On Landmarks for Introducing 3D SLAM Structure to VPR

by

Matthew Bradley

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2024

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Simultaneous Localization and Mapping (SLAM) is a critical foundation to a wide variety of robotic applications. Visual SLAM systems rely on Visual Place Recognition (VPR) for map maintenance and loop-closing so their quality suffers when VPR performance is impacted. In most VPR systems images are described compactly and stored for later comparison, with matches indicating that a scene has been seen before and has been revisited. Changes in illumination are a common difficulty for VPR image descriptors based on vocabularies of local features. Global descriptors which incorporate high-level structure are more robust to illumination, but are often sensitive to changes in viewpoint. There is an overall focus in VPR on describing single images despite the fact that SLAM systems recover 3D structure from the environment, and that this structure is both illumination invariant and remains the same regardless of vantage point. Work leveraging SLAM-recovered structure in the form of 3D points, in conjunction with LiDAR scan descriptors, has demonstrated superior VPR performance under harsh illumination compared with SoTA visual vocabulary descriptors. However, performance in general is not as high. A significant observed limitation was difficulty matching pseudo-LiDAR scans with significantly differing sub-regions. This is due to an assumption by the LiDAR descriptors used, that the entire volume of two corresponding scans should match. This does not fit well with the inherent sparsity of accumulated pointclouds from traversal by visual SLAM, due to differences in route, incomplete coverage, and the inherent sparsity of SLAM feature tracking in general. What is needed is an approach based on matching sub-regions which are common between pseudo-scans, in other words an approach performing place recognition based on landmarks. Here we explore generation of landmarks from accumulated SLAM structure through various clustering-based techniques, as well as the application of SoTA Grassmannian Graph-based association to match them. We present the challenges and successes of this approach to introducing 3D structure into VPR and propose various avenues of exploration to address the challenges faced. One of the foremost challenges is that pointclouds derived from SLAM are very sparse and uneven, making reliable and repeatable clustering difficult to achieve. We make significant improvement in landmark quality by using semantic labeling to provide better separation before clustering. While this has a noticeable impact on the number of outlier landmarks, we also find that there is an extreme sensitivity to outliers in the association method used. This sensitivity persists across data sets and seems inherent to this method of association. This precludes effective place recognition at this time, however in future work we expect this will be alleviated through the use of landmark descriptors for more effective outlier rejection. Descriptors can also provide putative associations which can be beneficial to landmark matching. We also propose various other enhancements to help improve landmark generation and association of landmarks for place recognition. It is our firm expectation that incorporation of 3D structure from SLAM systems into underlying VPR will be mutually beneficial, with VPR systems gaining additional descriptive capability which is fully invariant to illumination but more stable than viewpoint-sensitive 2D image structure.

Acknowledgements

I would like to thank the Natural Sciences and Engineering Research Council of Canada and the Ontario Graduate Scholarship Program for supporting my research.

Dedication

This is dedicated to my supervisor Dr. John Zelek and colleague Dr. Georges Younes for their mentorship during my studies.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Place recognition is the task of detecting when you have returned to a previously seen place based on repeated observation of the same environmental features. It is accomplished by taking ongoing measurements of the environment and comparing them against measurements made in the past. Matches indicate a revisit to a previously seen location, while representations of novel environments are stored for future comparison. The task of place recognition is critical to all self-contained navigation systems and is a key component of Simultaneous Localization and Mapping (SLAM). In this work we seek to improve upon solutions to the most pressing issues in visual place recognition (VPR) using structural information recovered by SLAM, such that widely used visual SLAM systems can be more accurate and reliable.

SLAM builds on simple sensor odometry which tracks movement continuously, constructing a map of the environment as it is explored. This requires the ability to detect revisits to previous places, both to simplify the overall map and avoid duplicates and to eliminate error accumulated over long periods of exploration. Error in the position estimate introduced during odometry (e.g. when performing bundle adjustment as in visual odometry) means that the same part of the environment will be estimated as having two different positions from one visit to the next. By optimizing the map to bring these same places together the error accumulated between visits can be greatly reduced, improving the accuracy of the overall map. Place recognition combined with this map can also allow a SLAM system to re-determine its position following interruptions, the core of the so-called "kidnapped robot" problem. Visual SLAM's ongoing and accurate estimate of camera position over a potentially large area (even when without a measure of scale) makes it core to many mobile robotic applications, from indoor robotic agents to self driving vehicles. Visual SLAM's reliance on only imagery from inexpensive cameras makes it possible to find a robotic system's position cheaply and in a wide variety of environments where aids are not present, including GPS-denied areas (e.g. indoors). For this reason, the success of VPR underpinning visual SLAM has an indirect impact on the flexibility and capability of a large number of robotic applications.

Visual place recognition is often formulated as an image retrieval problem where each "place" is equivalent to a single representative image. In this way the task becomes choosing which image in a database, if any, is most similar to the current view. To make this searching process efficient, each new image is converted into a compressed description vector, where similarity between these descriptors becomes a proxy for physical closeness. Processing into a separate description vector also provides an intermediate representation so that images can be compared more by their content than by similarity of their pixel values. All of this makes production of good image descriptors of paramount importance in VPR, with virtually all work proposing an enhancement or new kind of descriptor. The descriptors that have been proposed fall largely into two categories: those which extract many small regions (local features) and sort them into a visual vocabulary; and those global descriptors that seek to process whole images looking for large-scale structures and patterns. The former ignore the precise position of each local feature in an image and so are resistant to variations in viewpoint when recognizing a scene, while the latter have access to high-level features that insulate them from changes in illumination or even changes in weather condition and seasonal changes.

Parallel to place recognition and SLAM using cameras are similar systems which use LiDAR scanners. For this different sensor modality the same overall style of SLAM system also exists, including a reliance on place recognition. Where visual SLAM systems recover some sparse structural information from the environment (which is generally not used for place recognition, in favor of visual appearance), LiDAR scans are inherently structural and typically provide very dense coverage of the shape of all nearby surfaces. For this reason LiDAR systems employ structural place recognition, where instead of images the structure of each LiDAR scan is summarized into a descriptor. There is generally a preference here for handcrafted descriptors (as opposed to neural networks), since the extremely large number of points in each scan makes algorithmic efficiency important. LiDAR is generally invariant to illumination conditions and focus has been placed on allowing scans with some amount of misalignment to be matched.

1.1 Problem Description

The two main approaches to VPR, vocabularies of local features and whole-image global descriptors, both have strengths but neither is universally applicable. There are outstanding challenges which neither approach, local or global description, has been able to address simultaneously. Vocabularies of local features have significantly worse recall in the presence of strong illumination changes, even for some of the most robust methods available [57]. The small regions around each local feature can only make available relatively low-level information compared to the overall image, and so are not stable in the presence of strong appearance changes [51]. Meanwhile, global descriptors are able to incorporate large-scale 2D structure from across the entire image, and this provides superior robustness to illumination and other challenging appearance changes. This comes at the cost of increased sensitivity to viewpoint, where places seen from a slightly different viewpoint may go undetected. Global image descriptors typically assume that camera viewpoint remains similar [49], due to the role of viewpoint in how large-scale 3D scenes become projected into 2D structures on the image plane. This is in contrast to local methods' ignorance of geometric structure outside of small regions. [49] Work in VPR has sought to make local feature descriptors more robust to illumination changes using neural-based region descriptors [3] and there have been attempts to compensate for changes in viewpoint with global descriptors [50] [54]. However, there has not emerged a description approach that has been able to remedy both issues adequately at the same time, despite expressed need to escape the binary of local and global descriptors [24]. This lack of a universally-capable solution impacts VPR's versatility, and by extension the versatility of SLAM systems which rely on VPR for map maintenance and recovery from interruptions.

The robustness of global descriptors to illumination and other appearance changes is generally credited to their use of spatial information, however this is the same 2D geometric structure that leads to their viewpoint sensitivity. The 3D structure of the environment itself, as recovered by LiDAR is invariant to the illumination of a scene and the same regardless of where the observer is located. Despite this there has been very little exploration of using lessons from 3D LiDAR place recognition in VPR, despite the recovery of spatial information by SLAM systems. Some VPR systems have even discarded recovered 3D depth in favor of 2D arrangement [35].

One of the few works attempting to adapt 3D LiDAR techniques to SLAM-recovered 3D structure, the work proposed by [57], was able to demonstrate significantly better performance under severe illumination conditions as compared to SoTA visual vocabulary of local features [3]. They achieved consistent place recognition performance even in challenging conditions and without the use of global image descriptors, however they attained noticeably lesser performance in general. An issue they attribute this to is the intolerance of they global LiDAR scan descriptors they use to scans with slightly different coverage [57]. Due to being captured using a camera with limited field of view and other constraints, the imitation LiDAR scans they produce are sparser and can have missing regions. The LiDAR descriptors they use assume complete scan coverage, with a one-to-one relation between parts of the description vector and volumes inside the scan radius. What is needed is a method of comparison that can better tolerate sparsity and missing scan regions.

1.2 Objectives and Motivation

Our objective is to explore new ways of using the illumination and viewpoint-invariant 3D structure recovered by visual SLAM to perform place recognition. Rather than apply unmodified LiDAR global descriptors to pointclouds recovered from visual data, we have pursued a landmark-based approach to perform place recognition by associating sub-regions of greater sparsity in the recovered clouds.

We believe the application of 3D spatial information to VPR is critically under-explored, especially given its invariance to the effects that are core stated problems for the field. The underlying 3D structure of the environment is largely invariant to how illumination is cast onto it, and the physical features of the environment remain the same regardless of where the observer views them from. This promises to simultaneously address the two most prominent challenges in VPR. That SLAM systems also already recover spatial structure as part of their operation would seem too opportune to not explore this direction. Advocated for by [79] are solutions which combine modalities, with a combination of both LiDAR and cameras for example. However, our goal here is to empower robotic and other applications relying on SLAM through the improvement of place recognition with inexpensive camera sensors and so the ultimate goal is to improve VPR using the structure that can be recovered indirectly by visual systems.

The limited exploration by [57] to apply 3D structure to VPR focused on applying existing LiDAR global descriptors intended for a qualitatively different sensor domain. The authors encountered challenges with inconsistent coverage due to cameras' field of view, combined with the one-to-one mapping of traditional LiDAR descriptors to the totality of the area encompassed by a LiDAR scan (not typically an issue due to LiDAR having better coverage). This results in apparent mismatch of otherwise matching places. The resulting imitation-scans also remain very sparse with few points, clumped around windows, utility boxes, edges, and other localized structures. Compared to the very uniform and high-coverage scans produced by LiDAR scanning, 3D points are recovered by SLAM opportunistically on local features and areas of strong gradient. These two considerations combined have motivated our landmark-based approach, where individual, more-densely-populated sub-regions (corresponding to underlying portions of the environment) are associated to drive place recognition. The variation in sparsity makes separation of different structures difficult in some cases, which we improve on through the use of semantic labeling for better segmentation.

1.3 Contribution

The use of 3D structure in VPR promises to remediate two longstanding challenges which have proven hard to address simultaneously. This area is not well explored and previous work to do so [57] encountered challenges resulting from the use of unmodified LiDAR structural place recognition descriptors. Our contribution is the exploration of a new approach based on landmarks to address the challenges they encountered with the application of physical structure to VPR. We present here our findings from this exploration including methods to overcome various challenges and recommendations for future work to overcome those which remain.

Chapter 2

Background

2.1 Approaches to Visual Place Recognition

In VPR, the problem to be solved is that of detecting when a camera has returned to a prior-visited location. This is critically important to SLAM applications as it allows for the closing of loops and overall improvement of the system's map, improving accuracy. Most current approaches formulate VPR as an image retrieval problem [84] [24] [51], which simplifies localization of the camera to the task of image-to-image matching. A diagram is given in figure 2.2. Each new image is compared against all those which have previously been collected, and a sufficiently strong match indicates the new and old images were taken in the same place and a revisit has occurred. A side effect is that in many cases the database storing the accumulated images and their descriptions grows at roughly the rate that new area is explored. In the case of ORB-SLAM3 [10], only SLAM keyframes are kept as possible matching candidates.

To facilitate this process of image-to-image matching the use of image descriptors is essential, converting each new image to a description vector which can be efficiently stored and compared. These image descriptors should yield similar results for images of the same scenery and different results for different scenery, yet also be robust to various confounding factors like changes in illumination or viewpoint, or even minor changes to the scene and its appearance that occur over time [24]. The conversion to a compact description vector means that images can be matched based on measures of vector similarity which is much faster than comparing full images [84]. At worst it involves comparison with each stored description, although database techniques like hashing can improve search time significantly [84].

Two main kinds of image descriptor for place recognition have emerged, those which aggregate the information of many small local features through visual vocabulary, and global descriptors which seek to holistically describe each image at the high-level (see figure 2.1). Local features are detected in large numbers at points in an image, and a visual vocabulary groups them such that their aggregate properties can be measured. Their relative positions are generally discarded, and so changes in viewpoint which affect their arrangement have less effect. Global image descriptors encode the large-scale 2D structure of an image, discarding local details which are affected by changes in illumination, but this also makes them reliant on consistent viewpoint. Local and global methods are frequently seen as occupying a binary [24] based on this use of 2D structure, where leveraging it grants improved invariance to illumination changes and other appearance affects, but confers viewpoint sensitivity. That said, some methods based on local features have made use of 2D structure as a check to enhance their reliability [78] [35] [26] [76]. We have found use of 3D structure relatively rare, most likely due to the focus on matching single images which carry little 3D information.

2.1.1 Local Feature Vocabularies

The local features used in visual vocabulary come from small distinctive points found in an image. The classic example of such features is corner detection, where many points are found that correspond to corners in an image. By themselves they are not an image description, and manually comparing the features between two images can be costly in terms of time when many are detected. As such, visual vocabularies provide a useful summarization of the local features in an image by aggregating local features into a set of categories or visual ”words” which can be examined in bulk. This leads to an overall image description which can be used in image retrieval for the purpose of VPR.

One of the most widely known local features is SIFT [48], a handcrafted feature detector and descriptor which offers rotation and scale invariance. Other examples used in visual-vocabulary VPR systems include SURF [61] and ORB [68], the latter of which lends its name to the popular ORB-SLAM Simultaneous Localization and Mapping system(s) [10]. ORB-SLAM also demonstrates a useful synergy available to local features, where the same local features used for visual odometry are also used for VPR. This means that two subsystems of the SLAM can share features and feature extraction needs to only be performed once. SURF brought direct efficiency improvements through the use of integral images, while ORB employs binarization for much faster computation of the feature and its descriptor. The use of vocabulary trees [63] can significantly improve the speed of assigning each feature to a vocabulary word by splitting the clustering task into hierarchical levels where each step of clustering takes significantly less time.

While local feature detectors often come with their own or can be paired with local feature descriptors to describe the pixels around where a feature is located in an image, they are not on their own an image description. The number of features detected in each image will vary and may not be reliably re-detected or described the same each time. To construct an overall image description which is more reliable as an aggregate description of an image's local features, the common solution is visual vocabulary. This is where the local features are clustered into groups with similar descriptors, and the statistics of these groups is used to construct an image descriptor. Due to having similar descriptors, the local regions around the features in each group tend to share a similar visual appearances. Visual Vocabulary was pioneered by Bag of Visual Words [71] which used simple histogram descriptors recording the proportion of each group. Building on the approach of BoVW [71], VLAD [32] measures how the descriptors in each group deviate from its mean. This has improved recognition performance over BoVW, as it specifically encodes what makes the local features of a given image unique, and thus that is unique about the image itself [4]. NetVLAD [3] takes this a step further by using a neural network to describe the region around each local feature while also making the entire system end-to-end trainable. The use of neural networks in this way for local feature description is aimed at making best use of the small region around each local feature and obtaining the best possible illumination invariance. NetVLAD's high performance among visual vocabulary methods has resulted in it continually being widely cited in many comparisons [57] [81] [67].

With algorithmic techniques like vocabulary trees [63] to accelerate vocabulary clustering, local features are sufficiently fast for use in real-time SLAM systems [10]. As visual vocabulary techniques measure local features in aggregate and without regard for where in an image they occur, local feature-based descriptors maintain a high degree of viewpoint invariance [49]. It does not matter where they appear, only that they are counted, so local features are free to move with changes in viewpoint. However, incorporating some structural information has been shown to improve invariance to appearance changes (despite reduced viewpoint invariance) [49], and the small regions used to describe each feature are limited in this way. They only have relatively low-level information that is strongly dependant on pixel intensities, and so are unstable in the presence of strong appearance changes [51].

2.1.2 Global Feature Descriptors

As an alternative to describing an image using many small features, global descriptors seek to digest an image in its entirety and extract useful information about its large-scale structure and a qualitative description of its content. By looking at the image from a high-level perspective 2D structural information can be extracted that isn't available to local descriptors, making global descriptors generally more robust to lighting and other appearance changes that disrupt local image detail [49].

An early global descriptor to be used in VPR was GIST [65] which is based on applying Gabor filters to each image. The image descriptions produced by GIST were used for place recognition by [60], and later by SeqSLAM [55]. The former used GIST [65] descriptors on portions of panoramas to achieve place recognition. SeqSLAM is a place recognition technique (it does not propose a SLAM system) which uses GIST [65] as a lightweight global descriptor to compare images between portions of a camera trajectory. Neural networks were later introduced as a method of global image description for place recognition by [12] and have been the dominant means of global image description [84]. One exception is CoHOG [80], a relatively recent handcrafted global image descriptor which seeks to emulate the advantages of CNNs using HOG [16] descriptors. This approach has the advantage of significantly faster inference compared to a neural network while also not requiring training [80].

One of the first methods to demonstrate success at using a CNN for place recognition was [47]. CNNs are a widespread type of network for processing images, organized as a hierarchical series of learned filter kernels [47]. trained an existing CNN architecture to embed each image in a low-dimensional space. At inference time each image is converted to a 128-element descriptor which can be stored for later retrieval and compared to gauge similarity of the place depicted in the corresponding image. This was followed by the development of AMOSNet and HybridNet [12] which demonstrated the utility of fine-tuning an existing network rather than training from scratch for the task of VPR. AMOSNet's weights were trained from scratch for place recognition, while HybridNet's weights were initialized using weights from an object recognition network before receiving further place recognition training. This transfer learning gave HybridNet the benefit of object recognition features as a starting point for extracting information about a scene, yielding better quality place descriptions during inference, after training. The utility of attention mechanisms to direct attention of the network towards salient regions (large distinctive structures) of each image was demonstrated by [13]. Here attention mechanisms guide the extraction of features from salient regions at various scales which are then fused together. A final attention layer then produces a combination of the resulting features weighted by their importance, which is the basis of the final image description. A large scale CNN training dataset is proposed by [86] as well as a series of CNNs trained on it for place recognition. They are trained to treat place recognition as a classification problem, producing a final answer as to the current place rather than a descriptor. This does require all the places to be known at training-time. R-MAC [73] style features are used by [85] to generate a description for each image. At run-time they augment this by building a connected graph from sequences of images and their descriptions. This provides an additional check as nearby places along the route of the initial traversal are incorporated, and must also match those seen during a revisit. R-MAC [73] itself is a method of extracting and combining information from difference regions of an image at different scales within a neural network. Strong activations in the feature maps of higher CNN layers lead to the selection of features from corresponding regions in lower layers, which are then summed together before becoming the final image description. In this way it functions similarly to an attention mechanism, except that strong activations are the only attention criteria.

In addition to CNNs, encoder networks have also been used to digest images into compact descriptions for place recognition. These networks progressively compress images into more restricted representations, and in the case of autoencoders can also expand this information back to generate some image-like output. This concept is applied in CALC proposed by [52], who train an autoencoder to replicate the output of a gradient-based HOG descriptor and thereby force it to learn a illumination-robust representation. At inference time the pinch point in the middle of the network provides image descriptions. This was later improved upon in CALC 2.0 [53] where the autoencoder was trained to generate semantic masks for the image and thus learned semantic features, while also being forced to provide a full reconstruction of the image's appearance. This lead to additionally robust descriptors being obtained during inference from the pinch point at the end of the encoding stage.

A significant limitation of neural networks is their need for a lot of computation, typically being deployed on GPUs [84]. This is due to neural networks involving many floating point calculations for the values at each layer, with the number of computations increasing with the depth of the network. As such, a lot of recent work has been focused on reducing the computational footprint of neural global descriptors. To make more efficient use of compute resources without sacrificing too much VPR performance, [5] propose a voting scheme combining the outputs of multiple smaller networks. As binary weights are more compact and can often be computed more efficiently on traditional CPUs, [21] propose a network consisting almost entirely of binarized weights. This binarization is a similar approach to that taken by the ORB local descriptor to reduce computation [68]. An novel kind of network, the spiking neural network, is explored for place recognition by

[31], however they find it to be difficult to adapt to existing methodologies.

Global descriptors (particularly those based on neural networks [84]) provide significant robustness to illumination and other challenging conditions. However, global, whole-image descriptors tend to assume that camera viewpoint remains similar [49] and are less robust to changes in viewpoint and occlusions than descriptions based on regional landmarks [84]. The reason is that methods relying local regions are ignorant of geometric structure (as compared to global methods), resulting in improved viewpoint invariance. [49] With the use of 2D image structure comes a reliance on those structures remaining in consistent position in the image, which can only be obtained with consistent viewpoint. Various work has attempted to remediate this problem. The method proposed by [50] seeks to overcome the problem of requiring consistent feature locations by rotating CNN feature maps left and right to simulate the changes in viewpoint of looking left and right. This is so these simulated perspectives can also be searched for visual matches. The work of [54] uses a depth estimation network to produce additional synthetic views by shifting the pixels in an image directly based on their depth. The result is crude [54] but is able to simulate views from other lanes of a road scene, providing additional opportunities for place recognition.

2.1.3 Existing Work to Incorporate Structure Into VPR

While visual vocabulary approaches generally ignore the locations of local features (and achieve better viewpoint invariance as a result) the incorporation of geometric information can result in more robustness to challenging conditions like strong illumination change [49]. For this reason there are VPR methods based on local features which seek to improve their matching accuracy by incorporating spatial information. As image retrieval focuses on matching of single images as a proxy for place, this spatial information is frequently related to local features' 2D location in an image. This is despite 2D arrangement only being preserved when viewpoint is consistent, potentially introducing viewpoint sensitivity [49], although we find most methods incorporating 2D spatial information seek to mitigate this effect. The focus on single images seems to have served as a barrier to the use of 3D information in VPR despite the environment's underlying 3D structure being unaffected by viewpoint (as opposed to place recognition in domains outside VPR where special depth sensors like LiDAR or RGBD are used).

2D Structure

The method proposed by [82] seeks to improve matching by verifying the 2D spatial arrangement of local features between image pairs, even in the presence of viewpoint changes.

To this end they estimate a 2D motion field which best warps the viewpoint from one image to the other. The authors of [78] seek to build topological structures between local features which are robust to 2D affine transformations. This is in lieu of robustness to full non-rigid changes in perspective [78]. The matching of the complete topological structures and their included feature points is used to improve matching accuracy. In the process they also make use of a 2D motion consensus model to constrain 2D displacement vectors between matched local features. The method proposed by [35] takes the unusual step of discarding 3D depth for SLAM-tracked local features and taking their 2D position on the image plane. They assign these features using semantic segmentation to specific object instances, and from there construct 2D graphs. To overcome changes in 2D position due to viewpoint, they normalize the distances between graph objects. They propose a subgraph matching algorithm to leverage these graphs for place recognition. To aid place recognition under poor conditions that affect visual appearance, [23] propose a method which constructs 2D graphs with distance and angle relationships between patch features. These distance and angle relationships can then be compared to help ensure that matching of the patches is correct. For the purpose of performing a 2D spatial consistency check on top matches, the method proposed by [9] collects and stores a 13x13 grid of features for every image. These features are harvested from the intermediate layers of a CNN. For every potentially-matching pair of images during image retrieval, they match every cell in one image with the cell in the other image which has the most similar CNN features. If the corresponding cells in the same rows and columns as the matched cells also have similar features, then they contribute positively to the matching of that pair of images.

3D Structure

While 2D spatial information can be unstable with changes in viewpoint, the underlying structure of the scene itself is unaffected by the position of the observer. The use of 3D structural information ("structural place recognition" [33]) is unavoidable in domains leveraging depth sensors like LiDAR or RGBD cameras, but despite recommendations that structural information from visual SLAM or other forms of SFM can also be used [49] it remains rare to see 3D information used in purely visual localization or place recognition systems.

The method proposed by [1], while not VPR, seeks to address the problem of cross-view localization through the use of points generated by 3D SLAM. It localizes the camera's position by comparing the positions of nearby landmarks (cars) against those in a preexisting global map. Instances of cars are detected and 2D bounding-boxed by an object recognition network. The authors of [1] determine which 3D points tracked by a SLAM system project to within each car's bounding box to determine the vehicle's approximate position. Collecting vehicles near the camera, this local map is then compared with the global map. The need for a preexisting and accurate metric map of all vehicle locations makes this approach unsuitable for VPR, as a hard requirement of VPR is the ability to learn places online. The construction of accurate large-scale metric maps is also difficult without the support of VPR to help correct long-term drift in odometry. To better handle frames where an object's detection is disrupted by occlusions, we accumulate SLAM points over many frames and only then identify object instances in a single, shared 3D reference frame. The assignment of 3D points to 2D bounding boxes can also be very approximate, which is why when leveraging semantic segmentation we do so at the pixel level and on individual 3D points.

The experiments conducted by [57] attempt to use existing LiDAR descriptors [34] [27] [14] for the purpose of VPR by adapting them to use 3D points whose positions are estimated by a SLAM system [56]. Place recognition using these descriptors is similar to image retrieval in VPR, where 360-degree LiDAR scans are converted to description vectors, stored, and later matched against new scans' descriptors. To produce suitable "imitation-scans," [57] accumulate tracked 3D points from SLAM over a series of recent frames, then clip the resulting pointcloud to within a radius of the current camera pose. (see also pseudo-scans) This is done for each SLAM keyframe, creating a series of corresponding pseudo-scans which can be passed to one of the selected LiDAR descriptors. Comparing against NetVLAD [3] as a SoTA VPR method, [57] find that the best descriptor achieved a place recognition recall (at 100% precision) of 46% to NetVLAD's 2.89% under adverse illumination conditions. However, they found that in general case this rises to only 57% while NetVLAD typically achieves 80%. The LiDAR descriptors and SLAM-estimated sparse pointclouds representing the structure of the environment demonstrated strong invariance to illumination conditions however in general performed worse than SoTA VPR. Worth noting is that the points detected and tracked by sparse SLAM systems (of which [56] is an example) are significantly more sparse than 360-degree LiDAR scans, and depending on camera trajectory may not have complete surrounding coverage.

2.1.4 Conclusions on VPR

The majority of VPR methods being proposed follow the image retrieval format, with images from a camera being described and stored away for later comparison to find when a scene has been revisited. The descriptors used typically fall into one of two types, those which extract and summarize a series of local features from an image, and global descriptors which digest it at a more high level. Local features only have access to low-level details which are easily corrupted by changed in environmental conditions [51], however the spatial information which improves robustness for global features in challenging conditions also frequently leads to a need for stable viewpoint [49]. This has resulted in work to try to mitigate this in global descriptors [50], while local feature based methods have tried to extract more robust local descriptions [3] and some have tried to incorporate more structure [78] [35] [23].

These two classes of approach are often seen as opposing each other, with some calling for systems occupying a middle ground to be sought [24], however we see them more as being part of an incomplete progression. Local feature methods based on visual vocabulary have generally discarded all spatial information, achieving the highest viewpoint invariance at the cost of being the most susceptible to changes in environmental conditions [49]. Global descriptors which have gained prominence with the rise of neural networks incorporate some spatial information, but only 2D spatial information which is unstable as it becomes mutated by changes in viewpoint. For a description of the environment which is both more complete and stable, one needs to capture the true 3D structure of the elements that are visible, going beyond the 2D arrangement produced by their projection onto the image plane. This can be made possible by the SLAM systems which frequently leverage VPR and the structure they recover. The work by [57] demonstrates that significantly more stable all-around performance can be achieved. While average performance was lower, they obtained significantly better minimum performance in adverse lighting conditions where structural information is unaffected. What is needed is an approach beyond the use of off-the-shelf global LiDAR descriptors, which is tailored to the unique challenges of vision and the sparse data that SLAM systems recover and which makes more effective use of this 3D information.

2.2 Approaches to Structural Place Recognition

Parallel to VPR is place recognition using structural data. When using LiDAR scanners, RGBD cameras, and similar depth-measuring equipment, this sensor modality provides scans of nearby physical surfaces. This data is generally presented as a pointcloud containing many closely sampled points, wherever the sweeping beam of a LiDAR scanner or pixels of an RGBD camera have struck a surface in the environment and measured its position. Akin to the image retrieval formulation of VPR, LiDAR-oriented structural place recognition typically consists of matching current and past scans to determine when a place has been revisited. Descriptors are also employed in structural place recognition to convert scans into description vectors for easier and more efficient matching. The difference is that descriptors must now use the physical structure of the environment instead of extracting appearance-based information from images. Advocated for by [79] is also the field of radar-based place recognition, which is similar to LiDAR in producing scans, although at a significantly coarser level. Here we focus on LiDAR methods as the fine-grain nature of LiDAR-scanned points is much closer to the scale of image pixels and so it is easier to adapt concepts from one to the other and vice-versa.

Due to the very large number of points present in a typical scan, performance is paramount and many popular methods take the form of efficient, handcrafted descriptors [34] [14] [27]. Various approaches leveraging neural networks to process pointclouds have also been explored [74] [38] [11], as well as some methods which extract landmarks to perform comparison at a higher level [89] [40]. The majority of the following methods focus on LiDAR pointclouds but many concepts and conclusions are equally applicable to other spatial sensors, including RGBD cameras which likewise densely sample surfaces to produce similar pointclouds.

The taxonomy we present, of global handcrafted and neural descriptors along with those which operate on collections of local landmark regions is broadly comparable with the generally accepted taxonomy that exists in VPR with global image descriptors and vocabularies of local features. Other taxonomies have been proposed, for example that by [70] which is considerably more fine grained. Here we have excluded approaches such as those that try to match trajectory-to-trajectory, for example, as these could be independent of sensor modality so long as odometry is available and they generally rely on repeated similar traversals. They also separate segment-based approaches based on physical segments and those derived from semantic labels where as here we consider them to all be based on forms of landmark region. We do not classify those based on voxel grids and other small regions as automatically being separate ”local descriptors” as voxel grids etc. are frequently simply a means for processing a scan and methods employing variations on them like Scan Context [34] can often display properties more comparable to global image descriptors like embodiment of the entire scan in descriptors.

2.2.1 Global Handcrafted LiDAR Descriptors

As is the case with image descriptors, fast handcrafted descriptors have also been proposed for LiDAR pointclouds. These descriptors distill a LiDAR pointcloud into a descriptor which describes the physical structure surrounding the scanner, but do so using techniques which don't rely on neural networks for spatial description and thus can enjoy reduced computing requirements. The work by [57] leverages three well-known examples, Scan

Context [34], DELIGHT [14], and M2DP [27]. Scan Context [34] in particular has been extended many times in various directions [33] [44] [8].

Widely cited and with many derivatives, Scan Context [34] uses polar coordinates to divide the space around the LiDAR scanner. The use a series of rings 4 meters wide, then subdivide each ring into 60 bins. Points in the scan are collected into each bin based on their position and the maximum height of any point in each bin taken. The polar grid of bins can then be unrolled into a 2D descriptor, where the height of each bin provides the value of the descriptor at any given location. When comparing descriptors the best alignment needs to be found through circular shifting to account for rotation of the LiDAR sensor's orientation before the difference is computed. Scan matching is accelerated by performing a coarse search first, using a key derived from the rings of each scan.

The authors extend this work with Scan Context++ [33] with compensation for displacements in rotation and translation between scans. This is achieved through augmentation with assumptions of possible 2 meter road lane changes and 180 degree reversals in direction. They also provide an alignment between matching scans which have such a displacement. This allows for more flexible place recognition when the path taken is not the same between visit and revisit, but still permits accurate estimation of camera position after a place has been recognized.

A separate enhancement of Scan Context [34] is made by Weighted Scan Context [8]. LiDAR sensors sometimes provide intensity information for each point, based on the properties of the surface it sampled. Weighted Scan Context [8] uses the maximum intensity of a spatial bin to modulate the bin's maximum height. This additional feature increases the distinctiveness of each scan, as places with similar geometry but different surface materials will have more unique descriptors.

LiDAR intensity is also used by DELIGHT [14], another descriptor leveraged by [57] when they adapt visual data to structural place recognition. Like Scan Context [34], DELIGHT [14] also makes use of spatial binning by dividing the spherical space around the LiDAR scanner into 8 semi-spherical octants. These are then subdivided into two ranges, in a similar way to the rings used by Scan Context [34]. For each of these partitions of space, a histogram is produced from the intensities of the points that fall inside it. These histograms together make a complete DELIGHT descriptor. When adapting visual data for use with LiDAR descriptors like DELIGHT [14], the authors of [57] emulate LiDAR reflection intensity using the image pixel intensity of each tracked SLAM point that makes up a pseudo-scan.

The M2DP descriptor proposed by [27] takes a slightly different approach, projecting points into a set of orthographic views before developing a signature for each one. Within each 2D projection of the points a series of concentric rings of spatial bins is used to partition the points, very akin to the concept proposed by like Scan Context [34]. Unlike Scan context, D2DP [27] counts the number of points in each bin to obtain a density measurement, as opposed to a maximum height. The resulting density descriptors for each view are combined and subjected to SVD decomposition to reduce the dimensional of the description, with the left and right singular matrices forming the basis of M2DP's final scan descriptor. This descriptor was also used in the comparison conducted by [57].

A key commonality across all the methods discussed here, particularly those used by [57] (Scan Context [34], DELIGHT [14], and M2DP [27]), is spatial binning that gives the elements of each description vector a correspondence with the space around the LiDAR scanner. These methods divide the volume of each scan into many smaller volumes for representation by a descriptor, and this imposes the requirement that all or a majority of these sub-volumes must be similar between two scans for a match to be detected. Incomplete scans, occlusion, or different coverage (for example from a different entry or exit) can result in a failure to match. In the comparison conducted by [57] they find that LiDAR descriptors performed poorly when the portion of the route common to both visit and later revisit was very short, as it takes a series frames to accumulate a pseudo-pointcloud of tracked 3D SLAM points using their method. When there is only a short overlap before trajectories diverge again, it results in significant portions differing between overlapping pointclouds due to different entrances and exits to the scene [57]. An example situation is given in figure 2.7. We expect regions can also be missing (and thus disrupt one-to-one matching of described subvolumes) due to occlusions or differences in where the camera is pointing during the lead-up to an area of overlap, as traditional cameras are not 360 degree sensors (as with LiDAR). The approach proposed by [57] tries to overcome the very high sparsity of SLAM point tracking by accumulating points in 3D coordinates over multiple frames, but it cannot compensate for regions which are never seen. An example of the sparsity of accumulated visual features is given in figure 2.3 (seen from above in figure 2.4), while a traditional LiDAR scan appears in figure 2.5. A Velodyne LiDAR scan from the KITTI [25] dataset is depicted in figure 2.6, with its close and regular sampling of points visible on nearby surfaces (but limited vertical coverage).

