HAL
open science

Hybrid mapping for large urban environments

Ferit Üzer

To cite this version:

Ferit Üzer. Hybrid mapping for large urban environments. Other. Université Blaise Pascal - Clermont-Ferrand II; Sung Kyun Kwan university (Séoul), 2016. English. NNT : 2016CLF22675 . tel-01379603

HAL Id: tel-01379603

https://theses.hal.science/tel-01379603v1

Submitted on 11 Oct 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N° d'ordre: 2675

EDSPIC: 744

UNIVERSITÉ BLAISE PASCAL - CLERMONT II

SUNGKYUNKWAN UNIVERSITY

ECOLE DOCTORALE
SCIENCES POUR L'INGÉNIEUR DE CLERMONT-FERRAND

Thèse

présentée par

Ferit Üzer

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ

SPÉCIALITÉ: Vision pour la Robotique

Cartographie Hybride pour des Environnements de Grande Taille

Soutenue publiquement le 02/03/2016 devant le jury:

M. Ouiddad Labbani-Igbida	Professeur, Univ. de Limoges	Rapporteur
M. Cédric Demonceaux	Professeur, Univ. de Bourgogne	Rapporteur
M. Navrati Saxena	Professeur, Sungkyunkwan Univ.	Examinateur
M. June-Ho Yi	Professeur, Seoul National Univ.	Examinateur
M. Hemanth Korrapati	Chercheur, Blippar	Examinateur
M. Eric Royer	MCF, Univ. d'Auvergne	Co-Directeur de thèse
M. Sukhan Lee	Professeur, Sungkyunkwan Univ.	Directeur de thèse
M. Youcef Mezouar	Professeur, IFMA	Directeur de thèse

Contents

List of Figures ix

List of Tables xiii

1 Introduction 1
- 1.1 Motivation . . . 1
  - 1.1.1 Introduction . . . 2
  - 1.1.2 Robotic Mapping . . . 4
    - 1.1.2.1 Metric maps . . . 5
    - 1.1.2.2 Topological maps . . . 5
    - 1.1.2.3 Semantic maps . . . 7
    - 1.1.2.4 Hybrid maps . . . 7
  - 1.1.3 Loop Closure . . . 10
- 1.2 Thesis Roadmap . . . 11
  - 1.2.1 Loop closure for Topo-Metric Mapping . . . 11
  - 1.2.2 Hybrid Mapping . . . 12

2 Literature Review 17
- 2.1 Simultaneous Localization and Mapping . . . 17
- 2.2 Vision Based Metric SLAM . . . 19
  - 2.2.1 Structure From Motion Methods . . . 20
  - 2.2.2 Filtering Methods . . . 21
  - 2.2.3 Biologically Inspired Methods . . . 23
- 2.3 Vision Based Topological SLAM . . . 23
  - 2.3.1 Global descriptor based methods . . . 25
  - 2.3.2 Local features based methods . . . 26
  - 2.3.3 Bag of Words Based Methods . . . 28
    - 2.3.3.1 Offline Visual Vocabulary Building . . . 30
    - 2.3.3.2 Online Visual Vocabulary Approaches . . . 33
  - 2.3.4 Combined Methods . . . 33

2.4 Vision based Semantic Mapping 35
2.4.1 Semantic mapping based on Metric Approaches 36
2.4.1.1 Indoor Semantic Mapping 36
2.4.1.2 Outdoor Semantic Mapping 37
2.4.2 Semantic mapping based on Topological Approaches . . . 39
2.5 Vision Based Hybrid SLAM 40
2.6 Conclusion 42

3 Hierarchical Loop Closure Detection 45
3.1 Framework Overview 46
3.2 Vector of Locally Aggregated Descriptors 47
3.3 Node Level Loop Closure 49
3.3.1 NLLC 49
3.3.2 Parameter Learning 51
3.4 Image Level Loop Closure 54
3.4.1 VLAD similarity 56
3.4.2 Spatial Similarity 57
3.4.3 Similarity of Local Odometry 59
3.4.4 Minimum Matches Constraint 61
3.4.5 Post-Processing 62
3.5 Experiments 62
3.5.1 Features and Parameters 64
3.5.2 Node Level Loop Closure 67
3.5.3 Image Level Loop Closure 69
3.5.4 Computational Time 72
3.6 Conclusion and Future Work 76

4 Hybrid Mapping 79
4.1 Hybrid Topo-Metric Framework 80
4.1.1 Image Sequence Partitioning Module 82
4.1.2 Loop Closure Detection 83
4.1.2.1 Node Level Loop Closure Detection 83
4.1.2.2 Image Level Loop Closure Detection 84
4.1.2.3 Statistical Matching Constraint 86
4.1.2.4 VLAD similarity 87
4.1.2.5 Geometric similarity 88
4.1.2.6 Correlation Between Trajectories 88
4.1.3 Adding a new node 89
4.1.4 Parameters 89
4.2 Hybrid Map Based on Road Semantics 90
4.2.1 Feature Extraction and 3D Point Cloud Generation 91

4.2.2 Robot Path Classification 92
4.2.2.1 Conditional Random Fields 93
4.2.2.2 Feature Functions 94
4.2.2.3 Parameter Estimation & Inference 97
4.2.3 Loop Closure Detection 97
4.2.4 Parameters 100
4.3 Experiments 101
4.3.1 Results 103
4.3.1.1 Hybrid Topo-Metric Framework 103
4.3.1.2 Hybrid Semantic Framework 105
4.4 Conclusion 106

5 Conclusion **119**
5.1 Summary of the Thesis 119
5.2 Future Work 121

References **123**

List of Figures

1.1 Figure 1.1a: Stanley - the winner of DARPA desert challenge. Figure 1.1b: Google car - is a self driving car project by Google. Figure 1.1c: The robot truck - that can drive itself is under the test by big truck manufacturers. Figure 1.1d: Amazon Drone is being tested to deliver goods to customers. Figure 1.1e: It is designed for transport tasks within the hospital. Figure 1.1f: A Mint autonomous floor cleaning robot. 3
1.2 Metrical maps. 6
1.3 Topological maps. 7
1.4 Semantic maps. 8
1.5 A traditional example of hybrid maps is given which is constructed by adding scaled distances and geometry to a standard subway map. 9
1.6 Loop closure affect the global consistency by minimizing the drift error. (Figure Courtesy of Brian Williams) 10
1.7 Global Mapping strategy is illustrated. Our map contains three different (at the same time mutually connected) steps: local metric reconstruction, topological map building and semantic path classification. **Top:** Topo-metric structure is shown under the graph. **Bottom:** Path classification step is illustrated. 13
1.8 Three junction types examples considered in our path classification step. **Top:** a left turn. **Middle:** 4 way junction. **Bottom:** T junction. Mostly one or two lanes urban roads along buildings are considered. 14

2.1 SLAM problem: Landmarks being observed at different positions along the robot's trajectory. Figure courtesy: Time Bailey (DWB06) 18
2.2 Taxonomy for classifying vision based metric SLAM schemes based on the model chosen for representing the environment. 20
2.3 Classification of vision based topological SLAM is shown based on the descriptor type chosen to model the appearance information from images. 24

2.4 A toy example of bag of words model. How it is constructed and how features are quantized. Figure 2.4a: Two dimensional feature descriptors are extracted on training images on which . Figure 2.4b: K-means clustering are used ($k = 4$). Figure 2.4c: Extracted feature descriptors are quantized to their closest clusters and those cluster ids are the visual words of the corresponding descriptors. Figure 2.4d: A histogram representation of the given image is built based on the extracted visual words and is used for image matching. (Figure courtesy of Kristen Grauman's lecture notes.) 29
2.5 A toy example of vocabulary tree building. Figure 2.5a: A two dimensional feature descriptor space. Figure 2.5b: Hierarchical clustering of the feature space with two levels($l = 2$) and a branching factor($k = 3$). Top level clusters are represented by green circles and separated by green lines and similarly, the second level by blue. Figure 2.5c: A vocabulary tree representation of the hierarchical clusters. Figure 2.5d: The quantization of leaf nodes is shown. (Figure courtesy of David Nister (NS06).) 31
2.6 Semantic maps built on topological maps. Figure courtesy: A. Pronobis (PJ12) 35
2.7 An illustrative representation of the classification of semantic mapping approaches is shown. 36
2.8 As an example of outdoors semantic mapping: pixel wise labeling with the help of dense 3D reconstruction. The top row shows the pixel wise semantic labeling. The middle row shows the dense 3D reconstruction and the bottom one is one of the given image to algorithm. Figure courtesy: S. Sengupta (SGST13) 38
2.9 Toy example illustrating how the environment can be represented with metric, topological and topo-metric model, respectively. Figure courtesy: Ioannis Kostavelis (KG15) 40

3.1 Loop Closure Detection Framework 48
3.2 Plot of scaled conditional entropies against 100 different choices of k with the minimum value highlighted as k^*. 53
3.3 Graphical model of our Naive Bayes framework. 55
3.4 Feature Shift analysis of a true match and a false match. 3.4a shows features matched across a pair of images which are acquired in the same place. Matches are shown with blue lines. Shift in X and Y coordinates of a matched feature pair is demonstrated with a red dashed line. 3.4b illustrates a false match of a pair of images acquired at different places. 3.4c and 3.4d show the plots of matched feature shifts ($(\delta x, \delta y)$ in Figure 3.4a) corresponding to the true and false match cases respectively. . . . 58

3.5 Figure 3.5a shows a re-traversal. Green trajectory is the previous traversal and the red trajectory is the current traversal. Figure 3.5b shows the components of the trajectory structure evaluation function $O(.)$. 60
3.6 Graphical representation of CPDs for conditional probability of minimum matches constraint. 63
3.7 Two-level feature descriptor quantization structure. The first level quantizes descriptors using a float 128 word vocabulary while the second level of quantization further quantizes the descriptors using vocabulary tree structure. 66
3.8 Precision Recall graphs of image level loop closure with and without odometry constraint. 71
3.9 PAVIN (Trajectory in red, loop closures in green) 72
3.10 Cezeaux (Trajectory in red, loop closures in green) 73
3.11 PAVIN-Jonco (Trajectory in red, loop closures in green) 73
3.12 Cezeaux-Sealiz (Trajectory in red, loop closures in green) 74
3.13 NewCollege (Trajectory in red, loop closures in green) 74
3.14 Bicocca (Trajectory in red, loop closures in green) 75
3.15 Run-times for various modules of the loop closure algorithm. Figure 3.15a plots run-times for the feature quantization time, VLAD extraction time and Node similarity analysis. Figure 3.15b shows the run-times of image similarity analysis and the total time taken to process each image frame. 77

4.1 A global modular view of our hybrid topo-metric mapping framework. NLC and ILS stand for nood level loop closure and image level loop closure detection respectively. CN stands for current node. 81
4.2 Flow chart. There are four main blocks given as image sequence partitioning 'ISP', local 3D reconstruction 'metric', path classification and loop closure detection. 108
4.3 Toy example illustrating how the environment (point cloud) is divided into a set $\mathbf{S}$ of $n = 8$ segments of uniform-length and parallel to the ground plane. 109
4.4 Spatial Hierarchical Inverted File models node N_j and image I_t membership information of visual words w_i as well as gives the keypoints coordinates which constructs that particular word. In this toy example, visual word w_i is observed at the pixel coordinates x_p^1, y_p^1 of the image I_p as well as at the pixel coordinates x_t^1, y_t^1 of the image I_t under the node N_k and at the pixel coordinates x_m^1, y_m^1 of the image I_m under the node N_l. 110
4.5 Snapshots from Pavin dataset. 111

4.6 Example frames from selected sequences are shown. 111
4.7 Precision-Recall graphs on the reference datasets. 112
4.8 The overall map of first PAVIN sequence. The green areas represent the loop closure and therefore the metric nodes. 112
4.9 The experimental results. The first figure shows the overall map. The green areas represent the topological parts and the red areas represent the metric parts. The second figure zooms into the roundabout while robot is turning around more than one tour and the third zooms into the area while robot is driving into a bend. . 113
4.10 Simplified level of path classification step. The first column shows the three different type of junctions from the sequences. The second column shows the 3D point cloud extracted while robot is traversing these junctions. The third column shows the dominant line segments relative to the robot trajectory coming from the road surface. The last column shows the result of our CRF model in response of the feature functions which are using 3D and 2D available information. 114
4.11 It shows that we miss the two right turns at the first passes. However, using the loop closure detections between first and second passes as well as second and third passes, these right turns are also added to our map. 115
4.12 Precision Recall curve for the sequence 00 and 05 respectively. The red curve shows the performance of our algorithm while the blue is showing the perfomance of FABMAP. 116
4.13 Dataset sequence trajectories plotted in blue. The detected loop closure regions are shown in green. 116
4.14 The results of path classification are shown on the real trajectory of sequence 00. There are 25 different junctions and some of them are visited by robot multiple times. Yellow squares show the detected junction and the red one shows the missed junction at which robot traverse through a straight road with a narrow right turn. 117
4.15 The results of path classification are shown on the real trajectory of sequence 05 on the left and sequence 07 on the right. There are 12 and 7 different junctions in each sequence respectively. Yellow squares show the detected junction and there are no missed junctions in these sequences. 118

List of Tables

3.1 Datasets Description. (Traj.=Trajectory Length, Vel.=Average Acquisition Velocity, FPS=Images Frames acquired Per Second) . 65
3.2 Manually set Parameters 67
3.3 Node Statistics with the preliminary framework. #(Nodes) - Number of nodes of the map built on the sequence. #(im.)/node - Average number of images represented by each node. (Traj.)/Node - Average trajectory length represented by each node. 68
3.4 Node Statistics. #(Nodes) - Number of nodes of the map built on the sequence. #(images)/node - Average number of images represented by each node. (Traj.)/Node - Average trajectory length represented by each node. 68
3.5 Recall values on six sequences. The recalls of our approach listed above are the best values obtained of the two variants of ILLC that are discussed. 75
3.6 The computational time of each step of our algorithm is given for the longest test sequence Cezeaux. Considering that the code is not thoroughly optimized and the code runs on a laptop with Intel Core I7 processor. 76

4.1 Parameters 90
4.2 The proposed feature functions are listed above and it is also marked that they are used in calculation of unary or pairwise potential. 95
4.3 Ground truth transcript of loop closure intervals of the Sequence 00 102
4.4 Ground truth transcript of loop closure intervals of the Sequence 05 103
4.5 Ground truth transcript of loop closure intervals of the Sequence 07 103
4.6 Node Statistics. #(Nodes) - Number of nodes of the map built on the sequence. #(images)/node - Average number of images represented by each node. (Traj.)/Node - Average trajectory length represented by each node. 104
4.7 Structure of the topo-metric map for PAVIN. 104

4.8 Structure of the topo-metric map for PAVIN-Jonco. 104
4.9 Computational Time per frame in milliseconds. 105
4.10 Comparison of ground truth number and detected number of loop closures are given. First row shows the total number of loop closures in overall dataset. Then the following rows give the values for each sequence separately. As it is seen our algorithm detects the loops without any false positives. 106

1

Introduction

1.1 Motivation

Future mobile robots such as intelligent vehicles are expected to navigate autonomously in complex large scale urban environments. In fact, recent promising examples illustrated in the figure 1.1 show that human drivers or more general guidance will be gradually replaced by autonomous systems with their developing perception, algorithmic, computing, memory and learning capacities in the not too distant future. The first row of figure 1.1 illustrates the development of autonomous cars from the winner of DARPA desert challenge 1.1a to the test vehicles on public roads 1.1b, 1.1c. The second row of figure 1.1 is a good illustration of that autonomous robots are already in our everyday life for delivering the goods we ordered online, cleaning our houses 1.1f, carrying specimens, pharmacy supplies, surgical equipment and other items between departments in hospitals 1.1e.

