm. In *International Conference on 3-D Digital Imaging and Modeling (3DIM)*, pages 260–267, 2003.

[25] A. Geiger et al. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[26] J. Choi et al. Kaist multi-spectral day/night data set for autonomous and assisted driving. *IEEE Transactions on Intelligent Transportation Systems*, 19(3):934–948, 2018.

[27] S. Wenzel et al. 4seasons: A cross-season dataset for multi-weather slam in autonomous driving. In *DAGM German Conference on Pattern Recognition*, pages 404–417. Springer, 2020.

[28] J. Sturm et al. A benchmark for the evaluation of rgb-d slam systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 573–580, 2012.

[29] R. Schubert et al. The tum vi benchmark for evaluating visual-inertial odometry. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1680–1687, 2018.

[30] T. Schops et al. Bad slam: Bundle adjusted direct rgb-d slam. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[31] A. Handa et al. A benchmark for rgb-d visual odometry, 3d reconstruction and slam. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1524–1531, 2014.

[32] Opencv. *https://opencv.org/*, (date accessed 11-3-2023).

[33] Facebook's AI research lab. Pytorch. *https://pytorch.org/*, (date accessed 11-3-2023).

[34] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José M M. Montiel, and Juan D. Tardós. ORB-SLAM3: An accurate Open-Source library for visual, Visual–Inertial, and multimap SLAM. *IEEE Trans. Rob.*, pages 1–17, 2021.

[35] Arandjelovic, Gronat, Torii, and others. NetVLAD: CNN architecture for weakly supervised place recognition. *Proc. Estonian Acad. Sci. Biol. Ecol.*

[36] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017.

[37] Yingying Zhu, Jiong Wang, Lingxi Xie, and Liang Zheng. Attention-based pyramid aggregation network for visual place recognition, 2018.

[38] Junyi Ma, Jun Zhang, Jintao Xu, Rui Ai, Weihao Gu, and Xieyuanli Chen. OverlapTransformer: An efficient and Yaw-Angle-Invariant transformer network for LiDAR-Based place recognition. *IEEE Robotics and Automation Letters*, 7(3):6958–6965, July 2022.

[39] Jacek Komorowski. Improving point cloud based place recognition with ranking-based loss and large batch training. March 2022.

[40] Stephen Hausler, Sourav Garg, Ming Xu, Michael Milford, and Tobias Fischer. Patch-NetVLAD: Multi-Scale fusion of Locally-Global descriptors for place recognition, 2021.

[41] Hao Zhang, Xin Chen, Heming Jing, Yingbin Zheng, Yuan Wu, and Cheng Jin. ETR: An efficient transformer for re-ranking in visual place recognition. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 5665–5674. IEEE, January 2023.

[42] Philippe Weinzaepfel, Thomas Lucas, Diane Larlus, and Yannis Kalantidis. Learning Super-Features for image retrieval. January 2022.

[43] Wenzheng Song, Ran Yan, Boshu Lei, and Takayuki Okatani. SuperGF: Unifying local and global features for visual localization. December 2022.

[44] Haowen Lai, Peng Yin, and Sebastian Scherer. AdaFusion: Visual-LiDAR fusion with adaptive weights for place recognition. November 2021.

[45] Meng-Hao Guo, Cheng-Ze Lu, Zheng-Ning Liu, Ming-Ming Cheng, and Shi-Min Hu. Visual attention network, 2022.

[46] Tian-Xing Xu, Yuan-Chen Guo, Yu-Kun Lai, and Song-Hai Zhang. TransLoc3D : Point cloud based large-scale place recognition using adaptive receptive fields. May 2021.

[47] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning cnn image retrieval with no human annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(7):1655–1668, 2019.

[48] Bai, Hu, Zhu, Huang, Chen, and others. Transfusion: Robust lidar-camera fusion for 3d object detection with transformers. *Proc. Estonian Acad. Sci. Biol. Ecol.*

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[50] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[51] Ruotong Wang, Yanqing Shen, Weiliang Zuo, Sanping Zhou, and Nanning Zheng. TransVPR: Transformer-based place recognition with multi-level attention aggregation. January 2022.