A more robust system for sparse SLAM data derived from vision must essentially perform association between incomplete scans, via the remaining distinctive regions that remain. In other words, to obtain robustness to these effects would require a system that operates on landmarks instead of whole scans. The detection of points by a visual SLAM system is somewhat random, and so by chance it can occur that there are regions with fewer points. The approach of [57] also doesn't have any backwards effect on the SLAM system, to drive additional sampling of points in 3D regions where few local features have been detected.

2.2.2 Neural LiDAR Descriptors

In VPR, neural networks have presented the opportunity for powerful learned features. Various place recognition methods have sought the same for structural data as well. Where as the grid of pixels that makes up an image has made it relatively straightforward to apply CNNs, applying neural networks to unordered collections of tens (or hundreds) of thousands of points requires more work. With such a large number of points processing efficiency is also an important concern given that neural networks by themselves can be compute intensive. As with many CNN-based descriptors used in VPR, neural descriptors producing descriptors from the structure of a point cloud are trained to produce more similar descriptors for scans of close physical locations, and more distinct descriptors for scans that do not come from the same area. At inference time description methods produce descriptors from scans of each area visited, which are stored for later comparison in a similar way to image retrieval.

Many neural networks which process pointclouds take the approach of voxelizing the pointcloud in some way, summarizing the contents of each voxel into features and converting the cloud into a form traditional convolutions can be applied to. MinkLoc3D [38] takes this approach, voxelixing pointclouds and applying a series of convolutions to generate local features. At the end of the networks average pooling is used to generate a final global descriptor for the pointcloud. The authors extend this work with [39], where they process both pointclouds and traditional camera images into a combined descriptor. The images and pointcloud undergo separate, parallel network convolutions before finally being merged in the latter layers of the network. MinkLoc3D-SI proposed by [90] takes a different approach, representing point in spherical coordinates for voxelization, producing concentric shells of spatial bins similar to the operation of Scan Context [34]. This change is made to better reflect the change in density of points generated by a LiDAR scanner, where surfaces are sampled more densely closer to the sensor origin. The cause is a LiDAR scanner's use of a sweeping laser and consistent arc (not distance) between measurements. Similar to DELIGHT [14], the authors of [90] also incorporate the intensity of each point as an additional feature for each point fed to the bottom convolutional layers of the network.

LCDNet [11] also makes use of point cloud voxelization, with several initial layers of feature extraction based on 3D convolutions. It then splits two task heads, one that generates global descriptors for the point cloud for place recognition, and one that aids in relative position recovery. The place recognition descriptors are trained using triplet loss, where each training pointcloud's descriptor is pushed towards a positive matching example and away from a negative non-matching example. To assist with alignment, point features from the feature extraction stage are used to find associated points between pointclouds. From these an alignment can be recovered.

Instead of applying voxelization so that convolutions can be used, PointNet [66] is a general-purpose architecture that transforms points into a learned embedding directly. Each incoming point is embedded in a 64-dimension embedding space by a shared, learned MLP, before further convolutional feature extraction. PointNetVLAD [74] builds on this network, adapting it for place recognition by adding a differentiable VLAD [32] layer as is used for VPR by NetVLAD [3]. The initial features for each point in the pointcloud pass through multiple layers of convolutional feature extraction, before the VLAD layer generates the final place recognition description from these local features. SOE-Net proposed by [77] also builds on the work of PointNet [66] as a pointcloud-processing network architecture by adding orientation-encoding units in front of each layer's MLPs. These use a series of convolutions to introduce features from neighboring points in each of three directions, improving the representation ability of the network. SOE-Net [77] also incorporates a self-attention network after local feature extraction. This layer learns to apply a softmax to the extracted features based on long-range contextual information, improving the encoding of spatial relationships. The network concludes with a VLAD core to aggregate local point features (as was also the case with PointNetVLAD [74]) and a fully connected layer to generate the final scan descriptor for place recognition.

The use of neural networks to generate descriptors of LiDAR scans allows for the extraction of learned features, leading to better performance over handcrafted descriptors [74]. This is achieved through layers of learned feature extraction, as opposed to handcrafted descriptors' generally much more direct representation of the structure of the environment. An example would be Scan Context, where features of the environment are recognizable in its 2D descriptors [34]. This processing comes at a significant cost however, in the form of a large increase in processing time over methods like Scan Context [34] and/or a reliance on powerful GPUs for inference during VPR [11]. Many neural network based descriptors have required multiple gigaFLOPs of computing power during evaluation [30]. Despite improved performance over handcrafted methods, this makes it difficult to recommend neural network based place recognition descriptors in practical applications where mobility is required, including SLAM.

2.2.3 Landmark Based LiDAR Place Recognition

Rather than produce a descriptor for an entire scan pointcloud, and alternative approach is to describe and match them based on extracted landmarks. These are specific distinctive objects and structures that are identified and localized in each scan. Processing scans in terms of semantic objects and capturing the relations between them can be much closer to how humans recognize scenes [40]. It may also help to overcome issues with partial scans or scans of the same place with portions which differ, as landmarks provide an explicit route to represent subregions of a scene that could change.

The method proposed by [40] performs a semantic segmentation of a dense LiDAR scan, generating a series of semantically-labeled object instances. These are converted into a a graph of semantic landmarks, making the problem one of semantic graph to semantic graph comparison. To perform this task [40] propose a graph similarity network which computes graph-to-graph similarity for place recognition. This graph similarity network first generates an embedding for both graphs in each graph-to-graph comparison. This is done through parallel convolutions of both spatial and one-hot semantic features, before the features are combined and undergo additional convolution to produce the graph embeddings. The authors of [40] then employ a Neural Tensor Network [72] for the task of assessing the embeddings' similarity, instead of a simple inner product.

Semantic Scan Context [44] builds on Scan Context [34] by semantically labeling pointclouds and determining better alignments before performing Scan Context-style matching. They use semantic information to segment and extract semantically-relevant landmarks from the pointcloud data and from there find the best alignment between the current scan and each previous scan. They then use Scan Context [34] to determine the final place-matching score.

SegMap [17] is a place recognition method and extension of earlier work [18] by the same author which extracts and describes segments from pointclouds. These segments are contiguous collections of points representing objects and structures in the environment, and are grown over time with during traversal. To extract segments, first a dynamic voxel grid is implemented to which newly scanned points are continually added. Surface normals are incrementally computed as the accumulated cloud grows, and these are used to perform ongoing segmentation between parts of the pointcloud. To produce candidate associations between segments, descriptors are used which are derived from each segment using inference of a 3D-convolution encoder network. During training this network serves as the encoder portion of an autoencoder that describes and then reconstructs segments. This trains the encoder to preserve important information about each segment. During training the network is also required to generate classification labels. For place recognition the descriptors for each segment are used to obtain candidate matches, which are then checked for geometric consistency. The result is a series of correctly-recognized places. In addition to more efficient processing of pointclouds at the level of segments, [17] claim further use cases in reconstructing dense 3D maps and extracting semantic information.

The method proposed by [89] uses neural networks trained for semantic segmentation of dense LiDAR scans to obtain labels for the extraction of landmark objects. This is in contrast to our use of semantic labels projected from the visual domain, as the SLAM-derived pointclouds we work with are significantly more sparse. For each possible combination of semantic labels, histograms of the distances between those objects are concatenated to form a global descriptor for each LiDAR scan. This allows for a coarse matching of LiDAR scans.

To narrow down these possible scan matches into confirmed revisits, [89] make use of per-object descriptors and a RANSAC-based geometric verification process. The descriptors are composed of a series of histograms, each containing the distances from that object to other objects with different semantic labels. The RANSAC geometric verification seeks a transformation matrix which minimizes the re-projection error between corresponding objects in each pair of scans, while also refining the correspondence sets. Unfortunately, how this process works (including how object descriptors are used) is only very briefly explained and all work after descriptors are obtained is omitted from the provided reference implementation. After a best-fit transformation matrix and set of true object correspondences is selected for each pairing of scans, [89] propose that straightforward alignment loss be used to pick each LiDAR scan's top match.

2.2.4 Conclusions on Using Structure

The physical structure of the environment is invariant to effects like illumination that change its appearance, and underlying physical structure remains the same regardless of where the observer is located (unlike images' viewpoint-dependant projected 2D features). The work done by [57] finds that LiDAR scan descriptors have very high robustness to adverse illumination changes, even when adapted for use on data from visual SLAM. This invariance consistently exceeded what is available from other SoTA visual methods under such conditions [57]. Their accumulation of 3D points over time also helps to deal with occlusion. However, the structural descriptors used by [57] (Scan Context [34], DELIGHT [14], and M2DP [27]) operate on the principle of spatial binning, where the descriptor is ultimately derived from the subdivision of space into sub-volumes. This means that much or all of two scans of the same location need to be the same to be successfully matched.

When two visits to a scene entered and exited by different paths with a small overlap, [57] found that the way they accumulate SLAM points over time meant that scans were often too dissimilar to be matched. Core overlapping regions remained the same, but not the entire scan. The most obvious remedy would be to match scans based on matching of smaller sub-regions, with their spatial arrangement as an additional check. This would in effect mean moving towards use of distinctive landmarks and pointcloud regions for matching of places. This is necessary to more effectively handle the large-scale sparsity that results from using visual SLAM to generate points for use in structure-based place recognition, which can have varying coverage due to differences in viewing direction of traversal path.

LiDAR place recognition using landmarks has been explored, though segmentation into landmark regions is often not unassisted. Works like [40], Semantic Scan Context [44], and GOSMatch [89] perform a semantic segmentation to provide additional separation in dense LiDAR scan pointclouds between potential landmarks, while SegMap [17] makes use of estimated surface normals. In practice with visual data we have found it difficult to estimate surface normals from the much sparser point clouds generated by visual SLAM's tracking of distinct local features. The points are often far too sparse in terms of spacing to even discern a continuous surface. This problem persists when accumulating points as done by [57]. Semantic segmentation of visual data however is a well-explored area, with work like Segment Anything [36] providing fairly generalized solutions.

2.3 Conclusions on Place Recognition

Mainstream VPR relies on summarizing images into description vectors to store for later matching. Among existing VPR descriptors, there are broadly two categories of approach. Descriptors based on local features provide good viewpoint invariance due to discarding the arrangement of features in an image [49]. At the same time, their access to only the low-level features of an image makes local features vulnerable to strong illumination changes which affect pixel intensities [51]. Global image descriptors can make use of the high-level structure of images which provides robustness in challenging conditions, but the integration of this structure makes global descriptors less flexible with a tendency to assume consistent viewpoint [49]. A consistent viewpoint is required to ensure consistent image structure, with similarly-arranged contents. This makes whole image descriptions less robust to changes in viewpoint and occlusion than descriptions based on local regions [84].

The two approaches are generally seen as mirroring each other, with opposing strengths and weaknesses, and there have been calls to find some middle ground between the two [24]. However, we see the two methods as being part of a progression. Local features fail to capture any large scale structure, and so are most vulnerable to condition changes that affect the pixel intensities of local regions [51]. Global descriptors introduce structure into their descriptions which provides improved robustness to these effects, but the image structure that they capture is of a simplified 2D nature that results from viewing the scene from a particular viewpoint. Beyond capturing the 2D structure that is projected onto the image plane, capturing true 3D structure from a scene should be expected to be near-fully illumination invariant while not incorporating the effect of particular viewpoints into the matching process.

Some methods in VPR have explored manually introducing 2D structure into local feature methods [78] [35] [23], however it is rare for visual methods to explore incorporating 3D structure into something resembling recognition of place [57] [1]. Indeed, some methods leveraging 2D structure do so while discarding 3D depth [35]. Among the existing work in this direction, [57] adapt 3D structure estimated by visual SLAM for use with descriptors [34] [14] [27] intended for place recognition on LiDAR pointclouds. They demonstrate that these approaches (when adapted to visual data) provide significantly more robustness to harsh illumination changes than existing SoTA VPR methods. They do not perform as well under general, good conditions as NetVLAD [3], however provide decent, stable performance which strongly indicates use of 3D structure in VPR should be explored further.

The method proposed by [57] is not without its limitations. It has lower overall performance than SoTA methods in the absence of drastic illumination changes, and [57] note various difficulties. Most notably [57] find that the need to build up a pointcloud over time due to the sparsity of SLAM feature points means that matching is difficult when a visit and later revisit have very little overlap. The two accumulated pointclouds contain detected features at different exits and entrances, meaning a significant portion of the pointclouds differ despite containing common area. Visual SLAM also typically does not have 360 degree view, so coverage of the environment seen during two independent visits could potentially have incomplete overlap. The repurposed LiDAR descriptors employed by [57] all establish a correspondence between the full scan and the description vectors themselves, so any regions which differ negatively impact matching. We conclude from this that matching based on smaller regions, in other words recognition based on association of landmarks, is the best way to address this.

There have been some methods which extract landmarks from LiDAR scans and use them to perform place recognition, however the sparsity of SLAM pointcloud compared to LiDAR is a challenge. In our experience the data is too sparse to easily identify continuous surfaces, so segmentation based on surface normal estimation as by [17] is difficult to achieve. Semantic segmentation is employed by various methods [40] [44] [89] to obtain better separation for the purpose of extracting landmarks and semantic segmentation is very well explored when applied to visual data [36]. We explore semantic segmentation here as a way of improving the clustering process by which we generate landmarks.

2.4 Pointcloud Association Methods

It is a challenging task to associate and align sets of points in a global fashion which is fully invariant to any rotation or translation between them. ICP [6], perhaps the most famous method of aligning pointsets, is widely known to suffer from local minima and so requires an initial guess which is very close to the correct alignment.

GrassGraph [58] is a SoTA method which unlike ICP [6] is fully invariant to the relative transformation between the sets it aligns. It also also requires no descriptors or other features to label the points and provide hints as to which may correspond to each other. GrassGraph [58] leverages the concept of Grassmannian manifolds for its underlying proofs. It operates by transforming pointsets into a space where affine transformations have no effect before associating points directly using nearest neighbors. From that set of correspondences an alignment transformation can also be estimated. We use it here for the task of aligning sets of landmark points.

GrassGraph [58] has been demonstrated on computer graphics problems, aligning the points of two 3D models or two 2D shapes (like those from [42]). In these tests the sets are the same set, re-positioned and with added artificial noise/random outliers and it is guaranteed that the total number of points will be the same in both sets. This is worth noting in comparison with visual SLAM, where the points estimated by a SLAM system can have some position error, or may randomly appear or disappear between frames due to visual effects which impact detection [41], or in extreme cases may be mismatched requiring checks for consistency [10]. These properties can in turn affect the landmarks we generate through clustering, as instability in the position or detection of individual SLAM points can shift a cluster center or affect clustering and cause it not to be found. They can also vary between kinds of SLAM system, e.g. direct, indirect, and hybrid SLAM as they estimate structure differently. The indirect system DSO [19] for example must mainly estimates a pixel's depth, as opposed to the direct ORB-SLAM3 [10] which must estimate all three elements of a point's coordinates.

We did find that the global pointcloud alignment provided by Open3D [88] (which is an implementation of [69] and [87]) could reliably align SLAM-derived pointclouds. However, this method is based on extracting local feature descriptors from pointclouds and required the use of the full pseudo-pointclouds derived from SLAM. It could not operate on reduced sets of landmark points and still achieve correct alignment. The computational overhead and time required to compute features for pseudo-pointclouds with tens of thousands of points and then perform association and alignment was also far too large to be practical and we need to perform many such alignments.

2.5 Visual SLAM

Visual SLAM is a class of navigational systems that seek to build and maintain an accurate map over a large area, and to localize a camera at all times within it. Simpler visual odometry systems estimate the movement of the camera and it's current position by tracking only elements of the environment which are currently visible, but fail if interrupted by an obstruction or fast movement. SLAM extends this concept by continually capturing the tracked scene geometry and camera trajectory in a large-scale map. This map allows for recovery from interruptions through another system which SLAM incorporates: visual place recognition. As the environment is explored, each new place is captured, described, and compared against a database of all previously seen places. A novel place represents a new area in which to extend the map and is added to the place recognition database as well, where as a matching place is an opportunity for map refinement. If a place is visited again after a significant amount of travel then the position error that is accumulated during that time can be reduced using the knowledge that both ends of the path are the same. Bringing them together and adjusting the map accordingly makes it overall more accurate. Should visual odometry be interrupted, place recognition can be used to find places which have been seen before and thus rejoin the larger map.

Specific SLAM systems can be divided into direct (e.g. DSO [19]) or indirect (e.g. ORB-SLAM3 [10]) based on the operation of their visual odometry. Indirect systems operate on detected visual features when determining position, rather than directly on pixel intensities. A typical indirect system detects a large number of 2D visual feature points, matches them across frames, and utilizes bundle adjustment to jointly determine the camera position of each frame and position of each feature in 3D space. Indirect systems operate by jointly determining the camera positions and 3D geometry which best explain measured pixel intensities when they are re-projected from captured images. Here we are most concerned with the direct system Direct Sparse Odometry, or DSO [19], which has many similarities with more indirect SLAM. Like indirect methods, DSO begins by sampling points uniformly across an image and matching them. Given that the ray angles are known for every pixel in a calibrated camera, it jointly determines the depth of each point as well the camera positions such that the reprojection of that point into other images closely reproduces the measured pixel values. It is by matching these point features and determining their position in 3D space that both indirect SLAMs and DSO [19] are able to construct a map of the scene geometry, consisting of a point cloud.

The determination of these points' positions can fail in various ways, however. Detection of points is opportunistic for most direct SLAM systems, looking for corners and similar 2D features, so the number of points that can be collected is more limited (sparse) and non-uniform than that of a LiDAR scanner. (Indirect methods like DSO [19] can sometimes gain more points by allowing points on areas that are merely a gradient) There can be error in the estimated position of each point, either because of error in localizing it in each image, error in the overall solution, or other causes. Points can also be mismatched between frames and fail to be localized for failing consistency checks [10]. Points can also arbitrarily appear and disappear, being detected or not detected from one frame to the next due to noise, variation in illumination, and other visual effects. Examination by [41] found that feature points detected by algorithms like SIFT [48] could be readily disrupted by noise. Points not only could appear or disappear from frame to frame, making localization of each point difficult, but often fail to reappear reliable when a place is revisited later. [41] This can make it hard to pick up the original reference frame when returning to an old location, as indicated by place recognition. The authors of [41] propose additional criteria for selecting points more likely to be redetected in the future based on symmetry. Work by [29] over a selection of point detection/description algorithms found that illumination could affect the rate at which features were repeatedly found at a nearby position in subsequent frames. They also found that matching of points across frames could be impacted by spatial transformations like rotation or scaling. Any points which fail to be detected or which are detected in only one frame are of course hard or impossible to localize. Similarly, those which are not matched accurately may be rejected. The combination of the opportunistic detection of points, inaccurate estimation of their position, and the unreliability of their repeated detection means that the resulting collected pointcloud can be very sparse compared to LiDAR, non-uniformly distributed, and can shift and changes as points are detected or missed. This has a direct impact for this work, as when clustering these SLAM-derived points and converting cluster centers to landmarks the position of the clusters can vary with the detected positions of the points. The appearance and disappearance of points can also affect the clustering solution, causing whole clusters to disappear or be recombined with their neighbors. We find that this can make association of clusters significantly more difficult.

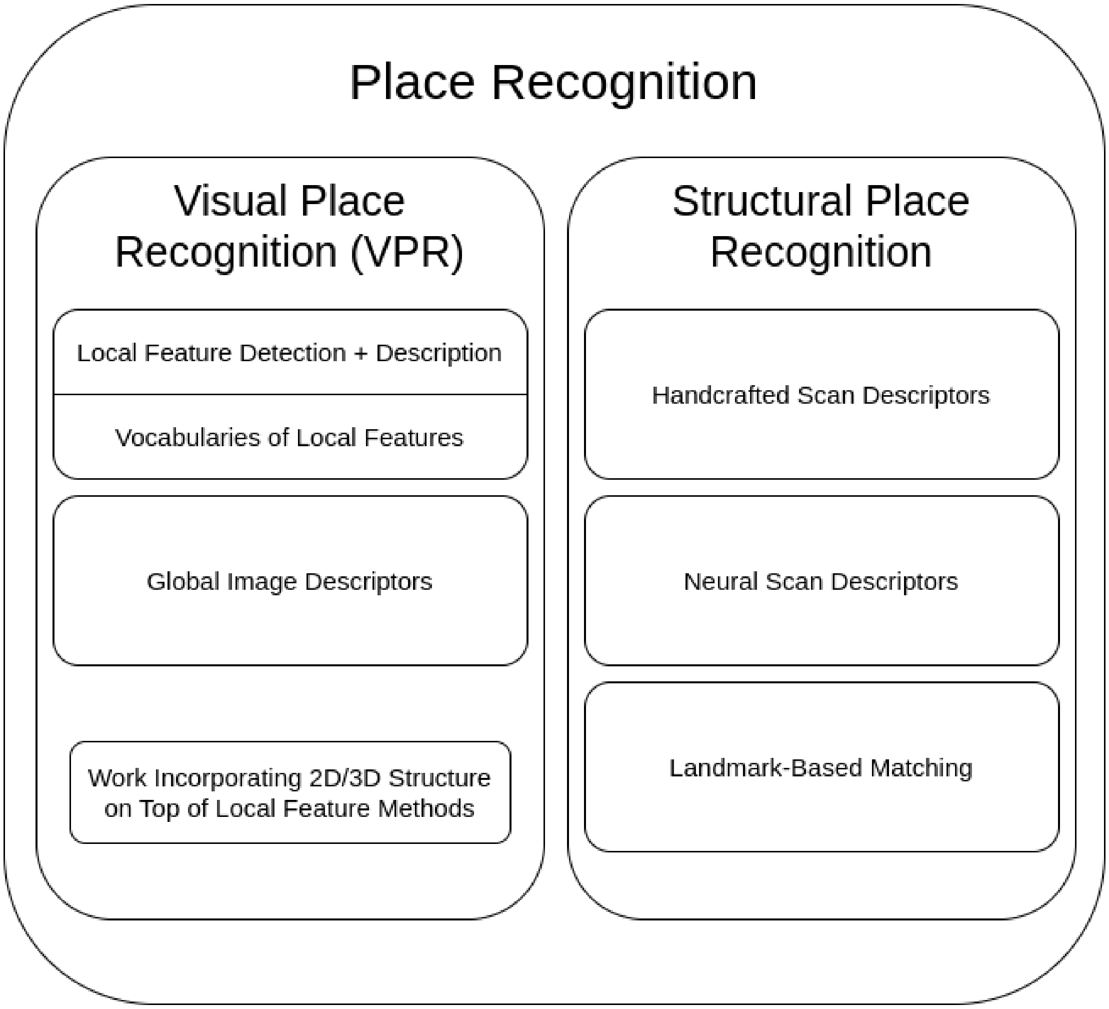
Place Recognition
Visual Place Recognition (VPR)
Local Feature Detection + Description
Vocabularies of Local Features
Global Image Descriptors
Work Incorporating 2D/3D Structure on Top of Local Feature Methods
Structural Place Recognition
Handcrafted Scan Descriptors
Neural Scan Descriptors
Landmark-Based Matching

Figure 2.1: A taxonomy of mainstream VPR approaches, plus approaches in the related field of place recognition using physical structure.

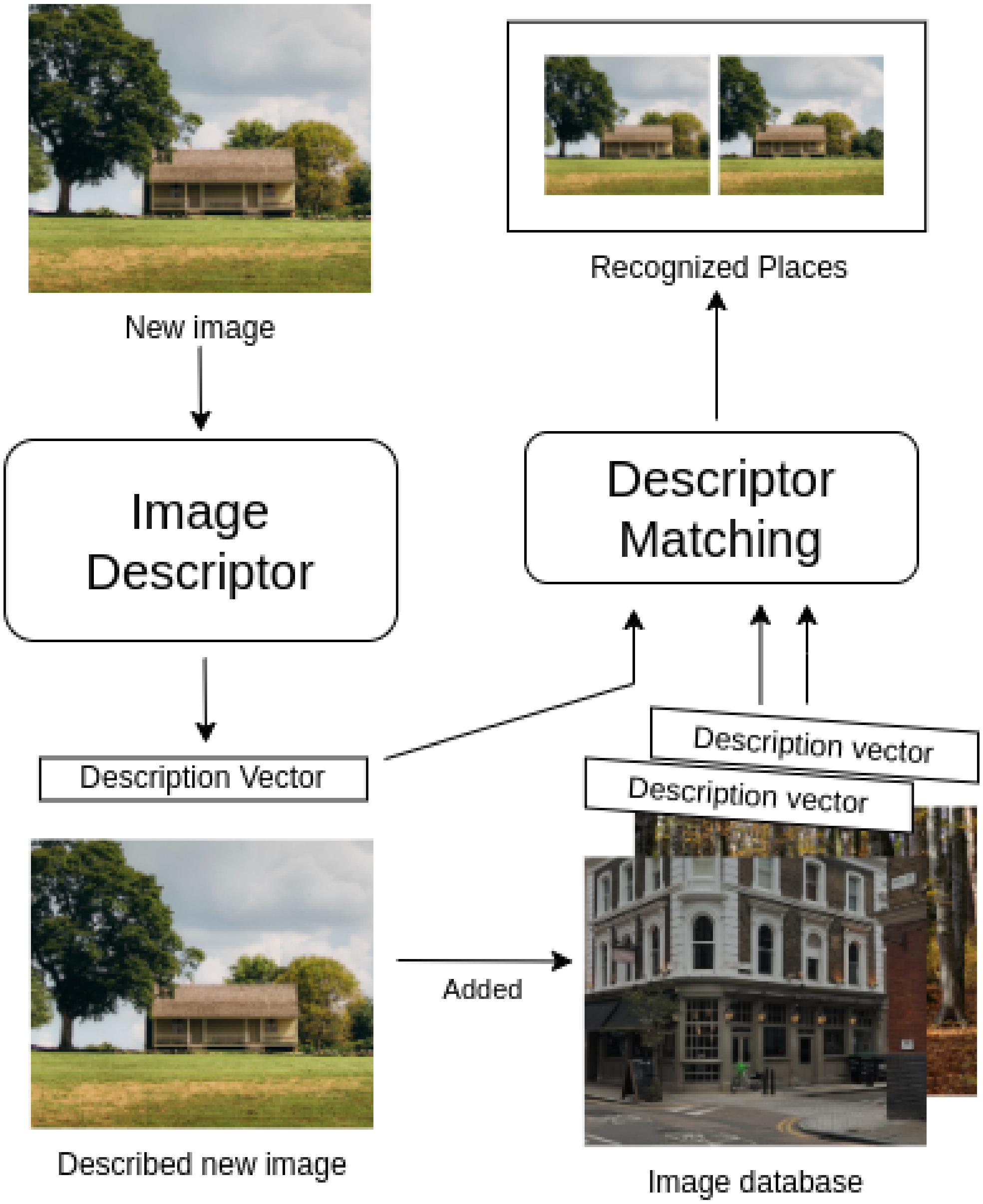

Figure 2.2: VPR is frequently cast as an image retrieval process. Under this system, each new image representing the current place is described into a description vector, which is then compared against the descriptors of all previously seen images. Sufficiently strong matches are taken as previous times the current place was visited. After matching, the current image is stored so that it can be matched against future images.

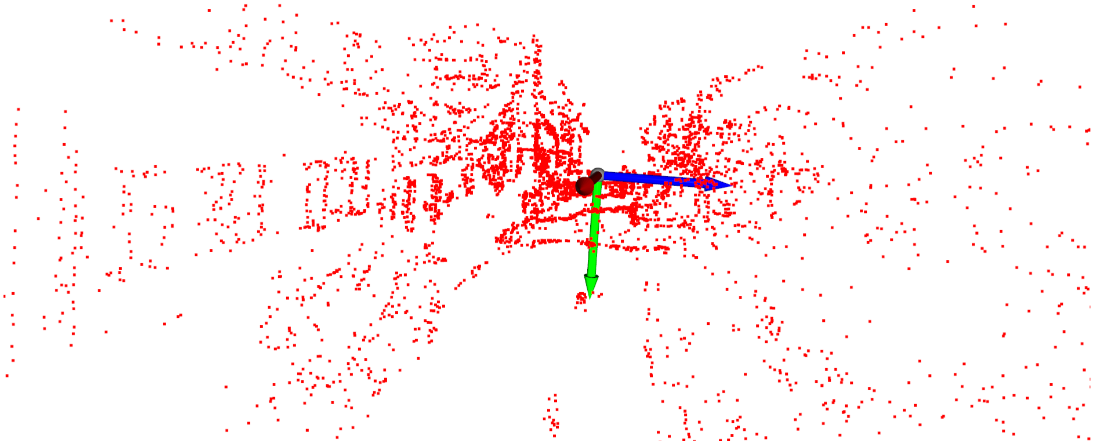

Figure 2.3: An example of a pseudo LiDAR scan produced by accumulating visual SLAM points over 100 keyframes. Note how large areas remain empty, with estimated points congregating on objects like windows and the edges of buildings.

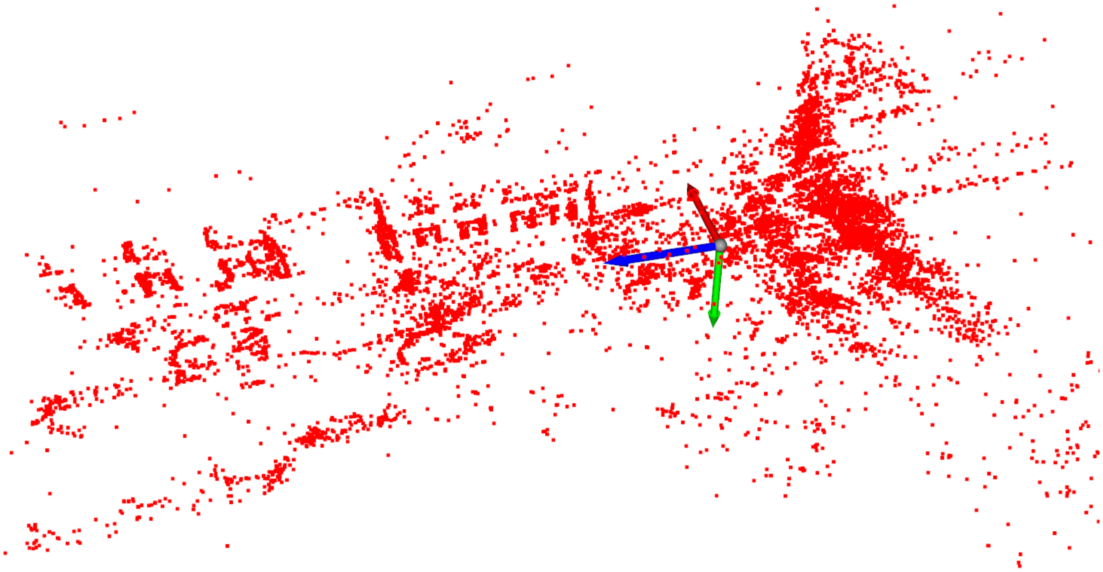

Figure 2.4: In this case a pseudo LiDAR scan is seen from above. This scan was created by accumulating points over 100 keyframes from a visual SLAM system. Note how large areas remain empty, with estimated points congregating on objects like windows and the edges of buildings.

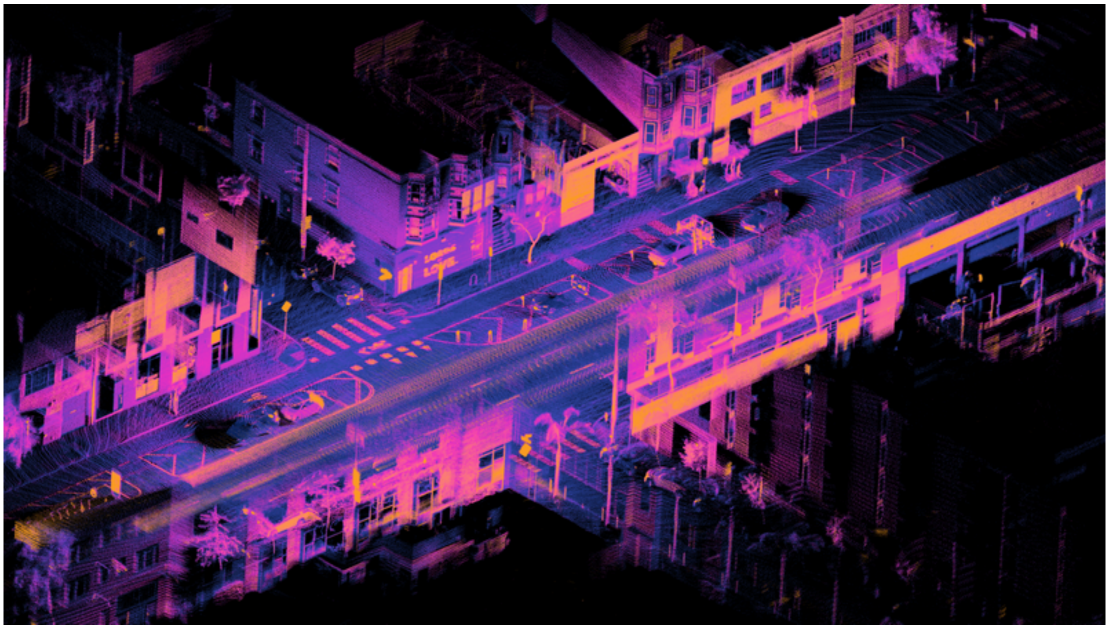

Figure 2.5: An example of a dense LiDAR scan, scanned at an intersection in San Francisco. Taken by Daniel L. Lu, licensed Creative Commons Attribution 4.0 https://creativecommons.org/licenses/by/4.0/deed.en

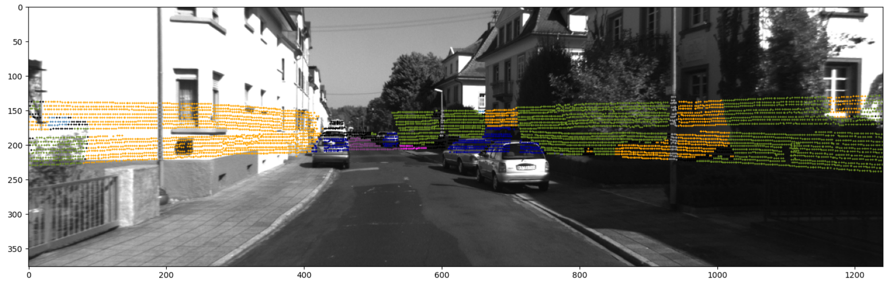

Figure 2.6: An example of a Velodyne laser scan from the KITTI [25] dataset. Note the close and regular sampling of points on every surface. These LiDAR scans happen to be in a narrow band around the vehicle due to the particular design of scanner used in the creation of KITTI, and in this case happens to be semantically labeled.