Fully autonomous cars are expected to be ready to set out on a journey as soon as 2017 or perhaps sometime in the 2020s. Although the timing may be uncertain, cars are already utilizing a lot of intelligent systems such as adaptive cruise control and assisted parallel parking. In fact, these are forerunners of future vehicle concept which can navigate from A to B while the people inside the vehicle just enjoy their journey. The biggest attention has focused on the sensors, other technological equipment and systems inside the cars as well as on the legal issues such as if an autonomous car causes an accident, who is to blame? what if the car was hacked? However, there is another crucial element: maps. While navigating in any kind of environment, a robot needs to be constantly computing its pose in its map of that particular environment to perform what is known as map based navigation. In other words, either a priori map of the environment is completely available or a map is being constructed at the same

time with exploration by the robot. In both cases, the map has an essential place in navigation or other autonomous tasks. In this work we aim to obtain a navigation oriented map for future autonomous driving applications at large scale urban environments by combining metric, topological and semantic information while the robot is autonomously or manually driven through the environment. .

1.1.1 Introduction

A robot which has the ability to perform the tasks as humans do in unstructured environments was only an utopic dream 20 years ago. However, autonomous capabilities of robots are increasing so fast as well as people's belief and acceptance of the integration of robots into daily life are becoming stronger nowadays. In fact, we start experiencing the first examples of these such as automatic vacuum cleaners, service robots, unmanned aerial vehicles and driver-less car projects etc. Therefore, robots moving around in future everyday environment and performing time consuming tasks of the daily life turns out to be a realistic expectation rather than an utopic dream.

The fascinating navigation and interaction capabilities of humans as well as animals are still unreachable for a robot. Especially how they achieve this by using noisy and partial information coming from their vision, tactile and audio sensors is an open question. To be able to achieve this level of autonomy in robot world, the limits of perception have been questioned in the artificial intelligence domain since the 1950s. What to conceive from surrounding environment, how to represent the conceived information and how computing as well as memory resources limit the perception are essential questions for a robot on executing simple and daily tasks (for an human or an animal) such as walking, driving, flying, climbing and swimming.

The first step of perception is to choose on-board sensor/sensors of robot. The most common ones are sonars, laser range finders and cameras. Among these, cameras are the most prominent and have been becoming the primary sensors in autonomous robotics recently as vision is our important sense for understanding the world around us. Endowing the robots with this ability opens a vast range of beneficial new applications because it contains rich information with a wide field of view and gives high potential on more general 3D spatial awareness and scene understanding. Therefore; cameras are potentially a very good choice as the main outward-looking sensor for a mobile robot. The main challenge lies in processing this huge amount of data to extract meaningful information in real time (within an exposure time) where it is still relevant.

A vision sensor creates an image which is a 2D projection of a 3D scene in its field of view over time. Computer vision is concerned for example with the inverse problem which is recovering 3D structure of a scene from sequence of 2D images.

(a) Autonomous Ground Vehicle

(b) Google car

(c) The Robot Truck

(d) Amazon Drone

(e) Robocourier

(f) Autonomous Floor Cleaning Robot

Figure 1.1: Figure 1.1a: Stanley - the winner of DARPA desert challenge. Figure 1.1b: Google car - is a self driving car project by Google. Figure 1.1c: The robot truck - that can drive itself is under the test by big truck manufacturers. Figure 1.1d: Amazon Drone is being tested to deliver goods to customers. Figure 1.1e: It is designed for transport tasks within the hospital. Figure 1.1f: A Mint autonomous floor cleaning robot.

It is known as structure from motion and there is a vast amount of work which can be seen as a step of imitating the visual understanding ability of a human. Solutions to the scene reconstruction and understanding are essential parts of localization, mapping and navigation for any autonomous robot application. For example, an autonomous car has to estimate its ego motion and then localize itself by analyzing the complicated traffic scenes continuously and real time as a human driver does.

Many of the complex tasks that are necessary for survival of robots in everyday life environments depend on general and fundamental capabilities such as mapping, localization, path planning and navigation capabilities. The first two of these are highly dependent to each other. In other words, a robot that estimates its localization by using only its on-board sensors needs to have the map of its environment. On the other hand, a robot that is to extend or build a map of its environment definitely needs to have its current localization. This problem is called as simultaneous localization and mapping (SLAM) and there is a vast amount of work on this especially at autonomous robotics literature.

Solutions to this problem using visual sensor alone is listed under Visual SLAM or Vision based SLAM. Although, the first solutions are concentrating on estimating camera poses and representing the structure of environment with a sparse 3D point cloud, the proposed solutions have diverged by time. They can be grouped as metric, topological, semantic and hybrid solutions. The main differences between them on the representation of measurement, level of abstraction and their general purposes. While metric maps concentrate on accurate localization within a local area, topological maps concentrate on global connectivity information. On the other hand, semantic maps focus on assigning a predefined set of labels to semantically describe various parts of the environment. Finally, hybrid maps focus on turning the weak points into a strength by combining specific advantages of the above discussed mapping strategies to solve a problem at hand.

In the remaining sections of this introduction we outline the visual solutions to mapping and localization problem.

1.1.2 Robotic Mapping

Cartography (mapping) has been integrated with human history. From cave paintings to 21st century, we mapped out the world and used these maps to define and navigate our way. As it is for us, mapping is also important part of autonomous navigation for robotics. Let's assume a robot which observes its previously unknown and uncontrolled environment by using its on-board sensor set. Integrating these partial observations coming from its environment into a consistent model is called as mapping. There are different kinds of maps developed by researchers in the mobile robotics community. The most popular ones are listed

as

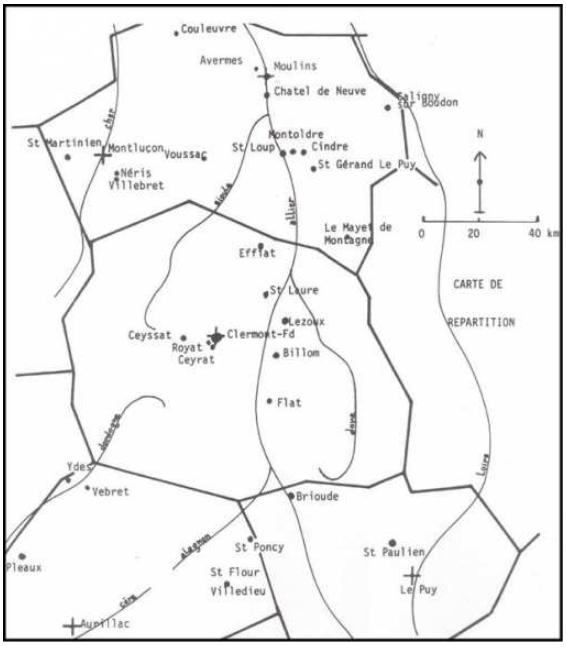
- Metric maps
- Topological maps
- Semantic maps
- Hybrid Maps

1.1.2.1 Metric maps

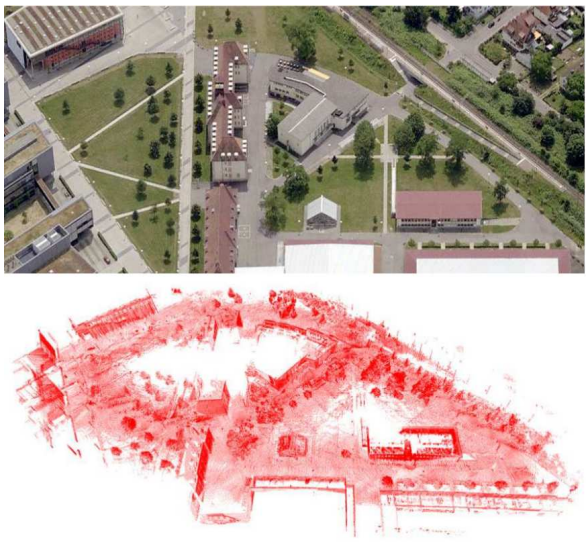
Metric maps model the environment in terms of distances which corresponds to actual real world geometry. A common and well-known example of metric maps is a city map as it is shown in figure 1.2a in which the actual distance between two places in a metric map of a city can be measured. In a similar way, this can be observed from mobile robot point of view. As it is shown in figure 1.2b, a metric map provides the relative distances between the surrounding objects and the robot poses under a common coordinate system. Generally, the resulted map contains 3D points, planes or objects in 3D based on the level of complexity and the robot poses given in a default Cartesian coordinate system. Metric maps are preferred when precise self-localization and path planning are extremely important and the size of the area is limited.

1.1.2.2 Topological maps

Topological maps model the environment in terms of its structure and connectivity independent of metrical information. In fact, it does not need metric information and a global coordinate frame. Generally, the resulting map has a graph structure which has nodes and edges. Nodes correspond to specific places in the environment and edges represent their connectivity to each other which marks the possibility to traverse from one node to another. The most common examples of topological maps in everyday life are subway maps. As it is shown in figure 1.3a, each station is given as a node and connection information between them is given as edges. In robotic mapping, topological maps are common for the cases which involve searching using connectivity such as loop closure detection, path planning and etc because they mainly focus on connectivity between places. A toy example is shown in figure 1.3b which illustrates how an environment can be modeled in an abstract manner by using a graph structure. The simple and compact structure of topological maps allow efficient scalability and optimization capabilities in robotic mapping.

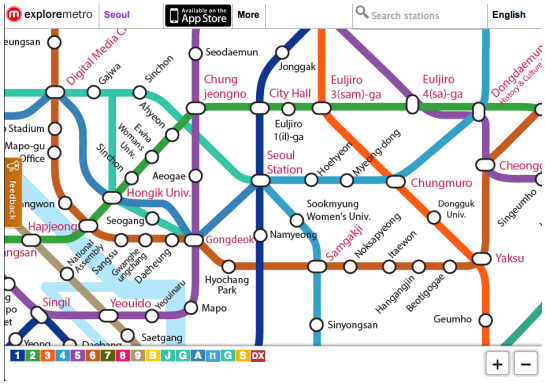


(a) A traditional example of metric maps is a geographic city map.

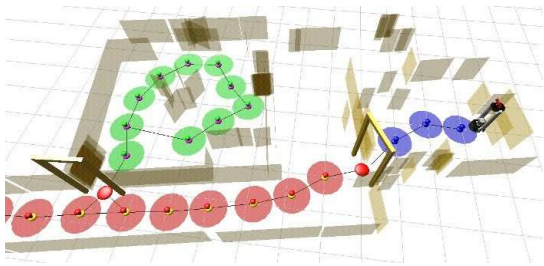


(b) An example metric map (consists of 3D point cloud) built by a robot at the university of Freiburg (courtesy of Kai. M. Wurm) and its relative satellite image.

Figure 1.2: Metrical maps.



(a) A traditional example of Topological maps is a subway map.



(b) An example topological map built by a robot in an indoor environment. (courtesy of Andrej Pronobis)

Figure 1.3: Topological maps.

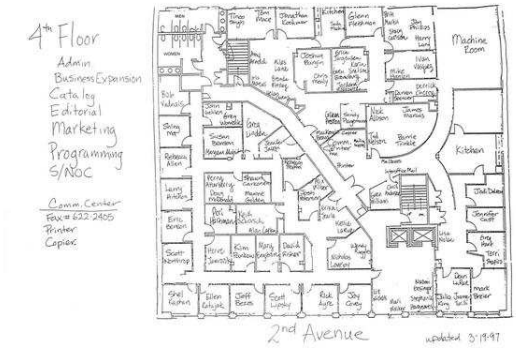
1.1.2.3 Semantic maps

Semantic information means that the relations between spatial entities and a set of predefined abstract concepts which are meaningful for humans. For example, we use room, corridor bathroom, toilet, etc as abstract concepts to simplify the layout of indoor environments. Similarly, road, turn, intersection, road bend etc are the common examples to these for outdoor environments. Thus, semantic mapping is a way of modeling the environment which gives the relation between spatial world and these concepts.

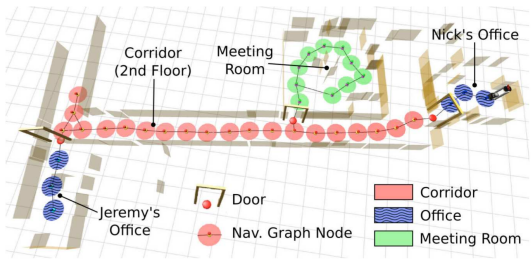
For example, a home layout tagged by names such as kitchen, bedroom, corridor can be considered as semantic map 1.4a . From robotic point of view, a semantic map similarly provides the identification of the signs, symbols, objects, places which are meaningful concepts for humans as well as gives the relationship between all of these. In other words, semantic maps model the world intelligently by employing these concept as humans do. They are used for making decisions at a high level and they provide better robot human interaction capability as well as they endow a robot with the capacity of understanding the functionality of the surrounding environment.

1.1.2.4 Hybrid maps

Hybrid maps are simply defined as a combination of different maps. A well known example of that kind are topo-metric maps.Although there is not a common definition of it and they are more specific to implementation, they are basically exploiting metric and topological information under the same map. The main reason for hybridization metric and topological maps is that they both have ad-



(a) A traditional example of semantic maps is a layout of an office floor.



(b) An example semantic map built by a robot in an office. (courtesy of Andrej Pronobis)

Figure 1.4: Semantic maps.

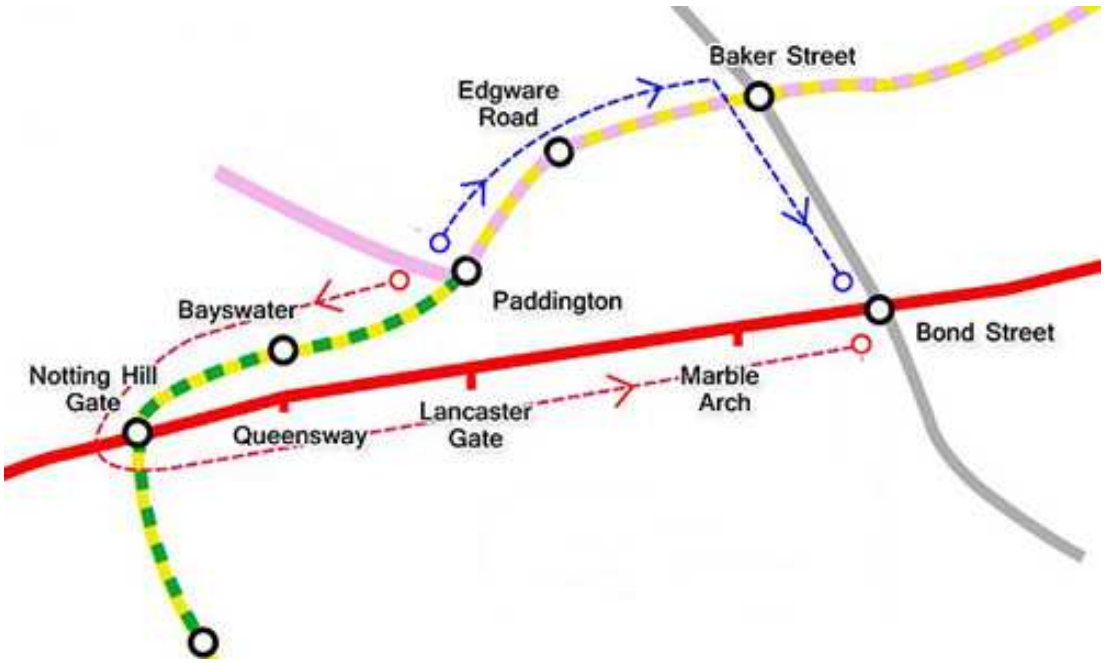
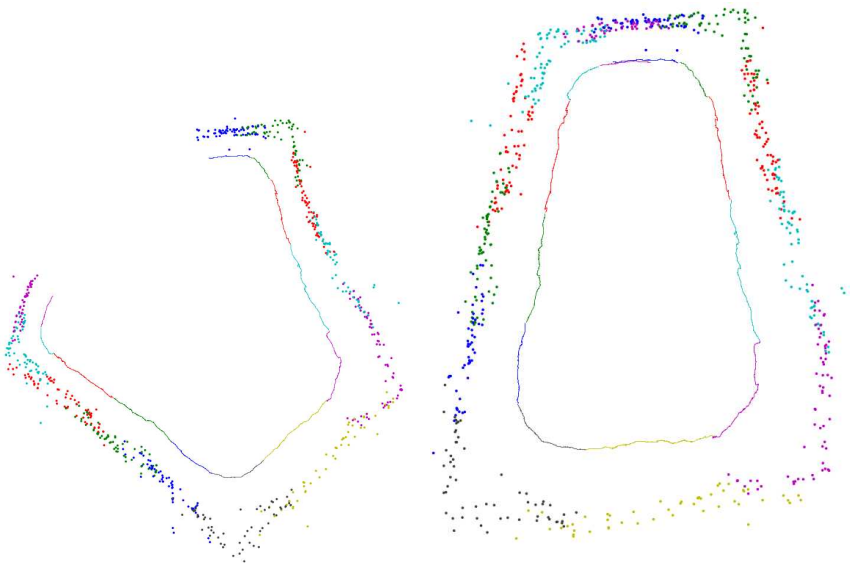


Figure 1.5: A traditional example of hybrid maps is given which is constructed by adding scaled distances and geometry to a standard subway map.

vantages over each other due to their different characteristics in ontology. Moreover, what is an advantage for one approach might be a disadvantage for the other. If we combine a standard subway map with a geographic map by adding scaled distances and geometry between each stop, then we obtain a hybrid map as it is illustrated in the figure 1.5. As it is seen in topo-metric map example, hybrid maps generally focus on turning the weak points into a strength to solve a problem at hand by connecting elements in one map to elements in another map.