[52] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International Conference on Computer Vision*, pages 2564–2571, 2011.

[53] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *CVPR*, 2020.

[54] Eric Brachmann and Carsten Rother. Learning less is more-6d camera localization via 3d surface regression. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4654–4662, 2018.

[55] Eric Brachmann and Carsten Rother. Visual camera re-localization from rgb and rgb-d images using dsac. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):5847–5865, 2021.

[56] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020.

[57] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning, 2020.

[58] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[59] Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. ConvNeXt v2: Co-designing and scaling ConvNets with masked autoencoders. January 2023.

[60] Keyu Tian, Yi Jiang, Qishuai Diao, Chen Lin, Liwei Wang, and Zehuan Yuan. Designing BERT for convolutional networks: Sparse and hierarchical masked modeling. January 2023.

[61] Xinli Xu. Dawei Zhao. Liang Xiao. Yiming Nie. Chen Min and Bin Dai. Voxel-mae: Masked autoencoders for pre-training large-scale point clouds. *arXiv e-prints*, 2022.

[62] Giseop Kim and Ayoung Kim. Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4802–4809. IEEE, 2018.

[63] David Arthur and Sergei Vassilvitskii. K-means++ the advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035, 2007.

[64] Sivic and Zisserman. Video google: a text retrieval approach to object matching in videos. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 1470–1477 vol.2, 2003.

[65] Alexander Hermans*, Lucas Beyer*, and Bastian Leibe. In Defense of the Triplet Loss for Person Re-Identification. *arXiv preprint arXiv:1703.07737*, 2017.

[66] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.

[67] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning cnn image retrieval with no human annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(7):1655–1668, 2019.

[68] Marvin Chancán, Luis Hernandez-Nunez, Ajay Narendra, Andrew Barron, and Michael Milford. A compact neural architecture for visual place recognition, 10 2019.

[69] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3146–3154, 2019.

[70] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V Le. Attention augmented convolutional networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3286–3295, 2019.

[71] Zetao Chen, Lingqiao Liu, Inkyu Sa, Zongyuan Ge, and Margarita Chli. Learning context flexible attention model for long-term visual place recognition. *IEEE Robotics and Automation Letters*, 3(4):4015–4022, 2018.

[72] Bingyi Cao, Andre Araujo, and Jack Sim. Unifying deep local and global features for image search. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, pages 726–743. Springer, 2020.

[73] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 213–229. Springer, 2020.

[74] Dan Barnes, Matthew Gadd, Paul Murcutt, Paul Newman, and Ingmar Posner. The oxford radar robotcar dataset: A radar extension to the oxford robotcar dataset. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6433–6438. IEEE, 2020.

[75] Tong Qin, Peiliang Li, and Shaojie Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, 2018.

[76] Tong Qin and Shaojie Shen. Online temporal calibration for monocular visual-inertial systems. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3662–3669. IEEE, 2018.

[77] Tong Qin, Shaozu Cao, Jie Pan, and Shaojie Shen. A general optimization-based framework for global pose estimation with multiple sensors, 2019.

[78] Simon Lacroix, Andrea De Maio, Quentin Labourey, Ellon Paiva Mendes, Pierre Narvor, Vincent Bissonette, Clément Bazerque, Fabrice Souvannavong, Raphaël Viards, and Martin Azkarate. The erfoud dataset: a comprehensive multi-camera and lidar data collection for planetary exploration. In *15th Symposium on Advanced Space Technologies in Robotics and Automation*, 2020.

[79] Maximilian Denninger, Dominik Winkelbauer, Martin Sundermeyer, Wout Boerdijk, Markus Knauer, Klaus H. Strobl, Matthias Humt, and Rudolph Triebel. Blenderproc2: A procedural pipeline for photorealistic rendering. *Journal of Open Source Software*, 8(82):4901, 2023.