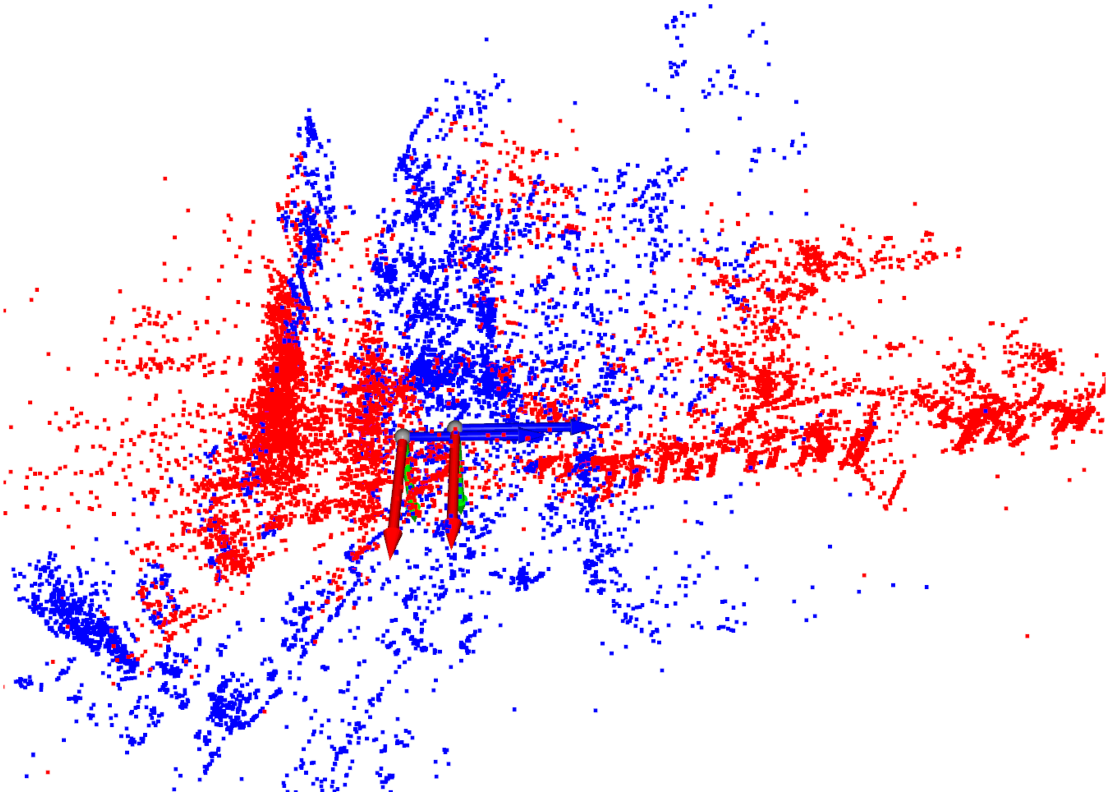

Figure 2.7: An example of a pair of pseudo-LiDAR scans in the same place captured during different routes, resulting in regions of significant difference. (manually aligned using ground-truth position) This situation is one of the cases observed by [57] which hamper recognition using global LiDAR scan descriptors.

Chapter 3

Methodology

3.1 Sparse Pseudo-Scans from SLAM Points

In the traditional formulation of VPR only single images are considered and these do not contain many 3D cues. Here we are concerned with making use of 3D structural information and so we collect 3D points representing tracked features from a SLAM system. We use the same SLAM system as [57] which makes available the 3D points being tracked in each keyframe. [1] This system is a modified version of SO-DSO [56], which itself extends the SLAM system DSO [19] with scale optimization based on stereo vision. The original DSO [19] SLAM system is simultaneously direct (operating on pixel intensities) and sparse. Because of their sparse nature, DSO and its derivatives produce clouds of tracked 3D points which are qualitatively very similar to the features of more common direct sparse SLAM systems like ORB-SLAM [59] [10]. The sampled points occur frequently at areas of strong gradient, which in turn results in them appearing on textured surfaces and on objects like cars, windows, doors, curbs, and the edges of buildings. Because of the stereo nature of SO-DSO [75], consistent scale is maintained during the runtime of the SLAM system. Other systems with monocular vision have been able to maintain consistent scale through the use of an IMU [64].

For each SLAM keyframe we accumulate together all of the tracked points as well points from the previous 100 keyframes. (roughly 10 seconds of real time) This produces an accumulated pointcloud of points representing nearby visual features which is considerably more complete. We clip it to a range of within 45 meters of the camera center. This is

[1] https://github.com/IRVLab/so_dso_place_recognition

essentially the same procedure described by [57]. The resulting ”imitation pointclouds” have the same size and shape as a real LiDAR scan and likewise every point represents part of a physical surface. (See also: pseudo-pointclouds) They do remain extremely sparse and unevenly distributed owing to when and where visual features are detected, and do not surround the camera as uniformly as an ordinary LiDAR scan. We perform filtering to remove points that are extremely close together, being duplicate detections of the same underlying visual feature. At the end of this procedure every keyframe has an associated pseudo-pointcloud. In the work done by [57] these pseudo pointclouds were passed directly to various pre-existing LiDAR pointcloud descriptors, where we extract landmarks through clustering instead.

The dataset we use as a source of video sequences for SLAM to operate on is the KITTI [25], specifically sequences 00, 02, 05, and 06. The KITTI dataset features revisits to the same location on these sequences as well as sequence 07, however the area of revisit in 07 is extremely short comprising only a few meters at the beginning and end of the sequence. This makes sequence 07 both not very valuable due to having very few frames in common between visit and revisit, and also makes quality results from this sequence difficult due to the need to initialize the SLAM system at the start of the sequence. We use KITTI as it is the dataset where [57] found that structural methods (LiDAR pointcloud descriptors) had difficulty competing with SoTA visual methods. In the other sequences they consider, structural methods succeeded because traditional VPR (particularly NetVLAD [3]) was rendered unusable by harsh illumination conditions [57]. We are concerned with the development of structurally-augmented VPR which is universally capable. This implies competitive operation in the general case and not only under uniquely difficult conditions. For this purpose KITTI is the dataset which best reflects normal conditions for a VPR system.

3.2 Semantic Labeling of Pseudo-Scans

The pseudo-pointclouds produced are initially unlabeled, possessing only the 3D coordinates of each visual feature relative to the camera. Some of the landmark-generation methods which we explore require that these points need be semantically labeled. The first step is to semantically label all of the pixels in every keyframe image, for which we use YOSO [28] pre-trained on the Cityscapes dataset [15]. For each SLAM keyframe containing tracked 3D points (which were accumulated to produce the pseudo-pointclouds) we project the points back into the viewing area and clip them to the image bounds. There cannot be visual points with an estimated position behind the camera, so clipping to be in front of the camera is not a concern. Once the 3D feature points obtained from SLAM are projected into the 2D plane of the image in which they were detected, the semantic label of the nearest pixel to each one can be obtained to label the 3D point. This label can then be propagated to all the pseudo-pointclouds of which that point is a member. Among the labels available ”building,” ”vegetation,” ”road,” ”car,” and ”sidewalk” are the most common five, in descending order. For our purposes we select buildings, cars, and vegetation as being both common and relatively self-contained (thus easier to generate landmarks from).

3.3 Landmarks from Pseudo-Scans

The generation of landmarks provides a more compact representation of a raw pointcloud while also providing a representation more akin to how humans see and recognize the places around them, as a collection of reference landmarks [89] [40]. The comparison of arrangements of landmark points by current graph association methods is much more feasible than comparison of raw point clouds, as simple testing revealed that associating 120 landmarks using [58] took well under a second while association between two complete pointclouds with tens of thousands of points took in excess of 20 minutes to compute. Here where we are making use of pseudo-pointclouds of SLAM-provided 3D features, the use of landmarks is also persued to address the extreme sparsity these pointclouds exhibit. They contain large regions with no points, where no visual features could be detected, but islands of density where more points congregate on visually complex objects. For a traditional LiDAR descriptor employing binning techniques, for example those compared by [57], these empty regions within the pseudo-scan result in many empty bins. When extracting landmarks through clustering however, the focus is on finding the specific regions with point density which we explore here.

3.3.1 Clustering of Unlabeled SLAM Points into Landmarks

Relying on the concept that the tracked SLAM points represent visual features in the underlying scene, and that these features' detection is determined by the abundance of visual detail on underlying objects and textured regions, we predict that visual features will frequently re-occur in places where they have been densely detected before. This leads to regions of space where one can expect to always find a higher proportion of detected points in a pseudo-pointcloud, even if the exact same points (with same positions) are not recovered every time. Thus, our first approach is to apply clustering to segment pseudo pointclouds into clusters which can be converted to final landmark points. To accomplish this, two clustering algorithms have been evaluated, DBSCAN [20] and BIRCH [83]. DBSCAN is a density-based approach, finding seeds of high density and then growing them to nearby points. BIRCH meanwhile applies a tree approach to break the entire pointcloud into N clusters based on distance. Once clusters are identified, their centers are found by taking an average of their constituents' positions, with these new points becoming the landmark 3D point that represents each cluster. As the one of the association techniques used [58] requires a consistent number of landmark points, and DBSCAN doesn't produce a consistent number of clusters, some additional post-processing is needed to correct the total number of landmarks. We randomly remove excess clusters or create additional ones as needed. Generally the number of clusters desired is set so that removal is balanced with addition. With BIRCH no such processing is required as the number of clusters is predetermined as a clustering parameter.

Parameter selection was carried out using clustering results from early in Sequence 00 of the Kitti dataset [25]. When tuning the parameters of DBSCAN [20], the range used for cluster growth is the most important while the number of initial points can be set as low as practical. If the range is set too small then many small and random clusters result, while if it is too large single clusters tend to grow until the consume the majority of the points, merging all detail into a single landmark point. For the dataset and SLAM feature detector considered, a range of 0.75 meters proved the best compromise for repeatable clustering that preserved underlying detail. The target number of clusters (120) was selected by observing the average number of clusters that DBSCAN produced under the above conditions, which also yielded the most repeatable clustering when specified as a target for BIRCH [83] clustering. Other BIRCH parameters were left at their defaults. Clustering should have similar properties over frames sampled from a sequence, and should also be tuned to be as similar as possible between those frames and corresponding revisits.

3.3.2 Clustering of Semantically Labeled SLAM Points

We also explore semantic labeling of points prior to clustering. This provides a sharper boundary between clusters belonging to different objects and helps address the problem of parameter selection for DBSCAN's radius of inclusion. This process begins with the semantically labeled clouds of SLAM feature points, from which label-specific subclouds are extracted for buildings, vegetation, and cars. These three are the most common labels, with the exception of roads for which there are more points than cars. Road and sidewalk points are excluded as these structures are extensive and rarely have a clear centroid and so cannot be clustered effectively. Depending on which underlying clustering technique is used, DBSCAN [20] or BIRCH [83], these three label-specific subclouds are handled differently to produce the clustering result. The target for correction of the total number of clusters is set lower at 20 instead of 120 clusters when using semantic labels. This is because semantically-guided clustering is more suited to finding whole semantically-relevant object instances, of which there are fewer in a scene compared with simpler regions of increased density. The latter may represent a wider range of smaller objects (or parts of objects in the case of buildings) which accumulate SLAM feature point detections.

In the case of semantically-assisted DBSCAN, DBSCAN [20] clustering is performed on each label-specific subcloud separately, with the clusters combined at the end. This permits different clustering parameters to be applied when clustering points belonging to each semantic class. The range at which new points can be added to a cluster seed can be extended to 3.0, 3.0, and 2.0 meters for buildings, vegetation, and cars respectively (up from 0.75 meters) where cars tend to be parked relatively close together and thus a shorter range is best. The starting number of seed points can also be raised to 25, 35, and 25 points respectively (up from 5) as with this approach the focus is on segmentation of whole semantically-relevant objects. With the improved separation of potential clusters when filtering points by semantic label these much more lax clustering criteria (particularly range) do not result in whole-cloud consuming super-clusters through unrestrained cluster growth. When the clusters have been generated they are combined across semantic labels to produce an equivalent list of clusters and their points to previous DBSCAN clustering.

In the case of BIRCH [83] clustering, the advantage of BIRCH wherein a specific, consistent number of total clusters can be specified is lost if multiple iterations of BIRCH are run in parallel. It is not possible to reliably estimate in advance how many clusters should be apportioned to each semantic class for a given scene. Instead of cluster each set of labeled points individually (buildings, vegetation, and cars), they are instead recombined but with physical separation between them to discourage cross-label combination into clusters. The three subclouds are vertically stacked in 3D space with a whole pseudo-pointcloud scan radius of 45 meters between them. BIRCH is then run on the combined cloud to produce clusters otherwise as before.

3.4 GrassGraph Association of Point Sets

While the structural information recovered by a SLAM system (and landmarks derived therefrom) is in the same frame of reference during a single visit of a scene, this does not hold true across multiple visits. SLAM systems experience long-term error in their estimate of camera position and use VPR to correct it. In order to compare sets of landmarks across visits, either their relative transformation must be recovered or they must be compared in a transformation-invariant way. We employ GrassGraph as proposed by [58] to obtain landmark correspondences and to recover the transformation.

GrassGraph [58] performs affine-invariant association between sets of points in 2D or 3D based on their geometric structure, and provides an estimated transformation matrix between them. GrassGraph [58] is derived from the insight that orthogonal projectors derived from a point set X or any affine-transformed image of X are theoretically equivalent, but due to practical limitations instead differ by an unknown rotation and sign flips [58]. This makes GrassGraph [58] a two-step process where a series of rotated and sign-flipped (but not affine-transformed) coordinates are first obtained from the same SVD decomposition required to construct a projector. In a second step the unknown rotation and sign flips are overcome to obtain correct point-to-point correspondences. The latter is by constructing a rotation-invariant graph so that the eigenvectors of the graph can be used to correctly identify points.

The first stage of GrassGraph [58] comes from the authors' search for a way to relate two sets of points X and Y which are identical except for an affine transformation between them (namely a rotation and translation). The authors of [58] observe that in homogeneous coordinates, every affine transformed pointset Y of a point set X is a linear combination of the columns of X, and that all affine transformed points remain members of a superset W which contains all possible linear combinations of these columns. There the Grassmannian is introduced is to prove that this superset W is not only a linear subspace but also a member of the Grassmannian manifold $Gr(d, \mathbb{R}^N)$ (where N is the number of points in each pointset). The beneficial result is that the linear subspace W must be unique, and that orthogonal projectors constructed from X or Y or any other affine-transformed image of X) are unique and equivalent [58]. This is embodied in the first step of the GrassGraph [58] algorithm where new coordinates for every point in initial sets X and Y are taken from a partial construction of this projector so that they can be related in a space which is affine-invariant. The new coordinates are taken from the rows of the U matrix which results from taking the SVD decomposition of X and Y as per Equation 3.1 [58]. The problem is that the construction of such a projector requires linear independence, which is not guaranteed in arbitrary sets of points. GrassGraph [58] uses SVD to extract linear independence from the columns of X and Y, but the act of utilizing SVD introduces an unknown rotation and sign flips into the generated projector and thus between these new coordinates [58].

$$X = U_X S_X V_X^T \text{ and } Y = U_Y S_Y V_Y^T \tag{3.1}$$

To overcome the unknown rotation in what would otherwise be new affine-invariant coordinates for X and Y, the second stage of GrassGraph [58] is to construct weighted graphs for both sets. The connections of these graphs are weighted with the new coordinates' euclidean distances so that rotations have no effect, making them rotation-invariant. If X and Y are indeed affine-transformed images of each other, then at this point their graphs should be the same and all that is required to associate them is to derive an identity for each point. The last step of GrassGraph [58] is to compute the graph Laplacian for the graphs of both X and Y, and the graphs' top three eigen vectors are obtained which have a row for each coordinate. These eigenvectors' rows provide a final representation for the points in X and Y in the graphs' eigenspaces where affine transformations are nullified and the same points should have the same positions [58]. Association between coordinates of X and Y is performed through a simple search for mutual nearest-neighbors. The process of performing this search is repeated for each of the possible SVD-induced sign flips (up to 8 possible cases) and the set of sign flips which results in the largest number of mutual nearest-neighbors is taken as correct. With the resulting set of correspondences between X and Y in the form of matching indices, GrassGraph [58] concludes with the regression of an affine transformation matrix relating the two sets.

The GrassGraph [58] process is independent of which specific two pointsets are being compared until the final nearest-neighbor association. Whenever we have attempted to match large numbers of pointsets to each other (for example when we have attempted place recognition), we have modified the approach of [58] for the sake of a large computational saving. We process all pointsets in advance and store their eigenvectors, such that they can be recalled later at match-time and only the nearest-neighbor associations must be performed per pairing. This means that a significant portion of the run-time cost of GrassGraph [58] only needs to be done once per set of landmark points, instead of for every pairing (which is proportional to the square of the number of pointsets).

3.5 Experimental Setup

3.5.1 Characterizing Generated Landmarks

When attempting place recognition we encountered difficulty with graph association methods [58] which were unable to successfully associate sets of landmarks from two different visits of the same place. This is despite apparent similarity when the two are overlaid using ground-truth (eg. GPS) poses. These graph association methods associate landmark points based on the graph of distances between them, where a correspondence between landmarks is identified when both share a similar configuration with their neighbors. Thus, they can be disrupted by differences between sets such as points without a corresponding match or which are offset from the expected position.

To identify areas of improvement we seek to know what aspect(s) of these landmarks make association difficult, to quantify these, and to examine where the tolerances are for existing graph association. Here we consider two measures: the number of outlier landmarks that exist in one set but not in the other, and the displacement between landmarks that were extracted in roughly the same place (across visits) and are thought to correspond to the same underlying physical feature. It is challenging to develop a ground truth map of what features of the environment ought to be extracted and used, as this act of extraction is essentially what the proposed methods were intended to achieve.

In practice however, we have found that a given landmark will either be re-detected closely enough (relative to other landmarks) that the original and re-detection will share a mutual-nearest-neighbor relation, or it will have no neighbors in the vicinity because no re-detection in that area occurred. This parallels the final association mechanism in use by the graph association technique proposed by [58], which looks for mutual-nearest-neighbor relationships in the final affine-invariant vector space. As such we use mutual-nearest-neighbor relations as a rule of thumb for determining which landmarks are or are not an outlier with no corresponding re-detection, and the distance between mutual nearest-neighbors as a measure of the error in re-detection of the underlying environmental region. In addition to measures based on a mutual-nearest-neighbor heuristic, we also show a distribution of the distances from all landmarks to their (mutual or non-mutual) nearest neighbor during a revisit. This gives an idea of what proportion of landmarks are close or far from landmarks found during another visit.

To collect these measures a series of detections and re-detections are needed. For a given sequence we collect all keyframes for which at least one other keyframe is at least 100 frames away while being closer than 10 meters. Out of these candidate revisits for the keyframe, we pair it with the revisit keyframe which is physically closest. Landmarks are extracted from the pseudo-pointclouds associated with these two keyframes, and their landmarks are compared. These revisit criteria parallel those which [57] employ. We do not restrict whether ”revisits” can be earlier or later in the sequence. For simplicity we find one closest revisit for every frame in the sequence. A summary of the number of frames with revisits found in each sequence is given in Table 3.1.

The data is presented in aggregate from all pairs of frame and matching revisit. When considering outliers from looking at mutual-nearest-neighbor relations, we present a distribution showing how many frames had what number of outliers. When considering the

Sequence 00	Sequence 02	Sequence 05	Sequence 06	Total
1353	753	791	453	3350

Table 3.1: The number of frames with revisits in each sequence meeting the criteria of the camera being within 10 meters of its past position and having occurred at least 100 frames (10 seconds) apart [57].

typical distance between these mutual-nearest-neighbor pairs, we aggregate all landmark pair distances from all frame pairs before presenting them. We do the same for distances from each landmark to its nearest (mutual or non-mutual) neighbor, collecting together the distances for every landmark in every pair of frames.

3.5.2 Sensitivity Testing of GrassGraph Association

To quantify what aspects of the generated landmarks make association with methods like GrassGraph [58] difficult, trials were conducted with controlled outliers and positional noise over varying ranges. For each frame with a valid revisit we take each set of real clustered landmarks (in this case using unassisted BIRCH [83]) and derive from it a corresponding synthetic set of landmarks with a random rotation/translation and desired number of outliers and degree of noise. GrassGraph [58] association is then attempted on this pair of landmark sets to obtain performance metrics. We do so for each frame for which a valid revisit exists and aggregate the performance results across all of these samples, for each sequence of frames. This was repeated for each combination of outlier percentage and noise standard deviation that we tested. The criteria for what qualifies as a revisit remain the same (another frame within 10 meters and occurring at least 100 frames apart [57]) and the number of qualifying frames in each sequence is given in Table 3.1.

The percentages of outliers tested are stepped over the same range as considered by [37] in their analysis of GrassGraph [58], with an additional consideration of the low end where there are few outliers as there was observed a sharp drop in performance in this region. The analyses conducted by [58] and [37] appear to ignore the scale of the noise added relative to the scale of the pointcloud itself when selecting values. Here we anchor the range of positional noise standard deviations to the standard deviations we measure in our own landmark clusters. The range covers from zero to six meters, twice the standard deviation observed of semantically-assisted clustering and four times that of traditional clustering. Increased detail is also given to the 1.5 meter range encompassing the first standard deviation of traditional clustering.

To create each frame's pair of real and synthetic landmarks for testing, we start with its BIRCH clustering result. We subtract the mean position of the landmark cluster points to center them on the origin. GrassGraph [58] estimates an aligning transformation between sets of landmark points, and so we generate a ground truth rotation and translation. The rotation matrix is uniformly sampled across the unit sphere (taking care not to excessively sample its poles). The translation is a uniformly sampled vector between (-45,-45,0) and (45,45,8), the typical extents of an (origin-centered) frame's pseudo-pointcloud. These are expressed as homogeneous transformation matrices and applied to transform the centered BIRCH landmark points to become this frame's synthetic set.

To finalize each frame's pair of landmark sets, outliers and noise must be added to the transformed landmark points. To introduce outliers the number of outliers was determined from the desired outlier percentage and the number of landmark points. After determining the minimum and maximum XYZ bounds of the transformed points, the required number of outliers were uniformly sampled from within that volume. These random outlier points then replace a random subset of the transformed landmarks. To introduce positional noise into all transformed points a unit vector is generated for each one and then randomly rotated. The length of each rotated vector is then multiplied by a sample from a half-normal distribution with the desired noise standard deviation. This mimics the random offsets expected of positional noise, where the measured displacement distances are expected to follow a half-normal distribution (as distances are strictly positive). Each transformed point is added with one of the rotated/scaled unit vectors to give its new "noised" position. The end result is the set of original BRICH clusters and a set of clusters that have been rotated and translated in a known way before addition of outliers and noise to the desired degrees.

We measure the success of landmark association attempts performed on these two sets by GrassGraph [58] in three ways. The first is the percentage of landmarks which were associated between sets, a general health metric. The second is the Frobenius norm between the alignment matrix estimated by GrassGraph [58] and the true combined rotation and translation used to generate seccond set of landmarks. This is the metric that [58] themselves report. Third is the more easily interpreted angular difference in orientation between the GrassGraph-proposed alignment and the true alignment of landmark test sets.

$$\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{i,j}|^2} \tag{3.2}$$

The number of associations found is determined directly, while to compute the Frobenius norm the true transformation between landmark sets is subtracted from the estimated transformation before applying Equation 3.2. The true transformation is the combination of the randomly selected rotation and translation used to generate this frame's set of synthetic set of landmarks. To estimate the angular error between the true alignment and estimated alignment for this frame's test sets, the estimated affine transformation provided by GrassGraph [58] is decomposed into a rotation, translation, zoom, and shear using Transforms3d [7]. The recovered translation is multiplied with the transposition of the true rotation. If the two are equal this yields an identity matrix, however if they are not it yields a rotation matrix with some angle of rotation representing their difference. To compute this angle of rotation the identity in Equation 3.3 is used. In some rare cases the transformation estimated by [58] is not valid and cannot be decomposed, in which case we assign an angular error value of 180 degrees, the maximum possible.

$$Trace(R) = 1 + 2cos(\theta) \Rightarrow \theta = arccos(\frac{Trace(R) - 1}{2}) \tag{3.3}$$

3.5.3 Use of SHREC Data in Sensitivity testing

When initially proposed, the authors of GrassGraph [58] made use of objects from the SHREC [42] dataset as a source of 3D pointsets when testing its resistance to outliers and noise. These sets of points consist of 250 points sampled on the surface of common objects. An example is Figure 3.1. During the course of our own testing of GrassGraph's [58] robustness to noise and outliers, we repeat our analysis on this dataset as well. The authors of [58] give little indication of how the noise they add was scaled relative to some aspect of these pointsets, so we independently determine a scaling factor when applying noise ourselves. The clustered landmarks we used were drawn from pseudo-pointclouds of tracked LiDAR points and are clipped to a radius of 45 meters. To estimate an equivalent maximum radius of this other dataset we find the average center of each pointset from SHREC [42] (as conveyed by [58]) and then compute distances from the center for each point. We take three standard deviations from the mean distance as the assumed maximum radius of SHREC pointsets. We take this approach instead of simply finding the furthest point as a couple of the 50 pointsets used by [58] have extreme outliers. For parity with our other sensitivity testing results we randomly sub-sample the SHREC pointclouds from 250 points to 120. This ensures that the same number of points is used through our testing.

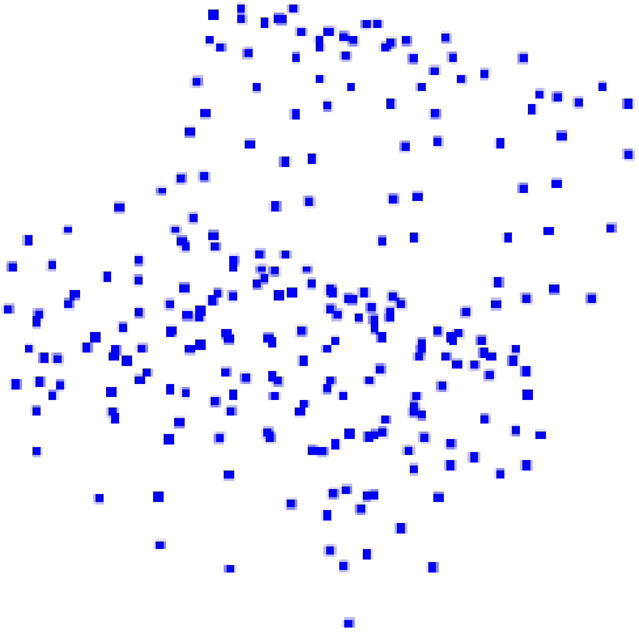

Figure 3.1: An example of a common object found in the SHREC [42] dataset (a chair), subsampled to 120 points from the default 250.

Chapter 4

Experimental results and Suggestions for Future Work

4.1 Characterizing Generated Landmarks

Here we characterize the landmarks which we generate from clouds of tracked 3D SLAM features. An example of the underlying clustering generated by each landmark generation method is given in Figure 4.1, with a side-by-side comparison of how two visits have clustered. We are interested in the repeatability of landmark detection as this has a direct effect on the success of landmark association. We express repeatability in two measures. First is the number of outliers generated, where a landmark is found during one visit but not in the same place during a later revisit. The second is the amount of positional error which is how far away a landmark is from its initial position when redetected. Both measures are ideally zero, but during clustering landmarks can fail to be detected, spurious ones can be added, or due to differences clustering a landmark may not be found in precisely the same position. These effects all have a negative effect on the ability to associate two sets of landmark points.

To obtain these measures a set of ground truth associations between landmarks is needed, but no such set initially exists for the landmarks we generate. There is also no ground truth set for what or where landmarks should be detected. It is difficult to create a ground truth set of associations using thresholding because of the varying density of visual pointclouds and of the extracted landmarks. For these reasons we rely on a heuristic that two landmarks probably refer to the same underlying physical feature and should be associated when they are both each other's nearest neighbors from the opposing set. In this way we can create our own synthetic ground truth set of associations where none existed. While this method does not have the most satisfying theoretical basis, in practice we have found it to be a good way to find landmarks that correspond to the same approximate physical feature.

To obtain a set of ground truth associations using this heuristic, we start with two sets of landmarks across different visits. We then manually align them using ground-truth camera poses (derived from GPS or similar). We then test pairs of landmarks to see if they are mutual nearest-neighbors (see Figure 4.2). That is, a landmark in one visit's set and a landmark in the other visit's set are both each other's nearest neighbors out of all the landmarks in the other set. Identifying these pairs gives us landmarks probably corresponding to the same physical feature (the distance between which we can measure). The largest caveat to this method is that distant outliers can rarely present as mutual nearest neighbors if no alternative is near to either of them.

4.1.1 Landmarks Clustered without Semantic Assistance

The key measures when generating landmarks via the clustering of SLAM points are the number of outliers and the position error. Outliers are the landmarks generated through clustering which appear in one visit but not in a revisit. The position error describes the degree to which a landmark changes position when it is found both in a visit and revisit. For landmarks generated through clustering with DBSCAN [20] and BIRCH [83], the distribution of frames by number of outliers is shown in Figure 4.3 and described in Table 4.1. As alignment using [58] requires that the number of landmarks in both sets be consistent, a consistent total for every cloud of SLAM features must be maintained by adding or removing some at random. Table 4.2 shows the corresponding figures with total correction performed, which has the effect of increasing the number of outliers. The corresponding figure is Figure 4.4. The targets for all methods (120 for traditional clustering, 20 with semantic assistance due scenes containing only about that many high-level semantic elements) were set based on the typical number of clusters found in the first 100-200 frames of sequence 00.

Outlier Measurements

It can be seen in the two leftmost columns of Figure 4.3 (clustering with DBSCAN [20] and BIRCH [83] with no semantic assistance) that the number of outliers has a low peak near 20% followed by a long tail of assorted frames with a larger number of outliers. This long tail brings the average number of outliers to 25-56%, as collected in Table 4.1. Such a large number of outliers is quite challenging, and as can be seen in Table 4.1 standard deviation for the number of outliers is also significant indicating a sizable spread. The lowest average number of outliers plus one standard deviation reaches close to 50%, the limit of what graph association methods have typically been tested to [58] [37]. This is a serious challenge for comparing sets of landmarks, as outliers make it difficult to correctly associate sets of landmarks and the GrassGraph [58] method of association suffers a particularly severe impact from outliers in 3D space [58]. GrassGraph [58] also requires a consistent total number of landmarks, and correcting this total has a noticeable impact on DBSCAN [20] which unlike BIRCH [83] doesn't target a fixed total. We provide supplementary post-correction results in Figure 4.4 and Table 4.2.

		DBSCAN	BIRCH	Semantic DBSCAN	Semantic BIRCH
Seq 00	Min	16.66	12.50	0.00	0.00
	Avg	43.67	32.73	28.23	28.87
	Std Dev	15.86	16.32	20.49	18.07
Seq 02	Min	26.47	24.16	0.00	5.00
	Avg	55.88	46.16	36.61	32.97
	Std Dev	14.51	12.88	24.56	16.16
Seq 05	Min	17.59	10.00	0.00	0.00
	Avg	42.28	31.97	26.52	28.05
	Std Dev	17.16	16.01	20.67	16.98
Seq 06	Min	0.00	9.16	0.00	5.00
	Avg	33.22	25.39	21.73	27.64
	Std Dev	17.88	18.49	21.84	17.99

Table 4.1: Percentage of outlier landmarks. These are the landmarks produced through clustering which appear in one visit but have no corresponding mutual-nearest-neighbor redetection in a subsequent revisit. The implementation of semantic assistance significantly reduces the average number of outliers for both methods of clustering. They also perform more closely indicating that the specific kind of clustering becomes less important when semantic info can help provide cluster separation. These measurements were taken **before** the total number of detected landmarks was corrected through random addition/removal.

		DBSCAN	BIRCH	Semantic DBSCAN	Semantic BIRCH
Seq 00	Min	27.50	12.50	0.00	0.00
	Avg	46.67	32.67	33.43	28.87
	Std Dev	13.74	16.08	19.81	18.07
Seq 02	Min	29.16	24.16	0.00	5.00
	Avg	55.26	46.16	42.84	32.97
	Std Dev	14.42	12.88	21.67	16.16
Seq 05	Min	20.00	10.00	0.00	0.00
	Avg	44.26	31.97	32.31	28.05
	Std Dev	16.90	16.01	22.03	16.98
Seq 06	Min	23.33	9.17	0.00	5.00
	Avg	40.38	25.44	30.29	27.68
	Std Dev	16.17	18.52	21.94	18.09

Table 4.2: Percentage of outlier landmarks. These are the landmarks produced through clustering which appear in one visit but have no corresponding mutual-nearest-neighbor redetection in a subsequent revisit. This table is supplementary to Table 4.1 and shows the effect of outliers introduced when correcting for a consistent total number of landmarks. This impacts BIRCH less severely as it intrinsically targets a set number of clusters. These measurements were taken **after** the total number of detected landmarks was corrected through random addition/removal.

Tuning Challenges

Comparing the performance between DBSCAN [20] and BIRCH [83], BIRCH consistently produced 8-11% fewer outliers on average. While BIRCH (which is based on the partitioning of a spatial tree) has a strong tendency to produce many approximately-spherical local clusters from larger regions of many SLAM feature points, this segmentation was often overall more consistent than the result of DBSCAN's density-based clustering. It is challenging to set an appropriate, universal scale/density parameter for DBSCAN as nearby objects are often not well separated (eg. between a tree or car and the ground, curb, or a nearby wall), with their closest distance being similar to the distance between member points. If the cluster growth range is set too large then large portions of the pointcloud are consumed by single clusters (Figure 4.5). If the range is too short then many very small and unpredictably placed clusters proliferate (Figure 4.6). In either case, the result is clusters which have no corresponding match from other visits, in other words an increase in outliers when using DBSCAN. With BIRCH the way that large structures are broken up into consistent-sized regions occupied by each cluster are somewhat more similar between visits. BIRCH is also essentially unaffected by the need to correct the total number of landmarks with random additions or deletions, as it targets a predefined total number of clusters. This is an important practical consideration when using GrassGraph [58] as it requires a consistent total.