Especially if mapping is investigated in terms of autonomous driving application, they have to be reassuring.

Autonomous cars need maps that differ in several important ways from the traditional robotic maps we use today. They should be hi-definition in the sense of containing different types of information and models. Metric maps can be good enough for GPS-based navigation in a limited area, however autonomous cars will need maps that can tell them where the curb is or where the intersections are located within a few centimeters. They also need to be updated regularly with information about accidents, maintenance and traffic management, and lane closures. Therefore, hybrid mapping as combining different models is a must to achieve autonomy in real environments.



(a) The mapping result before loop closure.

(b) The mapping result after loop closure.

Figure 1.6: Loop closure affect the global consistency by minimizing the drift error. (Figure Courtesy of Brian Williams)

1.1.3 Loop Closure

Loop closure detection plays an essential role in all mapping approaches in order to construct a consistent model of an environment. It consists in recognizing if the robot is currently revisiting a previously mapped area. Especially for performing large scale mapping, it is a crucial component which recovers the mapping system from critical errors. Figure 1.6 shows the effect of loop closure detection clearly. Figure 1.6a illustrates the result of a metrical mapping algorithm without loop closure detection component. There is a critical level of drift error which results with a divergence of the map in large scale. Only with adding loop closure detection step to the same algorithm, it is shown that this error is minimized and the map keeps the global consistency 1.6b. This effect is observed more strongly in topological mapping. In fact, loop closure is an inevitable step of capturing the topological structure of the environment by discovering connections between nodes that are not explicitly adjacent.

While solving this problem, other challenges such as perceptual aliasing, measurement noise and scalability issues arise. Main issue is to eliminate false positives due to the fact that even a single false can result with fatal errors in mapping process.

1.2 Thesis Roadmap

This thesis mainly focuses on two problems each of which are discussed in the following subsections.

1.2.1 Loop closure for Topo-Metric Mapping

Problem Description: Appearance based loop closure detection for topological and hybrid map building in large scale outdoor environments is proposed. Instead of using fixed thresholds a learning algorithm for inferring the vital parameters of loop closure is employed.

The loop closure process is defined under hierarchical map representation aiming to achieve efficient detection. A hierarchical topological map builds a graph whose nodes represent a group of images. We can understand these maps as a two level hierarchy with the first level composed of nodes (node level) and the second level composed of images belonging to each node (image level). In fact, it is the last step before passing to the hybrid approach which will be explained in chapter 4. Given a query image acquired at the current location, our algorithm retrieves the most similar node/place(s) and then retrieves the most similar image among the images that belong to the most similar place(s). The process of retrieving the most similar node is called the *Node Level Loop Closure* and that of retrieving the most similar image is called the *Image Level Loop Closure*. The size of the search space for node level loop closure is the number of nodes and that of image level loop closure is the number of images belonging to the retrieved nodes. Both of these search spaces are far smaller than the total number of images acquired, which is the search space size of many existing loop closure techniques (AFDM08), (ADMF09), (CN08b), (CN09), (CN10b), (GLT11). Given that the individual computational complexities of node and image level loop closures do not scale linearly with the number of images acquired, our hierarchical loop closure process is bound to be faster than most of the traditional approaches that will be discussed later in chapter 2.

In order to effectively capture visual similarity across images in a region, Omnidirectional/Panoramic cameras with their 360 degree field of view is a natural choice as compared to the conventional cameras with a limited and unidirectional field of view. Apart from that, omnidirectional cameras are also useful in detecting loop closures when the robot is re-traversing the environment in a reverse direction. This is impossible with traditional cameras. These two reasons justify our choice of omnidirectional cameras for the present approach.

The primary contributions of this method are the following:

- A node level loop closure algorithm (section 3.3) using Vector of Locally Aggregated Descriptors (chapter 3.2). The parameters required for node level

loop closure are automatically learned as opposed to empirical evaluation.

- An image level loop closure algorithm posed as a Naive Bayes classification problem (chapter 3.4) using four similarity metrics. This approach bypasses the need for geometric consistency check popular with many traditional approaches.
- Parameters for the two levels of loop closure are automatically learned from training data. In the majority of loop closure detection approaches this process was done empirically by studying the precision-recall graphs.
- Experimental results and analysis of accuracy of our loop closure algorithm on various public datasets(Chapter 3.5).

1.2.2 Hybrid Mapping

Problem Description: A novel vision based hybrid mapping framework which exploits metric, topological and semantic information is presented.

Two strategies on development of hierarchical mapping framework are proposed. The first strategy is the combination of our hierarchical loop closure algorithm and 3D reconstruction algorithm. Therefore, it is called as hybrid topo-metric map. The second upgrades this combination by adding semantic information into it. In other words, the semantic information allows us to tag the environment and also to decide automatically between metric and topological model. Therefore, it is named as hybrid semantic map.

The main contribution of the first strategy is how to combine metric and topological information by focusing on separability of maps and hierarchy. It proposes a hierarchical map representation which uses our image sequence partitioning (ISP) technique. The hierarchical map built can be understood as a topological map with nodes corresponding to certain regions in the environment.

The main contribution of the second strategy is how to utilize the semantic information into topo-metric approach. In order to make semantic domain utilizable in outdoor environments, we need to endow robots with human point of view. An important strategy used by humans to describe locations in a city is to abstract high level concepts from it and then represent the environment with them such as junctions, straight roads and turns. Especially considering urban areas and road networks, each of these concepts has significant functionality to integrate semantic knowledge over space, however the traditional robot maps concentrate only on how to represent the spatial structure and they miss these important concepts. For example, a metric map may represent the structure of a road but it does not pinpoint whether this road is straight, bent, turn or a junction. Moreover, it does not even mark that the given structure is a road. It is

Figure 1.7: Global Mapping strategy is illustrated. Our map contains three different (at the same time mutually connected) steps: local metric reconstruction, topological map building and semantic path classification. **Top:** Topo-metric structure is shown under the graph. **Bottom:** Path classification step is illustrated.

also similar in topological map. It gives the connection information between two nodes but it does not indicate the type of the connection if it is through a junction turn or a straight road. In fact, we call this extra information as semantic and the maps which are integrating this into the traditional robotic maps as hybrid semantic maps. Thus, we propose to use the same strategy for a large scale robot map and we propose a multi-layer approach in which each layer corresponds to different level of abstraction and represents the environment as precise as it is required.

As the first layer, we construct local metric maps that are represented as 3D point clouds while the robot traverses through the environment.These local 3D point clouds are combined with their appearance based information to semantically label the local robot path (road in our case) as straight road, road junctions

Figure 1.8: Three junction types examples considered in our path classification step. **Top:** a left turn. **Middle:** 4 way junction. **Bottom:** T junction. Mostly one or two lanes urban roads along buildings are considered.

etc. Meanwhile, the environment is also divided into discrete areas named as nodes by using an appearance based similarity measure. Nodes consist of visually similar image groups in which visual appearance is almost identical and form a topological graph as the second layer. To obtain the final layer, we use the extracted semantic information and combine the topo-metric nodes under semantic labels which facilitate loop closure detection faster and represent the map in a human friendly way. These layers are illustrated in the figure 1.7.

The final layer has the highest abstraction level. It is constructed by combining the adjacent and semantically identical nodes under discrete places. Our map can be understood as a graph with these places, and edges which connect spatially adjacent places in the environment. The places are classified into three types depending on the semantic information they carry:

1. *Straight nodes*, that contain images acquired on a straight road.

2. *Curved nodes*, that contain images acquired on a road that is curved.

3. *Junction nodes*, that contain images acquired at a junction of roads. The

junction types which are considered in our map is also illustrated in the figure 1.8.

In the map, straight nodes are represented with 2D features extracted from images. In addition to the 2D information, 3D information is also used for curved and junction nodes. The motivation behind this configuration and representation of our map is to facilitate quick loop closure and keep the complexity under control.

A novel map representation which exploit metric, semantic and topological information under a common model for urban area mapping is proposed. To this aim, local metric maps constructed in dynamically defined windows and estimated camera path as well as 3D points are used for classification such as straight road, bends and junction areas. This 3D reconstruction based classification step is supported by road/curb border detection to increase the robustness of junction detection. Especially separating the junctions and checking the loop closure only at these areas facilitate quick and more robust loop closure detection. Finally, it is also a suitable model for map matching and merging algorithms given well extracted junction nodes.

2

Literature Review

This chapter provides the necessary literature review in the lines of the work presented in this thesis. A brief review of mobile robotic SLAM will be provided followed by a detailed review of appearance based topological mapping. Subsequently, hybrid mapping approaches and visual memory based navigation approaches are reviewed.

2.1 Simultaneous Localization and Mapping

Initially, mapping and localization which are the two main steps of autonomous navigation of mobile robots were investigated separately. However, they are directly dependent which means that a correct map is necessary for localization and similarly a precise self-localization is necessary for constructing a map. Chatila and Laumond (CL85) and Smith et al. (SSC87) are the first works which considers this dependency. Based on these works, Hugh Durrant−White and John J. Leonard (LDW91) call this problem originally as SMAL and then change it to SLAM for a better impact. On the other hand the same problem is called as CML (Concurrent Mapping and Localization) by Newman et al. (NCH06) and Andrade and Snafeliu (ACS02). SLAM or CML is concerned with the process of building a map of an unknown environment by a mobile robot while using this map to localize itself at the same time. Figure 2.1 visualizes the dependency of localization and mapping.

The sensor suit of the mobile robot is quite important in order to build a map from the environment. These sensors which are used to perceive the surrounding world are grouped under exteroceptive and proprioceptive sensors. The well−known examples of exteroceptive sensors, which determine the measurements of surrounding objects relative to a robot's frame of reference, are sonar (TNNL02, RRTN08) , range lasers (NLHS07, TBF98), cameras (Dav03, SLL05,

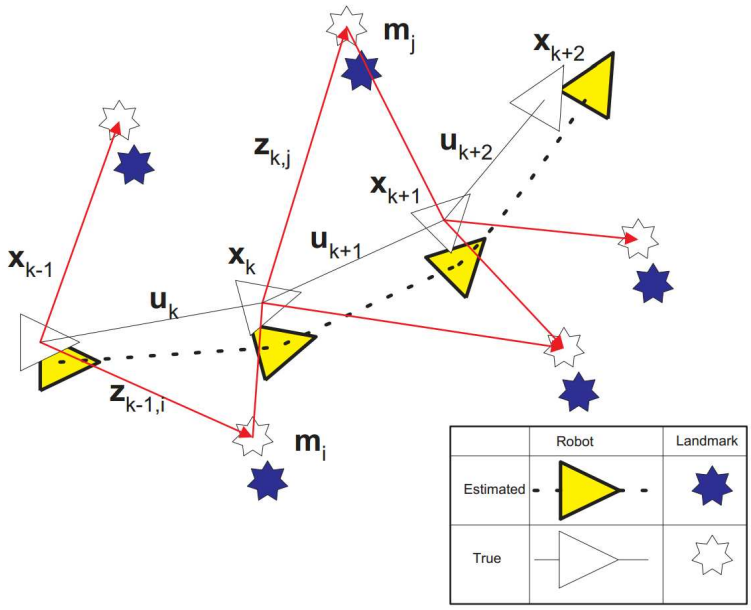


Figure 2.1: SLAM problem: Landmarks being observed at different positions along the robot's trajectory. Figure courtesy: Time Bailey (DWB06)

LBJL07, BRSM+09) and global positioning systems GPS (TBF05). Although the first two aforementioned sensors provide accurate and dense information of the surrounding world, they are impotent in highly cluttered environments as well as they are expensive and heavy. On the other hand, a GPS sensor alone is insufficient to update the map although it provides a global location of a robot. The second group of sensors measure a signal originating from itself and are responsible for monitoring self maintenance as well as controlling internal status. Encoders, accelerometers and gyroscopes are the most common examples of proprioceptive sensors used in mobile robots. These provide an incremental estimate of robot's motion which is known as dead reckoning navigation. However, they are insufficient to estimate the robot's position due to the cumulative error factor. Finally, there are some works which propose to fuse the information obtained from different types of sensors in order to obtain an accurate and robust perception of the surrounding environment and robot's state (CNT01, TBF05, NWSS11). The disadvantage of these methods are increasing cost, weight, computational and power requirements of a mobile robot. Therefore, we concentrate on investigation of systems which can locate itself and build a map with using cameras alone as exteroceptive sensors.

Historically, SLAM methods are divided into metric and topological approaches. Recently hybrid and semantic methods are also added to these. Each of these approaches will be investigated in the following sections

2.2 Vision Based Metric SLAM

During the last decade, vision sensors become a prominent external sensor suit of mobile robots for localization and mapping tasks (DRMS07, KM07, PT08, PPTN08). Camera based systems provides rich amount of information such as color, texture, appearance and structure of their field of view as well as range information.Moreover, they are cheaper, lighter and need less power to operate compared to the other sensors. Besides these advantages which make them popular, they have also some limits coming from sensitivity to lighting changes, insufficient camera resolution, motion blur and so on.

Before going into the details of vision based metric SLAM approaches, we will shortly mention about camera calibration process. Calibration which means that estimation of intrinsic (focal length and principal point) and extrinsic parameters (rotation and translation with respect to a reference coordinate system) of cameras is an important step for vision based systems which aims to achieve doing SLAM. A group of images that captures a calibration pattern from different angles and distances are used to estimate these two group of parameters at calibration step (Zha00). This is called as off−line calibration while it can be also carried out online. Assuming that intrinsic parameters are fixed, offline calibration methods are more popular in SLAM applications because it decreases the parameters to estimate online.

The early examples of vision based navigation offers to use a binocular stereo system (SLL02, OMSM03). The main disadvantage of using a binocular or trinocular stereo systems is its high cost. If we continue the classification based on camera systems, there are recent multi camera based systems (KD10, CAD11). Other ways of augmenting visual field of view are to use wide−angle (fisheye) cameras (Dav03) and to use omnidirectional cameras (SS08). Finally, RGB-D sensors which provides color and depth information together are used in indoor SLAM applications (HBH+11).