[80] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[81] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023.

[82] Robot operating system (ros). *https://www.ros.org/*, (date accessed 11-3-2023).

[83] CoppeliaRobotics. Coppeliasim. *https://www.coppeliarobotics.com/*, (date accessed 11-3-2023).

[84] Li He, Xiaolong Wang, and Hong Zhang. M2dp: A novel 3d point cloud descriptor and its application in loop closure detection. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 231–237. IEEE, 2016.

[85] Lin Li, Xin Kong, Xiangrui Zhao, Tianxin Huang, Wanlong Li, Feng Wen, Hongbo Zhang, and Yong Liu. RINet: Efficient 3D Lidar-Based place recognition using rotation invariant neural network. *IEEE Robotics and Automation Letters*, 7(2):4321–4328, April 2022.

[86] Paul-Edouard Sarlin, Ajaykumar Unagar, Måns Larsson, Hugo Germain, Carl Toft, Viktor Larsson, Marc Pollefeys, Vincent Lepetit, Lars Hammarstrand, Fredrik Kahl, and Torsten Sattler. Back to the feature: Learning robust camera localization from pixels to pose. pages 3247–3257, March 2021.

[87] Qunjie Zhou, Sergio Agostinho, Aljosa Osep, and Laura Leal-Taixe. Is geometry enough for matching in visual localization? March 2022.

[88] Grégoire Mialon, Dexiong Chen, Alexandre d'Aspremont, and Julien Mairal. A trainable optimal transport embedding for feature aggregation and its relationship to attention. *arXiv preprint arXiv:2006.12065*, 2020.

[89] Songtao Liu, Zeming Li, and Jian Sun. Self-emd: Self-supervised object detection without imagenet. *arXiv preprint arXiv:2011.13677*, 2020.

[90] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[91] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*, pages 778–792. Springer, 2010.

[92] Michael Calonder, Vincent Lepetit, and Pascal Fua. Keypoint signatures for fast learning and recognition. In *Computer Vision–ECCV 2008: 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part I 10*, pages 58–71. Springer, 2008.

[93] Dominik Winkelbauer, Maximilian Denninger, and Rudolph Triebel. Learning to localize in new environments from synthetic training data. In *2021 IEEE*

International Conference on Robotics and Automation (ICRA), pages 5840–5846, May 2021.

[94] Noha Radwan, Abhinav Valada, and Wolfram Burgard. VLocNet++: Deep multitask learning for semantic visual localization and odometry. *IEEE Robotics and Automation Letters*, 3(4):4407–4414, October 2018.

[95] Eric Brachmann and Carsten Rother. Visual camera Re-Localization from RGB and RGB-D images using DSAC. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(9):5847–5865, September 2022.

[96] Cem Akkus, Luyang Chu, Vladana Djakovic, Steffen Jauch-Walser, Philipp Koch, Giacomo Loss, Christopher Marquardt, Marco Moldovan, Nadja Sauter, Maximilian Schneider, Rickmer Schulte, Karol Urbanczyk, Jann Goschenhofer, Christian Heumann, Rasmus Hvingelby, Daniel Schalk, and Matthias Aßenmacher. Multimodal deep learning. January 2023.

[97] Tao Xie, Kun Dai, Ke Wang, Ruifeng Li, and Lijun Zhao. DeepMatcher: A deep transformer-based network for robust and accurate local feature matching. January 2023.

[98] Montiel J. M. M. Mur-Artal, Raúl and Juan D. Tardós. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[99] Hao Yu, Zheng Qin, Ji Hou, Mahdi Saleh, Dongsheng Li, Benjamin Busam, and Slobodan Ilic. Rotation-Invariant transformer for point cloud matching. March 2023.

[100] Yash Bhalgat, Joao F Henriques, and Andrew Zisserman. A light touch approach to teaching transformers multi-view geometry. November 2022.