Noise Measurements

Separate from the number of outlier landmarks which are not redetected during later visits, those landmarks which are redetected may not be found at exactly the same place. This positional error is the other measure we quantify affecting association of landmarks from different visits. Positional error is ideally zero (when landmarks are consistently detected in identical locations), and because there is an implicit absolute value in determining distance error measurements are expected to follow a half-normal distribution. This is generally observed here in Figure 4.7. For DBSCAN [20] and BRICH [83] without semantic augmentation the measured standard deviations are comparable at around 1.5 meters, with BIRCH being only ten to twenty centimeters more. Sequence 02 is a noticeable departure with overall higher outliers which shift the distributions in the second row to the right and result in a peak with most distance measurements at two meters. When treated as a half normal distribution this makes the measured standard deviation more pessimistic, as more samples have been moved away from the assumed peak at zero. For this reason, while in Table 4.3 we assume a half-normal distribution centered at zero and report the computed standard deviation for all the topmost figures, for sequences 02 (and 00 as it shows some of the same tendency) we provide supplementary average and standard deviation figures (assuming a normal distribution) at the bottom of the table. In general the standard deviation is very comparable between BIRCH and DBSCAN. Table 4.4 and figure 4.8 have results post-total-correction, although the addition and removal of landmarks to achieve a specific total rarely affects these distance results.

Abnormality of Sequence 02

Across all results, the performance of methods on sequence 02 stands out as being particularly poor. There is a large number of frames with high outlier percentages in sequence 02, spanning from 20% outliers all the way to 80% or more. The standard deviation of observed positional error in Table 4.3 is also typically twice as high as in other sequences. This is indicative that there is particularly poor repeatability. Upon examining the sequence itself, we find that in the segments which are visited more than once (922-1036,

	DBSCAN	BIRCH	Semantic DBSCAN	Semantic BIRCH
Seq 00	1.49	1.58	2.13	2.78
Seq 02	2.50	2.38	5.61	3.82
Seq 05	1.31	1.47	2.72	2.94
Seq 06	0.85	1.24	2.14	3.01
Average	1.57	1.67	3.15	3.14
Avg (excl. 02)	1.22	1.43	2.33	2.91

		DBSCAN	BIRCH	Semantic DBSCAN	Semantic BIRCH
Seq 00	Avg	1.26	1.43	1.61	2.23
	Std Dev	0.78	0.68	1.39	1.66
Seq 02	Avg	2.31	2.20	4.51	3.40
	Std Dev	0.96	0.92	3.31	1.73

Table 4.3: The positional noise in terms of distance between landmark and mutual-nearest-neighbor redetection, expressed as zero-centered half-normal standard deviation. For clustering methods without semantic augmentation the standard deviation for positional error (essentially the average distance between pairs) is quite similar at around 1.5 meters. This does roughly double for the semantic methods though, most likely because the whole-object clusters semantic labeling encourages tend to be much larger. The units are in meters, to scale. Average and standard deviation are also given for sequences 00 and particularly 02, where the assumed half-normal distribution is a more poor fit due to not being centered on zero. These measurements were taken **before** the total number of detected landmarks was corrected through random addition/removal.

1794-2003, 3329-3412, 4192-4625), we find that these are near-universally comprised of a scene with a narrow road with no parked vehicles or other objects present. As illustrated in Figure 4.9 buildings and other possible visually distinctive landmarks are obscured by continuous and thick vegetation which is both lacking in breaks and visually distinct features, yet remains textured and so generates many SLAM points. Under these conditions it is exceptionally difficult to obtain a distinctive description, particularly with clustering as there are no objects near the road or breaks in the vegetation which would allow clustering into distinct blobs. In our semantic approaches we extract to three classes (buildings, cars, and vegetation) which also makes these sorts of environments more challenging as it means additional objects cannot become landmarks during clustering. Its worth noting that [57] found performance on 02 was also worse, especially for NetVLAD [3] visual vocabulary (the poorest performance, with a maximal recall of 1.2%) which otherwise performed bet-

	DBSCAN	BIRCH	Semantic DBSCAN	Semantic BIRCH
Seq 00	1.64	1.58	2.35	2.78
Seq 02	2.73	2.38	4.53	3.82
Seq 05	1.55	1.47	2.47	2.94
Seq 06	1.13	1.24	2.47	3.01
Average	1.76	1.67	2.95	3.14
Avg (excl. 02)	1.44	1.43	2.43	2.91

		DBSCAN	BIRCH	Semantic DBSCAN	Semantic BIRCH
Seq 00	Avg	1.37	1.43	1.77	2.23
	Std Dev	0.90	0.68	1.54	1.66
Seq 02	Avg	2.45	2.20	4.00	3.40
	Std Dev	1.19	0.92	2.10	1.73

Table 4.4: The positional noise in terms of distance between landmark and mutual-nearest-neighbor redetection, expressed as zero-centered half-normal standard deviation. This table is supplementary to 4.3 where here we include the effect of correcting the total number of landmarks for consistency. However, this has little to no effect on positional error. The units are in meters, to scale. Average and standard deviation are also given for sequences 00 and particularly 02, where the assumed half-normal distribution is a more poor fit due to not being centered on zero. These measurements were taken **after** the total number of detected landmarks was corrected through random addition/removal.

ter than all other methods on this dataset. Such a visual vocabulary would be expected have difficulty finding distinctive features in this very indistinct environment.

4.1.2 Landmarks Clustered with Semantic Assistance

When clustering clouds of SLAM points it is important to repeatably separate portions that should belong to different landmarks. Inconsistent clustering creates outlier landmarks which do not have a matching landmark when revisited, while inconsistent boundaries of a landmark's cluster affects the estimation of its position. With DBSCAN the primary challenge was to select a proper cluster growth range, a balancing act between single large clusters that obscure much of the pointcloud into a single landmark coordinate, or many small clusters that spawned unpredictably. When adding points to growing clusters there is no consideration for what elements of the scene these points belonged to, and so the boundaries of objects were not respected. BIRCH clustering escaped the range selection problem by generating fairly consistently-sized clusters but could neither take advantage of the human, semantic boundaries of underlying objects given only point coordinates. Thus, here we evaluate a second pair of semantically-assisted DBSCAN and BIRCH based clustering mechanisms, where clouds of 3D SLAM features are first semantically labeled by class (car, building, vegetation) before clustering to provide more separation and better clustering results.

The performance of these semantically-augmented DBSCAN and BIRCH methods is presented in the right two columns of Figure 4.3 regarding outliers and Figure 4.7 regarding positional error, and Tables 4.1 and 4.3. The performance between the two augmented clustering methods in terms of outliers and positional noise is very comparable (when not correcting the total number of landmarks). This suggests that with improved separation of clusters thanks to semantic labeling the particular performance of either clustering method is much less important. BIRCH retains some practical advantage, simply because it targets a specific total number of clusters which is a requirement of GrassGraph [58] landmark association. The impact of correcting the total number of clusters is more pronounced for the version of DBSCAN with semantic assistance due to the reduced number of clusters, giving each cluster added or removed more impact. For supplementary results including the extra effect of correcting the total number of landmarks, see Figures 4.4 and 4.8 and Tables 4.2 and 4.4.

When comparing clustering methods with and without semantic assistance, the former exhibit an 8-10% reduction in outliers (see Table 4.1). This is a fairly significant fraction of the outliers present, and brings the range of average outliers from 25-56% down to 22-37%. We attribute this to the improved separation between clusters through labeling, leading to more consistent clustering of each cloud of SLAM feature points. In general it is beneficial to clustering if additional separation can be achieved through additional (in this case semantic) clustering features. This reduction in outliers can be visualized in Figure 4.3 where the distributions pertaining to semantic methods are noticeably shifted left towards zero. In addition to the improved number of outliers however, the estimated standard deviation for positional error is approximately doubled for these semantically-assisted clustering methods compared to those without semantic assistance, as per Table 4.3. White there are a few possibilities for this, the most likely explanation is the larger size of clusters generated. These clusters represent more complete semantically-relevant objects instead of smaller segments of them which happen to accumulate tracked SLAM feature points. As such, their larger size magnifies in real-world meters any variation in the extent and coverage of the points assigned to them. Similarly with other methods, Sequence 02 remains more challenging to cluster due to the lack of distinctive features. Semantic labeling is limited to cars, buildings, and vegetation and only these points are kept, however there are few other elements of the scene to cluster regardless. With or without semantic labeling, the continuous nature of the dense vegetation remains difficult to cluster into separate portions reliably.

With respect to gaps in the distributions for outliers when considering semantic methods, this is due to the fewer number of clusters generated (20). This is in turn owing to the naturally fewer number of high level semantic elements in a scene (trees, cars, buildings) compared with the many more partial regions clustered by the purely spatial traditional clustering methods. As such measures like the percentage of outliers are less precise since each cluster represents a larger proportion.

4.1.3 Discussion of Clustered Landmarks

For the sets of landmarks generated from clouds of tracked SLAM features, we are interested in two measures. The first is the percentage of outliers, where outliers are the landmarks present during an initial visit but not seen in the same location during a subsequent revisit. The second measure is the typical error in position when a landmark is detected again in nearly the same place. As a heuristic to differentiate the two, we look for nearest-neighbor relations landmark sets. If two landmarks (found during visit and revisit, respectively) are mutual nearest neighbors, then neither has a closer landmark in the other set. We thus assume that they are detections of the same landmark.

The most prominent result encountered while characterizing the landmarks we generate from clustering is the high number of outliers typically generated. This was reduced considerably through augmentation with semantic labels, though remains higher than would be desirable for reasonable matching techniques. Graph association methods for comparing sets of landmarks can be sensitive to outliers, particularly the method proposed by [58] in 3D coordinates. The need for a consistent total number of landmarks by this method introduces further complication, as the addition/removal of landmarks to achieve a specific total introduces additional outliers. We find that semantic labels provide improved segmentation between elements of the environment with different semantic classes, making it easier to cluster them separately whenever they are in close proximity or physically connected. Without this separation, selection of tuning parameters for clustering is difficult, as with DBSCAN [20] frequently over-segmenting the SLAM feature pointclouds into many small parts or growing clusters large enough to consume most of the cloud. The increase in cluster size when segmenting entire semantic objects appears to lead to an increase in positional error, and the reduction in the number of clusters to match the number of semantic objects could increase sensitivity each cluster overall.

Also encountered were particularly poor results on sequence 02, where the environment is lacking in distinctiveness. The environment consists of a narrow street with dense vegetation that obscures all other landmarks, man-made or otherwise. This presents a challenge for many place recognition systems including those considered by [57] (especially visual vocabulary [3]), but the lack of objects or breaks in the vegetation is also challenging for clustering. Without breaks it becomes ambiguous how to cluster the vegetation for each visit, and the addition of semantic labels is ineffective when the vegetation shares a single label. Adding semantic labels can also restrict clustered landmarks to particular classes of object, though few objects are present in sequence 02's revisited scenes.

For the following examination of graph association [58] performance given the landmarks produced by each of the clustering-based methods proposed, representative ranges of outlier percentage and the degree of positional error are required. The outlier percentages measured for the methods here have covered a fairly wide range from 22-56%, with those results included at each standard deviation beyond that to consider, although above 50% association is unlikely to succeed. With respect to the standard deviation, the values measured have have typically been around 1.5m for standard clustering and 3m for semantic clustering. For further examination we take 1.5m as a base measure, generating a scale covering tenths of this value and quarter-multiples up until at least two standard deviations of the semantic methods are covered.

4.2 Measuring Sensitivity of Graph Association

The ultimate goal of extracting landmarks from 3D data gathered from SLAM is to perform place recognition. This requires that landmarks be associated from each possible visit to each possible revisit and compared for similarity. It is also very convenient that any relative transformation be recovered so that the two sets can be aligned with each other. In this way landmarks can be more directly compared, for example through their Euclidean distance. GrassGraph [58] has been the most promising method for doing so, however it has been enormously challenging to reliably obtain sufficient correct associations between landmarks to make place recognition practical. To examine why this is the case, we construct synthetic sets of landmarks with known properties. By varying the amount of noise or number of outliers artificially included, we discover what aspects of our real landmarks are most problematic for GrassGraph [58].

The pointcloud association technique GrassGraph proposed by [58] associates 3D points between pointclouds in an afine-invariant way by comparing the distances to their neighbors, resulting in likely associations and a recovered set-to-set alignment transformation.

To better understand why it is difficult to associate the landmarks we generate using GrassGraph, we examine using this technique under controlled conditions while varying the number of outliers and amount of positional noise. These conditions are created synthetically by taking two copies of the same set of clustered landmarks and altering one to have the desired number of outliers or desired positional noise, as well as a geometric transformation to be recovered. For each combination of noise and outliers the performance of GrassGraph is measured, across all revisits in all four previously considered sequences.

The outlier percentages are initially stepped over the same range as [37], however due to outliers' severe impact we also provide a more detailed look at the low end of the scale which is stepped in single outliers. Both [58] and [37] seem to ignore the scale of the noise that they add to points during testing relative to the spacing between points, the scale of the cloud, or other some other metric. Here we sweep based on the standard deviations of noise we measured in the preceding section. Three meters and six meters are twice the observed standard deviation of the non-semantic and semantic landmark clustering methods respectively. The range covers from zero to six meters, roughly twice the standard deviation measured in semantic-assisted clustering and four times that of non-semantic. Additional detail is given at the low end of the scale from 0.0 to 1.5 meters (roughly the first standard deviation of non-semantic clustering).

When associating sets of landmarks GrassGraph [58] generates a collection of believed associations, the number of which is a coarse metric for the algorithm's success. We report the average percentage of points associated GrassGraph as one of the metrics of its performance. GrassGraph also estimates the transformation matrix between sets of landmarks, the matrix required to align them. We report the median Frobinus norm between the transformation matrix recovered by GrassGraph and the matrix representing the true transformation, which is the primary measure provided by [58]. We also report the average angle between the transformation estimated by GrassGraph [58] and the true transformation. This angle should ideally be zero and provides a more intuitive indication of the success of the algorithm than the Frobinus norm. The median Frobenius norm is used instead of average Frobenius norm, as average results are disrupted by occasional very poor estimated transformations. Likewise, it is occasionally impossible to decompose poor transformation estimates provided by GrassGraph [58] to provide a rotation estimate in which case we assume the maximum of 180 degree.

4.2.1 Effect of of Outliers and Noise

To gauge the effect of adding outliers or noise to the sets of landmarks associated by GrassGraph [58] we consider three metrics: the number of landmark points GrassGraph associated, the Frobenius norm between its estimated transformation and the true one, and the angular error of the alignment GrassGraph generates. The effects of both noise and outliers on these metrics can be seen in figures 4.10, 4.11, and 4.12. It can be seen in all three measures that the introduction of noise or outliers has a sharp negative impact on the performance of GrassGraph [58], however the impact of even 5% outliers is considerably more drastic than low levels of noise. The impact of noise is also overall noticeably more gradual. Numerical figures for these three plots can be found in the first row and column of Tables A.1, A.2, and A.3, as well as figures for varying combinations of the two effects.

In terms of the number of associations found, only 5% of landmarks have to be replaced with outliers before only 3% of landmarks are associated by GrassGraph [58]. This is in contrast to requiring noise with a standard deviation of many meters before the same effect is seen. (see Table A.1 for details) A similar effect can be seen on the median Frobenius norm which jumps to 35+ in the presence of 5% outliers while requiring noise with a large standard deviation to achieve the same result. (first row/column of Table A.2) Finally we also see the same jump with angular error, with the average becoming nearly 120 degrees with 5% outliers. All of this points to a a very high sensitivity to outliers by GrassGraph [58]. A sharp increase in registration error in the presence of outliers was also observed by [58], particularly in 3D datasets, though their measure of Frobenius norm registration error is not directly comparable due to pointcloud scale.

We also see a drop in the performance of GrassGraph [58] when positional noise is added, however it is not as significant. Noise with a standard deviation of 15 centimeters reduced the number of associated landmarks to around 30% rather than 3% in the case of 5% outliers, increased the Frobenius norm to only 5, and introduced an average registration angular error of around 60 degrees instead of 120. The latter is still quite large for a registration technique in practical terms, but is common when GrassGraph [58] finds associations for less than half the landmarks (yielding a poor registration solution). The analysis given by [58] also notes a gradual decrease in the performance of GrassGraph [58] with increased noise, although they generate noise by re-sampling on a circle rather than vary displacement randomly, as we do to better match real-world conditions.

4.2.2 Closer Look at Outliers

Due to the much more abrupt decrease in performance caused by the presence of outliers, we provide a more detailed examination with fewer outliers present. To do so we introduce outliers one at a time and collect the same measures as before. However, in Figure 4.15 we see that the introduction of even one outlier has an immediate effect, reducing the number of landmarks associated to less than 10%. (Table A.4) This also yields a rise in the alignment's Frobenius norm error to 15, roughly half that observed with 5% outliers. (Figure 4.16, column one of Table A.5) Finally, the effect of one outlier on the angular alignment error is also immediate, jumping to the full value of nearly 120 degrees seen with a larger number of outliers. (Figure 4.17 and Table A.6) Only the Frobenius norm shows any kind of gradual transition with additional outliers. This highlights a severe intolerance of the GrassGraph [58] method to the presence of outliers, perhaps because in many parts of it's formulation there is an assumption that point sets being aligned are fundamentally the same albeit translated and rotated.

4.2.3 Sensitivity Analysis on GrassGraph Dataset

As a baseline of comparison, we preform the same sensitivity analysis of GrassGraph [58] to outliers and positional noise but with the SHREC dataset [42] used by [58] as a source of test 3D pointclouds. The SHREC dataset as used and provided by [58] contains sets of 250 points, sampled on the surface of various objects. Here we sub-sample these to 120 points for proper parity with our other results, although we have performed the same tests with the full 250 points and found the performance to be the same within a margin of error.

Looking at the association performance of GrassGraph [58] from the standpoint of the number of points it associates in each SHREC [42] object, we can see by comparing Figure 4.18 to Figure 4.10 that while GrassGraph suffered a very similar drop in performance in the presence of 5% outliers, it is slightly less deep when testing using SHREC [42] pointclouds. At 5% outliers roughly 13% of points were associated, versus around 3% when testing using clustered landmarks (figures can be found in Tables A.7 and A.1). The effect of noise on the number of SHREC [42] points/clustered landmarks is very similar, though with larger amounts of noise GrassGraph [58] performs better with SHREC [42] points. At the high end of the noise standard deviations considered roughly 12% of SHREC [42] points were associated versus 6% of clustered landmarks. This is notable, though firm conclusions with such small values are difficult due to uncertainty in estimating the maximum radius of SHREC [42] pointclouds and thus appropriately scaling the noise standard deviation appropriately.

When considering the Frobenius norm error between the alignment matrix estimated by GrassGraph [58] and the true alignment matrix, again similar behavior but better alignment is obtained with SHREC [42] points. With only 5% outliers the Frobenius norm is around 35 with clustered landmarks but only around 5 when testing with SHREC [42] pointsets, and this trend continues with more outliers. A similar situation exists with sensitivity to noise as before, with SHREC [42] pointsets showing some improvement that widens with increased noise. Figures can be found in Tables A.8 and A.2.

The situation with the angular error between true and GrassGraph-estimated alignments is also similar, with the average angular error being significantly lower with SHREC [42] pointsets (90 vs 120 degrees). The same dramatic increase with the presence of outliers is observed in both cases. As before, there is also somewhat better performance observed with SHREC [42] points in the presence of positional noise. Figures can be found in Tables A.9 and A.3.

Overall we see that the association and consequently alignment performance of GrassGraph [58] is noticeably better in many respects when using SHREC [42] pointsets instead of clustered landmarks. This remains even when the number of points in the SHREC dataset is sub-sampled to be consistent with the number of clustered landmarks. It implies that there is something about the shape of the 3D coordinates which is important to the success of GrassGraph [58], occasionally providing a small improvement to robustness to outliers and noise.

4.2.4 Discussion of Sensitivity Analysis

The ultimate goal of this work is to perform place recognition using landmarks extracted from SLAM's recovered structural data, however doing so requires an effective way to associate landmarks between visits. We have found that GrassGraph [58], while apparently well-suited to this task as it is entirely invariant to the transformation between sets of points, is unable to consistently deliver the number of correct associations required. To better understand this challenge, we conduct and present a sensitivity analysis examining what effects (outliers, noise) are responsible for degraded performance of GrassGraph [58] and to what degree.

We have found that even a small number of outliers severely degrades the ability of GrassGraph [58] to associate sets of landmarks. While the initial proposal of the method [58] shows a dramatic drop in performance under the presence of outliers (when specifically working with 3D coordinates), significant impact from single outliers (as we have repeatedly observed) is unexpected. For this reason, this method of directly associating sets of points based on their positions is insufficient for the purposes of place recognition with the landmarks we obtain. GrassGraph [58] is also likely unsuitable in many other real-world applications as well, when any number of outliers are present. Instead, use of GrassGraph [58] requires that the same set of landmarks can always be consistently detected. The effect of positional noise reinforces this conclusion as moderate positional noise produces alignment errors which would likely be difficult to manage. These effects (particularly strong intolerance to the presence of any outliers) remain when testing is repeated on the SHREC [42] dataset with which GrassGraph [58] was proposed, with outliers and noise introduced in the same way and in the same combinations. This would indicate that the intolerance of GrassGraph [58] to outliers is a systemic issue and not dependant on the data in question, whether the landmarks we generate using SLAM data gathered from KITTI [25], or sets of points taken from other datasets.

It is interesting to note that under the same conditions and with the same number of points, GrassGraph [58] does appear to perform slightly but consistently better on the specific 3D dataset it was proposed with [42] than on synthetic sets of points derived from our clustered landmarks. As the only remaining difference is the shape of the pointcloud before outliers, noise, and random transformations are added, this would suggest that the success of GrassGraph [58] is somewhat dependant on the shape of the pointcloud it is associating. This difference in distribution may come from the different domains that SHREC [42] and KITTI [25] are developed for. The landmarks obtained from clustering come from an "inside looking out" task (visual SLAM, during exploration of the environment) while the points on the surface of objects in SHREC [42] come from an "outside looking in" task, namely scanning of objects. This does yield a difference in the general shape and distribution of the 3D pointclouds, for example because household objects more compact compared to long and branching city streets. Unfortunately though, from a practical standpoint, there is limited control in most domains over the shape of the points being associated.

4.3 Experimental Conclusions

We have found that generating landmarks from visual slam points is quite challenging to do through clustering. The varying sparsity of the pointclouds makes it hard to find a one-size-fits-all solution that works consistently, even across the same cloud. At the same time, the rich information available through visual systems can help guide the segmentation process. This improves the repeatability of segmenting the pointcloud into landmarks and the added separation can also help more clearly define the boundaries of landmarks. The latter in turn helps to estimate their position. Here we have experienced success improving segmentation results using visual semantic segmentation, reducing the number of outlier landmarks through more repeatable clustering. Semantic segmentation does have the limitation that inbuilt classes should match the domain of the environment. To obtain improved flexibility and further improvements in landmark extraction, it is desirable to continue to explore the range of visual techniques available and broaden the range which can be applied.

It remains an open problem how to handle environments like those in KITTI [25] sequence 02. The environment presented was both indistinct and lacking in small segmentable objects, instead containing only large-scale continuous structures (hedges and roads) where it is unclear how they should be subdivided. This is essentially the worst-case scenario for landmark extraction, as logically there are no reliable ways to segment such structures to create them. State of the art VPR methods also struggle under the same conditions. [57]

The GrassGraph association mechanism proposed by [58], while powerfully affine transformation invariant, is ill-suited for the task of matching landmarks for place recognition. This is primarily owing to its high degree of intolerance to outlier points, combined with its requirement of a consistent total number of points to be associated. The former requires that identical sets of landmarks be extracted during every visit, which isn't practical when different trajectories through the same area can be taken [57]. Visual systems capture very similar but not identical data in such a scenario making identical extraction difficult. The requirement for a consistent total also prevents matching of incomplete landmark sets as introduction/removal of additional landmarks becomes required, which creates outliers. Overall GrassGraph [58] is strongly suited to matching of point sets where it can be guaranteed that there exists a 1:1 correspondence for every member point. The realities of place recognition, from freedom of movement to inevitably-imperfect detection, dictate that while there may be substantial similarity between landmark sets there is unlikely to be such a perfect correspondence. Instead methods of association are needed which can exploit the correspondences which exist.

4.4 Future Directions

Clustering of 3D SLAM feature points is challenging due to their sparsity but also non-uniform density. They tend to be detected on objects and textured surfaces, but between such regions can be large empty expanses. The spacing between points can also be similar to the separation between objects or structural elements in the environment. Plain clustering on the points' spatial coordinates can therefore be difficult to tune for reliable and repeatable results, with few outlier clusters between visits. This is particularly true when scale parameters must be selected for clustering. We achieved success in partially remediating this problem by the introduction of semantic labels to the points, providing artificial separation between nearby clusters of points with different labels. Clustering whole objects can help to pin the expected scale of each cluster, though it may not help with spacing between clusters of the same class.

In examining the clusters we generate however, we were not able to sufficiently address the problem of repeatability to enable association of set of of landmarks by GrassGraph [58] and subsequent place recognition. While the positional noise was fairly large compared to the scale of the landmarks considered, it could be improved through better clustering and estimation of a representative cluster/object center. Outliers, those landmarks without a match during revisit, proved to be a far more significant challenge. The clustering methods considered generate a quite variable number of outliers, and under testing the presence of any outliers at all had a significant negative impact on GrassGraph's ability to find associations and recover an accurate alignment between sets of landmarks.

4.4.1 Addressing Outliers With Spatial Descriptors

To address the problem of outlier landmarks which have no correspondence between visits and frustrate matching, we propose the use of landmark descriptors. With a description of each landmark based on its local neighborhood or its intrinsic properties, landmarks with sufficiently different descriptions (outliers) can be rejected and putative associations between landmarks in different sets can be obtained. With respect to local neighborhoods, certain geometric properties can be expected to remain consistent across detections of a set of landmarks like the distances and relative angles between them. As in the case of [89] when working with LiDAR-derived landmarks, these descriptors can be used to find initial associations and from there bootstrap the association of all landmarks, rather than rely on an approach like GrassGraph [58] which seeks to associate all landmarks simultaneously using nothing but their arrangement.

Histogram descriptors for each landmark are proposed by [89] and take into account the distance from each landmark to its neighbors, and provide initial correspondences which are then iteratively refined using RANSAC [22]. A similar method is proposed by [62] when matching 2D overhead maps of trees, wherein polygons are drawn connecting nearby trees and the angles of each polygon are used to produce a local descriptor to eventually total association. Semantic object instances are extracted by [46] from the fusion of dense RGBD scan data with with semantic segmentation masks, but once extracted the matching descriptor for each object instance is a collection of random walks from that object to others nearby. The methods proposed by [89], [62], and [46] are readily applicable to collections of landmarks beyond those derived from dense RGBD pointclouds. All of these methods can be generalized to derive landmark descriptors for landmarks from any source, including SLAM-estimated structure. They can thereby serve to reject outliers which have no similar corresponding landmark, as well as provide putative associations.

There are also various methods which integrate of 2D spatial relationships between local features, which may map to 3D landmarks. Many build structures between local features, and the resulting graphs could be described using descriptors proposed by [46], [89], or [62]. Topological structures (essentially small graphs) are built between local features by [78]. Beyond description, a similar matching of topological structures between landmarks may also enable rejection of single outliers. Comparing angles and lengths in small graphs between local features is used by [23] as verification technique, and these could also be used to describe a landmark and its neighbors. Descriptors could also be applied to the graph generated by [35], who project 3D points to a 2D surface to construct an object graph before normalizing lengths to avoid viewpoint effects. 3D coordinates maintain the same length and angle relationships regardless of viewpoint, and so if substituted into these methods the resulting graph descriptors can include these physical quantities.

4.4.2 Addressing Outliers With Visual Descriptors

It is possible that outliers may disrupt local geometric relationships and so spatial descriptors may not be sufficient. In that case, descriptors derived from visual depictions of each landmark are an alternative. Such descriptors could also serve to reject outlier landmarks which have no similar match, and provide possible matches to drive association. A coarse form limited to only a few object classes would be to apply semantic labels to landmarks, derived from a neural networks which provide object detection or pixel-wise semantic labeling. The method proposed by [46] makes use of semantic segmentation masks, while [35] and [2] make use of 2D bounding boxes to identify candidate object landmarks. Semantic labels are used to distinguish landmarks during association by [89] and semantic labels are present in the random-walk descriptors proposed by [46]. Semantic labels could provide a simple filter during the final nearest-neighbor based association stage performed by [58].

As a more general approach without fixed object classes, it is likely possible to leverage the global descriptors already being developed in VPR and other areas of image retrieval which provide good illumination robustness. For maximum possible viewpoint invariance this should be constrained to describing image regions depicting each landmark specifically, which can be estimated through the projection of 3D data. Description vectors are to be inferenced for image patches which depict each landmark, with networks pre-trained to generate similar descriptors given patches depicting similar objects. In some ways this simplifies use of these descriptors to an object recognition or matching problem. Describing landmark-depicting patches is an approach taken by Patch-NetVLAD [26], which sought to build on NetVLAD [3] by describing larger image regions than is typical for a local feature method. Practically any global descriptor could be used to describe image regions depicting detected landmarks. CNNs [12] [13], autoencoders [52] [53], and even handcrafted descriptors [80] are potential candidates.

4.4.3 Visual Aids When Forming Landmarks

Just as we found semantic labeling provided helpful distinction when clustering SLAM points, visual techniques can also provide labeling that aids the segmentation of pointclouds into landmarks. A simple approach would be to collect 2D semantic-segmentation regions and correlate them to recovered underlying 3D geometry. A similar approach is taken by [1] using 2D bounding boxes from an object detection network. This has the advantage of an external source of object instances, however also poses the challenge of correctly merging detections from multiple views. As with most object detection and image segmentation methods, it also generally works for a preselected set of semantic classes. The method Segment Anything proposed by [36] provides a more flexible segmentation framework able to segment any number or type of region in the environment. Many regions are semi-randomly extracted for objects and components of objects, and some additional filtering and description will probably be required to make effective use of them. Similar in theme is earlier work proposed by [43] or [45] in salient instance segmentation, which focuses on segmenting entire salient object instances even when they may not have occurred in the network training set. The focus on segmentation of whole objects may lend some additional stability, preventing over-segmentation where many sub-components of objects are independently segmented by Segment Anything [36].

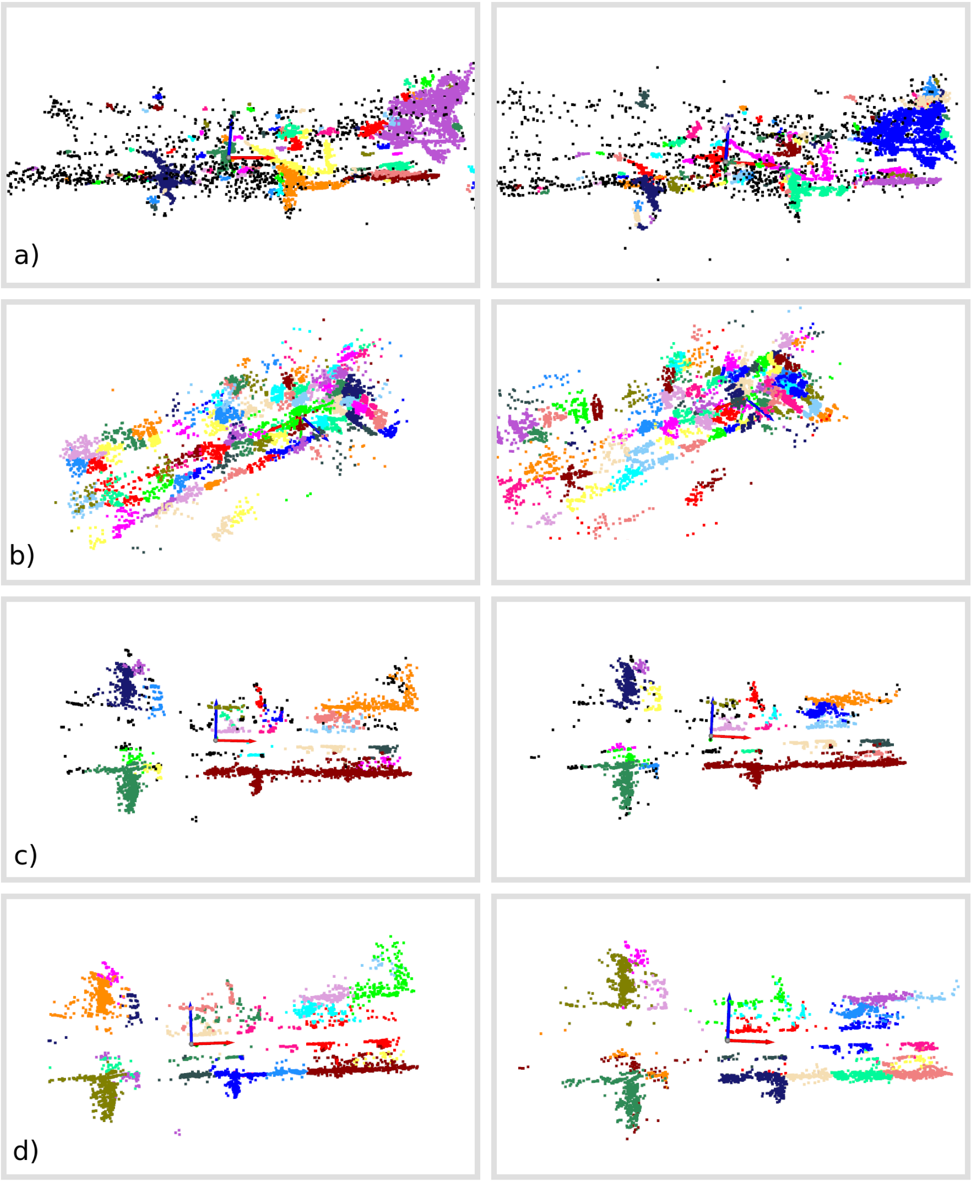

Figure 4.1: A side-by-side comparison of the four clustering approaches characterized. Left is a visit, right a later revisit. Methods are unassisted DBSCAN [20] (a), unassisted BIRCH [83] (b), semantically-assisted DBSCAN (c), and semantically-assisted BITCH (d). The latter two use semantic labeling to achieve better clustering through improved separation.