Binocular, trinocular and multi camera systems except of the ones without any overlapping between the views are grouped under stereo systems. Their main advantage is to yield an estimation of real 3D positions of the landmarks in their field of view. The prominent and recent examples of stereo SLAM approaches are the work of Konologie (KA08, KBC+10) and Mei et al. (MSC+09).

On the other hand, single camera based systems are grouped under monocular SLAM or MonoSLAM (Dav03) approaches. It brings a simple, flexible and computationally and economically low−cost solution to the SLAM problem. The weakness of monocular SLAM is the unknown scale problem (Nis04, SMD10a) which means that the depth of a 3D point can be estimated only up to scale. In other words, it is a partially observable problem in which a single observation is insufficient to calculate the depth of a landmark in the field of view. Therefore,

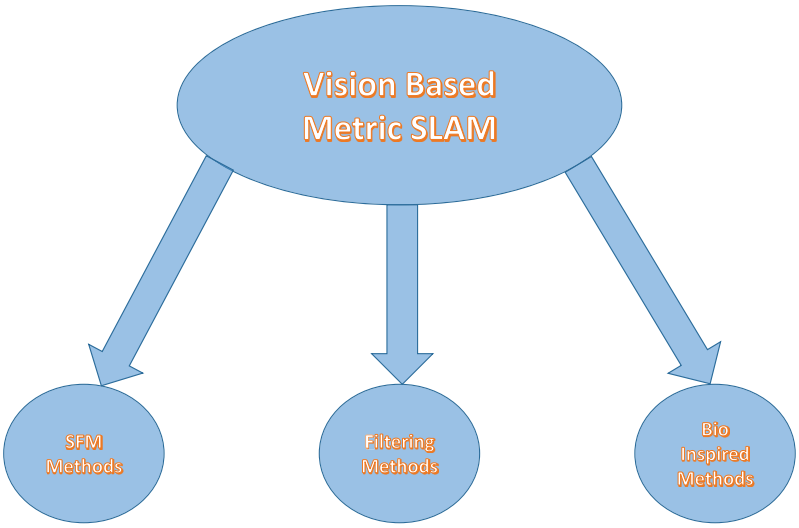


Figure 2.2: Taxonomy for classifying vision based metric SLAM schemes based on the model chosen for representing the environment.

matched features must be tracked across multiple views to be able to obtain 3D structure of the environment.

The approaches in the literature which aim to solve the visual SLAM problem will be given in the following sections. We mainly review a number of related works under three title,

- Structure from Motion SFM based solutions.
- Filtering methods for SLAM.
- Biologically inspired methods for SLAM

All approaches are used for stereo vision as well as monocular vision. A graphical description of this classification is shown in the figure 2.2.

2.2.1 Structure From Motion Methods

Structure from motion SFM is to obtain 3D scene reconstruction as well as camera position from small sets of images (PGV+04). It is a well−studied research area in computer vision and its principles are derived from photogrammetry. It has mainly 4 steps such as feature extraction, matching, triangulation and a non-linear optimization. The last step is known as Bundle adjustment BA (TMHF00)

which performs a batch optimization over camera poses and 3D points to minimize the re-projection error.

SFM permits precise estimation of camera pose which can be seen as localization step. It does not guarantee to construct a consistent 3D structure of the environment which can be seen as mapping step. In fact, using vanishing points is a good example of estimating rotation of camera without 3D struture information (BSD+12). Visual odometry (Nis04) which aims to calculate the camera pose of each video frame and 3D positions of matched features is a good example to this fact. By adding Local bundle adjustment step to visual odometry system, Mouragnon et al. (MLD+09) construct trajectories up to 500 meters. While the main advantage of visual odometry is to permit to utilize thousands of features per frame which is a huge number compared to filter based methods, its main disadvantage is the lack of loop closure detection ability which is an important step to obtain consistent maps.

Instead of using point features, there are line-based SFM solutions which are either using only line features or the mixture of line and point features (Har97, AD03, LDVP14). Especially for spherical images and heterogeneous cameras, line features are more stable than point features in the detection and matching due to the severe distortion (LDVP14). Another advantage of using line features is that they are are common in urban environments.

Parallel tracking and mapping PTaM (KM07) is a well-known recent monocular SLAM technique. The structure of the system is based on two parallel threads. First one is to track the features and the second one is to construct 3D point maps by using BA.

Rather than using individual 3D features, there are approaches which consist of a non linear constraint graph between selected images. FrameSLAM (KA08) and View−based maps (KBC+10) are implementation of this idea by using a stereo vision based system in literature. They give promising results especially for long trajectories passed through challenging environments thanks to their graph based representation model and loop closure ability.

Strasdat et al. compares SFM and filter based methods (SMD10b). This work shows that the accuracy is related with the number of matched features rather than the number of selected images which is an essential property of SFM. Their results show that the techniques which use BA gives more accurate result while probabilistic filters are more convenient for the environments in which there are high level of uncertainty.

2.2.2 Filtering Methods

Filtering approaches use probability distributions defined over extracted features and estimated camera poses to fuse and summarize the information up to the

current time. Extended Kalman Filter EKF, Factored Solution to SLAM FastSLAM, Maximum Likelihood ML and Expectancy Maximization are the examples of probabilistic filters that are used in SLAM solutions (TBF05). Although they give good results in a limited environment, they are not appropriate for large environment.

Stochastic mapping concept first introduce by Smith et al. (SSC90) as an Extended Kalman Filter based incremental solution to the SLAM. Using nonlinear models of observation and transition EKF SLAM recursively update a state vector which consist of camera pose and 3D points. Its main assumption is that the recursive propagation of the mean and covariance of probability density functions which represent the uncertainty are close to the optimal solution. Therefore, it is sensitive to the bad associations of measurements. Another weak point of EKF SLAM is its quadratic complexity growth with the size of the map. The works such as Atlas Framework (BNLT04), Compressed Extended Kalman Filter CEKF (GN01), Sparse Extended Information Filter SEIF (TBF05), Divide and Conquer (PPTN08) and Conditionally Independent Submaps CI−Submaps (PT08) focus on this weak point by using submapping techniques.

FastSLAM (MTKW02, MTRW03) utilizes Rao-Blackwellized particles for pose distribution and EKF for its map. The main blocks of the algorithm are a particle generator, re sampling process which prevents the degeneration of the particles over time. It achieves a logarithmic computational cost $O(plogn)$ where p and n are the number of particles and features on the map respectively. The deficiency of the algorithm is that using many particles increase the requirement of memory and computational cost however using few particles is insufficient to obtain accurate results. Moreover, it is impossible to determine the optimum number of particles which is necessary beforehand.

MonoSLAM proposed by Davison (Dav03) is a first monocular EKF SLAM system which works real time.It achieves to construct a 3D metric map and to estimate 6 degrees of freedom camera poses simultaneously at 30 frames per second.

A constant velocity model is used as a motion model. Although it limits camera mobility because the model can not correctly deal with sudden movements, it is good for simplicity and real time constraint. Especially for vehicles, it is a sufficient model, however; it fails under the rapid hand held or wearable camera movements cause severe errors due to the limits of the chosen motion model. Moreover, it works in indoor environment in which there are a limited number of features with a small displacement between frames due to the limits of EKF formulation. Gee et al. (GCCMC08) updates the MonoSLAM by using an advanced motion model which can deal with acceleration and also optimizes the system so that operating speed can go up to 200 HZ. However, it can work real time only for a few seconds due to its extremely growing computational cost and memory

requirements.

Inspired by FastSLAM Eade and Drummond (ED06) also propose to use particle filter to increase the number of features on the map. By modifying MonoSLAM with a hierarchical mapping technique and Geometric Constraints Branch and Bound GCBB which allows to detect large loops, Clemente et al. (CDR+07) propose a SLAM approach for large outdoor environments. Finally, Civera et al. (CDM08) focuses on the feature depth initialization problem of monocular SLAM. While Davison (Dav03) utilizes a delayed initialization technique, Civera achieves to perform an undelayed feature initialization in an EKF-SLAM system.

2.2.3 Biologically Inspired Methods

Many animals have the ability to follow habitual routes between important locations. Although it is still not known how they achieve to encode their routes, there are few interesting work inspired from this fact.

RatSLAM (MWP04) which use models of the hippocampus of rodents can build consistent and accurate maps of complex environments by using a single camera alone. Then he extends his work with adding promising results obtained in large indoor and outdoor environments (MW08). Moreover, his work has a strong loop closure detection capacity even using the sequences which are captured at different hours of day.

Another interesting work investigates navigational mechanism of desert ants (Col10). Their research focuses on understanding how ants navigate using visual information than the implementation on a robot. However, they claim that the proposed solution has potential to implement as a SLAM system on a robot.

2.3 Vision Based Topological SLAM

In topological SLAM approaches, the environment is modeled in an abstract manner compared to the metrical approaches. In other words a graph consists of discrete locations called as nodes and edges link them by modeling the relations between them to represent the environment. In contrast to the metrical SLAM approaches, they result simple, scalable and compact representations of the environment. Although we talk about the graphical structure of topological SLAM approaches, it does not mean that all graphical models can be in the field of topological methods. For example,the pose graph SLAM which represent the environment with a graph is a metrical approach because the nodes of a pose graph represent sampled metric positions of a robot. However, the nodes of a topological graph represent distinct places based on appearance information.

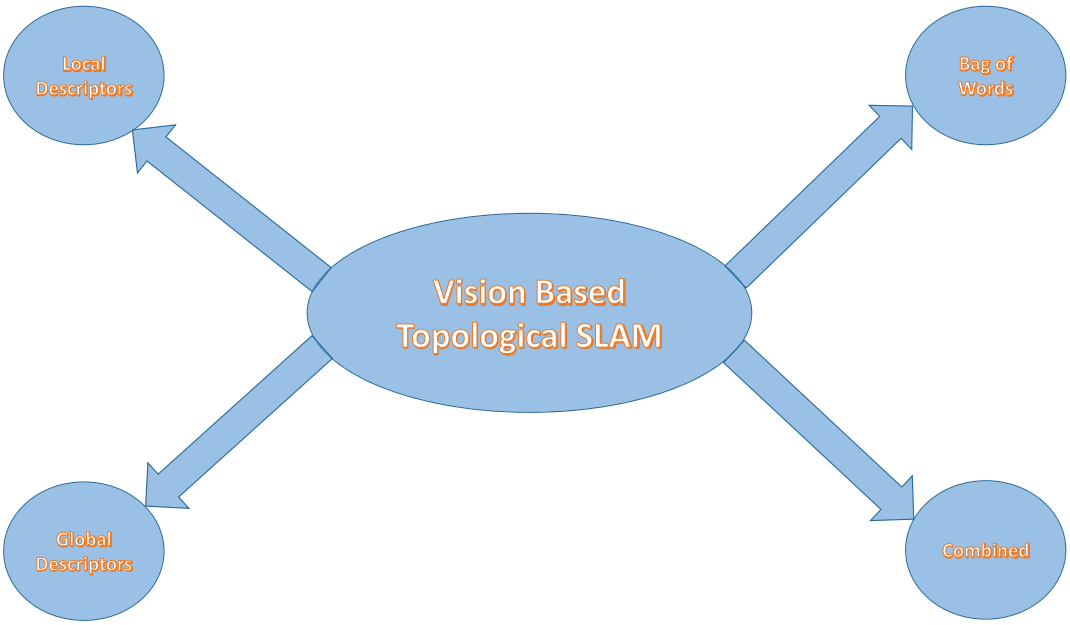


Figure 2.3: Classification of vision based topological SLAM is shown based on the descriptor type chosen to model the appearance information from images.

Loop closure detection component stays in the center of topological mapping approaches. In fact, pure topological mapping absolutely depends on this step for accurate map generation. Given a query image, the loop closure step searches if the image belongs to a previously mapped place. If it finds a match from map, then the current image and the previous image are linked in the topological graph. Wrong loop closures may have fatal impact on the accuracy of the maps. In fact, these create false edges between nodes or create a graph of redundant nodes which does not represent the topology of the environment. Hence an accurate loop closure approach which focuses on obtaining zero false positives and minimizing the number of false negatives is essential for topological SLAM algorithms.

We classify the topological SLAM approaches based on the visually describing method. The works in the literature can be grouped under four titles such as,

- Global descriptor based methods.
- Local features based methods.
- Bag of words based methods.
- Combined methods.

A graphical description of this classification is illustrated in the figure 2.3.

2.3.1 Global descriptor based methods

Global descriptors are used to encode the complete image in a way that allows it to be compared and matched to other images. Although they are generally not very robust, they are fast in computation and matching which directly accelerate the SLAM process. Therefore, they are used in several applications such as scene classification, image registration and retrieval and topological mapping.

First examples of global descriptor based SLAM approach are the ones which use histograms. Histograms represent an image in a compact way by using a particular feature such as color or gradient orientation information. A topological localization method in which each image is represented by six one dimensional color histograms is proposed by Ulrich and Nourbakhsh (UN00). Using color histogram in combination with a Bayes Filter, Werner et al. (WMS09) proposes a topological SLAM approach in which the Bayes filter helps to handle places with similar appearance. Histograms that are used for localization and mapping systems are not only based on color information. A topological map is built by using gradient orientation histograms as global image descriptor (KLY05). He builds a graph structure whose nodes consist of sets of representative views and at the navigation step histogram of each query image is compared with each node representatives to localize the robot. Bradley et al. (BPVT05) improve this work by using Weighted Gradient Orientation Histogram and they test their approach in large outdoor environment. Similar to this work, Weis et al. (CAHA07) propose a topological localization approach for outdoor environments by dividing image into grids and computing 8×8 histogram of integral which are invariant features to translations and rotations. Orientation Adjacency Coherence Histograms which are computed based on Harris detector response are another type used in topological localization approach (WZC06).

The global GIST descriptor which is initially developed for scene recognition (OT01, OT+06) is another popular global descriptor used in topological SLAM approaches. Inspired by how humans classify images, the spatial envelope of the scene which consists of a set of perceptual dimensions namely naturalness, openness, roughness, expansion and ruggedness is estimated by using a bank of filters. Finally , a global descriptor whose dimension is reduced by using Principal Component Analysis is derived.

Singh and Kosecka (SK10) are the first researchers who propose GIST based similarity measure between panoramic images and they utilize this for loop closure detection in Manhattan world like environments. Murillo et al. (MCKG10) proposes omni−gist which is modified version of GIST descriptor for omnidirectional images and then proposes a hierarchical topological SLAM approach which uses omni−gist. Another interesting work combines GIST and BRIEF which is a binary feature descriptor (CLO+12) and introduce to use BRIEF−GIST descrip-

tor in a SLAM algorithm designed for large environments (SP11). Projecting GIST descriptors into a low dimensional space and using it in particle filter is also proposed for efficient loop closure detection (LZ12). GIST is also tested on spherical images but the researchers concludes that it is not suitable for this kind of images due to lost of sphere spatial periodicity (CRF12, CRF13).

Extracting fingerprints of places and using them for topological SLAM application (LNJS01) is a specific method developped for omnidirectional images. A vertical edge detector is used with a color patch detector to obtain a global descriptor and then a minimum energy algorithm is proposed to match these descriptors. Then Tapus et al. (TS05) augment this work by adding a feature uncertainty model in order to obtain better localization results. Similarly, Fast Adaptive Color Tags algorithm (LSP+09) which first divides an omnidirectional image of an indoor environment based on vertical image in order to compute the average color value in the U−V space for each region and then connects each region descriptor in a global vector is used for a mobile robot mapping system. An improved version of this work using a Dirichlet Process Mixture Model is called as DP−FACT (LZ12). Computing invariant signatures of omnidirectional images based on Haar invariant integrals (LICM11) is the most recent method of extracting fingerprints of places. Later on, these signatures are strengthen against perceptual aliasing by using a different representation of omnidirectional images and a new integrative kernel and they are used in placed recognition and robot localization (MLIM12).