[101] Wang Yifan, Carl Doersch, Relja Arandjelović, João Carreira, and Andrew Zisserman. Input-level inductive biases for 3D reconstruction. pages 6176–6186, December 2021.

[102] Jose M Facil, Benjamin Ummenhofer, Huizhong Zhou, Luis Montesano, Thomas Brox, and Javier Civera. CAM-convs: Camera-aware multi-scale convolutions for single-view depth. pages 11826–11835, April 2019.

[103] T. Nam, J. Shim, and Y. Cho. A 2.5 d map-based mobile robot localization via cooperation of aerial and ground robots. *Sensors*, 17(12):2730, 2017.

[104] R. Arandjelovic et al. Netvlad: Cnn architecture for weakly supervised place recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5297–5307, 2016.

[105] Z. Chen, A. Jacobson, N. Sünderhauf, B. Upcroft, L. Liu, C. Shen, I. Reid, and M. Milford. Deep learning features at scale for visual place recognition. In *IEEE International Conference on Robotics and Automation*, pages 3223–3230, 2017.

[106] Z. Chen, F. Maffra, I. Sa, M. Chli, et al. Only look once, mining distinctive landmarks from convnet for visual place recognition. In *IEEE International Conference on Intelligent Robots and Systems*, pages 9–16, 2017.

[107] Z. Chen, L. Liu, I. Sa, Z. Ge, and M. Chli. Learning context flexible attention model for long-term visual place recognition. *IEEE Robotics and Automation Letters*, 3(4):4015–4022, 2018.

[108] M. Zaffar, A. Khaliq, S. Ehsan, M. Milford, K. Alexis, and K. McDonald-Maier. Are state-of-the-art visual place recognition techniques any good for aerial robotics? *arXiv preprint arXiv:1904.07967*, 2019.

[109] A. Torii, R. Arandjelovic, J. Sivic, M. Okutomi, and T. Pajdla. 24/7 place recognition by view synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1808–1817, 2015.

[110] A. Khaliq, S. Ehsan, Z. Chen, M. Milford, and K. McDonald-Maier. A holistic visual place recognition approach using lightweight cnns for significant viewpoint and appearance changes. *IEEE Transactions on Robotics*, pages 1–9, 2019.

[111] G. Tolias, R. Sicre, and H. Jégou. Particular object retrieval with integral max-pooling of cnn activations. In *Proc. International Conference on Learning Representations*, 2016.

[112] R. Bishop. *Intelligent vehicle technology and trends*. 2005.

[113] S. Chen, L. Huang, J. Bai, H. Jiang, and L. Chang. Multi-sensor information fusion algorithm with central level architecture for intelligent vehicle environmental perception system. Technical report, SAE Technical Paper, 2016.

[114] G. Milford and G. Wyeth. Persistent navigation and mapping using a biologically inspired slam system. *The International Journal of Robotics Research*, 29(9):1131–1153, 2010.

[115] J. Biswas and M. M. Veloso. Localization and navigation of the cobots over long-term deployments. *The International Journal of Robotics Research*, 32(14):1679–1694, 2013.

[116] S. Thrun et al. *Robotic mapping: A survey*, volume 1. 2002.

[117] Tong Qin, Jie Pan, Shaozu Cao, and Shaojie Shen. A general optimization-based framework for local odometry estimation with multiple sensors, 2019.

[118] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[119] D. Nister, O. Naroditsky, and J. Bergen. Visual odometry. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pages I–I, 2004.

[120] P. Dellenbach et al. What's in my lidar odometry toolbox? *arXiv preprint arXiv:2103.09708*, 2021.

[121] W. Maddern et al. 1 year, 1000 km: The oxford robotcar dataset. *The International Journal of Robotics Research*, 36(1):3–15, 2017.

[122] J. Wang et al. Tartanair: A dataset to push the limits of visual slam. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4909–4916, 2020.

[123] P. Furgale et al. The devon island rover navigation dataset. *The International Journal of Robotics Research*, 2020.