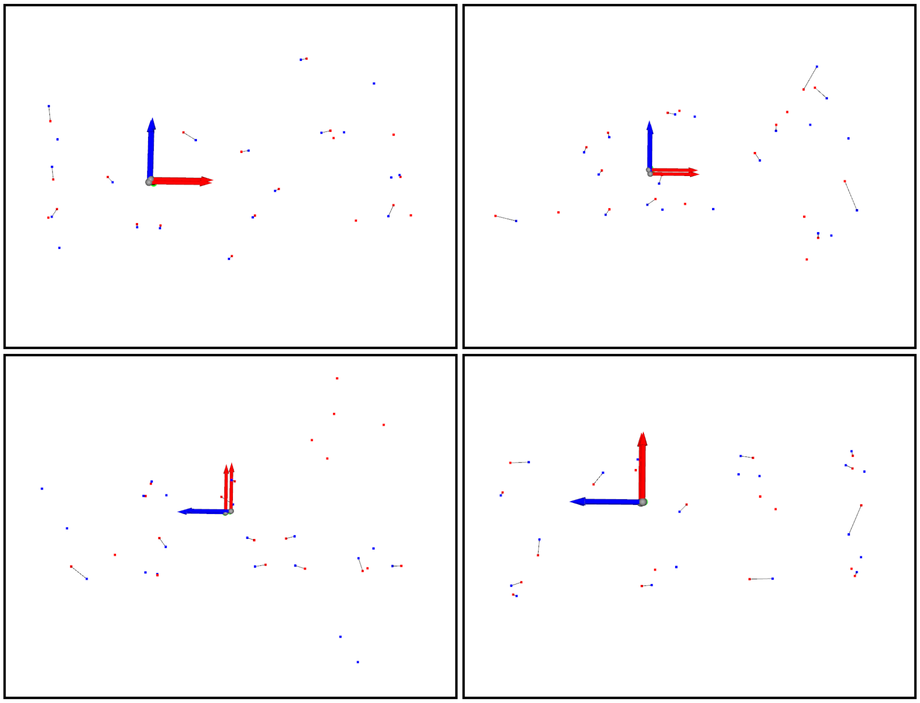

Figure 4.2: An illustration of the heuristic used to determine valid landmark correspondences between visit and revisit, in the absence of ground-truth landmark detections. Shown from overhead are landmarks points (blue and red) extracted from a street scene in Sequence 00. A line is drawn between them if they are mutual nearest neighbors. Unpaired landmarks are considered outliers, while distance between pairs is assumed to be noise in estimated position. Ground truth alignment between visit and revisit coordinates is required (from GPS or similar).

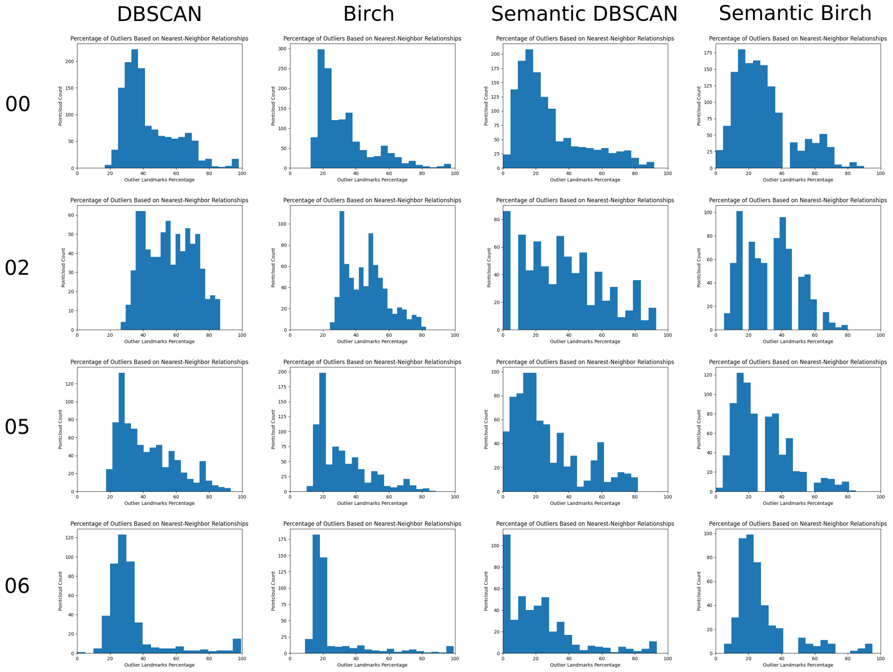

Figure 4.3: Histograms for each method and sequence, showing the proportion of frames that have a particular number of outliers (fewer outliers being better). There is typically a peak near 20% outliers followed by a long tail which brings the average to the range of 25-56% for non-semantic approaches depending on the sequence and method. BIRCH [83] clustering typically has around 10% fewer outliers than DBSCAN [20]. The high end of this range of outliers is a formidable challenge, however the introduction of semantic assistance does bring this range down to 22-37%. DBSCAN and BIRCH also become much closer together in performance when assisted, indicating that the specific clustering method becomes less important. This can be seen in the histograms as the distributions for semantic methods being overall shifted left. Imagery from sequence 02 stands out as being particularly difficult to cluster into landmarks due to its indistinct vegetation. These measurements were taken **before** the total number of detected landmarks was corrected through random addition/removal.

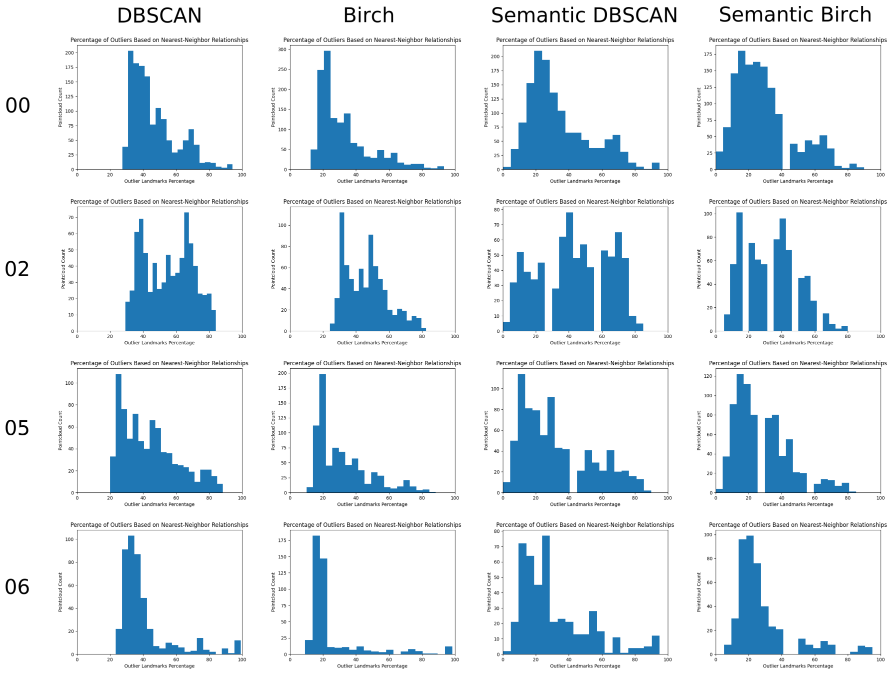

Figure 4.4: Histograms for each method and sequence, showing the proportion of frames that have a particular number of outliers (fewer outliers being better). These plots are supplementary to Figure 4.3, where these plots include the effect of outliers created to reach an expected total number of clustered landmarks. This is an issue for DBSCAN [20] much more than BIRCH[83], as BIRCH enforces a fixed number of clusters and so typically does not need total correction. These measurements were taken **after** the total number of detected landmarks was corrected through random addition/removal.

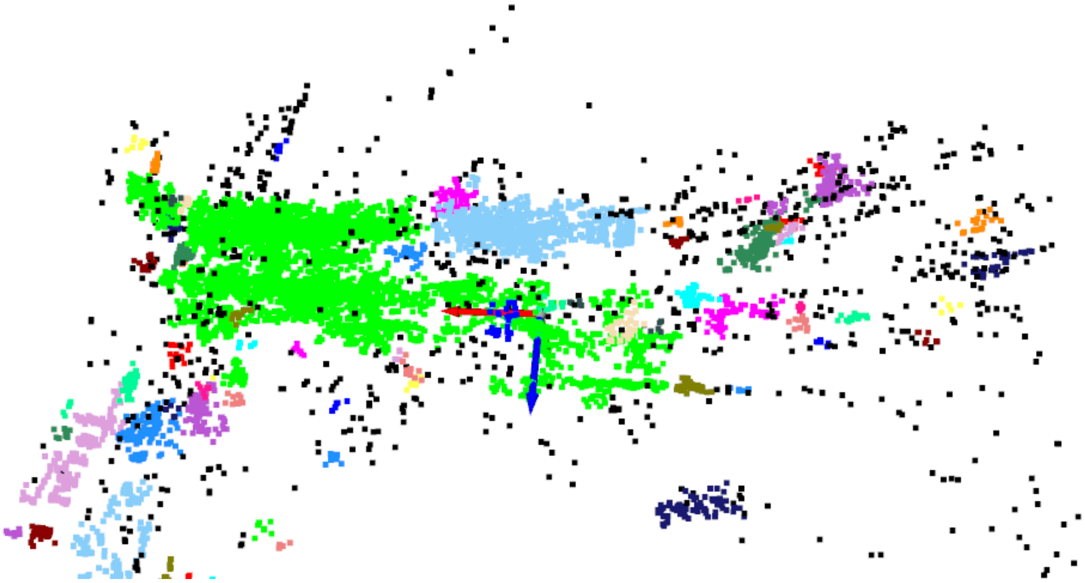

Figure 4.5: If DBSCAN [20] is tuned with two large a growth range, large clusters can result as initial seed clusters grow out of control. Here a SLAM-derived pointcloud has developed such clusters. It can be a challenge to find a balance though to avoid many small random clusters.

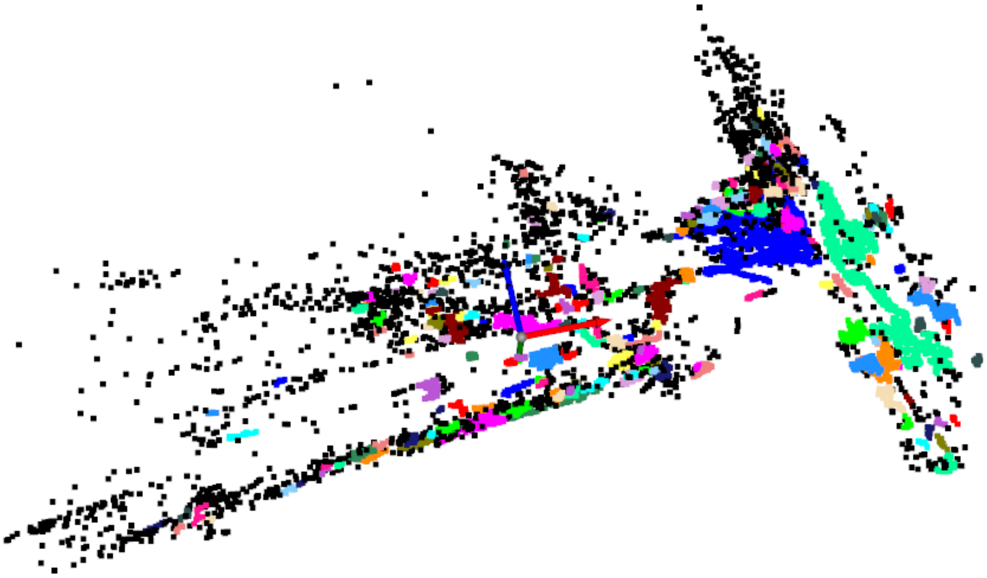

Figure 4.6: If DBSCAN [20] is tuned with two small a growth range, many small random clusters will result. Here a SLAM-derived pointcloud has developed such clusters. It can be a challenge to find a balance though to avoid many large all-consuming clusters.

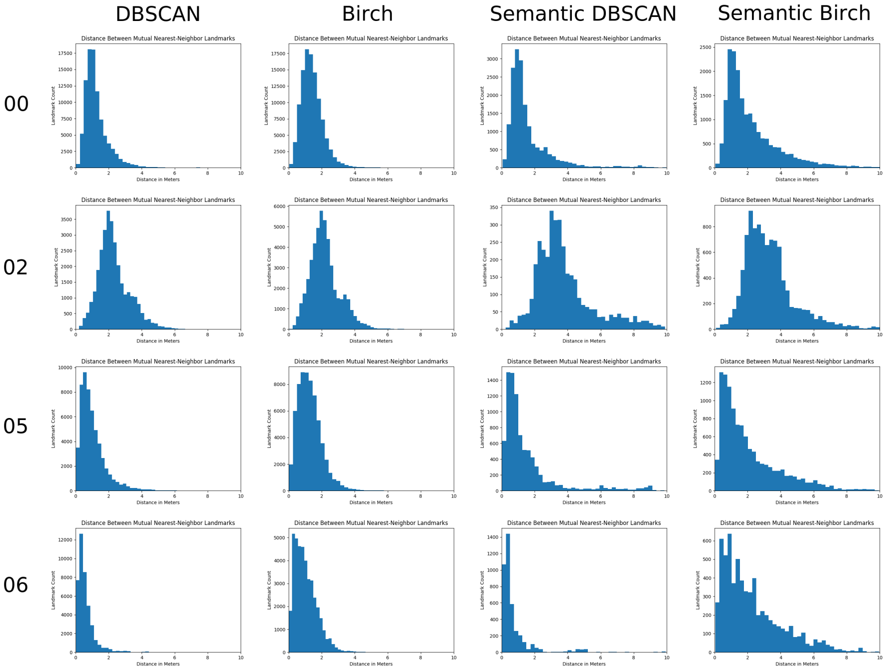

Figure 4.7: Histograms for each method and sequence, showing the distribution of detection-redetection pairs with a particular inter-landmark distance between them. (meters) Smaller distance is better. For clustering methods without semantic augmentation, the standard deviations for positional error (essentially the average distance between pairs) are quite similar at around 1.5 meters. This does roughly double for the semantic methods though, most likely because the whole-object clusters semantic labeling encourages tend to be much larger overall. Sequence 02 of KITTI [25] stands out as being particularly difficult to cluster into landmarks well. These measurements were taken before the total number of detected landmarks was corrected through random addition/removal.

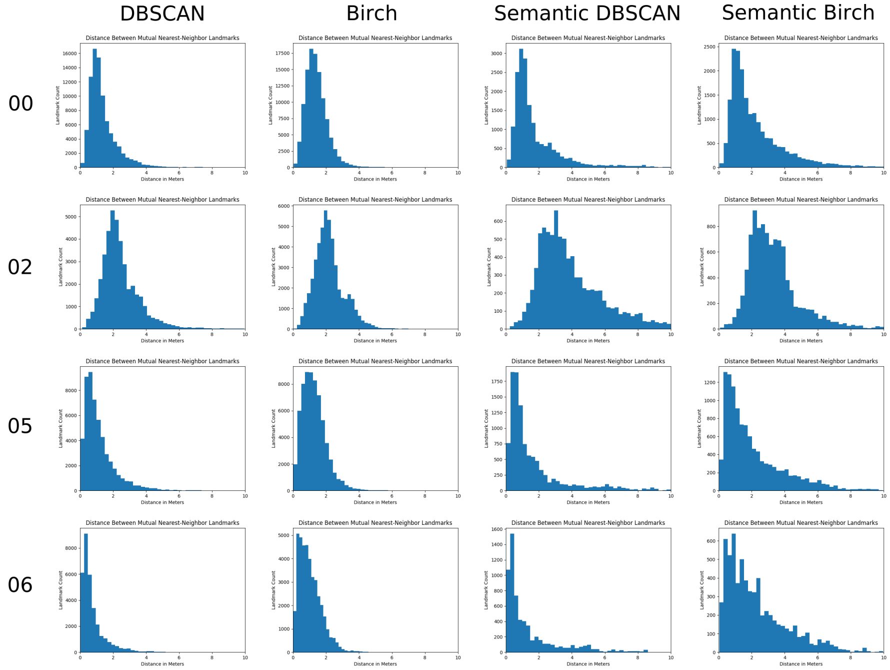

Figure 4.8: Histograms for each method and sequence, showing the distribution of detection-redetection pairs with a particular inter-landmark distance between them. (meters) Smaller distance is better. This set of histograms is supplementary to Figure 4.7, where here we include the effect of correcting the total number of landmarks. This has little to no effect on the positional noise, though. Sequence 02 of KITTI [25] stands out as being particularly difficult to cluster into landmarks well. These measurements were taken after the total number of detected landmarks was corrected through random addition/removal.

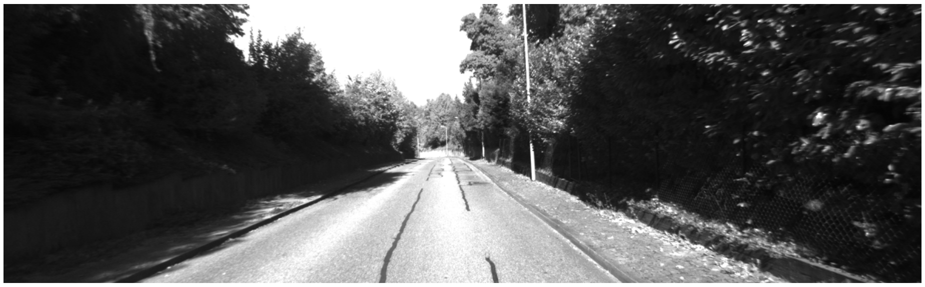

Figure 4.9: Frame 1820 from sequence 02 of KITTI [25], one of the earliest positions that is later revisited. This frame is typical of areas with revisits in 02. Pictured is a narrow street dense, featureless, but textured vegetation which obscures buildings and other man-made structures, as well as a shortage of objects which might stand out as distinctive landmarks. No parked (or mobile) vehicles are present. The continuous but indistinct nature of the vegetation and the lack of any other visible buildings, objects, or other landmarks makes this sequence's environment particularly hard.

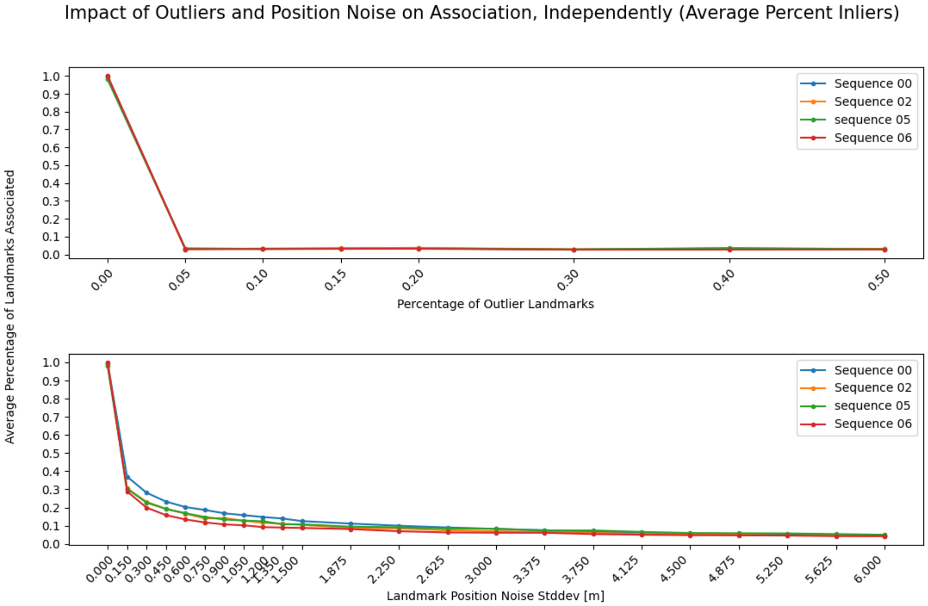

Figure 4.10: The percentage of landmarks associated by GrassGraph [58], measured independently given the percentage of outlier landmarks or positional noise (associating more landmarks is generally better). Noise standard deviation is in meters. Of significant note is that the presence of any outliers produces a sharp drop, where as the effect of noise is more gradual.

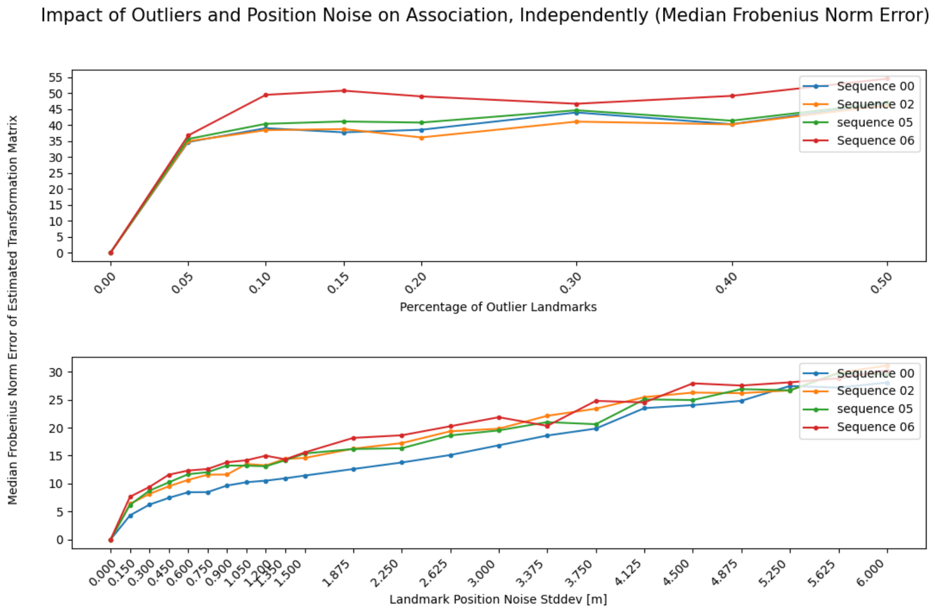

Figure 4.11: The median error (computed using the Frobenius norm) between the true registration matrix for aligning sets of landmarks and that estimated by GrassGraph [58], measured independently given outlier landmarks or positional noise. Less error is better. Noise standard deviation is in meters. Of significant note is that the presence of any outliers produces a sharp rise in error, where as the effect of noise is more gradual.

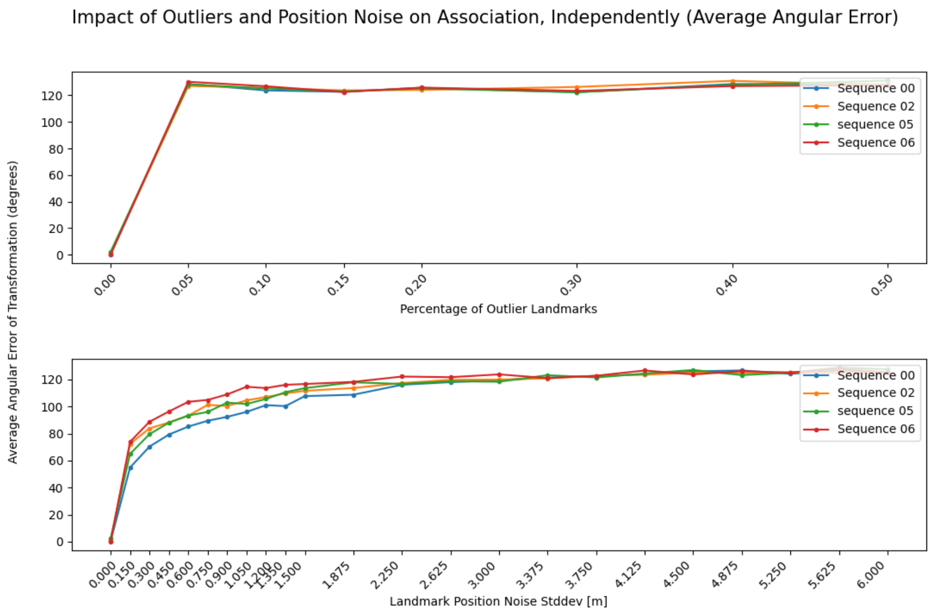

Figure 4.12: The angular error between true registration matrix for aligning sets of landmarks and that estimated by GrassGraph [58], measured independently given outlier landmarks or positional noise. Less error is better. Noise standard deviation is in meters. Of significant note is that the presence of any outliers produces a sharp rise in error, where as the effect of noise is more gradual.

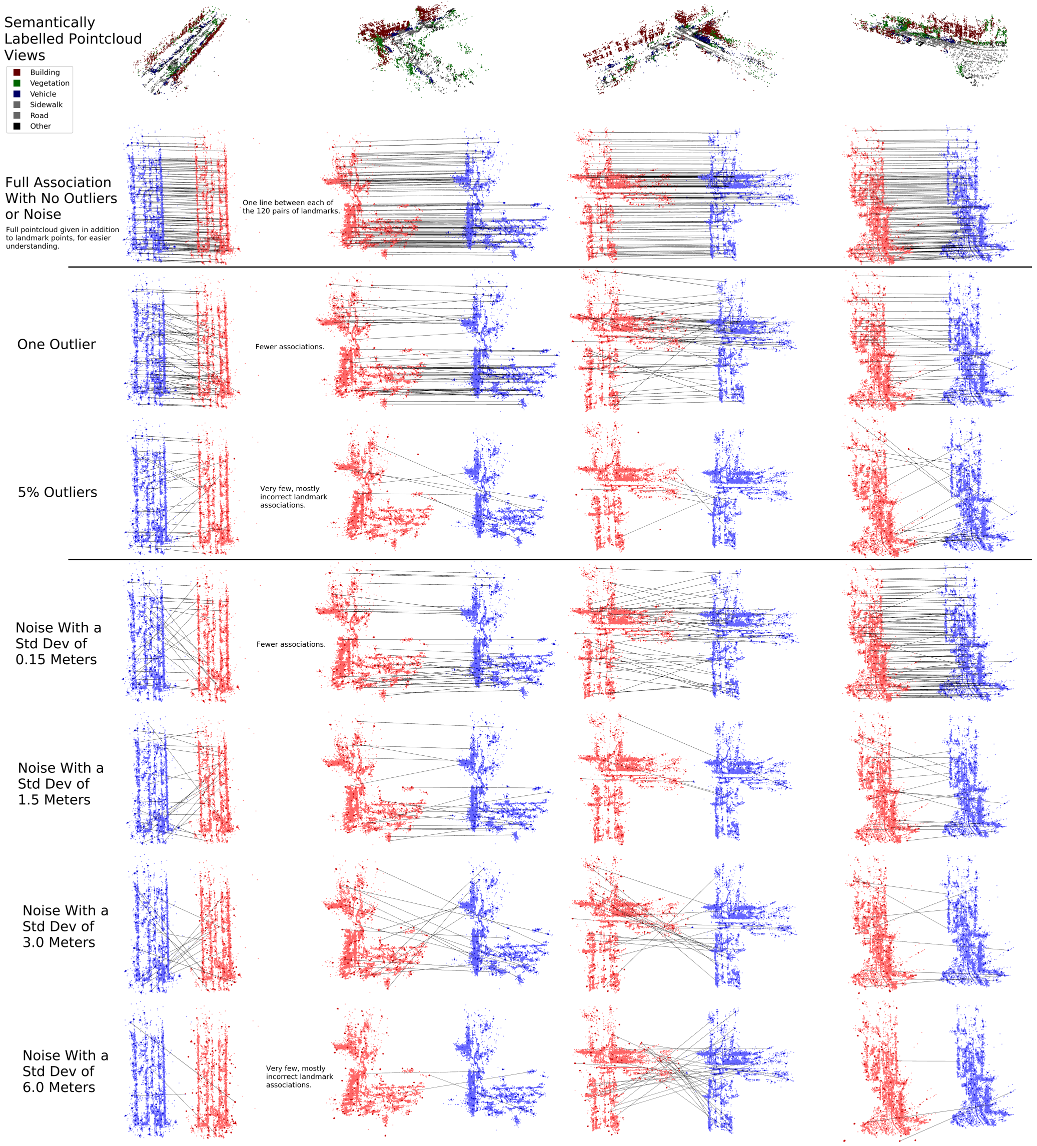

Figure 4.13: Shown are associations between landmark points (darker circles) recovered by GrassGraph [58] given various levels of noise and outliers, for four scenes. The full pseudo-pointcloud is also shown (lighter colors) for easier understanding. The presence of outliers GrassGraph [58] quickly reduces the number/correctness of associations. degradation caused by noise is more gradual. An illustration of the quality of the resulting alignment estimates for the first column is given in figure 4.14.

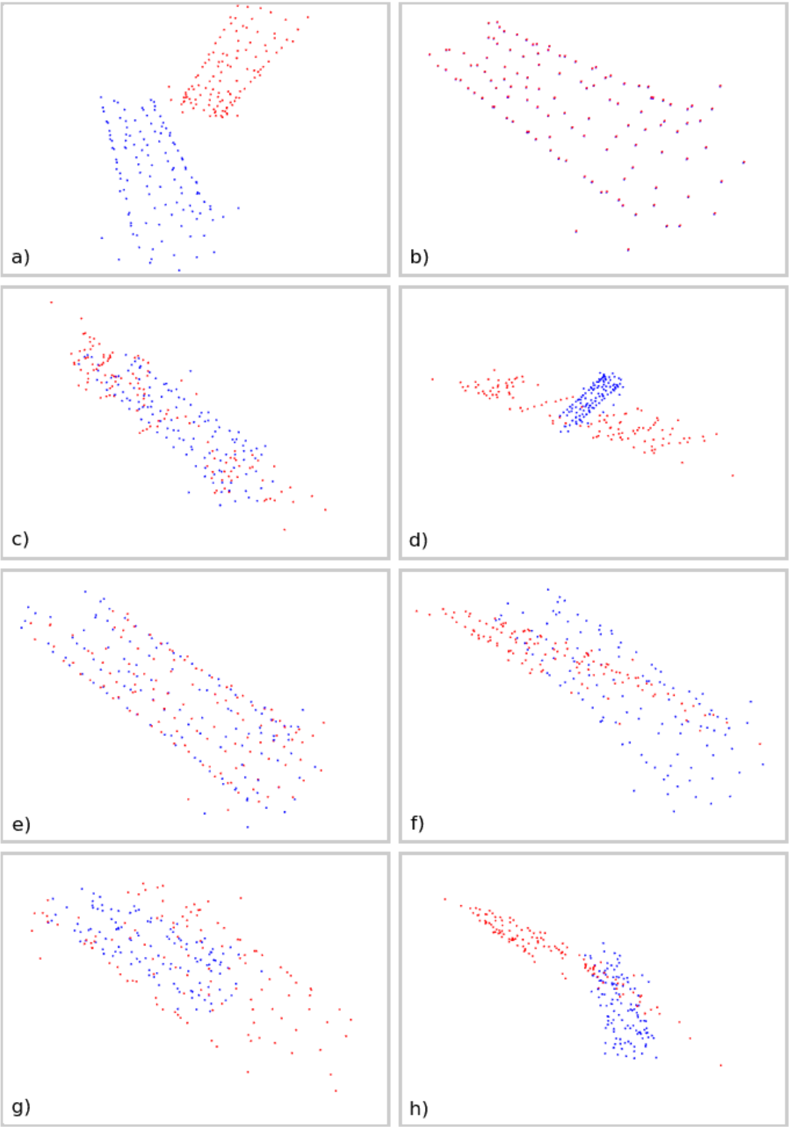

Figure 4.14: An illustration of GrassGraph [58] estimated alignment quality given outliers or noise, corresponding to the first column of Figure 4.13. Image a) shows pre-alignment landmarks. b) shows a perfect alignment by GrassGraph [58] given no outliers or noise. In c) one outlier degrades alignment, while 5% outliers causes a very poor alignment in d). Varying noise standard deviations of 0.15, 1.5, 3.0, and 6.0 meters are shown in e), f), g) and h). Increased noise decreases quality more gradually, maintaining a somewhat correct alignment longer.

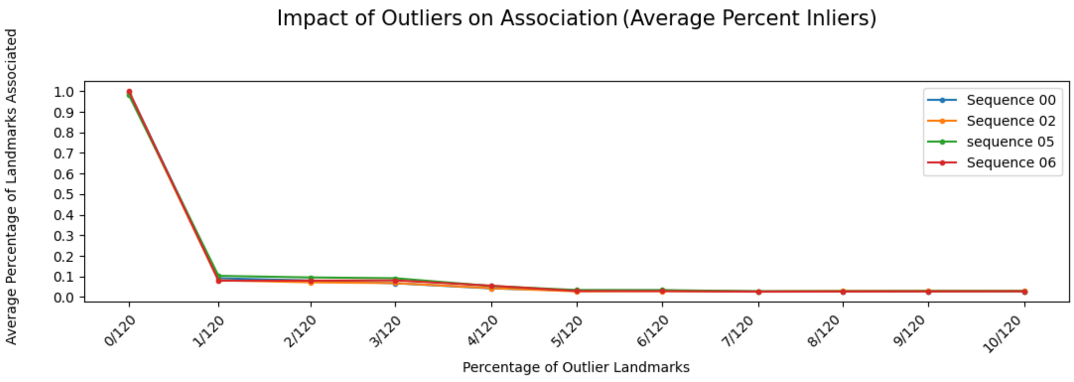

Figure 4.15: The percentage of landmarks associated by GrassGraph [58], measured independently given outlier landmarks or positional noise. Associating more landmarks is generally better. When increasing the number of outliers one at a time having even one noticeably impacts the algorithm's performance. From this it is clear that GrassGraph [58] has a significant sensitivity to outlier points/landmarks.

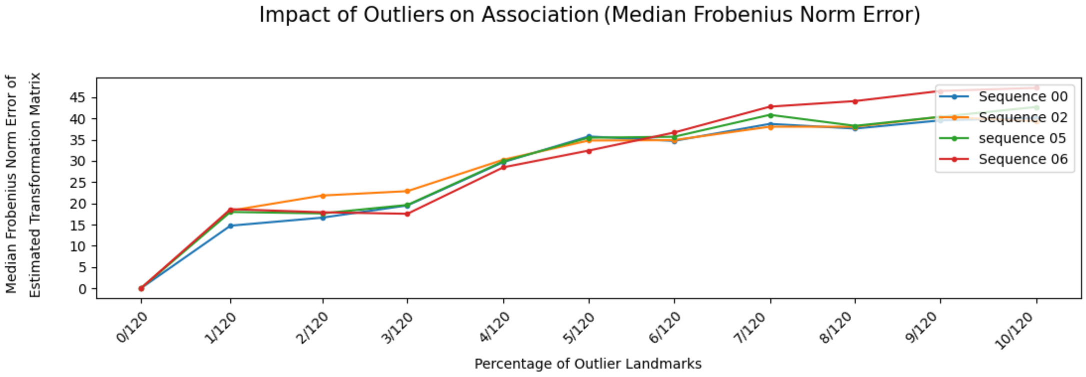

Figure 4.16: The median error (computed using the Frobenius norm) between the true registration matrix for aligning sets of landmarks and that estimated by GrassGraph [58], measured independently given outlier landmarks or positional noise. Less error is better. When increasing the number of outliers one at a time having even one noticeably impacts the algorithm's performance, though not as sharply as in Figures 4.15 or 4.17.