Exploiting the 360 degree FOV and the specific geometry features of omnidirectional images, an active contour algorithm (MLIM11) is proposed to extract the navigable space in the surrounding environment. Similar to the construction of local Voronoi diagram extracted from a laser range finder, this method is utilized in extracting the topology of the environment based on omnidirectional images (MLIM14).

Aiming to obtain robust descriptors to illumination changes and weather changes for long term SLAM system, there are also recent works which can be counted under global descriptor based methods. Dird is an Illumination Robust Descriptor (LBKS13) and SeqSLAM (MW12) are recent example of these.

2.3.2 Local features based methods

A local image feature captures an interesting pattern which shows difference within a patch of images. It is a specific structure such as a corner, edge, blob or region. After extracting local image features, a descriptor building step which is based on the measurements from the neighborhood of each feature starts. The resulting descriptor can be a floating point vector or a simple binary vector as bit strings. Local descriptors are more robust to changes, partial occlusions and

camera rotations between the matched images. Survey on the local image feature detectors (TM08) lists the features of a good detector such as repeatability, distinctiveness, locality, quantity, accuracy and efficiency. Among these repeatability is the most important feature which proves its robustness to small changes and its invariance against large changes.

Using local image features for topological SLAM approaches starts with the Scale Invariant Feature Transform SIFT (Low04a). Kosecka and Yang (KLY05) propose to use SIFT for their global localization algorithm designed for indoor environments. They improve their work by adding a feature selection strategy which measures the ability of each extracted SIFT features in the sense of describing places (LK06).

Extracting and matching features for each image is a costly task therefore some researchers focus on accelerating this task by maintaining only persistent features instead of all. In order to construct a topological map incrementally, Rybski et al. (RZL+03) apply this idea by using Kanade−Lucas−Tomasi KLT feature tracker. For the same purpose, manifold constraints (HZM06) are used to find the best features for representing the topological places. Similarly, a dense continuous topological map is proposed by Johns and Yang (JY11). By tracking persistent features, they construct a set whose members are named as landmarks and learn discrimination properties of each landmark in order to build their probabilistic mapping and localization approach.

Position Invariant Robust Features PIRF which are derived by taking the average of tracked SIFT features across multiple images are proposed to be used in place recognition (KTH10) and SLAM algorithms (KTTH11). The environment is divided into places in which variation of their Pirfs are negligible and based on majority voting scheme PIRF-Nav is proposed as a SLAM method. Modifying the PIRF dictionary management step, Tongsprasit et al. (TKH11) achieves to accelerate the PIRF−Nav 12 times. 3D−PIRF (MYH11) which is the latest version of PIRFs based SLAM approaches are proposed for navigational purposes in challenging indoor environment. The performance of SIFT for indoor and outdoor environment is investigated (VL10). This work shows that SIFT performs better in indoor environments than in outdoor environment.

A particular version of SIFT algorithm M-SIFT is proposed for being used with omnidirectional images. SIFT is simplified by eliminating unnecessary scale and translation invariance feature. Therefore, it is interesting to use on navigation systems of land robots. For example, Valgren et al. (VLD06) prefers to use M−SIFT descriptors while building image similarity matrix that represent the environment. Then they extend their work by adding an incremental spectral clustering algorithm which accelerates localization process (VDL07). Image similarity matrix is used also by Anati and Daniilidis (AD09). A new image similarity measure is introduced and Markov Random Field MRF model is used to detect

loop closure.

Graph Transformation Matching GTM based topological SLAM is proposed by Romero and Cazorla (RC10, RC12). Similar to the idea of representing an environment with a graph, they represent each image with a graph whose nodes attach to the image segments. In fact, each image is divided into segments and extracted invariant features set constructs the node of the graph.

Air−ground matching localization problem for a Micro Aerial Vehicle MAV is related with topological SLAM (MASS13). Local image feature based histogram voting algorithm is used to match the images from Google Street View and images captured by MAV.

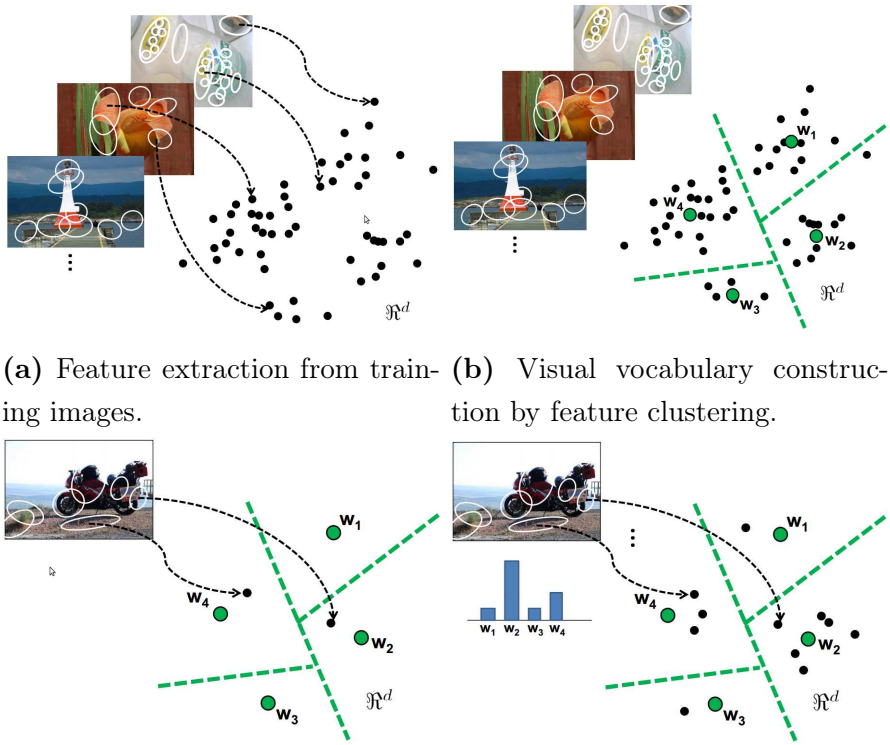
Similar to the metrical SLAM Partical filters (SLA07, KKG09) are also used to model the localization problem probabilistically for topological case. A topological navigation system working indoor is the most recent example which employs a particle filter using local feature matching (LSC13).

Finally, an appearance based method which defines a similarity measure between images and the places in the map (GFO15) maps out the environment. Randomized KD trees (CL68) facilitate the matching process and using history of mapped areas a discrete Bayes filter predicts loop candidates. Therefore, memory and computational costs are optimized.

2.3.3 Bag of Words Based Methods

The Bag of Words BoW method is first proposed as an efficient solution of indexing the documents that is finding all pages of the document on which a word occurs. Therefore, this algorithm is employed successfully in content−based image retrieval (MRS08) in the computer vision community. In fact the aim is to find all images in which a visual feature or a set of visual features occur. Recently, the majority of appearance based SLAM approaches exploit BoW (SZ03) or BoW based algorithms (NS06).

Local image features are quantized by using a visual vocabulary which consists of a set of representative features in order to extract visual words in an image. In other words, visual features are mapped to the nearest visual word in the vocabulary. Clustering algorithms such as k−means are the most common way of building a visual vocabulary. Figure 2.4 shows a toy example of a bag of words training and representation. Generating visual vocabulary is done in an offline or in online fashion. Further improvement is done by classifying the words based on how discriminative they are. Term Frequency−Inverse Document Frequency TF−IDF is an algorithm in order to weight visual words for this aim.



(a) Feature extraction from training images.

(b) Visual vocabulary construction by feature clustering.

(c) Feature quantization using visual vocabulary on a query image.

(d) Visual word histogram of the query image.

Figure 2.4: A toy example of bag of words model. How it is constructed and how features are quantized. Figure 2.4a: Two dimensional feature descriptors are extracted on training images on which . Figure 2.4b: K-means clustering are used ($k = 4$). Figure 2.4c: Extracted feature descriptors are quantized to their closest clusters and those cluster ids are the visual words of the corresponding descriptors. Figure 2.4d: A histogram representation of the given image is built based on the extracted visual words and is used for image matching. (Figure courtesy of Kristen Grauman's lecture notes.)

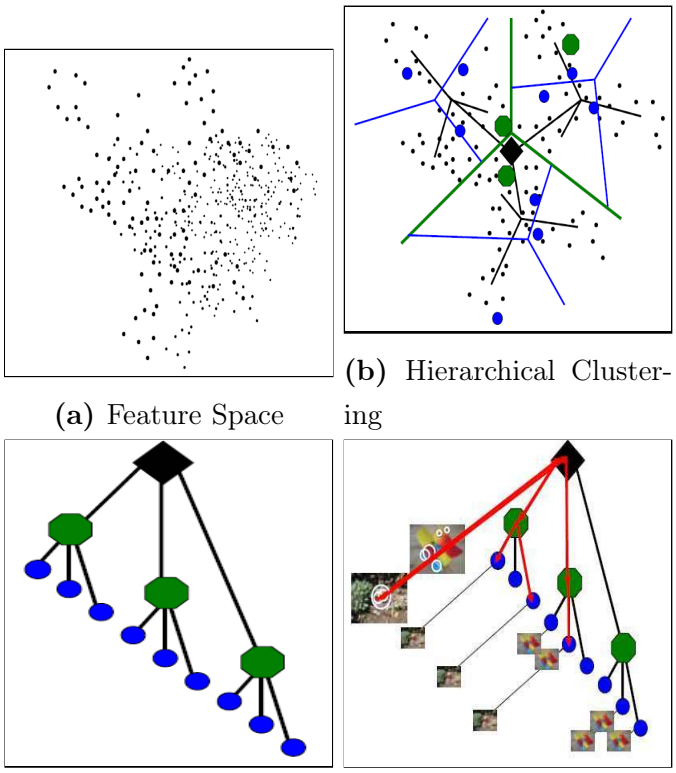
2.3.3.1 Offline Visual Vocabulary Building

The first work which implements this idea in visual search algorithms is published by Sivic and Zisserman (SZ03). SIFT features are extracted and then quantized to the visual words. Inverted file structure which is a look up table and shows the map between each word and their source images is another contribution to accelerate the image retrieval process. Wang et al. (WCZ05) utilizes this technique in a global localization problem. The vocabulary and inverted index file are built offline for localization step. In order to decrease the false positives, he adds a post verification step which checks the epipolar geometry constraints.

The size of vocabulary has a direct effect on the performance of retrieval process. The accuracy of the algorithm increases with the growing size of the vocabulary however this also causes growing computational cost. Therefore, building vocabulary like a tree structure is proposed (NS06). Each extracted descriptor of a query image is assigned to a visual word by traversing the vocabulary tree from the root to the leaf node. This tree structure in combination with inverted index algorithm makes BoW scalable which can handle millions of images. Figure 2.5 illustrates a toy example of a vocabulary tree. Adding RANSAC procedure for geometry check to this algorithm, it is implemented in vision based SLAM approaches for large environments (FEN07). Konolige et al. (KBC+10) implement this process on a stereo vision based SLAM system and test it in indoor and outdoor environments. The results show that high loop closure detection rates are achieved by using this method with a geometric filter.

Fast Appearance Based Mapping FAB−MAP (CN08b) which models the occurrence of visual words in probabilistic manner is the most prominent work in this category. The maximum co-occurrence probabilities defined over visual words are estimated by a Chow Liu tree constructed by using a set of training data. Observation likelihoods which are combined using a recursive Bayesian filter are calculated based on these probabilities. Posterior probabilities obtained from the Bayesian filter are then passed through geometric verification step and finally they are used to predict loop closure candidates. However, likelihood calculation for each location in the map is a costly process. Therefore, a probabilistic bail−out test is introduced in order to fix this problem. Moreover, they propose a modified version of their probabilistic model which is compatible with inverted index algorithm to accelerate their algorithm further (CN10b).

OpenFABMAP (GMMW10) is developed as an open source implementation of FABMAP algorithm because only the binaries of FABMAP are published initially. It plays a central role in CAT−SLAM algorithm (MMW11) in which an appearance based SLAM approach is supported with an extra odometry sensor data. Adding the ability of handling multiple traverses of the same place to the CAT−SLAM CAT−Graph (MMW12) is developed and tested in indoor environ-



(a) Feature Space

(b) Hierarchical Clustering

(c) Vocabulary Tree

(d) Quantization

Figure 2.5: A toy example of vocabulary tree building. Figure 2.5a: A two dimensional feature descriptor space. Figure 2.5b: Hierarchical clustering of the feature space with two levels($l = 2$) and a branching factor($k = 3$). Top level clusters are represented by green circles and separated by green lines and similarly, the second level by blue. Figure 2.5c: A vocabulary tree representation of the hierarchical clusters. Figure 2.5d: The quantization of leaf nodes is shown. (Figure courtesy of David Nister (NS06).)

ments.

The positions of local features in Bag of Words algorithm are lost. The drawback of this comes across at localization step of SLAM algorithms. FAB−MAP 3D (PN10) deals with this problem by taking into consideration the word co-occurrences as well as their pairwise distances. Another interesting work which augments the BoW approach with spatial information is the Feature Co−occurrence Maps (JY13b). The quantization process is done in feature and image space and co-occurrence probabilities are calculated for different times of the day. Their results also demonstrate that accurate long term localization can be achieved by learning the properties of local features captured instead of representing a place with a single image (JY13a).

To accelerate building the vocabulary process, Galvez Lopez and Tardos (GLT11) propose to extract FAST features with binary descriptor BRIEF. They perform loop detection with an average speed of 16 ms per image using big sequences by using their new direct index method which is an efficient way of finding correspondences between images. Another online algorithm which updates the posterior with each new measurement by using a Rao−Blackwellized particle filter (RD11). It is a sensor independent solution therefore it can be used with laser range finder or a vision sensor. In fact their algorithm is the online version of Probability Topological Maps (RD06) with an upgraded inference step.

Instead of using epipolar geometry based verification step Conditional Random Fields are used for a stereo vision based place recognition system (CGLR+10). This idea is also employed in a system which can rectify the map by eliminating past false positive loop closures (LCN12).

Using a discriminative criterion is introduced by Ciarfuglia et al. (CCVR12). As a part of training, weights assigned to the visual words are learned. These learned weights facilitate and increase the precision of a loop closure detection step. Majdik et al. (MGLLC11) uses this idea by introducing updatable weights. In other words, weights of visual words change according to their importance while detecting the loop closures.

There are BoW based SLAM methods which aim to work in more challenging environment such as urban places. To be able to index images captured in cities the training data set is used to select the most discriminative features such as the ones that can be assigned to some specific places all the time (SBS07). A more recent approach proposes to identify features coming from moving object in the field of the vision sensor (AJK11). Therefore, it can handle the situation arisen due to dense traffic or pedestrians.

Instead of extracting interest points to obtain visual words, Lee et al. (LZLS13) utilizes lines in his indoor place recognition system. Mean Standard Deviation Line descriptors MSLD are chosen and a hierarchical visual dictionary was constructed.

2.3.3.2 Online Visual Vocabulary Approaches

Building visual vocabulary simultaneously while the robot is exploring the word is called as online approaches. Filliat (Fil07) first proposes to build a visual vocabulary dynamically. Given a local feature a simple linear search algorithm is used to find the closest visual word in the vocabulary. If the distance between the closest word and the given local feature is high, then a new word is added to the vocabulary. He used this algorithm in mapping and localization tasks but for small areas due to the limits of the linear search algorithm. Then Angeli et al. (ADMF08) augments this approach to detect loop closures in real time. Considering the temporal coherency a discrete Bayes filter is employed to estimate the loop closure probabilities. Using two visual vocabularies as input of Bayes filter (AFDM08) is published as an extension of this approach and finally a topological SLAM system which utilizes this idea is completed (ADMF09)

A long term SLAM approach which is inspired by Angeli et al. is proposed for large environments (LM11). They develop a novel memory management system which separates memory as working memory and long term memory. While working memory consists of the most recent and frequently occurred words and it is used for loop closure detection in the first case, the rest is kept in long term memory. Besides working at real time, their results show high recall rates at 100 % of precision.

A modified agglomerative clustering algorithm is utilized to build visual vocabulary simultaneously as exploration of the environment is ongoing (NG09, NG12). Using the tracked features the preliminary clusters are formed. Then considering the global distribution of the given features Fisher's linear discriminant based criterion is used to merge these preliminary clusters in order to obtain more distinctive visual words. They test their system in outdoor environments and also in underwater environments.

2.3.4 Combined Methods

There are several works which utilize different types of image descriptors in order to develop better topological SLAM approach. Using the combination of global and local descriptors is a common example of this approach. The main idea behind this is to exploit fast image matching ability of using global descriptors and robust matching of using local descriptors at the same time.

A localization and mapping system which based on the combination of different descriptors is proposed by Goedeme et al. (GTV+05). That is, extracted vertical column segments which are described with ten different descriptors are clustered and then they are given to a kd−tree structure which is used for localization step. Given a query image, loop closure candidates are first retrieved by

using the vertical structures. Then, a matching distance is calculated among these candidates in order to find the exact loop closure pair. Afterwords, they achieve to develop a complete navigation system by adding SIFT features and applying the Dempster−Shafer probabilistic theory to their approach (GNTVG07).

OACH global descriptor combined with Harris Laplace local features described by the SIFT is proposed to be used in topological localization (WZC06). Two databases such as the first with OACH for initial localization and the second with SIFT for final localization steps are constructed. Finally a geometric verification step such as a RANSAC based fundamental matrix estimation is applied to eliminate false detection.

An outdoor localization system (WMZ07) is implemented by using a particle filter which utilizes two global descriptors such as WGOH and WGII. Given two images, the similarity between them is estimated based on the comparison of each global descriptor separability. This approach obtains loop closure recall rates which is close to the SIFT based methods while it is four times faster than them. They further modify this combination with adding a local SIFT descriptor in order to handle the cases in which global descriptors are insufficient to localize the robot alone (WTMZ07).

Inspired by biological concepts Siagian and Itti (SI09) propose to use Gist as a statistical measure of an image and salient features as measures of interest at each image location. Then they are given to the particle filter based localization method.

A loop closure detection algorithm which utilizes SIFT as local features and histograms of features distribution as global features is introduced by Chapoulie et al. (CRF11). A Bayes filter combining these representations is used to detect loop closures in outdoor environments.

Hull Census Transform HCT based scene change detection and topological map building algorithm for being used with omnidirectional images is proposed by Wang and Lin (WL10). The algorithm starts with SURF extraction and the convex hulls are computed over extracted features. Using a vector magnitude comparison coding statistics for coding are calculated and finally binary codes are formed. The codes are employed for scene changes detection and building topological graph. Further, Extended HCT (LLY13) in which color information and structure information of the convex hulls are injected to the framework is proposed as expanded version of HCT. Another location recognition approach which is a combination of edges, local features and color histogram is developed by Wang and Yagi (WY12).

In our previous work (KUM13), we propose a hierarchical topological map which represents an environment with a graph structure by using a global descriptor and visual words. It utilizes Vector of Locally Aggregated Descriptors (JDSP10b) for node level loop closure detection as a first step and then utilizes

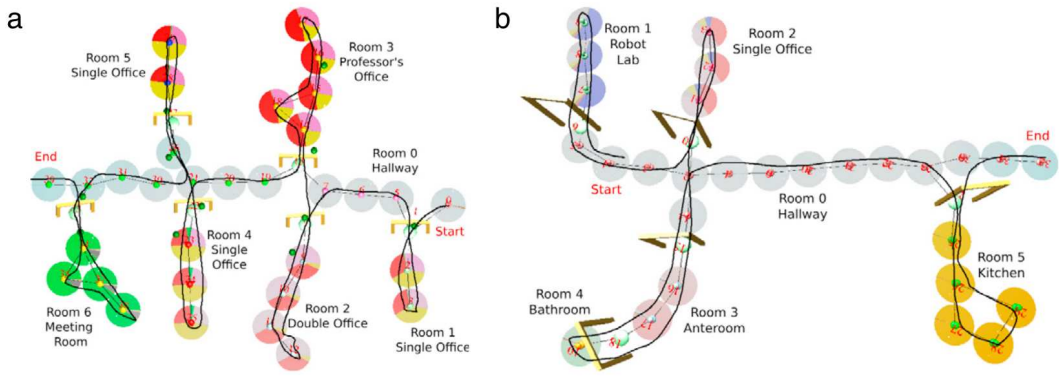


Figure 2.6: Semantic maps built on topological maps. Figure courtesy: A. Pronobis (PJ12)

BoW method for image loop closure detection. It is an efficient appearance based loop closure detection algorithm which can handle over 11000 images by using modified inverted file structure. Another contribution is that a spatial similarity constraint which exploits an advantage of the omnidirectional image is utilized instead of epipolar geometry based verification step.

2.4 Vision based Semantic Mapping

Semantic mapping is a qualitative way of representing the environment which targets to achieve better navigation and task planning as well as intersection between humans and robots capabilities. A qualitative way of representing the environment can be explained as identifying and keeping the track of the signs, the symbols and the objects which involve meaningful concepts for human beings. Therefore, semantic mapping is a hybrid mapping technique which requires combination of different information models. An intuitive way of building semantic maps is shown in figure 2.6.

We classify semantic mapping literature based on their applications in mobile robotics. An illustrative representation of this classification of the most common semantic mapping approaches is illustrated in the figure 2.7. First we divide the literature into two groups. The first category consist of the ones which use 2D or a 3D metric map of the environment. Metric model usually is utilized as a supplementary information which facilitate the process of identification and labeling of the environment. Semantic mapping approaches which are constructed on topological map is the second category. An abstraction of the explored environment in terms of graph whose nodes are organized in a geometrical manner is proposed to maintain conceptual knowledge about the mapped areas. These

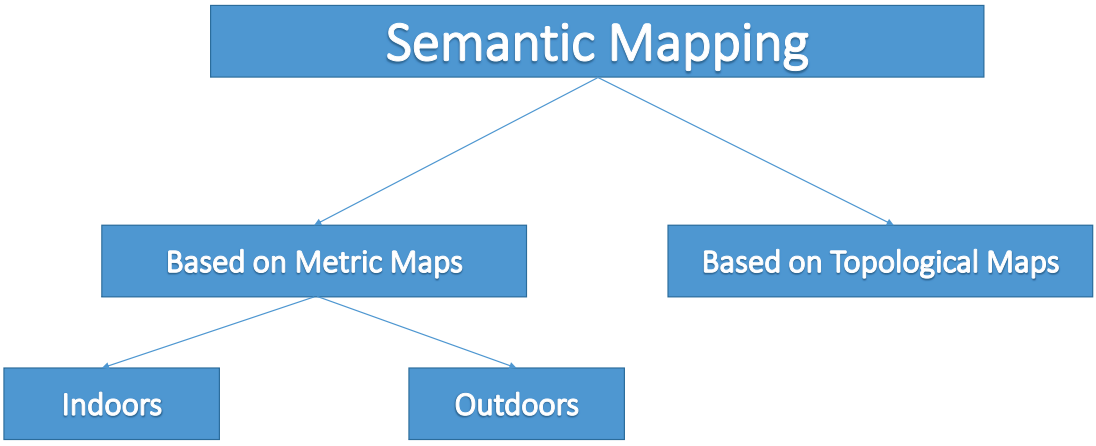


Figure 2.7: An illustrative representation of the classification of semantic mapping approaches is shown.

topological graphs can further be divided into two category. The ones which only keep geometrical features unrestrictedly or the ones which are restrained in agreement with the semantic attributes that they enclose.

2.4.1 Semantic mapping based on Metric Approaches

The majority of the semantic mapping approaches in the literature are constructed on top of metric maps. They are separated into two groups. The paradigms are designed for indoor environment and outdoor environments.

2.4.1.1 Indoor Semantic Mapping

The approaches which are designed for indoor environments are further classified as single scene and large scale scene. The approaches which take an instance frame into account and provides semantic concepts with respect to a local coordinate frame is named as the single scene. On the other hand, the approaches which progressively build a metric map with respect to a global coordinate frame at the same time define high level concepts are named as the large scene.

A single frame snapshot technology (NRG+04) is proposed as an example of semantic map which is considered as an interaction between robots and humans. Real word images are captured and kept to augment a metric map. Icons or symbols which provide importance of places and objects in the field of view are used to augment the metric maps. Kostavelis et al. (KGBN12) is proposed to learn if the mapped places are traversable by using an SVM based memorization algorithm. It is designed to work in damaged indoor environments after being

hit by a natural disaster. Further this method is implemented on a stereo vision system and shows remarkable performance in indoor and also outdoor environments (KNG12). Visual place categorization using a single RGB−D frame can be given as examples of this category (MMKH12, RTMF08).

On the other hand, large scale indoor semantic mapping literature can be classified based on the sensor choice and the methods used for building the metric map. Although there is a strong literature which utilizes laser scanners, we exclude them here because we concentrate on vision based solutions. RGD−B sensors are another popular way to construct 3D structure of the surrounding environment. In the first example of this (KG13), a support vector machine SVM in conjunction with bag of words method is applied to identify dissimilar places after building a 3D map. In the second example SLAM6D toolkit is modified to register subordinate point clouds into a consistent full scene points (GWAH13).

Civeraa et al. (CGLR+11) proposes a semantic mapping approach which employs monocular SLAM and object recognition process in parallel. This can be seen as a feature based map enhanced with different types of furniture identified in the mapped area.

A graphical model is first built to represent semantic information by Pronobis et al. (PJ12) and further it is augmented by introducing an SVM based cue integration to the framework (PMCJ10). Finally, a multi layered semantic mapping approach (PJ11) which combines multiple visual and geometrical information is presented as a final version of this work.

Stereo vision based methods which aim to obtain global and consistent metric map are also good examples of the large scale indoor semantic maps. For example, object labels are identified and they are used to augment the metric map constructed by the SLAM module (VGNS07). The same objective for an office environment is reached by exploiting text detection (CSCN11). In an other work a Gaussian model is used to label the spatial regions of the metric map in order to improve the navigation and mobile manipulation ability (NGRTC10). A framework based on homography and context based image retrieval techniques is another similar work proposed by Feng et al. (FRJ+12) in order to deal with difficulties coming from viewpoint and camera position changes. A long term metric map is used to categorize the places by Ranganathan et al. (RL11).

2.4.1.2 Outdoor Semantic Mapping

Compared to the approaches developed for indoor environments, there are less number of semantic mapping approaches designed for outdoor environments. Most of them deal with street images captured by a camera or multi camera setup. The result of a recently proposed method is shown in figure 2.8 as an example.

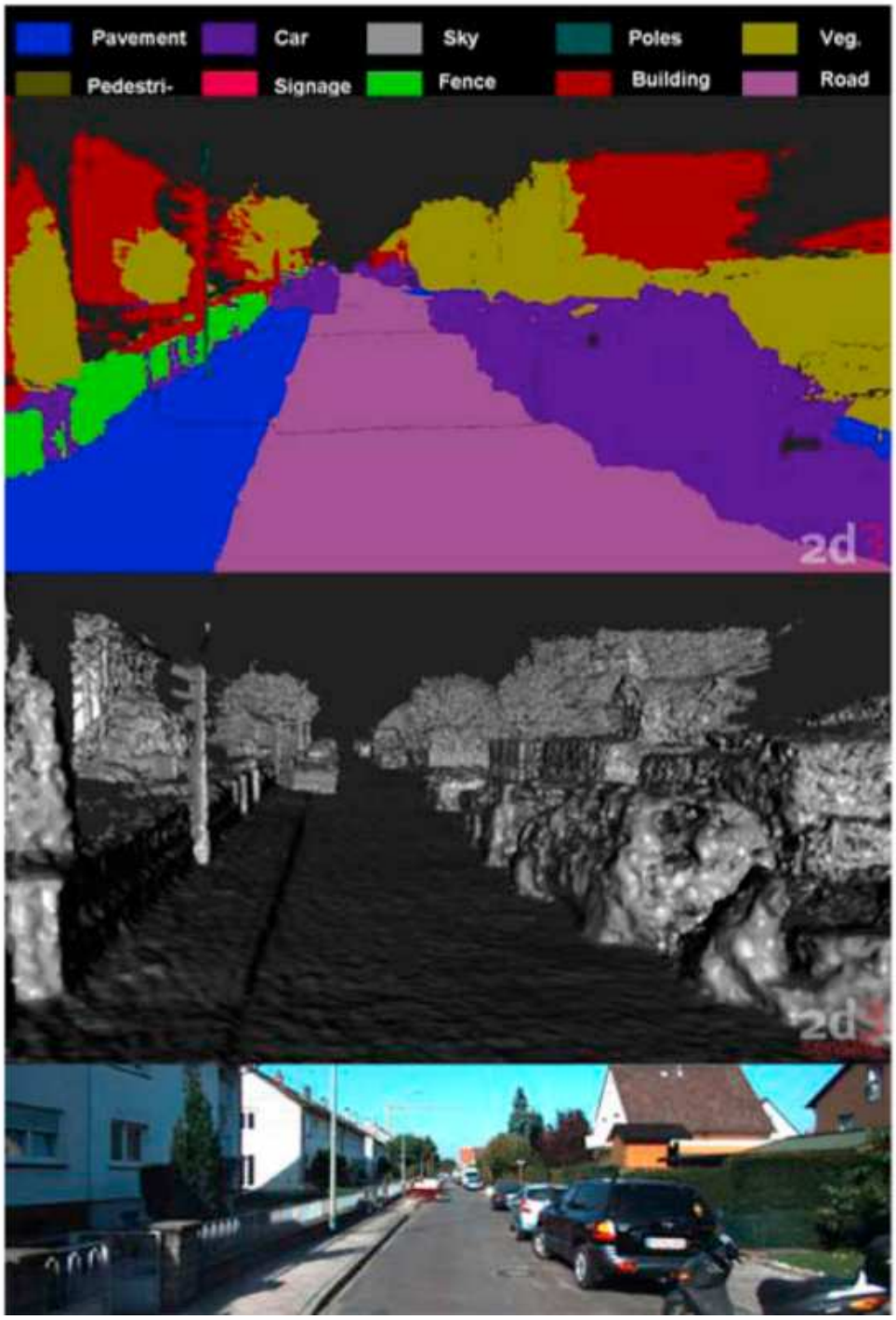


Figure 2.8: As an example of outdoors semantic mapping: pixel wise labeling with the help of dense 3D reconstruction. The top row shows the pixel wise semantic labeling. The middle row shows the dense 3D reconstruction and the bottom one is one of the given image to algorithm. Figure courtesy: S. Sengupta (SGST13)

The approach proposed in (BXD+13) aims to classify scenes by using multiple sensors. Similar to this work, Conditional Random Fields are used on stereo images captured through streets in order to label them (SGST13). Another approach which is also designed for labeling of street images is developed based on segmenting each given image (SSLT12). A large scale semantic map is constructed based on the segmentation of successive images as the final result of the algorithm. As another probabilistic approach supervised multi−class Gaussian process GP classification is applied on the 3D point cloud in order to categorize the objects (PTRN12).

Katsura et al. (KMHS03) introduces a vision based outdoor navigation system enforced with object detection. The main contribution of their work is that the object recognition is robust to the weather and season changes. LadyBug multi camera system is used by Singh and Kosecka (SK12) in order to build large scale semantic map. The street scenes are clustered with predefined labels based on a trained classifier. An online gradient boost algorithm which depicts concept dependent detectors (LSS13) is utilized to build a semantic map based on the images captured by UAV.