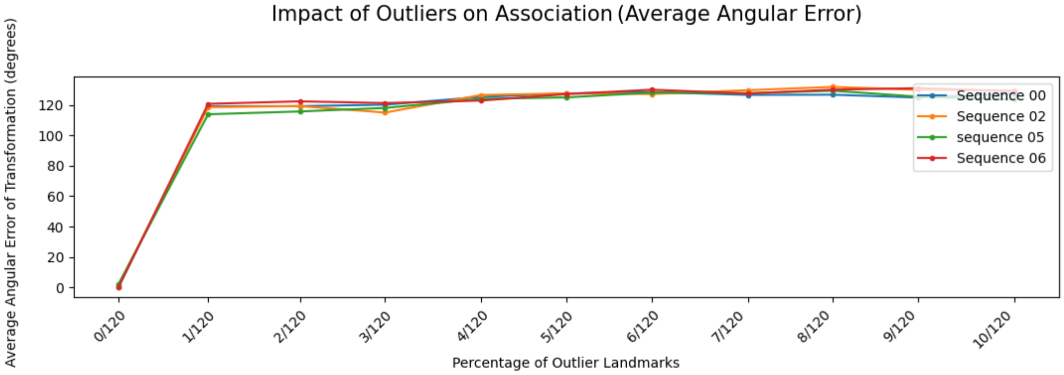

Figure 4.17: The angular error between the resulting orientation of true registration matrix for aligning sets of landmarks and that estimated by GrassGraph [58], measured independently given outlier landmarks or positional noise. Less error is better. When increasing the number of outliers one at a time having even one noticeably impacts the algorithm's performance. From this it is clear that GrassGraph [58] has a significant sensitivity to outlier points/landmarks.

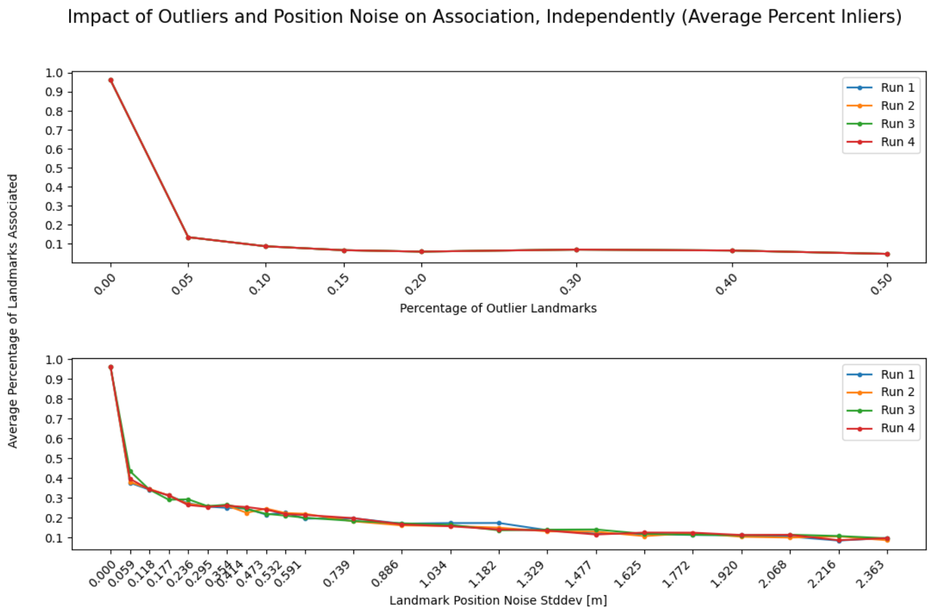

Figure 4.18: The percentage of points associated by GrassGraph [58], measured independently given the percentage of outlier points or positional noise. Associating more landmarks is generally better. The point sets are drawn from the SHREC dataset [42] as used by [58], randomly sub-sampled from 250 to 120 points for parity with our other results. Noise standard deviation is in meters. Each run of noise/outlier combinations used a different random seed for the added noise. Here the same sharp drop in associations seen in Figure 4.10 persists despite the change in dataset. This lends credence to the issue being systemic to GrassGraph [58].

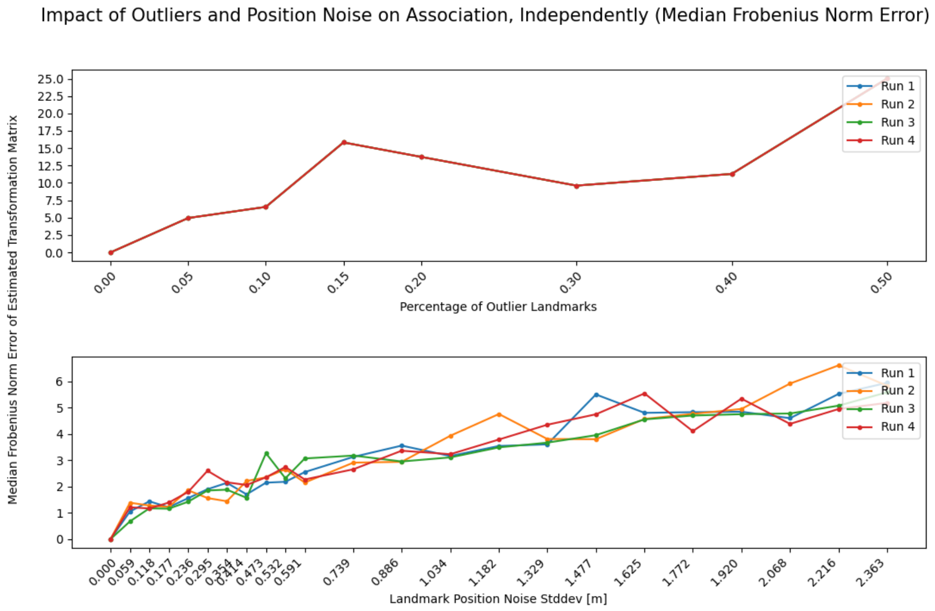

Figure 4.19: The median error (computed using the Frobenius norm) between the true registration matrix for aligning sets of points and that estimated by GrassGraph [58], measured independently given outlier points or positional noise. Less error is better. The point sets are drawn from the SHREC dataset [42] as used by [58], randomly sub-sampled from 250 to 120 points for parity with our other results. Noise standard deviation is in meters. Each run of noise/outlier combinations used a different random seed for the added noise.

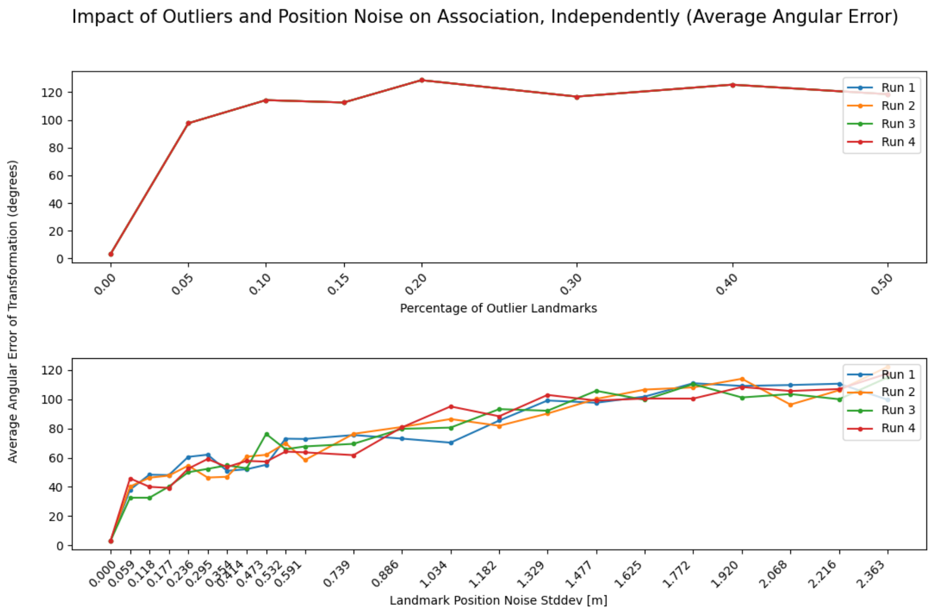

Figure 4.20: The angular error between true registration matrix for aligning sets of points and that estimated by GrassGraph [58], measured independently given outlier points or positional noise. Less error is better. The point sets are drawn from the SHREC dataset [42] as used by [58], randomly sub-sampled from 250 to 120 points for parity with our other results. Noise standard deviation is in meters. Each run of noise/outlier combinations used a different random seed for the added noise. Here the same sharp rise in error seen in Figure 4.12 persists despite the change in dataset. This lends credence to the issue being systemic to GrassGraph [58].

Chapter 5

Conclusion

5.1 Restatement of Problem

Visual Place Recognition is a critical component of any large-scale navigation system, of which SLAM is no exception. Current VPR techniques based on vocabularies of local features and global image descriptors both fall short of providing a versatile solution. The small regions described by local features (Section 2.1.1) result in instability under illumination change, even as discarding their position in the overall image results in better robustness to viewpoint change. Global descriptors (Section 2.1.2) meanwhile are able to leverage high-level structure in an image to overcome illumination and other appearance changes, but are sensitive to viewpoint due to the connection between viewpoint and how these structures are projected onto the 2D image plane. (See Section 2.1.4) As discussed in Section 2.2.4, the 3D structure of the environment recovered as a byproduct of SLAM provides a path to overcoming these conflicting challenges as it is invariant to illumination and remains fixed regardless of the observer. However, there has been very little exploration in the direction of leveraging it for place recognition in visual systems. While [57] found found a large improvement in minimum performance under difficult conditions, they found that the nature of off-the-shelf structural LiDAR scan descriptors were unsuitable for the sparser data generated by camera-based SLAM. Mismatches in coverage, missing regions, and sparsity contributed to an overall lower recall [57], as we discuss at the end of Section 2.2.4.

5.2 Contributions and Results

Our contribution is the exploration of a new approach applying the structure recovered by glsSLAM to visual place recognition. Structural information has invariance to illumination and viewpoint which promises to help address the longstanding issues with existing VPR methods. While the limited existing work [57] was able to validate a meaningful improvement in worst case performance, they encountered difficulty directly applying existing methods from LiDAR. To overcome the difficulty of LiDAR descriptors requiring dense, consistent, and complete coverage of a scene, we instead have persued a landmark-based approach which seeks to associate regions of higher density where more points are available. This has not been without its own set of challenges and which we discuss along with solutions and recommendations for further work in this area. It remains an open problem how best to leverage the spatial information recovered by SLAM for VPR, for which more exploration is warranted.

To locate regions of higher density for use as landmark candidates, our approach has been clustering-driven. We found that clustering using density [20], while the natural choice, produced noticeably less stable results than tree-based partitioning [83] (Section 4.1.3). This is partially due to the uneven density of SLAM pointclouds presenting a problem with parameter selection, but the lack of repeatability in the distribution of points in the underlying pointcloud also makes the density-based clustering solution itself unstable over time. The points detected and tracked by SLAM are of a random distribution owing to the underlying visual features of objects in the environment and their textures, and may or may not be re-detected essentially at random due to various visual effects. It is for this reason we focused on clusters of features rather than particular individual points, and while clustering using the tree-based [83] yielded more stable results, purely spatial clustering of SLAM points remains very difficult. To remedy this we employed semantic labeling of the points from visual pixel-wise semantic labels, such that artificial separation could be introduced during clustering (Section 4.1.2). This produces more-isolated islands of density and helps to cluster structures which may be sparsely populated with points but which are physically close. For structures in the environment which are continuous for long distances (e.g. hedges) it remains an open question if or how these can be converted to landmarks (Section 4.1.1).

The ultimate result of clustering instability is not only inaccuracy in landmark position, but also the production of single landmarks which will not be detected again anywhere nearby. These outlier landmarks were more common with density-based clustering [20] which tends to produce less stable results, however it also produces another challenge. A target number of clusters can be achieved by [83], but this is less feasible with approaches like [20]. When the particular association method used requires equal landmarks in both sets, as [58] does, this means some must be randomly removed or added pre-association to reach the necessary total all sets of landmarks must have (Section 3.3.1). This adding/removing inherently creates a significant number of additional outliers (Section 4.1.3). In the best case, using clustering based on [83] and with semantic assistance (which noticeably reduced outliers), we found that it was still impractical to obtain enough reliable associations via [58] to drive place recognition matching. In examination of the sensitivity of [58] to various effects (Section 4.2.4), we found that it was much more sensitive to outliers than expected despite being initially presented on sets with up to 50% outliers. The presence of even one outlier has a significant effect on the ability to recover associations (Section 4.2.2). There was a modest but noticeable improvement in this regard (and in general) when testing sensitivity on the object pointclouds that [58] was demonstrated on [42] versus the landmarks we generate (Section 4.2.3). This suggests that there is some sensitivity to the type of task which generated the sets of points being associated via their shape, where the two represented here are ”outside-looking” (camera navigation) and ”inside-looking” (object scanning). What is needed in future work is a method of association more robust to outliers (Sections 4.4.1 and 4.4.2). Also needed are methods of producing landmarks which overcome the inconsistency in SLAM pointcloud generation (Section 4.4.3).

5.3 Future Work

With the most difficult challenge being outliers which prevent successful association, future work should employ landmark-level descriptors to improve their ability to associate landmarks while rejecting outliers. We propose the use of various graph descriptors which can be adapted from landmark-based LiDAR place recognition. We also propose the use of various methods from VPR to generate graphs to which these descriptors can be applied. These attempt to incorporate 2D spatial relationships into local feature methods but could be adapted to use 3D relationships instead (discussed in Section 4.4.1). Parallel to this we also propose reuse of various visual descriptors to describe landmarks based on their appearance (discussed in Section 4.4.2), mostly drawing from global VPR descriptors for their illumination invariance. To best address the issue of viewpoint variation this should be constrained to image regions depicting each landmark, which essentially makes this an object-oriented task. Semantic labels can also be used to provide putative associations and reject outliers, as demonstrated by [89].

Upstream improvement to the formation of landmarks could also have a significant impact on various challenges, but is itself a challenging task when using sparse and unevenly-sampled SLAM-tracked points. To aid segmentation or clustering of landmarks there are various image segmentation methods from visual work which can provide additional features and guidance (discussed in Section 4.4.3). We already make some use of semantic segmentation, although this could be expanded to more classes. To provide a more generalized solution we also propose the use of various semantic-class-free visual segmentation methods, for example Segment Anything [36].

With invariance to illumination change, due to lack of effect on the underlying physical shape of the environment, it has been shown and can continue to be expected that incorporating structural information into VPR methods provides significantly enhanced robustness. Viewpoint invariance also becomes a more achievable property as physical structure in the environment is invariant to the point from which it is observed. However, successfully leveraging recovered structure from Simultaneous Localization and Mapping remains challenging, with open problems. Here we have explored ways to combine both spatial information and more traditional appearance-based techniques, and demonstrated some improvement in the face of the challenges it poses. More work in this direction is needed to overcome them and we propose many avenues to explore. With continued development we hope to see combinations of traditional VPR and structural approaches become competitive with and surpass SoTA methods in all use cases. Such a form of Visual Place Recognition will provide a significantly more robust contribution to overall SLAM performance in challenging environments, benefiting all users of SLAM systems.

References

[1] Jacqueline Ankenbauer, Kaveh Fathian, and Jonathan P How. View-invariant localization using semantic objects in changing environments. *arXiv preprint arXiv:2209.14426*, 2022.

[2] Jacqueline Ankenbauer, Parker C Lusk, and Jonathan P How. Global localization in unstructured environments using semantic object maps built from various viewpoints. *arXiv preprint arXiv:2303.04658*, 2023.

[3] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5297–5307, 2016.

[4] Relja Arandjelovic and Andrew Zisserman. All about vlad. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1578–1585, 2013.

[5] Bruno Arcanjo, Bruno Ferrarini, Michael Milford, Klaus D McDonald-Maier, and Shoaib Ehsan. An efficient and scalable collection of fly-inspired voting units for visual place recognition in changing environments. *IEEE Robotics and Automation Letters*, 7(2):2527–2534, 2022.

[6] K Somani Arun, Thomas S Huang, and Steven D Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on pattern analysis and machine intelligence*, (5):698–700, 1987.

[7] Matthew Brett and Christoph Gohlke. Transforms3d: Code to convert between various geometric transformations, 2022. Software available from pypi.org.

[8] Xin Cai and Wensheng Yin. Weighted scan context: global descriptor with sparse height feature for loop closure detection. In *2021 International Conference on Computer, Control and Robotics (ICCCR)*, pages 214–219. IEEE, 2021.

[9] Luis G Camara and Libor Přeučil. Spatio-semantic convnet-based visual place recognition. In *2019 European conference on mobile robots (ECMR)*, pages 1–8. IEEE, 2019.

[10] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021.

[11] Daniele Cattaneo, Matteo Vaghi, and Abhinav Valada. Lcdnet: Deep loop closure detection and point cloud registration for lidar slam. *IEEE Transactions on Robotics*, 38(4):2074–2093, 2022.

[12] Zetao Chen, Adam Jacobson, Niko Sünderhauf, Ben Upcroft, Lingqiao Liu, Chunhua Shen, Ian Reid, and Michael Milford. Deep learning features at scale for visual place recognition. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 3223–3230. IEEE, 2017.

[13] 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.

[14] Konrad P Cop, Paulo VK Borges, and Renaud Dubé. Delight: An efficient descriptor for global localisation using lidar intensities. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3653–3660. IEEE, 2018.

[15] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[16] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee, 2005.

[17] Renaud Dubé, Andrei Cramariuc, Daniel Dugas, Juan Nieto, Roland Siegwart, and Cesar Cadena. Segmap: 3d segment mapping using data-driven descriptors. *arXiv preprint arXiv:1804.09557*, 2018.

[18] Renaud Dubé, Mattia G Gollub, Hannes Sommer, Igor Gilitschenski, Roland Siegwart, Cesar Cadena, and Juan Nieto. Incremental-segment-based localization in 3-d point clouds. *IEEE Robotics and Automation Letters*, 3(3):1832–1839, 2018.

[19] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):611–625, 2017.

[20] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.

[21] Bruno Ferrarini, Michael J Milford, Klaus D McDonald-Maier, and Shoaib Ehsan. Binary neural networks for memory-efficient and effective visual place recognition in changing environments. *IEEE Transactions on Robotics*, 2022.

[22] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[23] Peng Gao and Hao Zhang. Long-term place recognition through worst-case graph matching to integrate landmark appearances and spatial relationships. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1070–1076. IEEE, 2020.

[24] Sourav Garg, Tobias Fischer, and Michael Milford. Where is your place, visual place recognition? *arXiv preprint arXiv:2103.06443*, 2021.

[25] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[26] Stephen Hausler, Sourav Garg, Ming Xu, Michael Milford, and Tobias Fischer. Patch-netvlad: Multi-scale fusion of locally-global descriptors for place recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14141–14152, 2021.

[27] 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.

[28] Jie Hu, Linyan Huang, Tianhe Ren, Shengchuan Zhang, Rongrong Ji, and Liujuan Cao. You only segment once: Towards real-time panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17819–17829, 2023.

[29] Chunjie Hua, Yanan Yu, and Zhongmin Wang. Study on feature extraction algorithm of mobile robot vision slam under dynamic illumination. In *AOPC 2019: Optical Sensing and Imaging Technology*, volume 11338, pages 446–451. SPIE, 2019.

[30] Le Hui, Hang Yang, Mingmei Cheng, Jin Xie, and Jian Yang. Pyramid point cloud transformer for large-scale place recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6098–6107, 2021.

[31] Somayeh Hussaini, Michael Milford, and Tobias Fischer. Spiking neural networks for visual place recognition via weighted neuronal assignments. *IEEE Robotics and Automation Letters*, 7(2):4094–4101, 2022.

[32] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3304–3311. IEEE, 2010.

[33] Giseop Kim, Sunwook Choi, and Ayoung Kim. Scan context++: Structural place recognition robust to rotation and lateral variations in urban environments. *IEEE Transactions on Robotics*, 38(3):1856–1874, 2021.

[34] 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.

[35] Jonathan JY Kim, Martin Urschler, Patricia J Riddle, and Jorg S Wicker. Closing the loop: Graph networks to unify semantic objects and visual features for multi-object scenes. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4352–4358. IEEE, 2022.

[36] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

[37] Alexander Kolpakov and Michael Werman. Robust affine feature matching via quadratic assignment on grassmannians. *arXiv preprint arXiv:2303.02698*, 2023.

[38] Jacek Komorowski. Minkloc3d: Point cloud based large-scale place recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1790–1799, 2021.

[39] 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. IEEE, 2021.

[40] Xin Kong, Xuemeng Yang, Guangyao Zhai, Xiangrui Zhao, Xianfang Zeng, Mengmeng Wang, Yong Liu, Wanlong Li, and Feng Wen. Semantic graph based place recognition for 3d point clouds. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8216–8223. IEEE, 2020.

[41] Gert Kootstra, Sjoerd De Jong, and Lambert RB Schomaker. Using local symmetry for landmark selection. In *International Conference on Computer Vision Systems*, pages 94–103. Springer, 2009.

[42] Bo Li, Tobias Schreck, Afzal Godil, Marc Alexa, Tamy Boubekeur, Benjamin Bustos, Jipeng Chen, Mathias Eitz, Takahiko Furuya, Kristian Hildebrand, et al. Shrec'12 track: Sketch-based 3d shape retrieval. *Eurographics 2012 Workshop on 3D Object Retrieval*, 2012.

[43] Guanbin Li, Yuan Xie, Liang Lin, and Yizhou Yu. Instance-level salient object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2386–2395, 2017.

[44] Lin Li, Xin Kong, Xiangrui Zhao, Tianxin Huang, Wanlong Li, Feng Wen, Hongbo Zhang, and Yong Liu. Ssc: Semantic scan context for large-scale place recognition. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2092–2099. IEEE, 2021.

[45] Xiaodan Liang, Liang Lin, Yunchao Wei, Xiaohui Shen, Jianchao Yang, and Shuicheng Yan. Proposal-free network for instance-level object segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2978–2991, 2017.

[46] Yu Liu, Yvan Petillot, David Lane, and Sen Wang. Global localization with object-level semantics and topology. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 4909–4915. IEEE, 2019.

[47] Manuel Lopez-Antequera, Ruben Gomez-Ojeda, Nicolai Petkov, and Javier Gonzalez-Jimenez. Appearance-invariant place recognition by discriminatively training a convolutional neural network. *Pattern Recognition Letters*, 92:89–95, 2017.

[48] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[49] Stephanie Lowry, Niko Sünderhauf, Paul Newman, John J Leonard, David Cox, Peter Corke, and Michael J Milford. Visual place recognition: A survey. *IEEE Transactions on Robotics*, 32(1):1–19, 2015.

[50] Feng Lu, Baifan Chen, Xiang-Dong Zhou, and Dezhen Song. Sta-vpr: Spatio-temporal alignment for visual place recognition. *IEEE Robotics and Automation Letters*, 6(3):4297–4304, 2021.

[51] Carlo Masone and Barbara Caputo. A survey on deep visual place recognition. *IEEE Access*, 9:19516–19547, 2021.

[52] Nate Merrill and Guoquan Huang. Lightweight unsupervised deep loop closure. *arXiv preprint arXiv:1805.07703*, 2018.

[53] Nathaniel Merrill and Guoquan Huang. Calc2. 0: Combining appearance, semantic and geometric information for robust and efficient visual loop closure. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4554–4561. IEEE, 2019.

[54] Michael Milford, Chunhua Shen, Stephanie Lowry, Niko Suenderhauf, Sareh Shirazi, Guosheng Lin, Fayao Liu, Edward Pepperell, Cesar Lerma, Ben Upcroft, et al. Sequence searching with deep-learnt depth for condition-and viewpoint-invariant route-based place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 18–25, 2015.

[55] Michael J Milford and Gordon F Wyeth. Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In *2012 IEEE international conference on robotics and automation*, pages 1643–1649. IEEE, 2012.

[56] Jiawei Mo and Junaed Sattar. Extending monocular visual odometry to stereo camera systems by scale optimization. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6921–6927. IEEE, 2019.

[57] Jiawei Mo and Junaed Sattar. A fast and robust place recognition approach for stereo visual odometry using lidar descriptors. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5893–5900. IEEE, 2020.

[58] Mark Moyou, Anand Rangarajan, John Corring, and Adrian M Peter. A grassmannian graph approach to affine invariant feature matching. *IEEE Transactions on Image Processing*, 29:3374–3387, 2019.

[59] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, 33(5):1255–1262, 2017.

[60] Ana C Murillo and Jana Kosecka. Experiments in place recognition using gist panoramas. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 2196–2203. IEEE, 2009.

[61] Ana Cris Murillo, José Jesús Guerrero, and C Sagues. Surf features for efficient robot localization with omnidirectional images. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 3901–3907. IEEE, 2007.

[62] Guilherme V Nardari, Avraham Cohen, Steven W Chen, Xu Liu, Vaibhav Arcot, Roseli AF Romero, and Vijay Kumar. Place recognition in forests with urquhart tessellations. *IEEE Robotics and Automation Letters*, 6(2):279–286, 2020.

[63] David Nister and Henrik Stewenius. Scalable recognition with a vocabulary tree. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 2161–2168. Ieee, 2006.

[64] Gabriel Nützi, Stephan Weiss, Davide Scaramuzza, and Roland Siegwart. Fusion of imu and vision for absolute scale estimation in monocular slam. *Journal of intelligent & robotic systems*, 61(1-4):287–299, 2011.

[65] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42:145–175, 2001.

[66] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[67] Zobeir Raisi and John Zelek. Text detection & recognition in the wild for robot localization. *arXiv preprint arXiv:2205.08565*, 2022.

[68] 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. Ieee, 2011.

[69] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *2009 IEEE international conference on robotics and automation*, pages 3212–3217. IEEE, 2009.

[70] Pengcheng Shi, Yongjun Zhang, and Jiayuan Li. Lidar-based place recognition for autonomous driving: A survey. *arXiv preprint arXiv:2306.10561*, 2023.

[71] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *Computer Vision, IEEE International Conference on*, volume 3, pages 1470–1470. IEEE Computer Society, 2003.

[72] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. Reasoning with neural tensor networks for knowledge base completion. *Advances in neural information processing systems*, 26, 2013.

[73] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. Particular object retrieval with integral max-pooling of cnn activations. *arXiv preprint arXiv:1511.05879*, 2015.

[74] Mikaela Angelina Uy and Gim Hee Lee. Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4470–4479, 2018.

[75] Rui Wang, Martin Schworer, and Daniel Cremers. Stereo dso: Large-scale direct sparse visual odometry with stereo cameras. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3903–3911, 2017.

[76] Ruotong Wang, Yanqing Shen, Weiliang Zuo, Sanping Zhou, and Nanning Zheng. Transvpr: Transformer-based place recognition with multi-level attention aggregation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13648–13657, 2022.

[77] Yan Xia, Yusheng Xu, Shuang Li, Rui Wang, Juan Du, Daniel Cremers, and Uwe Stilla. Soe-net: A self-attention and orientation encoding network for point cloud based place recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11348–11357, 2021.

[78] Xinyu Ye and Jiayi Ma. Visual place recognition via local affine preserving matching. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12954–12960. IEEE, 2021.

[79] Peng Yin, Shiqi Zhao, Ivan Cisneros, Abulikemu Abuduweili, Guoquan Huang, Micheal Milford, Changliu Liu, Howie Choset, and Sebastian Scherer. General place recognition survey: Towards the real-world autonomy age. *arXiv preprint arXiv:2209.04497*, 2022.

[80] Mubariz Zaffar, Shoaib Ehsan, Michael Milford, and Klaus McDonald-Maier. Cohog: A light-weight, compute-efficient, and training-free visual place recognition technique for changing environments. *IEEE Robotics and Automation Letters*, 5(2):1835–1842, 2020.

[81] Mubariz Zaffar, Sourav Garg, Michael Milford, Julian Kooij, David Flynn, Klaus McDonald-Maier, and Shoaib Ehsan. Vpr-bench: An open-source visual place recognition evaluation framework with quantifiable viewpoint and appearance change. *International Journal of Computer Vision*, 129(7):2136–2174, 2021.

[82] Kaining Zhang, Xingyu Jiang, Xiaoguang Mei, Huabing Zhou, and Jiayi Ma. Motion field consensus with locality preservation: A geometric confirmation strategy for loop closure detection. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 445–451. IEEE, 2021.

[83] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. Birch: an efficient data clustering method for very large databases. *ACM sigmod record*, 25(2):103–114, 1996.

[84] Xiwu Zhang, Lei Wang, and Yan Su. Visual place recognition: A survey from deep learning perspective. *Pattern Recognition*, 113:107760, 2021.

[85] Xiwu Zhang, Lei Wang, Yan Zhao, and Yan Su. Graph-based place recognition in image sequences with cnn features. *Journal of Intelligent & Robotic Systems*, 95:389–403, 2019.

[86] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017.

[87] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 766–782. Springer, 2016.

[88] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018.

[89] Yachen Zhu, Yanyang Ma, Long Chen, Cong Liu, Maosheng Ye, and Lingxi Li. Gosmatch: Graph-of-semantics matching for detecting loop closures in 3d lidar data. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5151–5157. IEEE, 2020.

[90] Kamil Żywanowski, Adam Banaszczyk, Michał R Nowicki, and Jacek Komorowski. Minkloc3d-si: 3d lidar place recognition with sparse convolutions, spherical coordinates, and intensity. *IEEE Robotics and Automation Letters*, 7(2):1079–1086, 2021.

APPENDICES

Appendix A

Full Numerical Results From Sensitivity Analysis

Here we provide the full numerical results for all the combinations of outliers and noise which are tested and presented during the sensitivity analysis in Section 4.2. For all plots showing only the effect of outliers or only the effect of noise, corresponding figures can be found in the first column and row of the appropriate table.

The results of our initial sensitivity analysis across a wide range of outlier and noise parameters is given in Tables A.1, A.2 and A.3. Table A.1 portrays the impact of outliers and noise on the average number of believed associations between landmark points GrassGraph [58] is able to recover, while Table A.2 portrays the effect on the median Frobenius norm error between the alignment matrix estimated by GrassGraph [58] and the true alignment. Table A.3 provides the average angular error in the estimated alignment as an alternative, which is potentially easier to interpret. All three show an impact as the standard deviation of the noise is increased, but they also show a dramatic drop in performance as soon as outliers are introduced. This dramatic drop in performance is further explored in Tables A.4, A.5, and A.6, which repeat the same experiments but while adding single outliers at a time. The clear conclusion is that the presence of any outliers at all has a significant impact on the success of GrassGraph [58].

We also repeat this same sensitivity analysis on the SHREC [42] dataset used by [58] when proposing GrassGraph [58]. This dataset is comprised of pointsets which represent everyday 3D objects, as opposed to the landmarks which we extract from street scenes. Table A.7 explores the impact of outliers and noise on GrassGraph's [58] ability to recover associations between these points, while Table A.8 gives the median Frobenius norm error of the estimated alignment matrices. Table A.9 gives the average angular error of the estimated alignments. These tables show the same overall result, including the same sharp drop as soon as outliers are introduced, however the impact of outliers and noise on GrassGraph's [58] performance is slightly less strong on this dataset. This would suggest that there is some effect of the shape of the points being associated on the robustness of the association. This is likely in part owed to the difference in domain, where KITTI [25] (used for the majority of our testing) depicts long and narrow street scenes intended for outward-looking navigation, while SHREC [42] depicts relatively self-contained objects for more inward-looking identification and object pose recovery.