2.4.2 Semantic mapping based on Topological Approaches

Due to the fact that metric maps are based on spatial information, the associated high level concepts are kept out of the sight. A topological graph which consists of nodes and edges is a way of revealing this hidden information.

A combination of semantic and topological maps is proposed by Krishnan et al. (KK10). A graph whose nodes assign to a label such as a room and edges assign to transition area label such as corridor is constructed at the top of a topological map. A similar graph is constructed as a semantic map (VMS+09) based on clustering the recognized object with respect to their spatial information. Again based on objects a hierarchical probabilistic representation of the environment which consists of semantic concepts is proposed (VS08). A semantically annotated topological map is another example of augmenting a topological map with semantic information (KG13).

Inspired by the way of human navigation a semantic mapping process which employs global landmarks is presented in (KYS13). Bayesian model of egocentric semantic map which consists of spatial object relationships and spatial node relationships is developed. Considering that, it is also seen as a hybrid map which contains topological, metric and semantic information.

Another type of building semantic maps is to use the constrained topological maps. The constrained topological maps can be defined as navigation graphs which express the connectivity and the transition feasibility among the places in the map. An example of this process is presented by Mozos and Burgard (MB06).

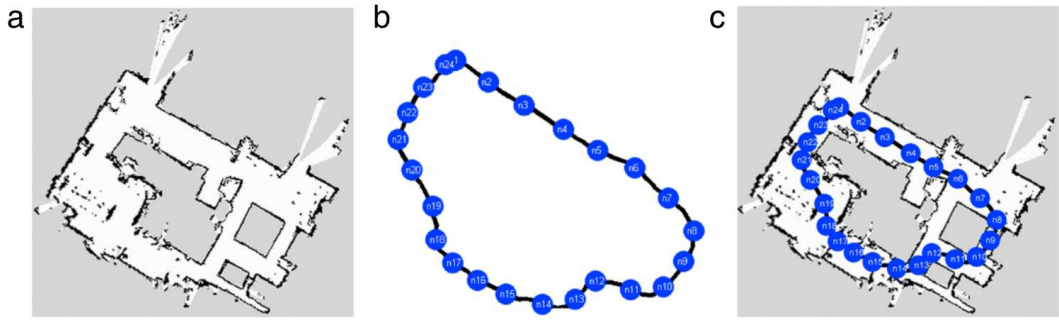


Figure 2.9: Toy example illustrating how the environment can be represented with metric, topological and topo-metric model, respectively. Figure courtesy: Ioannis Kostavelis (KG15)

Adaboost used for classifying the metric map into semantic classes and then the topological map constructed as the combination of the geometric and the semantic knowledge. In other words, a graph whose nodes and edges correspond to the semantically annotated regions and their connections respectively is their final result. In another work hidden Markov models HMM are used to semantically label the nodes in the topological map (MTJ+07). Recently, HMMs are also utilized to build a sparse topological map in which each node is accompanied by a place label (KCG13).

For outdoors environments which is captured by UAV, topological representation of the environment is induced by using different semantic concepts (LSS13). The second example is designed for being used in a wearable catastrophic system for an outdoors scenario (MGGR+12). That is, grouping the Markov models is utilized to semantically label the topological maps.

2.5 Vision Based Hybrid SLAM

Hybrid SLAM combine higher level conceptual maps and localization methods such as topological and semantic with lower level and spatially accurate maps and localization methods in order to maximize the advantages and minimize the weaknesses of each alone. For example, the lack of metric information is one limitation of the pure topological approaches. Meanwhile, the objective of hybridization of metric maps is to extend them for large environment and obtain better global coherency. As a toy example figure 2.9 visualizes the combination of metric and topological maps together.

A vision based hierarchical map is built automatically by (ZBK05). First, using SIFT features and geometrical constraints a low level map which is a graph

is constructed. Then, a high level map is obtained by clustering the nodes. Adding the epipolar geometry check and planar geometry constraint to this framework a navigation system is presented by (BTZK07). Further, it is augmented with an incremental data association algorithm which utilizes Connected Dominating Set concept (BZK09). An efficient loop closure detection is proposed by using this method. The same strategy is implemented on omnidirectional images along with odometry information by (DCD11). First a dense pose graph map of the environment is constructed by using a graph based SLAM approach. Then a two level hybrid map such as local and global is extracted from this pose graph. A global topological map is built by a dual clustering approach while the local level consists of spherical views represented by the extracted features. These spherical views are used for estimating the robot's heading.

An alternative hybrid representation is obtained by using metric sub−mapping approach. In other words, graph based models which divide the large environments into local metric maps are commonly used (BNL+03). Inspired by this work, hierarchical atlas which is a multilevel and multi−resolution representation of an environment is proposed by (LMS+05). There is a topological map at the highest level that organizes the free space into low level sub-maps while a collection of features constructs these submaps. Using the hierarchical atlas maps a hybrid localization approach in two steps is presented by (TMM+07). First, the most probable map is found based on calculation of a discrete probability. After finding the correspondent map, a metric position is estimated by using a Kalman Filter. Later on, the same authors also present a solution for a multi hypothesis topological loop closing problem as a part of SLAM (TKCW09). Finally, they combine all these in their recent SLAM framework (TKC12). In the same line, Hybrid Metric Topological SLAM HTM−SLAM (BFMG08) in which the environment is modeled with a graph. Nodes of the graph are in the form of sub−metric maps and edges of the graph represent the coordinate transformation between these local maps. Moreover, there is also a path estimation step which is based on an unified Bayesian approach. An extended version of this work (BGJFM09) uses spectral techniques in order to partition the map into sub-maps in an efficient way. Also they derive their formulation in order to use stereo vision.

A hybrid visual navigation approach is presented by (SRDC09) in order to map out large environments and to accurately localize using a monocular vision sensor. 2D image features and reconstructed 3D features are used together for performing a navigation task. In our previous work (UKR+14) a sparse topological map which estimates the loop closure likelihood hierarchically by using Bayesian filtering (KUM13) is enhanced with metric sub-maps at the nodes in which the utilized robotic platform is turning or passing through a bent.

Badino et al. (BHK12) proposes to integrate metric data directly into a topo-

logical map using GPS and odometry besides the vision sensors. Each node of the graph consists of a selected image which is also marked with its GPS tag. Selection of the images are done at each predefined Euclidean distance. Moreover, they also build a hybrid feature database which consists of feature's node information and its real position information. Therefore, real metric positions of the features extracted from these captured images are also available with the topological structure of the environment in their map. A Bayes filter is utilized to estimate the position while the vehicle is exploring the environment.

Another objective of hybridization is to detect loop closures which is an important constraint to keep the map globally coherent. Accumulating errors on robot position estimation and landmarks in metric maps makes the loop closure detection a difficult task without utilizing the topological methods. Therefore Lim et al. (LFP12) uses an appearance based loop closure detection module with a sparse bundle adjustment module to obtain a topo-metric map which also tries to perceive globally metric property.

A three step hierarchical localization method approach which is designed for omnidirectional images is proposed by (MSG+07). First, a global color descriptor is used to find loop closure candidates and then line features are matched in order to find the most similar image among the extracted candidates. Finally, metric localization is also implemented by using the 1D radial trifocal tensor. Further they augment this framework by adding SURF local invariant features (MGS07).

2.6 Conclusion

The previous sections surveyed the related literature in visual SLAM. As it is seen, a significant number of SLAM approaches are using vision sensors due to the fact that cameras are low cost, light, passive and low energy sensors and they provide a rich and distinctive data. To be able to exploit the full capacity of vision sensors needs to develop reliable methods which are robust to changes in illumination, in appearance due to people and other moving object in the field of view as well as in seasons. In fact, data association is still an open research area in the field of computer vision. The choice of detector and descriptor has a direct impact on building the consistent representation of the environment and the speed of the system. Therefore, visual SLAM is still a challenging and developing area.

We state that these approaches can be classified based on the model used for representing the environment such as;

- SLAM approaches which builds metric maps. They deal with the camera pose estimation and scene structure from multiple images. They concentrate on accurate localization within a local area.

- SLAM approaches which build topological maps. They deal with the structure of the environment which can be seen as building a graph with nodes and edges. They concentrate on global connectivity information.
- SLAM approaches which build semantic maps. They deal with high level concepts that can provide suitable information for modeling the environment to achieve the ability of human point of view. They focus on assigning a predefined set of labels to semantically describe various parts of the environment.
- SLAM approaches which build hybrid maps. They focus on turning the weak points into a strength by combining specific advantages of the above discussed mapping strategies to solve a problem at hand.

They are all active research area and researchers proposes new solutions for SLAM regularly. We discussed the strong and weak points of each area in detail. However, there is a lack of consensus on evaluation and comparison method in order to show global efficiency and effectiveness of the proposed SLAM approaches. Therefore we consider several parameters while discussing on their performance such as the degree of human intervention, localization accuracy, map consistency, computational time and so on and give importance to their results on well−known publicly available datasets.

3

Hierarchical Loop Closure Detection

The accuracy of a mapping system depends on the ability of knowing if the robot is revisiting a previously mapped area. It is called as loop closure detection and it plays an essential role in all mapping approaches. Therefore, in this chapter we discuss a hierarchical loop closure framework for building hybrid maps as proposed in the following chapter.

A hierarchical loop closure framework is that given a query image acquired at the current location, the algorithm retrieves the most similar node/place(s) and then retrieves the most similar image among the images that belong to the most similar place(s). The process of retrieving the most similar node is called the Node Level Loop Closure and that of retrieving the most similar image is called the Image Level Loop Closure. The aim of this formulation is to achieve efficient and accurate loop detection. For instance, the size of the search space for image level loop closure is the number of nodes and that of image level loop closure is the number of images belonging to the retrieved nodes. Therefore, both these search spaces are far smaller than the total number of images acquired which means faster loop closure detection process.

Considering these advantages, we construct a hierarchical framework which consists of VLAD (Vector of Locally Aggregated Descriptors) (JDSP10a) based node level similarity check and image level loop closure check using visual words. In other words, given a query image, firstly the most similar places/nodes in the map are retrieved. Then, an exhaustive similarity analysis is performed on the member images of the retrieved places. Here nodes/places are the structures which contains images in which visual similarity is almost constant. As the first phase, nodes are constructed by using Image Sequence Partitioning (ISP) which breaks a sequence of images using VLAD descriptor. This process happens very

fast and boils down the whole mapped area to a few important nodes. The second phase of loop closure that aims to find the most similar images is carried out using visual words based histogram scoring.

The main issue with the hierarchical formulation of loop closure problem is that it involves a lot of parameters which need to be tuned with respect to a given environment or camera type. The parameters can be reduced by enabling the algorithms to learn the parameters using a training dataset. Therefore, we take this hierarchical structure and modify the definition of loop closure problem to fit a classification problem. Using this modification, the parameters required for node level loop closure are automatically learned as opposed to empirical evaluation in a standard loop closure detection algorithm. Moreover, an image level loop closure algorithm posed as a Naive Bayes classification problem using four different similarity metrics, instead of using histogram scoring used in the first case. This approach bypasses the need for geometric consistency check popular with many traditional approaches. In other words, parameters for the two levels of loop closure are automatically learned from training data.

Experimental results obtained on various publicly available datasets acquired in challenging environments. In order to effectively capture visual similarity across images in a region by using only appearance based 2D methods, omnidirectional/panoramic cameras with their 360 degree field of view is chosen over the conventional cameras with a limited and unidirectional field of view. Apart from that, omnidirectional cameras are also useful in detecting loop closures when the robot is re-traversing the environment in a reverse direction. This is impossible with traditional cameras. These two reasons justify our choice of omnidirectional cameras for testing the generality of our approach. The computational efficiency and accuracy obtained is evaluated and compared between two frameworks and also two state of the art approaches such as FAB MAP 2.0 (CN09) and the work by D. Galvez-Lopez et al. (GLT11).

The remainder of this chapter is organized as follows. We start with an overview of the proposed framework. Before, we go into the details of the algorithm, we give an introduction of VLAD for the sake of completeness. Then we continue with explaining each step of the proposed framework.

3.1 Framework Overview

This section discusses our loop closure framework which can be seen as pure topological mapping framework. As discussed before, the incoming sequence of images is partitioned and each partition is represented as a node in the topological graph which is used to perform loop closure.

Given a newly acquired image I_t, the following two steps are performed:

1. A node level loop closure (NLLC) is performed with the references nodes $\boldsymbol{N^R} = \{N_1, N_2, ...\} - N_c$ (all nodes of the map except current place node N_c).

2. Up on successful NLLC, an image level loop closure (ILLC) is performed on the reference images $\boldsymbol{I^R} = \{\bigcup_{N_i \in \boldsymbol{N^*}} \boldsymbol{I^{N_i}}\}$, where $\boldsymbol{I^{N_i}}$ is the set of images belonging to the node N_i and $\boldsymbol{N^*}$ is the set of similar nodes from node level loop closure. If image level loop closure is successful, I_t is added to the corresponding node and the topological graph $\boldsymbol{T}$ is appropriately updated.

3. If either or both of the node and image level loop closures fail, the image is compared to the current place N_c and is added to it if similar. If dissimilar, a new place node is created to which the image is added. Since this process decides whether to create a new node or expand an existing node, we call it Image Sequence Partitioning (ISP).

The framework of our loop closure algorithm is given in figure 3.1.

One can note that the current place node is not included in the reference node set $\boldsymbol{N^R}$. The reason is that, due to temporal and spatial proximity, often newly acquired images are highly similar to the current place node. Therefore, a tighter similarity measure than the one used for node level loop closure has to be employed to decide whether a new image belongs to the current node.

We make use of Vector of Locally Aggregated Descriptors (VLAD) for node level loop closure (NLLC) and image sequence partitioning (ISP), and local image descriptors (like SIFT, SURF, etc) for image level loop closure (ILLC). The following sections discuss VLAD, NLLC, ISP and ILLC.

3.2 Vector of Locally Aggregated Descriptors

VLAD (Vector of Locally Aggregated Descriptors) (JDSP10a) is a global image descriptor constructed from local image descriptors like SIFT (Low04b) or SURF (BTG08). It has been successfully used for compact representation and search in web-scale databases. The basic intuition behind VLAD descriptors is to combine the quantization residues of the local feature descriptors into a single descriptor to use it as a global image descriptor. To the best of our knowledge, this is the first usage of VLAD in robotics and hence we describe it briefly here.

Algorithm 1 describes VLAD computation using local image feature descriptors. The inputs for VLAD computation are an image I, a bag of words quantizer Q (SZ03), feature descriptor length l and a PCA (Principal Component Analysis) matrix P. The quantizer $Q = \{c_1, c_2, \ldots, c_k\}$ is learned on training data, where each c_i is a cluster centroid and k is the vocabulary size. The PCA matrix P is

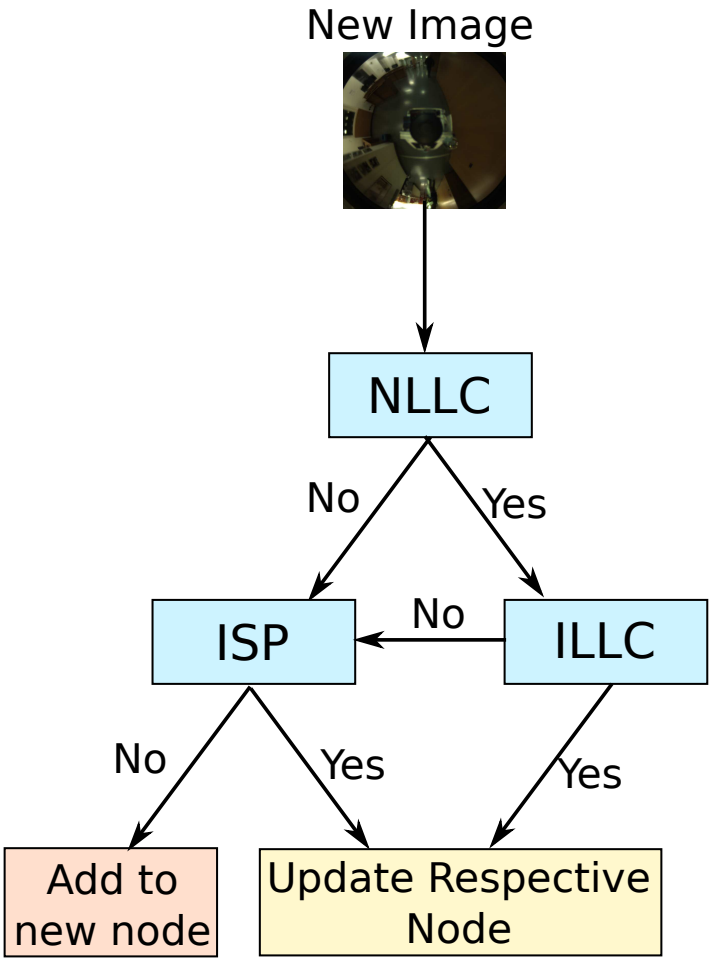


Figure 3.1: Loop Closure Detection Framework

also learned on the training data. In other words, Q and P are learned by using big datasets and more details are given in experiments section.