Sensitivity to Noise vs Outliers: Sequence 00																							
	Noise Standard Deviation [m]																						
% Outliers	0.0	0.15	0.3	0.45	0.6	0.75	0.9	1.1	1.2	1.4	1.5	1.9	2.2	2.6	3.0	3.4	3.8	4.1	4.5	4.9	5.2	5.6	6.0
0.0	100	37	28	23	20	19	17	16	15	14	13	11	10	9.0	8.2	7.5	6.8	6.2	5.7	5.5	5.1	4.9	4.7
5.0	3.2	3.2	3.2	3.2	3.2	3.2	3.1	3.0	3.2	3.0	3.0	2.9	2.9	2.9	2.9	2.9	2.9	3.0	2.9	2.9	3.0	3.0	3.1
10.0	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.0	3.0	3.0	3.0	3.0	2.9	2.9	2.9	2.9	2.9	2.9	2.9
15.0	3.3	3.3	3.3	3.3	3.2	3.3	3.3	3.3	3.3	3.3	3.2	3.2	3.2	3.1	3.1	3.1	3.1	3.0	3.0	3.0	3.0	2.9	2.9
20.0	3.4	3.4	3.5	3.5	3.5	3.4	3.4	3.5	3.4	3.4	3.4	3.4	3.4	3.3	3.2	3.2	3.1	3.1	3.1	3.0	3.0	3.0	3.0
30.0	2.7	2.7	2.7	2.7	2.8	2.7	2.7	2.7	2.8	2.8	2.7	2.8	2.8	3.0	2.9	3.0	3.1	3.1	3.1	3.2	3.2	3.2	3.3
40.0	3.3	3.4	3.5	3.6	3.6	3.6	3.6	3.7	3.7	3.6	3.6	3.7	3.9	4.0	3.8	3.9	3.9	4.0	4.0	3.9	3.8	3.9	3.8
50.0	2.9	2.9	2.9	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.9	2.8	2.9	2.9	3.0	3.1

Sensitivity to Noise vs Outliers: Sequence 02																							
	Noise Standard Deviation [m]																						
% Outliers	0.0	0.15	0.3	0.45	0.6	0.75	0.9	1.1	1.2	1.4	1.5	1.9	2.2	2.6	3.0	3.4	3.8	4.1	4.5	4.9	5.2	5.6	6.0
0.0	99	31	23	19	17	14	14	13	12	11	10	8.7	8.5	7.4	6.9	6.6	6.1	5.5	5.3	5.2	4.6	4.6	4.3
5.0	2.9	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.7	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.9	3.0	2.9	3.0	3.0	3.0	3.0
10.0	3.2	3.2	3.2	3.2	3.3	3.3	3.2	3.2	3.3	3.2	3.3	3.2	3.2	3.2	3.2	3.1	3.1	3.1	3.1	3.0	3.0	3.0	3.0
15.0	3.4	3.5	3.5	3.4	3.5	3.5	3.5	3.4	3.4	3.5	3.4	3.4	3.4	3.4	3.3	3.3	3.3	3.2	3.2	3.2	3.1	3.1	3.1
20.0	3.6	3.6	3.6	3.6	3.7	3.7	3.7	3.7	3.6	3.6	3.6	3.6	3.6	3.6	3.5	3.4	3.4	3.3	3.3	3.2	3.2	3.2	3.2
30.0	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.2	3.2	3.2	3.4	3.2	3.2
40.0	3.3	3.3	3.3	3.3	3.4	3.5	3.3	3.5	3.4	3.3	3.4	3.5	3.4	3.5	3.5	3.4	3.3	3.4	3.6	3.5	3.5	3.5	3.5
50.0	3.1	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.1	3.0	3.0	3.0	3.1	3.0	3.1	3.1

Sensitivity to Noise vs Outliers: Sequence 05																							
	Noise Standard Deviation [m]																						
% Outliers	0.0	0.15	0.3	0.45	0.6	0.75	0.9	1.1	1.2	1.4	1.5	1.9	2.2	2.6	3.0	3.4	3.8	4.1	4.5	4.9	5.2	5.6	6.0
0.0	98	30	23	19	17	15	13	13	12	11	11	9.5	9.1	8.5	8.3	7.3	7.4	6.6	6.0	5.8	5.7	5.4	5.0
5.0	3.4	3.4	3.3	3.3	3.5	3.2	3.3	3.4	3.3	3.2	3.3	3.3	3.2	3.0	3.1	2.9	2.9	3.1	3.1	3.1	3.1	3.0	3.2
10.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9
15.0	3.3	3.2	3.2	3.2	3.2	3.2	3.2	3.2	3.2	3.2	3.2	3.2	3.1	3.1	3.1	3.1	3.0	3.0	3.0	3.0	3.0	3.0	2.9
20.0	3.3	3.4	3.4	3.4	3.3	3.4	3.3	3.3	3.4	3.3	3.3	3.3	3.3	3.3	3.2	3.2	3.1	3.1	3.1	3.1	3.0	3.0	3.0
30.0	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.9	3.0	3.0	3.3	3.2	3.4	3.2	3.5	3.6	3.5	3.7	3.7
40.0	3.5	3.6	3.6	3.7	3.6	3.7	3.9	3.8	3.9	3.9	3.8	4.0	4.0	4.0	4.1	4.0	4.0	3.9	4.1	4.2	4.3	4.3	4.3
50.0	2.9	2.9	2.9	2.9	2.9	2.9	2.8	2.8	2.9	2.8	2.8	2.8	2.8	2.9	2.9	2.8	2.9	2.8	2.8	2.8	2.9	2.9	3.0

Sensitivity to Noise vs Outliers: Sequence 06																							
	Noise Standard Deviation [m]																						
% Outliers	0.0	0.15	0.3	0.45	0.6	0.75	0.9	1.1	1.2	1.4	1.5	1.9	2.2	2.6	3.0	3.4	3.8	4.1	4.5	4.9	5.2	5.6	6.0
0.0	100	29	20	16	13	12	11	10	9.2	9.0	8.8	8.1	7.0	6.3	6.1	6.1	5.4	5.0	4.8	4.7	4.6	4.3	4.3
5.0	2.9	2.9	2.9	2.9	2.9	2.9	3.0	2.9	3.0	2.9	2.8	2.8	2.9	2.9	2.8	2.8	2.8	2.8	2.7	2.8	2.8	2.8	2.8
10.0	3.0	3.0	3.0	3.0	3.0	2.9	3.0	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.8	2.8	2.9	2.8	2.8	2.8	2.8
15.0	3.1	3.2	3.1	3.1	3.2	3.1	3.2	3.1	3.2	3.1	3.1	3.1	3.1	3.1	3.1	3.0	3.0	3.0	2.9	2.9	2.8	2.9	2.9
20.0	3.1	3.2	3.2	3.2	3.2	3.2	3.2	3.2	3.2	3.2	3.2	3.2	3.2	3.0	3.0	3.0	3.0	3.0	2.9	2.9	2.9	2.9	2.8
30.0	2.6	2.6	2.6	2.6	2.6	2.6	2.6	2.6	2.6	2.7	2.7	2.7	2.8	2.7	2.9	3.1	2.9	3.1	3.1	3.0	3.3	3.2	3.0
40.0	2.7	2.7	2.7	2.7	2.7	2.7	2.7	2.8	2.9	3.0	2.9	3.1	3.2	3.4	3.4	3.3	3.5	3.4	3.6	3.5	3.5	3.5	3.4
50.0	2.8	2.7	2.7	2.7	2.7	2.7	2.7	2.7	2.7	2.7	2.6	2.7	2.6	2.6	2.6	2.7	2.7	2.6	2.7	2.6	2.7	2.7	2.7

Table A.1: The average percentage of landmarks associated by GrassGraph [58], given various percentages of the landmarks replaced with outliers and noise with various standard deviations added to their positions. The effect of only outliers and effect of only noise are given in the first column and first row.

Sensitivity to Noise vs Outliers: Sequence 00																							
	Noise Standard Deviation [m]																						
% Outliers	0.0	0.15	0.3	0.45	0.6	0.75	0.9	1.1	1.2	1.4	1.5	1.9	2.2	2.6	3.0	3.4	3.8	4.1	4.5	4.9	5.2	5.6	6.0
0.0	0.0	4.3	6.2	7.4	8.5	8.5	9.6	10.0	11.0	11.0	11.0	13.0	14.0	15.0	17.0	19.0	20.0	23.0	24.0	25.0	27.0	27.0	28.0
5.0	35.0	34.0	36.0	36.0	35.0	36.0	35.0	37.0	34.0	36.0	35.0	35.0	36.0	36.0	35.0	37.0	36.0	37.0	38.0	37.0	36.0	35.0	37.0
10.0	39.0	38.0	41.0	40.0	38.0	39.0	40.0	38.0	38.0	38.0	39.0	39.0	40.0	40.0	40.0	40.0	42.0	42.0	41.0	40.0	40.0	41.0	40.0
15.0	38.0	39.0	39.0	39.0	39.0	39.0	39.0	40.0	38.0	40.0	39.0	40.0	40.0	41.0	41.0	40.0	42.0	41.0	43.0	41.0	41.0	44.0	44.0
20.0	39.0	37.0	37.0	37.0	37.0	38.0	38.0	38.0	38.0	37.0	38.0	40.0	40.0	39.0	41.0	40.0	42.0	43.0	42.0	42.0	42.0	41.0	42.0
30.0	44.0	41.0	42.0	43.0	40.0	41.0	41.0	42.0	42.0	40.0	42.0	42.0	43.0	42.0	41.0	41.0	41.0	42.0	43.0	41.0	41.0	41.0	39.0
40.0	40.0	39.0	40.0	39.0	40.0	41.0	37.0	38.0	39.0	38.0	37.0	38.0	38.0	37.0	37.0	37.0	37.0	36.0	38.0	37.0	39.0	39.0	40.0
50.0	47.0	44.0	45.0	46.0	45.0	45.0	44.0	44.0	45.0	45.0	45.0	44.0	46.0	42.0	44.0	42.0	46.0	44.0	45.0	42.0	45.0	45.0	45.0

Sensitivity to Noise vs Outliers: Sequence 02																							
	Noise Standard Deviation [m]																						
% Outliers	0.0	0.15	0.3	0.45	0.6	0.75	0.9	1.1	1.2	1.4	1.5	1.9	2.2	2.6	3.0	3.4	3.8	4.1	4.5	4.9	5.2	5.6	6.0
0.0	0.0	6.4	8.1	9.5	11.0	12.0	12.0	14.0	13.0	14.0	15.0	16.0	17.0	19.0	20.0	22.0	23.0	25.0	26.0	26.0	27.0	30.0	31.0
5.0	35.0	36.0	35.0	35.0	36.0	38.0	36.0	36.0	35.0	39.0	37.0	36.0	37.0	36.0	37.0	36.0	40.0	38.0	39.0	38.0	37.0	36.0	38.0
10.0	38.0	38.0	38.0	40.0	39.0	39.0	40.0	39.0	38.0	39.0	38.0	38.0	40.0	37.0	39.0	42.0	38.0	37.0	43.0	39.0	42.0	39.0	40.0
15.0	39.0	36.0	37.0	37.0	36.0	38.0	35.0	36.0	37.0	38.0	37.0	38.0	36.0	35.0	37.0	39.0	38.0	39.0	40.0	39.0	41.0	41.0	43.0
20.0	36.0	36.0	36.0	37.0	37.0	37.0	36.0	37.0	35.0	37.0	35.0	35.0	39.0	37.0	36.0	40.0	38.0	41.0	39.0	38.0	39.0	40.0	40.0
30.0	41.0	45.0	41.0	42.0	44.0	40.0	42.0	43.0	42.0	41.0	41.0	42.0	39.0	45.0	42.0	43.0	40.0	41.0	42.0	42.0	44.0	39.0	43.0
40.0	40.0	39.0	39.0	39.0	39.0	40.0	38.0	38.0	40.0	41.0	41.0	39.0	38.0	42.0	38.0	41.0	41.0	41.0	41.0	41.0	42.0	41.0	41.0
50.0	46.0	43.0	43.0	44.0	43.0	45.0	43.0	44.0	43.0	45.0	44.0	45.0	46.0	43.0	43.0	46.0	43.0	45.0	42.0	45.0	45.0	43.0	44.0

Sensitivity to Noise vs Outliers: Sequence 05																							
	Noise Standard Deviation [m]																						
% Outliers	0.0	0.15	0.3	0.45	0.6	0.75	0.9	1.1	1.2	1.4	1.5	1.9	2.2	2.6	3.0	3.4	3.8	4.1	4.5	4.9	5.2	5.6	6.0
0.0	0.0	6.2	8.7	10.0	12.0	12.0	13.0	13.0	13.0	14.0	15.0	16.0	16.0	19.0	20.0	21.0	21.0	25.0	25.0	27.0	27.0	30.0	30.0
5.0	36.0	36.0	37.0	37.0	35.0	36.0	36.0	36.0	36.0	38.0	36.0	35.0	34.0	37.0	35.0	38.0	37.0	40.0	37.0	35.0	37.0	39.0	36.0
10.0	40.0	40.0	42.0	42.0	40.0	43.0	43.0	43.0	42.0	43.0	43.0	41.0	41.0	41.0	43.0	39.0	43.0	44.0	44.0	43.0	43.0	43.0	43.0
15.0	41.0	41.0	40.0	39.0	44.0	42.0	41.0	40.0	41.0	41.0	42.0	42.0	43.0	41.0	44.0	41.0	43.0	42.0	45.0	46.0	46.0	44.0	46.0
20.0	41.0	41.0	41.0	40.0	43.0	44.0	41.0	41.0	40.0	42.0	43.0	43.0	42.0	45.0	42.0	44.0	43.0	44.0	46.0	44.0	46.0	46.0	44.0
30.0	45.0	43.0	40.0	44.0	43.0	42.0	41.0	44.0	41.0	42.0	45.0	43.0	40.0	43.0	39.0	42.0	42.0	40.0	44.0	42.0	42.0	39.0	41.0
40.0	41.0	44.0	41.0	43.0	42.0	40.0	40.0	41.0	39.0	40.0	39.0	37.0	42.0	40.0	40.0	39.0	39.0	39.0	39.0	38.0	38.0	37.0	41.0
50.0	47.0	45.0	49.0	47.0	46.0	46.0	46.0	46.0	45.0	48.0	45.0	43.0	47.0	46.0	46.0	51.0	47.0	47.0	47.0	46.0	46.0	46.0	46.0

Sensitivity to Noise vs Outliers: Sequence 06																							
	Noise Standard Deviation [m]																						
% Outliers	0.0	0.15	0.3	0.45	0.6	0.75	0.9	1.1	1.2	1.4	1.5	1.9	2.2	2.6	3.0	3.4	3.8	4.1	4.5	4.9	5.2	5.6	6.0
0.0	0.0	7.6	9.4	12.0	12.0	13.0	14.0	14.0	15.0	14.0	16.0	18.0	19.0	20.0	22.0	20.0	25.0	25.0	28.0	28.0	28.0	29.0	30.0
5.0	37.0	39.0	39.0	38.0	35.0	38.0	36.0	35.0	36.0	35.0	37.0	38.0	37.0	37.0	37.0	37.0	39.0	39.0	42.0	39.0	39.0	38.0	41.0
10.0	49.0	45.0	49.0	47.0	49.0	45.0	48.0	49.0	48.0	45.0	47.0	46.0	51.0	47.0	47.0	47.0	52.0	49.0	52.0	45.0	49.0	46.0	46.0
15.0	51.0	49.0	51.0	50.0	49.0	49.0	50.0	50.0	49.0	51.0	49.0	52.0	51.0	48.0	47.0	51.0	52.0	52.0	50.0	53.0	53.0	48.0	50.0
20.0	49.0	49.0	51.0	51.0	47.0	51.0	47.0	49.0	50.0	49.0	51.0	49.0	48.0	52.0	52.0	50.0	50.0	51.0	51.0	49.0	58.0	55.0	46.0
30.0	47.0	48.0	46.0	51.0	48.0	47.0	47.0	46.0	49.0	47.0	51.0	44.0	46.0	46.0	47.0	47.0	42.0	42.0	46.0	45.0	45.0	46.0	45.0
40.0	49.0	48.0	48.0	45.0	52.0	47.0	51.0	48.0	48.0	47.0	43.0	44.0	45.0	44.0	43.0	44.0	44.0	44.0	43.0	45.0	43.0	42.0	39.0
50.0	55.0	60.0	58.0	53.0	56.0	57.0	55.0	51.0	55.0	49.0	47.0	50.0	50.0	50.0	50.0	46.0	51.0	51.0	51.0	48.0	51.0	54.0	45.0

Table A.2: The median Frobenius norm error between the alignment matrix estimated by GrassGraph [58] and the true alignment matrix, given various percentages of the landmarks replaced with outliers and noise with various standard deviations added to their positions. The effect of only outliers and effect of only noise are given in the first column and first row.

Sensitivity to Noise vs Outliers: Sequence 00																							
	Noise Standard Deviation [m]																						
% Outliers	0.0	0.15	0.3	0.45	0.6	0.75	0.9	1.05	1.2	1.35	1.5	1.88	2.25	2.62	3.0	3.38	3.75	4.12	4.5	4.88	5.25	5.62	6.0
0.0	0.396	54.9	70.3	79.2	85.1	89.5	92.3	96.0	101.0	100.0	108.0	109.0	116.0	118.0	119.0	121.0	122.0	124.0	126.0	127.0	124.0	127.0	125.0
5.0	129.0	129.0	130.0	127.0	128.0	129.0	129.0	129.0	127.0	130.0	128.0	128.0	129.0	130.0	129.0	130.0	129.0	130.0	130.0	130.0	132.0	129.0	130.0
10.0	124.0	123.0	124.0	123.0	123.0	125.0	125.0	123.0	123.0	124.0	125.0	125.0	125.0	123.0	125.0	123.0	125.0	124.0	126.0	127.0	126.0	125.0	126.0
15.0	123.0	125.0	126.0	125.0	124.0	124.0	123.0	123.0	124.0	124.0	125.0	127.0	123.0	124.0	124.0	123.0	123.0	124.0	125.0	124.0	125.0	126.0	125.0
20.0	126.0	123.0	124.0	122.0	123.0	123.0	123.0	125.0	124.0	125.0	122.0	124.0	124.0	123.0	125.0	124.0	125.0	124.0	123.0	126.0	124.0	125.0	126.0
30.0	122.0	123.0	124.0	124.0	122.0	123.0	125.0	126.0	124.0	125.0	122.0	125.0	124.0	123.0	125.0	126.0	123.0	126.0	123.0	124.0	125.0	126.0	125.0
40.0	129.0	126.0	129.0	128.0	128.0	129.0	128.0	128.0	126.0	127.0	127.0	126.0	127.0	128.0	125.0	127.0	124.0	127.0	126.0	127.0	126.0	127.0	124.0
50.0	128.0	128.0	131.0	130.0	131.0	128.0	132.0	130.0	128.0	128.0	129.0	127.0	128.0	129.0	127.0	129.0	128.0	126.0	129.0	128.0	127.0	128.0	127.0

Sensitivity to Noise vs Outliers: Sequence 02																							
	Noise Standard Deviation [m]																						
% Outliers	0.0	0.15	0.3	0.45	0.6	0.75	0.9	1.05	1.2	1.35	1.5	1.88	2.25	2.62	3.0	3.38	3.75	4.12	4.5	4.88	5.25	5.62	6.0
0.0	1.24	72.3	83.8	88.2	93.2	101.0	100.0	105.0	107.0	110.0	112.0	114.0	117.0	120.0	120.0	121.0	123.0	123.0	125.0	125.0	125.0	125.0	125.0
5.0	127.0	127.0	129.0	128.0	130.0	129.0	130.0	128.0	129.0	131.0	129.0	129.0	129.0	130.0	130.0	128.0	129.0	129.0	129.0	128.0	130.0	132.0	130.0
10.0	126.0	127.0	128.0	128.0	126.0	128.0	129.0	124.0	126.0	130.0	123.0	129.0	126.0	126.0	127.0	127.0	129.0	128.0	128.0	129.0	126.0	127.0	129.0
15.0	124.0	124.0	125.0	125.0	122.0	125.0	125.0	124.0	124.0	123.0	126.0	126.0	124.0	128.0	127.0	125.0	128.0	128.0	125.0	123.0	126.0	127.0	127.0
20.0	124.0	124.0	127.0	127.0	126.0	128.0	126.0	124.0	126.0	124.0	128.0	126.0	124.0	125.0	123.0	122.0	124.0	125.0	126.0	123.0	125.0	127.0	123.0
30.0	126.0	126.0	125.0	124.0	126.0	127.0	126.0	124.0	125.0	126.0	125.0	126.0	126.0	123.0	127.0	124.0	127.0	124.0	123.0	125.0	124.0	123.0	126.0
40.0	131.0	130.0	132.0	131.0	132.0	129.0	127.0	132.0	129.0	130.0	130.0	131.0	130.0	131.0	129.0	130.0	127.0	129.0	127.0	124.0	123.0	127.0	125.0
50.0	128.0	130.0	132.0	130.0	126.0	130.0	131.0	130.0	132.0	129.0	128.0	130.0	127.0	127.0	130.0	130.0	130.0	129.0	127.0	127.0	129.0	128.0	127.0

Sensitivity to Noise vs Outliers: Sequence 05																							
	Noise Standard Deviation [m]																						
% Outliers	0.0	0.15	0.3	0.45	0.6	0.75	0.9	1.05	1.2	1.35	1.5	1.88	2.25	2.62	3.0	3.38	3.75	4.12	4.5	4.88	5.25	5.62	6.0
0.0	2.23	64.9	79.5	88.0	93.3	96.0	103.0	102.0	106.0	111.0	114.0	118.0	117.0	119.0	118.0	123.0	121.0	124.0	127.0	123.0	125.0	129.0	128.0
5.0	128.0	130.0	126.0	130.0	132.0	127.0	129.0	128.0	130.0	130.0	129.0	128.0	129.0	131.0	131.0	129.0	127.0	130.0	128.0	128.0	127.0	128.0	130.0
10.0	125.0	123.0	123.0	124.0	125.0	122.0	124.0	123.0	123.0	124.0	124.0	121.0	126.0	122.0	125.0	124.0	124.0	124.0	128.0	123.0	125.0	124.0	126.0
15.0	123.0	125.0	123.0	121.0	122.0	122.0	123.0	124.0	124.0	124.0	124.0	124.0	124.0	121.0	125.0	128.0	126.0	124.0	125.0	127.0	122.0	126.0	121.0
20.0	125.0	123.0	124.0	125.0	126.0	127.0	124.0	125.0	123.0	127.0	125.0	123.0	122.0	127.0	123.0	123.0	122.0	122.0	125.0	122.0	125.0	124.0	123.0
30.0	122.0	124.0	125.0	124.0	123.0	123.0	122.0	122.0	124.0	123.0	122.0	120.0	124.0	125.0	125.0	124.0	124.0	125.0	125.0	124.0	124.0	125.0	124.0
40.0	128.0	129.0	128.0	128.0	129.0	128.0	125.0	126.0	126.0	125.0	127.0	126.0	128.0	126.0	125.0	126.0	125.0	125.0	123.0	125.0	126.0	126.0	126.0
50.0	131.0	132.0	130.0	131.0	132.0	128.0	128.0	129.0	127.0	127.0	129.0	129.0	131.0	129.0	126.0	129.0	128.0	127.0	128.0	126.0	126.0	125.0	127.0

Sensitivity to Noise vs Outliers: Sequence 06																							
	Noise Standard Deviation [m]																						
% Outliers	0.0	0.15	0.3	0.45	0.6	0.75	0.9	1.05	1.2	1.35	1.5	1.88	2.25	2.62	3.0	3.38	3.75	4.12	4.5	4.88	5.25	5.62	6.0
0.0	0.102	73.9	88.7	96.3	103.0	105.0	109.0	115.0	114.0	116.0	117.0	118.0	122.0	122.0	124.0	121.0	123.0	127.0	124.0	126.0	125.0	128.0	125.0
5.0	130.0	128.0	133.0	125.0	128.0	129.0	129.0	130.0	128.0	129.0	129.0	128.0	128.0	129.0	128.0	128.0	130.0	130.0	129.0	130.0	129.0	129.0	129.0
10.0	127.0	126.0	127.0	126.0	128.0	125.0	125.0	129.0	130.0	129.0	126.0	127.0	128.0	127.0	126.0	126.0	127.0	127.0	124.0	128.0	128.0	124.0	126.0
15.0	123.0	126.0	127.0	123.0	124.0	122.0	126.0	129.0	128.0	126.0	128.0	126.0	126.0	128.0	128.0	123.0	123.0	126.0	126.0	126.0	123.0	129.0	125.0
20.0	126.0	124.0	124.0	126.0	127.0	124.0	126.0	122.0	125.0	123.0	126.0	125.0	126.0	124.0	129.0	126.0	125.0	128.0	124.0	124.0	123.0	127.0	124.0
30.0	123.0	127.0	127.0	125.0	123.0	123.0	126.0	125.0	124.0	126.0	125.0	126.0	128.0	125.0	127.0	126.0	127.0	127.0	131.0	123.0	127.0	124.0	131.0
40.0	127.0	127.0	126.0	128.0	127.0	128.0	128.0	127.0	125.0	130.0	128.0	125.0	124.0	127.0	122.0	127.0	125.0	127.0	126.0	127.0	123.0	125.0	125.0
50.0	128.0	129.0	129.0	129.0	127.0	129.0	127.0	127.0	127.0	131.0	127.0	127.0	128.0	127.0	128.0	127.0	126.0	127.0	127.0	126.0	126.0	123.0	126.0

Table A.3: The average angular error between the alignment matrix estimated by Grass-Graph [58] and the true alignment matrix, given various percentages of the landmarks replaced with outliers and noise with various standard deviations added to their positions. The effect of only outliers and effect of only noise are given in the first column and first row.

Sensitivity to Noise vs Outliers: Sequence 00																							
	Noise Standard Deviation [m]																						
Outliers	0.0	0.15	0.3	0.45	0.6	0.75	0.9	1.1	1.2	1.4	1.5	1.9	2.2	2.6	3.0	3.4	3.8	4.1	4.5	4.9	5.2	5.6	6.0
0	100	37	29	23	21	19	17	15	14	13	13	11	9.9	9.1	8.3	7.2	6.7	6.2	5.9	5.3	5.2	4.9	4.8
1	9.1	9.1	9.1	9.3	9.5	9.2	8.7	8.8	8.9	8.6	8.5	7.8	7.4	6.5	6.0	5.3	5.0	5.0	4.5	4.3	4.2	4.3	4.2
2	8.0	8.0	8.1	8.2	8.0	7.9	7.9	7.8	7.9	7.9	7.4	7.4	6.5	6.1	5.7	5.5	5.0	4.8	4.6	4.4	4.5	4.4	4.2
3	6.7	6.7	6.7	6.8	6.7	6.9	6.7	6.9	6.7	6.8	6.9	6.2	6.1	5.8	5.3	5.1	5.0	4.4	4.4	4.2	4.3	4.1	4.2
4	4.2	4.2	4.2	4.3	4.3	4.3	4.1	4.3	4.3	4.4	4.3	4.2	4.1	4.0	3.8	3.7	3.5	3.5	3.4	3.4	3.4	3.4	3.4
5	3.2	3.2	3.2	3.1	3.2	3.1	3.1	3.1	3.1	3.2	3.1	3.2	3.2	3.2	3.0	3.2	3.1	3.1	3.2	3.3	3.3	3.4	3.4
6	3.2	3.2	3.2	3.2	3.2	3.2	3.0	3.1	3.1	3.1	3.0	3.0	3.0	3.0	2.9	2.9	2.9	2.9	3.0	3.0	3.0	3.0	3.0
7	2.7	2.7	2.7	2.7	2.7	2.7	2.8	2.7	2.7	2.7	2.7	2.7	2.7	2.8	2.8	2.8	2.8	2.8	2.8	2.9	2.9	3.0	3.0
8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.9	2.9	2.8	2.9	2.9	2.9	2.9	2.9	3.0
9	2.9	2.9	2.9	2.8	2.9	2.9	2.8	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.8	2.9	2.9	2.9	2.9
10	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9

Sensitivity to Noise vs Outliers: Sequence 02																							
	Noise Standard Deviation [m]																						
Outliers	0.0	0.15	0.3	0.45	0.6	0.75	0.9	1.1	1.2	1.4	1.5	1.9	2.2	2.6	3.0	3.4	3.8	4.1	4.5	4.9	5.2	5.6	6.0
0	99	31	24	19	17	15	13	13	12	11	11	9.7	8.6	7.6	6.8	6.3	6.1	5.7	5.3	5.0	4.9	4.6	4.3
1	8.1	7.9	7.8	7.7	7.8	8.0	7.9	7.5	7.5	7.8	7.3	6.9	6.4	5.8	5.3	5.0	4.8	4.8	4.4	4.4	4.1	4.0	3.9
2	7.2	6.9	6.9	7.0	7.5	7.4	7.0	7.2	7.0	7.0	6.7	7.1	6.1	5.8	5.5	5.7	4.6	4.4	4.4	4.3	3.9	4.1	4.2
3	6.8	6.8	7.0	7.1	7.2	7.0	6.7	6.9	6.9	7.2	6.8	6.4	5.8	5.5	5.5	5.0	4.6	4.4	4.3	4.4	4.2	4.0	3.8
4	4.3	4.4	4.3	4.4	4.5	4.5	4.4	4.9	4.5	4.6	4.6	4.6	4.2	4.3	4.0	3.6	3.6	3.5	3.5	3.5	3.4	3.5	3.4
5	2.8	2.8	2.8	2.9	2.9	2.9	2.9	2.8	2.9	3.1	2.9	3.0	3.0	3.0	3.1	3.0	3.1	3.3	3.1	3.0	3.3	3.2	3.2
6	2.9	2.9	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.7	2.8	2.8	2.8	2.8	2.9	2.9	2.9	2.9	2.9	2.9	3.0	2.9	3.0
7	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.9	2.9	2.8	2.9	2.9	3.0	3.0	3.0
8	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	3.0	2.9	2.9	3.0	3.0	3.0	3.0
9	3.0	3.0	3.0	2.9	2.9	2.9	3.0	3.0	3.0	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	3.0	3.0	2.9	2.9
10	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	2.9	3.0	3.0	3.0	3.0	3.0	3.0	2.9	2.9	3.0	3.0	3.0	3.0	3.0

Sensitivity to Noise vs Outliers: Sequence 05																							
	Noise Standard Deviation [m]																						
Outliers	0.0	0.15	0.3	0.45	0.6	0.75	0.9	1.1	1.2	1.4	1.5	1.9	2.2	2.6	3.0	3.4	3.8	4.1	4.5	4.9	5.2	5.6	6.0
0	98	30	23	19	17	15	14	13	12	11	10	9.7	9.3	8.3	8.0	7.8	7.2	6.9	6.0	5.6	5.7	5.1	5.4
1	10	10	9.8	10	10	9.3	9.3	9.6	9.6	9.2	9.0	8.4	8.2	7.3	6.9	6.3	6.1	5.4	5.0	4.8	4.7	4.7	4.6
2	9.5	9.6	9.3	9.6	9.0	9.0	8.7	9.2	8.8	8.4	8.6	7.7	7.4	7.4	7.0	6.5	5.8	5.3	5.0	5.1	5.0	4.9	4.6
3	9.1	9.0	9.0	9.1	8.8	9.2	8.9	8.8	8.1	8.5	8.4	8.4	7.4	7.4	6.7	6.3	5.7	5.7	5.1	5.2	5.1	4.7	4.5
4	5.4	5.5	5.4	5.1	5.2	5.3	5.7	5.1	5.5	5.6	5.1	5.6	5.6	4.7	4.6	4.4	4.1	4.1	4.2	3.9	4.0	4.0	3.6
5	3.4	3.4	3.4	3.4	3.3	3.5	3.6	3.4	3.5	3.4	3.4	3.6	3.5	3.6	3.3	3.4	3.5	3.5	3.4	3.4	3.6	3.7	3.6
6	3.4	3.4	3.4	3.4	3.3	3.5	3.3	3.4	3.2	3.2	3.3	3.2	3.1	3.1	3.0	3.0	2.9	3.1	3.0	3.2	3.3	3.1	3.2
7	2.8	2.7	2.7	2.7	2.8	2.7	2.7	2.8	2.7	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.9	2.9	2.9	2.9	3.1	3.0	3.1
8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.9	2.9	2.9	3.0	3.0	3.1	3.1
9	2.9	2.9	2.8	2.8	2.9	2.9	2.9	2.8	2.8	2.9	2.8	2.8	2.9	2.9	2.9	2.9	2.9	2.9	3.0	3.0	2.9	3.0	3.0
10	2.9	2.9	2.8	2.8	2.9	2.8	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	3.0

Sensitivity to Noise vs Outliers: Sequence 06																							
	Noise Standard Deviation [m]																						
Outliers	0.0	0.15	0.3	0.45	0.6	0.75	0.9	1.1	1.2	1.4	1.5	1.9	2.2	2.6	3.0	3.4	3.8	4.1	4.5	4.9	5.2	5.6	6.0
0	100	28	19	15	13	12	11	10	9.8	9.4	8.8	8.4	7.3	6.9	6.1	5.3	5.5	4.9	4.6	4.8	4.2	4.3	4.2
1	7.9	7.8	7.6	7.8	7.7	7.8	8.0	7.9	7.7	7.1	7.0	6.6	5.8	5.6	5.2	4.6	4.6	4.1	4.1	3.7	3.5	3.5	3.6
2	8.0	7.7	7.9	7.8	7.5	7.8	7.8	7.5	7.3	7.2	7.1	6.7	6.4	5.5	5.4	4.4	4.5	4.3	4.0	3.6	3.7	3.7	3.3
3	8.1	8.0	8.0	8.0	7.9	8.0	7.4	7.8	7.8	7.3	6.9	6.5	6.1	5.7	5.1	5.1	4.8	4.3	3.9	3.7	3.8	3.6	3.5
4	5.5	5.4	5.4	5.5	5.3	5.3	5.4	5.1	5.1	4.9	4.9	4.2	4.3	4.2	3.7	3.4	3.3	3.2	3.2	3.1	2.9	3.0	3.0
5	2.9	3.0	3.0	2.9	2.9	2.9	3.0	3.0	3.1	3.0	2.9	3.0	3.1	3.0	3.1	2.9	2.9	3.0	2.9	2.9	3.0	3.1	3.0
6	2.9	3.0	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.8	2.9	2.8	2.8	2.8	2.8	2.9	2.8	3.0
7	2.6	2.6	2.6	2.6	2.6	2.6	2.6	2.6	2.6	2.6	2.6	2.6	2.6	2.6	2.6	2.7	2.6	2.7	2.7	2.7	2.7	2.8	2.7
8	2.7	2.7	2.7	2.7	2.7	2.7	2.7	2.7	2.7	2.7	2.7	2.7	2.7	2.8	2.7	2.7	2.8	2.7	2.8	2.7	2.7	2.7	2.8
9	2.7	2.7	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8
10	2.8	2.8	2.7	2.7	2.7	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8

Table A.4: The average percentage of landmarks associated by GrassGraph [58], given landmarks replaced one-by-one with outliers. Noise with various standard deviations is added to their positions, as before. The effect of only outliers and effect of only noise are given in the first column and first row.

Sensitivity to Noise vs Outliers: Sequence 00																							
	Noise Standard Deviation [m]																						
Outliers	0.0	0.15	0.3	0.45	0.6	0.75	0.9	1.1	1.2	1.4	1.5	1.9	2.2	2.6	3.0	3.4	3.8	4.1	4.5	4.9	5.2	5.6	6.0
0	0.0	4.2	6.0	7.5	8.0	9.0	9.4	10.0	10.0	11.0	12.0	13.0	14.0	16.0	17.0	19.0	20.0	23.0	23.0	26.0	26.0	26.0	29.0
1	15.0	15.0	15.0	15.0	14.0	15.0	15.0	15.0	15.0	16.0	16.0	17.0	19.0	21.0	23.0	24.0	26.0	27.0	29.0	31.0	29.0	30.0	30.0
2	17.0	17.0	17.0	18.0	18.0	17.0	17.0	17.0	17.0	17.0	18.0	19.0	20.0	22.0	24.0	24.0	27.0	25.0	27.0	29.0	28.0	29.0	31.0
3	20.0	19.0	20.0	20.0	20.0	19.0	20.0	20.0	21.0	20.0	20.0	22.0	23.0	24.0	25.0	25.0	26.0	29.0	28.0	29.0	30.0	31.0	30.0
4	30.0	30.0	29.0	29.0	29.0	29.0	29.0	29.0	30.0	29.0	29.0	30.0	32.0	31.0	32.0	32.0	35.0	33.0	34.0	34.0	32.0	33.0	34.0
5	36.0	37.0	35.0	36.0	35.0	35.0	36.0	36.0	36.0	35.0	36.0	37.0	34.0	37.0	35.0	37.0	37.0	34.0	35.0	34.0	35.0	37.0	34.0
6	35.0	34.0	34.0	34.0	34.0	37.0	36.0	36.0	36.0	37.0	36.0	37.0	36.0	36.0	36.0	37.0	38.0	38.0	37.0	36.0	39.0	37.0	36.0
7	39.0	37.0	38.0	39.0	40.0	39.0	38.0	38.0	41.0	38.0	40.0	39.0	39.0	39.0	37.0	39.0	38.0	39.0	39.0	39.0	39.0	39.0	40.0
8	38.0	38.0	39.0	39.0	38.0	37.0	39.0	38.0	40.0	39.0	40.0	39.0	38.0	39.0	39.0	39.0	39.0	39.0	38.0	39.0	39.0	40.0	39.0
9	40.0	41.0	39.0	40.0	41.0	38.0	41.0	39.0	39.0	41.0	39.0	39.0	39.0	41.0	40.0	40.0	39.0	39.0	40.0	41.0	40.0	40.0	40.0
10	40.0	40.0	40.0	40.0	40.0	40.0	38.0	39.0	41.0	39.0	39.0	40.0	40.0	40.0	39.0	39.0	40.0	39.0	41.0	39.0	40.0	40.0	40.0

Sensitivity to Noise vs Outliers: Sequence 02																							
	Noise Standard Deviation [m]																						
Outliers	0.0	0.15	0.3	0.45	0.6	0.75	0.9	1.1	1.2	1.4	1.5	1.9	2.2	2.6	3.0	3.4	3.8	4.1	4.5	4.9	5.2	5.6	6.0
0	0.0	6.0	7.7	9.8	11.0	12.0	13.0	12.0	13.0	14.0	14.0	15.0	17.0	18.0	20.0	23.0	22.0	24.0	25.0	25.0	29.0	30.0	33.0
1	18.0	18.0	18.0	19.0	19.0	19.0	19.0	19.0	20.0	18.0	18.0	19.0	21.0	24.0	25.0	25.0	27.0	26.0	29.0	30.0	30.0	33.0	32.0
2	22.0	20.0	20.0	20.0	20.0	20.0	21.0	21.0	21.0	21.0	21.0	22.0	23.0	25.0	26.0	25.0	28.0	28.0	29.0	30.0	32.0	30.0	31.0
3	23.0	20.0	21.0	20.0	21.0	20.0	22.0	19.0	21.0	19.0	22.0	20.0	22.0	22.0	25.0	26.0	29.0	29.0	28.0	30.0	32.0	34.0	36.0
4	30.0	30.0	30.0	30.0	29.0	31.0	30.0	32.0	30.0	30.0	31.0	30.0	31.0	31.0	32.0	32.0	32.0	33.0	31.0	35.0	34.0	35.0	36.0
5	35.0	37.0	35.0	35.0	36.0	35.0	35.0	35.0	34.0	36.0	32.0	36.0	34.0	33.0	35.0	34.0	35.0	36.0	37.0	35.0	36.0	37.0	37.0
6	35.0	36.0	36.0	38.0	37.0	37.0	36.0	35.0	39.0	35.0	36.0	37.0	36.0	38.0	36.0	40.0	38.0	38.0	36.0	40.0	38.0	37.0	37.0
7	38.0	38.0	39.0	39.0	39.0	40.0	38.0	41.0	42.0	39.0	38.0	40.0	39.0	38.0	38.0	38.0	40.0	38.0	40.0	39.0	42.0	39.0	40.0
8	38.0	39.0	40.0	37.0	39.0	39.0	38.0	41.0	39.0	37.0	38.0	38.0	41.0	41.0	39.0	41.0	39.0	40.0	40.0	40.0	38.0	41.0	39.0
9	40.0	39.0	39.0	40.0	42.0	39.0	39.0	38.0	40.0	39.0	40.0	40.0	41.0	44.0	41.0	41.0	42.0	40.0	39.0	39.0	38.0	43.0	40.0
10	39.0	40.0	41.0	39.0	39.0	38.0	40.0	38.0	39.0	39.0	39.0	37.0	39.0	40.0	40.0	40.0	40.0	40.0	39.0	40.0	39.0	39.0	39.0

Sensitivity to Noise vs Outliers: Sequence 05																							
	Noise Standard Deviation [m]																						
Outliers	0.0	0.15	0.3	0.45	0.6	0.75	0.9	1.1	1.2	1.4	1.5	1.9	2.2	2.6	3.0	3.4	3.8	4.1	4.5	4.9	5.2	5.6	6.0
0	0.0	6.1	8.2	10.0	11.0	13.0	13.0	13.0	14.0	14.0	15.0	16.0	16.0	18.0	19.0	20.0	22.0	23.0	26.0	26.0	27.0	29.0	29.0
1	18.0	17.0	18.0	17.0	18.0	18.0	18.0	19.0	18.0	18.0	19.0	20.0	21.0	21.0	23.0	26.0	24.0	27.0	29.0	29.0	30.0	30.0	32.0
2	18.0	19.0	19.0	18.0	19.0	19.0	20.0	19.0	18.0	20.0	19.0	21.0	22.0	22.0	22.0	24.0	26.0	27.0	30.0	29.0	31.0	32.0	30.0
3	20.0	20.0	20.0	20.0	19.0	19.0	21.0	19.0	20.0	19.0	18.0	21.0	21.0	22.0	23.0	26.0	29.0	27.0	30.0	29.0	29.0	32.0	32.0
4	30.0	29.0	30.0	30.0	29.0	29.0	29.0	29.0	27.0	28.0	28.0	29.0	29.0	29.0	29.0	31.0	33.0	32.0	31.0	36.0	33.0	33.0	36.0
5	35.0	35.0	35.0	37.0	36.0	36.0	35.0	37.0	35.0	36.0	37.0	36.0	35.0	36.0	35.0	35.0	34.0	38.0	36.0	36.0	37.0	35.0	36.0
6	36.0	35.0	36.0	37.0	36.0	35.0	37.0	37.0	36.0	36.0	33.0	38.0	35.0	36.0	37.0	36.0	38.0	40.0	38.0	37.0	38.0	37.0	38.0
7	41.0	40.0	41.0	39.0	41.0	41.0	43.0	41.0	41.0	40.0	39.0	39.0	39.0	41.0	41.0	38.0	39.0	39.0	41.0	41.0	39.0	39.0	41.0
8	38.0	40.0	41.0	38.0	42.0	39.0	39.0	39.0	42.0	38.0	37.0	39.0	40.0	39.0	41.0	41.0	40.0	41.0	39.0	40.0	37.0	42.0	39.0
9	40.0	44.0	41.0	41.0	43.0	41.0	41.0	41.0	43.0	41.0	43.0	41.0	39.0	42.0	41.0	40.0	41.0	42.0	40.0	42.0	43.0	43.0	43.0
10	43.0	42.0	40.0	39.0	44.0	41.0	42.0	40.0	40.0	40.0	40.0	43.0	41.0	41.0	38.0	42.0	43.0	41.0	43.0	43.0	45.0	42.0	42.0

Sensitivity to Noise vs Outliers: Sequence 06																							
	Noise Standard Deviation [m]																						
Outliers	0.0	0.15	0.3	0.45	0.6	0.75	0.9	1.1	1.2	1.4	1.5	1.9	2.2	2.6	3.0	3.4	3.8	4.1	4.5	4.9	5.2	5.6	6.0
0	0.0	7.7	10.0	11.0	11.0	13.0	13.0	14.0	14.0	15.0	16.0	16.0	18.0	21.0	22.0	24.0	24.0	25.0	28.0	27.0	30.0	32.0	32.0
1	19.0	19.0	20.0	18.0	18.0	19.0	18.0	19.0	17.0	20.0	20.0	21.0	23.0	23.0	26.0	30.0	27.0	30.0	27.0	34.0	35.0	34.0	34.0
2	18.0	19.0	17.0	19.0	19.0	18.0	19.0	19.0	20.0	19.0	20.0	22.0	22.0	24.0	26.0	28.0	30.0	30.0	31.0	35.0	34.0	35.0	37.0
3	18.0	18.0	18.0	18.0	18.0	19.0	18.0	19.0	19.0	19.0	20.0	22.0	23.0	24.0	25.0	26.0	29.0	28.0	33.0	29.0	33.0	32.0	34.0
4	28.0	30.0	29.0	29.0	28.0	29.0	27.0	28.0	30.0	30.0	32.0	32.0	32.0	33.0	30.0	33.0	38.0	38.0	36.0	38.0	37.0	41.0	40.0
5	32.0	36.0	36.0	34.0	39.0	36.0	34.0	38.0	35.0	38.0	37.0	37.0	37.0	35.0	37.0	38.0	40.0	37.0	41.0	37.0	44.0	41.0	40.0
6	37.0	34.0	37.0	38.0	37.0	38.0	37.0	41.0	36.0	36.0	42.0	35.0	37.0	34.0	37.0	39.0	39.0	39.0	40.0	40.0	42.0	41.0	43.0
7	43.0	40.0	40.0	43.0	45.0	42.0	43.0	43.0	47.0	46.0	40.0	46.0	47.0	42.0	45.0	46.0	43.0	45.0	43.0	45.0	47.0	46.0	46.0
8	44.0	43.0	45.0	41.0	46.0	47.0	45.0	44.0	41.0	47.0	43.0	46.0	43.0	43.0	45.0	46.0	45.0	46.0	42.0	43.0	44.0	45.0	46.0
9	46.0	47.0	45.0	45.0	46.0	46.0	47.0	44.0	46.0	46.0	49.0	48.0	49.0	47.0	44.0	45.0	46.0	48.0	46.0	47.0	46.0	48.0	46.0
10	47.0	45.0	45.0	45.0	48.0	46.0	44.0	47.0	45.0	45.0	48.0	44.0	45.0	45.0	46.0	47.0	47.0	45.0	49.0	49.0	48.0	46.0	47.0

Table A.5: The median Frobenius norm error between the alignment matrix estimated by GrassGraph [58] and the true alignment matrix, given landmarks replaced one-by-one with outliers. Noise with various standard deviations is added to their positions, as before. The effect of only outliers and effect of only noise are given in the first column and first row.

Sensitivity to Noise vs Outliers: Sequence 00																							
	Noise Standard Deviation [m]																						
Outliers	0.0	0.15	0.3	0.45	0.6	0.75	0.9	1.05	1.2	1.35	1.5	1.88	2.25	2.62	3.0	3.38	3.75	4.12	4.5	4.88	5.25	5.62	6.0
0	0.396	54.5	68.2	77.8	84.3	89.3	94.9	99.1	100.0	105.0	106.0	111.0	112.0	114.0	120.0	123.0	124.0	123.0	125.0	126.0	127.0	125.0	127.0
1	119.0	117.0	118.0	117.0	119.0	118.0	117.0	117.0	116.0	118.0	119.0	119.0	120.0	121.0	119.0	119.0	122.0	124.0	122.0	122.0	123.0	122.0	123.0
2	119.0	119.0	119.0	120.0	119.0	119.0	121.0	119.0	119.0	118.0	120.0	120.0	120.0	121.0	123.0	119.0	124.0	124.0	123.0	124.0	125.0	123.0	126.0
3	120.0	120.0	119.0	120.0	120.0	120.0	120.0	120.0	121.0	119.0	121.0	120.0	118.0	122.0	122.0	121.0	123.0	125.0	123.0	123.0	123.0	125.0	125.0
4	125.0	125.0	127.0	128.0	127.0	125.0	126.0	125.0	128.0	126.0	127.0	126.0	127.0	126.0	126.0	127.0	127.0	126.0	126.0	127.0	128.0	128.0	127.0
5	128.0	128.0	127.0	127.0	128.0	127.0	127.0	128.0	127.0	126.0	125.0	127.0	128.0	128.0	126.0	125.0	126.0	126.0	126.0	127.0	129.0	128.0	125.0
6	128.0	128.0	126.0	127.0	129.0	127.0	127.0	127.0	128.0	128.0	130.0	129.0	128.0	130.0	128.0	129.0	126.0	129.0	132.0	131.0	129.0	132.0	130.0
7	127.0	127.0	127.0	126.0	126.0	128.0	126.0	127.0	128.0	128.0	124.0	128.0	130.0	127.0	128.0	128.0	129.0	131.0	129.0	130.0	131.0	129.0	130.0
8	127.0	126.0	127.0	127.0	129.0	127.0	130.0	127.0	126.0	127.0	126.0	127.0	130.0	129.0	128.0	127.0	129.0	129.0	130.0	130.0	131.0	133.0	129.0
9	125.0	125.0	127.0	126.0	124.0	126.0	127.0	125.0	127.0	128.0	127.0	127.0	126.0	126.0	128.0	127.0	129.0	129.0	129.0	131.0	128.0	128.0	130.0
10	127.0	126.0	126.0	127.0	129.0	126.0	127.0	125.0	124.0	125.0	127.0	127.0	127.0	126.0	126.0	127.0	128.0	128.0	128.0	129.0	129.0	130.0	130.0

Sensitivity to Noise vs Outliers: Sequence 02																							
	Noise Standard Deviation [m]																						
Outliers	0.0	0.15	0.3	0.45	0.6	0.75	0.9	1.05	1.2	1.35	1.5	1.88	2.25	2.62	3.0	3.38	3.75	4.12	4.5	4.88	5.25	5.62	6.0
0	1.24	70.7	79.9	89.6	99.6	99.9	102.0	103.0	105.0	111.0	110.0	109.0	117.0	119.0	120.0	124.0	122.0	124.0	125.0	123.0	125.0	124.0	128.0
1	118.0	117.0	115.0	116.0	116.0	117.0	118.0	120.0	118.0	118.0	115.0	118.0	119.0	119.0	120.0	120.0	122.0	122.0	121.0	124.0	125.0	125.0	124.0
2	119.0	120.0	117.0	115.0	120.0	118.0	118.0	118.0	115.0	118.0	117.0	118.0	120.0	119.0	120.0	121.0	121.0	124.0	121.0	125.0	123.0	125.0	127.0
3	115.0	116.0	118.0	118.0	118.0	116.0	118.0	116.0	117.0	118.0	118.0	118.0	118.0	122.0	121.0	121.0	120.0	122.0	125.0	124.0	126.0	122.0	124.0
4	126.0	124.0	124.0	123.0	123.0	124.0	127.0	124.0	125.0	124.0	123.0	125.0	124.0	127.0	127.0	125.0	127.0	127.0	130.0	129.0	129.0	128.0	131.0
5	128.0	125.0	125.0	124.0	125.0	123.0	128.0	125.0	126.0	126.0	123.0	126.0	122.0	126.0	123.0	128.0	125.0	128.0	126.0	125.0	127.0	127.0	127.0
6	127.0	131.0	128.0	129.0	131.0	129.0	129.0	128.0	130.0	128.0	131.0	131.0	129.0	130.0	129.0	131.0	131.0	129.0	130.0	131.0	132.0	130.0	129.0
7	130.0	128.0	129.0	130.0	129.0	133.0	129.0	132.0	130.0	129.0	131.0	130.0	129.0	131.0	132.0	129.0	132.0	131.0	131.0	130.0	132.0	134.0	133.0
8	132.0	128.0	128.0	129.0	128.0	130.0	130.0	130.0	129.0	129.0	132.0	135.0	132.0	132.0	130.0	133.0	131.0	132.0	133.0	130.0	131.0	130.0	134.0
9	130.0	131.0	129.0	128.0	129.0	129.0	128.0	130.0	129.0	129.0	128.0	130.0	129.0	130.0	131.0	130.0	129.0	130.0	132.0	130.0	132.0	131.0	132.0
10	129.0	128.0	131.0	131.0	128.0	128.0	129.0	130.0	129.0	131.0	128.0	130.0	130.0	129.0	129.0	130.0	131.0	131.0	130.0	130.0	133.0	131.0	129.0

Sensitivity to Noise vs Outliers: Sequence 05																							
	Noise Standard Deviation [m]																						
Outliers	0.0	0.15	0.3	0.45	0.6	0.75	0.9	1.05	1.2	1.35	1.5	1.88	2.25	2.62	3.0	3.38	3.75	4.12	4.5	4.88	5.25	5.62	6.0
0	2.23	65.2	76.1	88.6	94.3	98.2	106.0	105.0	111.0	107.0	108.0	115.0	118.0	118.0	123.0	124.0	122.0	124.0	124.0	125.0	125.0	125.0	127.0
1	114.0	117.0	119.0	114.0	117.0	120.0	116.0	116.0	117.0	117.0	115.0	117.0	118.0	122.0	121.0	120.0	121.0	121.0	122.0	126.0	124.0	123.0	127.0
2	116.0	116.0	120.0	120.0	117.0	115.0	116.0	117.0	118.0	117.0	118.0	120.0	123.0	117.0	121.0	125.0	121.0	121.0	123.0	125.0	124.0	126.0	126.0
3	118.0	119.0	118.0	115.0	119.0	117.0	115.0	117.0	118.0	115.0	118.0	120.0	118.0	120.0	121.0	121.0	120.0	123.0	124.0	127.0	123.0	123.0	127.0
4	124.0	125.0	126.0	124.0	127.0	124.0	126.0	125.0	126.0	127.0	127.0	124.0	128.0	127.0	125.0	129.0	127.0	126.0	127.0	127.0	127.0	128.0	127.0
5	125.0	125.0	128.0	126.0	128.0	125.0	128.0	124.0	125.0	127.0	125.0	126.0	127.0	124.0	126.0	126.0	127.0	125.0	125.0	126.0	127.0	127.0	130.0
6	128.0	129.0	127.0	133.0	129.0	127.0	130.0	130.0	130.0	129.0	127.0	130.0	130.0	130.0	127.0	129.0	131.0	129.0	130.0	130.0	130.0	131.0	127.0
7	128.0	128.0	128.0	127.0	125.0	130.0	129.0	129.0	128.0	130.0	128.0	128.0	129.0	128.0	129.0	128.0	130.0	129.0	130.0	130.0	129.0	130.0	131.0
8	129.0	128.0	128.0	129.0	128.0	128.0	127.0	127.0	127.0	124.0	129.0	123.0	129.0	127.0	130.0	127.0	128.0	129.0	130.0	130.0	130.0	130.0	129.0
9	125.0	124.0	124.0	123.0	126.0	126.0	125.0	126.0	127.0	128.0	125.0	127.0	128.0	126.0	125.0	128.0	126.0	129.0	128.0	128.0	129.0	131.0	128.0
10	123.0	128.0	125.0	124.0	128.0	127.0	125.0	126.0	125.0	127.0	126.0	127.0	126.0	126.0	125.0	126.0	128.0	128.0	129.0	127.0	129.0	129.0	132.0

Sensitivity to Noise vs Outliers: Sequence 06																							
	Noise Standard Deviation [m]																						
Outliers	0.0	0.15	0.3	0.45	0.6	0.75	0.9	1.05	1.2	1.35	1.5	1.88	2.25	2.62	3.0	3.38	3.75	4.12	4.5	4.88	5.25	5.62	6.0
0	0.102	76.3	89.4	98.0	104.0	105.0	111.0	114.0	109.0	114.0	114.0	117.0	118.0	120.0	122.0	123.0	123.0	127.0	127.0	126.0	126.0	125.0	127.0
1	121.0	123.0	125.0	123.0	122.0	121.0	123.0	126.0	122.0	119.0	121.0	120.0	126.0	122.0	123.0	123.0	124.0	124.0	124.0	124.0	125.0	126.0	125.0
2	122.0	124.0	120.0	126.0	121.0	122.0	123.0	125.0	123.0	125.0	120.0	122.0	123.0	123.0	124.0	123.0	124.0	125.0	123.0	124.0	124.0	124.0	124.0
3	121.0	122.0	122.0	122.0	122.0	125.0	125.0	124.0	120.0	121.0	122.0	121.0	122.0	122.0	123.0	122.0	122.0	123.0	124.0	120.0	120.0	123.0	126.0
4	123.0	123.0	126.0	123.0	122.0	124.0	127.0	123.0	121.0	124.0	122.0	125.0	123.0	127.0	125.0	127.0	127.0	123.0	125.0	129.0	127.0	125.0	124.0
5	127.0	124.0	128.0	126.0	128.0	125.0	125.0	122.0	126.0	128.0	124.0	125.0	124.0	124.0	126.0	123.0	126.0	128.0	126.0	125.0	125.0	127.0	125.0
6	130.0	126.0	126.0	129.0	127.0	129.0	127.0	129.0	128.0	132.0	128.0	129.0	125.0	129.0	130.0	132.0	127.0	128.0	128.0	129.0	127.0	133.0	131.0
7	127.0	130.0	132.0	130.0	129.0	130.0	130.0	129.0	129.0	132.0	130.0	130.0	132.0	132.0	126.0	128.0	131.0	129.0	131.0	130.0	131.0	132.0	130.0
8	130.0	127.0	129.0	128.0	128.0	130.0	130.0	130.0	127.0	131.0	128.0	130.0	131.0	130.0	130.0	130.0	132.0	130.0	128.0	130.0	129.0	132.0	128.0
9	131.0	129.0	127.0	128.0	130.0	129.0	129.0	128.0	131.0	129.0	127.0	129.0	131.0	134.0	132.0	127.0	129.0	127.0	129.0	129.0	131.0	127.0	131.0
10	129.0	129.0	130.0	131.0	128.0	126.0	127.0	130.0	129.0	126.0	130.0	130.0	129.0	130.0	130.0	129.0	131.0	133.0	130.0	132.0	132.0	132.0	129.0

Table A.6: The average angular error between the alignment matrix estimated by GrassGraph [58] and the true alignment matrix, given landmarks replaced one-by-one with outliers. Noise with various standard deviations is added to their positions, as before. The effect of only outliers and effect of only noise are given in the first column and first row.

Sensitivity to Noise vs Outliers: SHREC Dataset																							
	Noise Standard Deviation [m]																						
% Outliers	0.0	0.059	0.12	0.18	0.24	0.29	0.35	0.41	0.47	0.53	0.59	0.74	0.89	1.0	1.2	1.3	1.5	1.6	1.8	1.9	2.1	2.2	2.4
0.0	96	38	34	31	27	26	25	25	22	23	20	20	17	17	17	14	12	12	11	11	11	8.5	9.9
5.0	13	14	14	14	13	12	13	13	12	12	12	13	12	11	12	11	11	9.8	9.2	11	10	8.4	9.6
10.0	8.6	7.7	8.0	8.2	7.9	8.1	7.6	7.4	7.6	7.0	6.7	7.6	7.6	7.0	7.0	7.1	6.9	6.8	5.6	7.0	7.2	6.0	6.1
15.0	6.6	6.6	6.7	6.8	6.9	7.1	7.3	5.8	6.5	6.7	6.9	6.3	6.3	5.2	6.1	5.7	6.2	6.1	5.8	5.8	5.8	5.4	4.8
20.0	5.8	5.8	6.1	5.9	5.8	6.0	5.8	6.1	5.6	5.8	6.2	5.5	6.4	6.0	5.9	4.9	5.7	6.3	6.1	4.3	5.8	5.4	4.4
30.0	6.9	6.8	6.7	6.7	6.6	6.6	6.6	6.6	6.4	6.4	6.7	5.8	6.4	6.8	6.6	5.6	7.0	6.5	6.0	5.6	6.8	5.6	5.8
40.0	6.4	6.3	6.4	5.9	6.6	5.7	6.0	6.6	5.5	7.1	6.1	6.6	5.6	6.0	5.3	6.8	6.1	5.8	4.4	5.2	4.5	5.2	5.6
50.0	4.6	5.1	5.4	4.6	4.8	4.5	4.1	4.7	5.1	5.7	5.5	5.7	4.8	4.9	5.5	5.3	5.3	4.7	4.4	4.2	5.6	4.7	4.4

Table A.7: The percentage of points associated by GrassGraph [58], given various percentages of points replaced with outliers and noise with various standard deviations added to their positions. The effect of only outliers and effect of only noise are given in the first column and first row. The point sets are drawn from the SHREC dataset [42] as used by [58], randomly sub-sampled from 250 to 120 points for parity with our other results.

Sensitivity to Noise vs Outliers: SHREC Dataset																							
	Noise Standard Deviation [m]																						
% Outliers	0.0	0.059	0.12	0.18	0.24	0.29	0.35	0.41	0.47	0.53	0.59	0.74	0.89	1.0	1.2	1.3	1.5	1.6	1.8	1.9	2.1	2.2	2.4
0.0	0.0	1.1	1.4	1.2	1.6	1.9	2.1	1.7	2.1	2.2	2.5	3.1	3.6	3.2	3.5	3.6	5.5	4.8	4.8	4.8	4.6	5.5	5.9
5.0	4.9	4.6	5.5	4.9	5.3	6.2	5.6	5.4	5.8	6.0	7.2	5.9	5.9	5.9	5.6	5.9	5.8	5.1	7.8	5.2	6.8	8.1	6.7
10.0	6.5	11.0	10.0	11.0	7.2	10.0	7.7	9.0	10.0	9.8	14.0	8.6	13.0	7.9	8.0	11.0	12.0	11.0	13.0	12.0	16.0	9.6	9.2
15.0	16.0	12.0	17.0	19.0	10.0	13.0	17.0	27.0	16.0	14.0	16.0	16.0	14.0	11.0	11.0	16.0	14.0	11.0	12.0	13.0	15.0	16.0	19.0
20.0	14.0	11.0	13.0	12.0	14.0	14.0	20.0	19.0	13.0	16.0	18.0	14.0	17.0	19.0	13.0	24.0	11.0	20.0	13.0	19.0	13.0	12.0	16.0
30.0	9.6	8.1	7.6	8.6	13.0	8.8	8.9	8.0	9.7	11.0	11.0	11.0	8.1	7.9	8.9	11.0	10.0	9.8	10.0	14.0	10.0	10.0	11.0
40.0	11.0	13.0	12.0	13.0	12.0	11.0	12.0	14.0	12.0	12.0	15.0	12.0	13.0	10.0	13.0	11.0	11.0	12.0	20.0	12.0	20.0	14.0	12.0
50.0	25.0	21.0	15.0	28.0	13.0	19.0	28.0	15.0	19.0	16.0	17.0	14.0	20.0	17.0	16.0	14.0	16.0	15.0	16.0	15.0	16.0	15.0	18.0

Table A.8: The median error (computed using the Frobenius norm) between the true registration matrix for aligning sets of points and that estimated by GrassGraph [58], given various percentages of points replaced with outliers and noise with various standard deviations added to their positions. The effect of only outliers and effect of only noise are given in the first column and first row. The point sets are drawn from the SHREC dataset [42] as used by [58], randomly sub-sampled from 250 to 120 points for parity with our other results.

Sensitivity to Noise vs Outliers: SHREC Dataset																							
	Noise Standard Deviation [m]																						
% Outliers	0.0	0.059	0.118	0.177	0.236	0.295	0.354	0.414	0.473	0.532	0.591	0.739	0.886	1.03	1.18	1.33	1.48	1.62	1.77	1.92	2.07	2.22	2.36
0.0	3.26	38.1	48.4	48.1	60.6	62.1	51.0	52.1	55.1	73.0	72.8	75.5	73.1	70.3	85.4	99.2	97.6	102.0	111.0	109.0	110.0	111.0	99.8
5.0	97.5	96.6	107.0	107.0	102.0	114.0	96.4	96.7	100.0	99.2	106.0	103.0	106.0	116.0	102.0	114.0	116.0	115.0	115.0	111.0	115.0	120.0	115.0
10.0	114.0	123.0	116.0	111.0	108.0	112.0	125.0	121.0	113.0	131.0	120.0	120.0	121.0	114.0	121.0	115.0	122.0	125.0	128.0	124.0	124.0	134.0	121.0
15.0	113.0	120.0	123.0	126.0	121.0	119.0	113.0	119.0	115.0	120.0	126.0	125.0	123.0	125.0	124.0	122.0	125.0	115.0	128.0	117.0	123.0	118.0	123.0
20.0	129.0	122.0	124.0	125.0	127.0	129.0	126.0	121.0	125.0	124.0	119.0	122.0	126.0	129.0	129.0	133.0	126.0	134.0	131.0	120.0	126.0	127.0	136.0
30.0	117.0	130.0	128.0	114.0	126.0	113.0	120.0	120.0	124.0	112.0	123.0	117.0	124.0	127.0	116.0	124.0	116.0	128.0	129.0	121.0	131.0	125.0	120.0
40.0	125.0	114.0	123.0	119.0	133.0	118.0	128.0	122.0	130.0	124.0	123.0	125.0	130.0	121.0	124.0	131.0	115.0	111.0	132.0	129.0	125.0	114.0	132.0
50.0	119.0	121.0	118.0	122.0	126.0	114.0	132.0	132.0	130.0	130.0	119.0	127.0	130.0	124.0	122.0	131.0	126.0	123.0	120.0	123.0	129.0	135.0	118.0

Table A.9: The angular error between true registration matrix for aligning sets of points and that estimated by GrassGraph [58], given various percentages of the landmarks replaced with outliers and noise with various standard deviations added to their positions. The effect of only outliers and effect of only noise are given in the first column and first row. The point sets are drawn from the SHREC dataset [42] as used by [58], randomly sub-sampled from 250 to 120 points for parity with our other results.

Glossary

autoencoder A neural network architecture for learning efficient representations for unlabeled data. Consists of two portions, an encoder which generates compressed representations and a decoder which uses them for reconstruction. 64, 108, 109

CNN Convolutional Neural Network. A network architecture specialized for image processing. Each layer consists of a series of learned feature-extracting convolutional kernels. 9, 12, 18, 64, 109

DSO Direct Sparse Odometry. A SLAM system which samples points sparsely from the environment and from there tries to estimate ongoing camera position by minimizing the difference in pixel value when their reprojection is estimated. 25, 34

encoder See autoencoder. 10, 20, 109

global descriptor Can refer either to a process of summarizing the total content of an image into a description, or the description vector which embodies the resulting description. 2, 3

HOG Histogram of Oriented Gradients. A method of describing a region, of pixels based on the dominant direction of the image gradients in each of a grid of cells. 9

ICP Iterative Closest Point. A method for iteratively refining the alignment between two point clouds, based on making the distance between nearby points smaller. ICP is prone to poor solutions if the initial alignment is not close to the correct alignment, due to local maxima in ICP's loss function. 24

image retrieval A class of computer vision task that forms the basis of most VPR approaches. Given an image, retrieve images from a set which contain similar content.

Image retrieval is typically implemented through image descriptors which which summarize each image into a description vector which can be indexed for fast searching. 2, 6, 7, 11–14, 18, 63

IMU Inertial Measurement Unit. A combination of accelerate and gyroscope which can provide a coarse estimate of motion and therefore relative position. Very susceptible to drift over time, but can provide an absolute reference of scale as accelerometry is in meters per second squared. 34

keyframe As an optimization, visual SLAM systems primarily operate on keyframes which are a subset of all frames received from the camera. A variety of metrics are in use for selecting ”important” frames which are kept as keyframes. Neighboring keyframes frequently share portions of the environment which are mutually visible in both images. 6, 13, 29, 30, 34, 35, 41

LiDAR Light Detection and Ranging. A LiDAR scanner is a device which continually scans the surrounding environment with a laser beam. This produces a dense, regularly-sampled series of points on every surface, the positions of which are known and represented as a point cloud. 2–4, 11, 12, 14, 15, 18, 20, 21, 23, 26, 35, 36, 44, 84, 110

local descriptor Can refer either to a process for describing the pixels around a local feature, or the description vector which embodies the resulting description. 109

local feature Local features represent a specific location in an image, usually at a visual corner or other distinctive spot. They may include a description of nearby pixels, see local descriptor. 2, 109

metric map A map describing some area in terms of absolute position and distance, in coordinates. This can be difficult to maintain accurately over large areas, as opposed to more flexible maps describing relations between specific landmarks. 13

MLP MultiLayer Perceptron. The most basic architecture of neural network, consisting of layers of fully connected neurons where every neuron is connected to every neuron in the layer that precedes it. Frequently superseded by more specialized networks, such as a CNN or encoder/autoencoder. 19

place recognition A superset of VPR, place recognition is the general task of recognizing when one has returned to a previously visited location. One cannot rely on a reliable estimate of position, but instead recognize the environment through repetition of some measurable quantity. 1, 2, 7, 9, 12, 14, 16, 18, 20, 22, 23, 40, 55, 59, 62, 111

pseudo As in pseudo-LiDAR, pseudo-scan, or pseudo-pointcloud, terms used here to describe the pointclouds proposed by [57] which imitate the scans generated by a LiDAR scanner, produced from 3D data gathered by visual SLAM. 13, 16, 17, 25, 29, 30, 33, 35, 36, 38, 41, 43, 44, 76

R-MAC Regional Maximum Activation of Convolutions. A way to organize image-processing neural networks where activations in later layers select regions of lower-level features in earlier layers, to be summed together to produce a combined feature vector. 10

RANSAC Random Sample Consensus. An algorithm for determining the parameters of a model in the presence of outliers. For a series of randomly selected samples, models are generated and their fitness is scored based on the number of well-fitting inlier points. 21, 62

RGBD Red Green Blue Depth. A type of camera which densely measures depth at every pixel, such that each image is also a densely-sampled pointcloud akin to a directional LiDAR scan. 12, 14, 62

SLAM Simultaneous Localization and Mapping. The pose of a moving camera is continually tracked by using visual features in the environment. (See also: visual odometry) At the same time a large-scale map is constructed to improve global position estimates, the maintenance of which relies on loop closures provided by VPR 1–3, 6, 7, 9, 12, 14, 17, 19, 21, 23–25, 34, 36–38, 46, 47, 52, 54, 55, 59, 61, 64, 83, 111

SoTA The most recent proposed method which demonstrates improvement over all or most relevant previous methods for a given problem. 3

SVD Singular Value Decomposition. A common method of matrix factorization. 17, 39

transfer learning A way to boost the performance of a neural network in performing a task by leveraging training done for a related task. One example is starting with a neural network trained for object detection and retraining it to perform place recognition. In this case it may benefit from already possessing trained features relevant for object recognition. 9

visual odometry The process of continually tracking camera position by using features visible to a camera as reference. Has a strong tendency to accumulate error in global position estimates over time, which SLAM seeks to correct. 1, 7, 25, 110

VPR Visual Place Recognition. This is a class of technique which memorizes the scenery as it is first seen by a camera and detects when the camera returns to the same or a nearby location. It is an important requirement that VPR operate in unknown environments and not rely on an existing estimate of the camera's global position. VPR is a specialization of place recognition for visual camera imagery. 1, 3, 6, 7, 9, 11–14, 18, 22, 34, 38, 63, 83, 84, 108–110