Firstly, feature descriptors are extracted on image I which are then quantized (lines 3-6) using the quantizer Q. Subsequently, quantization residues are computed and cumulated for each quantized descriptor (lines 7-12). Quantization residue is the vector difference between the feature descriptor and the centroid of the cluster in the vocabulary Q to which it is quantized, and hence has the same dimensionality l as the feature descriptors. Then the quantization residues of all the descriptors assigned to each cluster are summed up and stored as column vectors of a matrix d. Therefore, the matrix d will have k columns each corresponding to a cluster in the vocabulary and l rows indicating the feature dimensionality. In case no descriptors are quantized to a particular cluster, it simply contributes a column vector of zeros to the matrix d. Finally, the k column vectors of d (sum of quantization residues) are augmented to a single vector $\mathbf{V}'$ (line 13) resulting in what we call a full VLAD descriptor of dimensionality $k * l$ which can be a huge number. For example, in our implementation, a $128-$word vocabulary (k) and $64-$dimensional (l) SURF descriptors are used and the resulting full VLAD descriptor is $8192-$dimensional. Therefore, we conserve only a few informative dimensions by projecting to a lower dimensional space using a PCA-projection matrix P (line 15), which is learned offline along with the bag of words vocabulary Q. In the present application, PCA-projection has been used to compress the full VLAD descriptor to a $128-$ dimensional VLAD descriptor $\mathbf{V}$. Hereafter in this paper, whenever a reference is made to VLAD descriptor it implies PCA compressed VLAD descriptor.

The quantizer Q and the PCA matrix P are the input parameters which are learned offline on the training data. It has been suggested in (JDSP10a) that very small vocabulary sizes like $k = 64$ to $k = 256$ are sufficient for constructing meaningful descriptors that facilitate accurate matching. A detailed description of the quantizer and PCA matrix learning is given in section 3.5.

Since VLAD only depends on the continuous quantization residues, it has been shown to bypass (JDSP10a) the side effects of hard quantization (PCI+08a) to some extent.

3.3 Node Level Loop Closure

3.3.1 NLLC

Node level loop closure NLLC aims to evaluate the similarities of a given image with each of the reference nodes (node-image similarity). We model any node N_i as an ellipsoid whose axes are defined by a multivariate gaussian distribution

Algorithm 1 VLAD Descriptor Computation

```
procedure GET_VLAD(I, Q, l, P)
    ▷ I - Image, Q - Quantizer, l - SURF descriptor dimension, P - PCA matrix
    F_surf = Extract_SURF(I)                    ▷ Extracts SURF features
    n = Num(F_surf)                             ▷ Number of SURF features extracted.
    k = Vocabulary_Size(Q)
    F_w = Quantize(F_surf, Q)                   ▷ Quantize features into words.
    d = [0]_{k×l}                               ▷ Initialize residue matrix with zeros.
    for i = 1 to n do                           ▷ For each SURF feature
        c^i = Get_Centroid(Q, F_w^i)            ▷ get centroid corresponding to the word.
        d(F_w^i) = d(F_w^i) + F_surf^i − c^i
                                                ▷ Accumulate quantization residue as columns of d.
    end for
    V' = [d(1)^T|d(2)^T|....|d(k)^T]_{1×(k*l)}
                                                ▷ VLAD descriptor computation by augmenting quantization residues.
    V = P × V'^T                                ▷ PCA-projection to compress the descriptor length.
end procedure
```

over VLAD descriptors $\mathbf{V^{N_i}} = \{V_j^{N_i}, V_{j+1}^{N_i}, \dots\}$ of its member images $\mathbf{I^{N_i}}$. Hence a node N_i can be represented by the mean vector μ^{N_i} and the diagonal covariance matrix $\Sigma_d^{N_i}$ computed over the member image VLAD descriptors.

Node-image similarities are evaluated using Mahalanobis distance (Mah36) which measures the distance between a given vector and a distribution. Given the VLAD V_q of a query image I_q, its similarity to a given node N_i is evaluated using Mahalanobis distance $\Delta_{V_q}^{N_i}$.

$$\Delta_{V_q}^{N_i} = \left(V_q - \mu^{N_i}\right)^T \Sigma_d^{N_i\,-1} \left(V_q - \mu^{N_i}\right) \tag{3.1}$$

We consider that image I_q is similar to a node N_i if the Mahalanobis distance $\Delta_{V_q}^{N_i}$ satisfies the similarity condition in equation 3.2.

$$|\Delta_{V_q}^{N_i} - \mu_{ns}^{\Delta}| \leq 3\sigma_{ns}^{\Delta} \tag{3.2}$$

Where μ_{ns} and σ_{ns} are the mean and standard deviation of a gaussian distribution over Mahalanobis distance thresholds that control the node similarity measure. We enforce $\Delta_{V_q}^{N_i}$ to be within three times standard deviation such that 99% of the possibilities governed by the gaussian distribution are covered.

The nodes $\mathbf{N}^*$ that satisfy the similarity condition are the winning nodes of the node level loop closure and represent possible places at which the query image

I_q might have been acquired. The member images of $\mathbf{N}^*$ are considered for image level loop closure.

However, if the similarity condition is not satisfied (no similar nodes were found), image sequence partitioning (ISP) is performed to verify if the image I_q belongs to the current place node. Equations 3.3 and 3.4 show the ISP conditions that determine the membership of I_q in N_c.

$$|\Delta_{V_q}^{N_c} - \mu_{isp}^{\Delta}| \leq 3\sigma_{isp}^{\Delta} \tag{3.3}$$

$$\|\mathcal{C}\left(\mathbf{V}^{\mathbf{N_i}}\right) - \mathcal{C}\left(\mathbf{V}^{\mathbf{N_i}} \cup \{V_q\}\right)\| \leq S_{isp} \tag{3.4}$$

Equation 3.3 employs a condition similar to that of node similarity evaluation using mean and variance of a gaussian distribution learned on training data. In equation 3.4, $\mathcal{C}(.)$ is a function that calculates centroid of a set of descriptors ; essentially this condition measures the centroid shift induced by the query VLAD descriptor and constrains it to be within a certain threshold S_{isp}.

For the preliminary case the parameters μ_{ns}^{Δ}, σ_{ns}^{Δ}, μ_{isp}^{Δ}, σ_{isp}^{Δ} and S_{isp} are fixed manually based on experimental observations. In the second part, this is changed with more sophisticated learning process which is discussed in detail in section 3.3.2.

3.3.2 Parameter Learning

The simplest way to automatically learn (via supervised learning) parameters that control the determination of node similarity and the sequence partitioning is to have an accurate ground-truth. However, it is nearly impossible to construct ground-truth with an ideal partitioning and similarity measure even by humans. Therefore, we propose a two-step approach to learn the parameters, using image acquisition ground-truth (GPS readings at acquired locations). The first step is an automatic partitioning of a training image sequence to the best possible accuracy. The second step involves estimation of parameters that best fit the training sequence partitioning.

For the partitioning step, training data is carved out from an original image sequence which constitutes a set of VLAD descriptors of the training image subsequence and the associated GPS coordinates(ground-truth). K-means clustering (HW79) is performed by varying the number of clusters k on the VLAD descriptor set and their GPS coordinates separately. The value of k which maximizes correlation between the sets of VLAD and GPS clusters is considered to be the best partitioning. Intuitively, the clustering of images is enforced to satisfy the ground-truth. The correlation between clusterings is measured using conditional

entropy (AR13) as in equation 3.5,

$$H\left(\mathcal{G}_k|\mathcal{V}_k\right) = -\sum_{i=1}^{k}\sum_{j=i}^{k} p_{ij}\log_2\frac{p_{ij}}{pc_i} \tag{3.5}$$

where $\mathcal{G}_k$ and $\mathcal{V}_k$ are the clusterings of ground-truth (GPS readings) and VLAD descriptors with k clusters respectively, p_{ij} is the probability of a data point in cluster i of $\mathcal{V}_k$ belonging to a cluster j of $\mathcal{G}_k$, and pc_i is the probability of cluster i in $\mathcal{V}_k$ which is the fraction of the data points in that cluster to the total number of data points.

Conditional entropy is zero when the two clusterings perfectly agree with each other and $\log_2 k$ in case of worst cluster correlation. Since the maximum conditional entropy depends on the value of k, the entropies should be scaled by a factor of $\log_2 k$ in order to compare different clusterings. The best clustering k^* is the one that minimizes scaled conditional entropy (equation 3.6).

$$k^* = argmin_k \frac{H\left(\mathcal{G}_k|\mathcal{V}_k\right)}{\log_2 k} \tag{3.6}$$

Figure 3.2 shows a plot of scaled conditional entropy plotted against k values. $\mathcal{V}_{k^*}$ may not represent the perfect clustering of images but automatically provides sufficient examples for the parameter estimation step.

The parameter estimation step first estimates the node similarity parameters μ_{ns}, σ_{ns} from the clustering $\mathcal{V}_{k^*}$. The idea is to fit a gaussian distribution represented by these parameters over observing Mahalanobis distances $\mathbf{\Delta_T} = \{\Delta_1, \Delta_2, \dots\}$ computed on various examples of node level loop closures from the clustering $\mathcal{V}_{k^*}$. Assuming that the distances are independent and identically distributed, the probability of all the distances are given by equation 3.7.

$$p\left(\mathbf{\Delta_T}|\mu,\sigma^2\right) = \prod_{i=1}^{n}\mathcal{N}\left(\Delta_i|\mu,\sigma^2\right) \tag{3.7}$$

$$where \qquad \mathcal{N}\left(\Delta_i|\mu,\sigma^2\right) = \frac{1}{\sqrt{2\pi\sigma}}\exp\left(-\frac{\left(\Delta_i-\mu\right)^2}{2\sigma^2}\right) \tag{3.8}$$

$\mathcal{N}\left(\Delta_i|\mu,\sigma^2\right)$ is a gaussian distribution defined by mean μ and variance σ^2. The parameters μ_{ns}^{Δ} and σ_{ns}^{Δ} are those that maximize the probability in equation 3.7. From the maximum likelihood estimates, they can be calculated in closed form as,

$$\mu_{ns}^{\Delta} = \frac{1}{n}\sum_{i=1}^{n}\Delta_i \qquad \sigma_{ns}^{\Delta} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\Delta_i-\mu_{ns}^{\Delta}\right)^2} \tag{3.9}$$

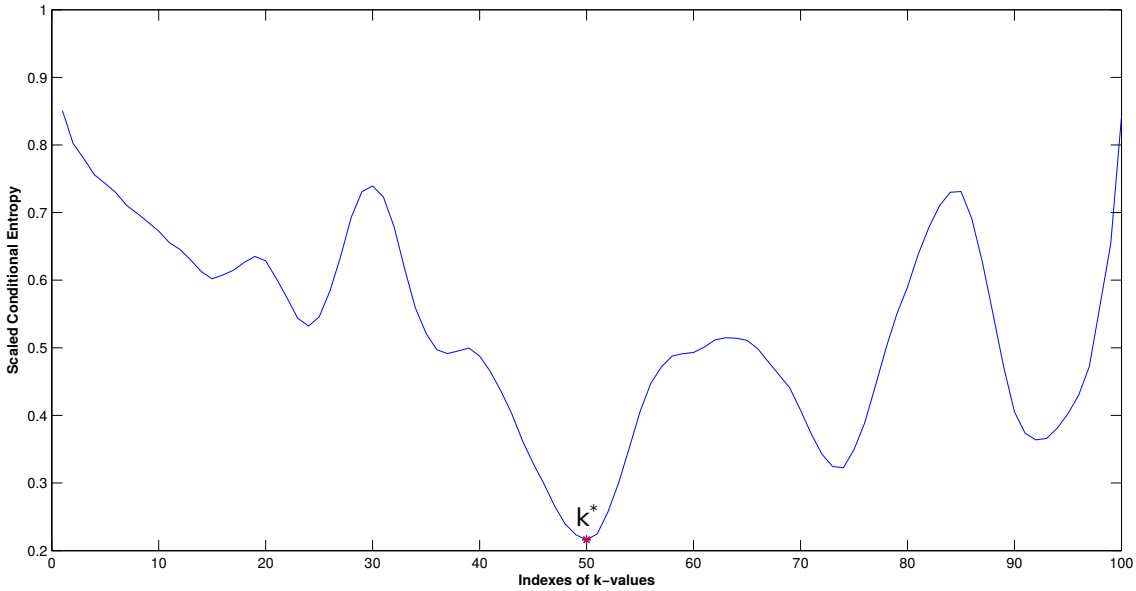


Figure 3.2: Plot of scaled conditional entropies against 100 different choices of k with the minimum value highlighted as k^*.

The parameters μ_{isp}^{Δ} and σ_{isp}^{Δ} are estimated in a similar fashion by maximizing the likelihood of Mahalanobis distances of descriptors farthest to the centroid of their respective clusters. However, Mahalanobis distance just measures the distance in standard deviations. For nodes with few member images/descriptors mahalanobis distance could result in over-confident results.

Therefore we use the centroid shift parameter S_{isp}. It provides an absolute distance between descriptors and is measured using centroid shifts. A centroid shift is the distance by which a centroid is shifted on adding a new descriptor. We consider centroid shifts computed between cluster centroids and their closest descriptors in neighboring clusters from the clustering $\mathcal{V}_{k^*}$; the shift parameter for ISP, S_{isp} is simply the median of these centroid shifts. To elucidate, we consider the cases where a cluster is broken and a new cluster is formed, identify the data points that lead to segmentation of the clusters, measure the centroid shift demanded by these data points, and finally recover the median centroid shift as S_{isp}.

3.4 Image Level Loop Closure

Image level loop closure aims to find the images most similar to the query image I_q and determine if they actually cause loop closure. Given the set of winning nodes $\mathbf{N}^*$ from node level loop closure, a reference image set $\mathbf{I^R}$ consisting of all the member images of the winning nodes is constructed. Each reference image in $\mathbf{I^R}$ is matched with the query image I_q.

For the preliminary case, image similarities are computed by combining two similarity measures namely the VLAD descriptor similarity and the spatial similarity. VLAD descriptor similarity is computed as the euclidean distance between the query image VLAD descriptor V_q and the reference image descriptor V_r. While algorithm 2 gives the brief structure of image similarity computation, VLAD descriptor similarity and the spatial similarity which evaluates spatial similarity between two omnidirectional images will be explained in detail as the random variables of classification algorithm.

Algorithm 2 Image similarity computation.

```
procedure LIKELIHOOD_EVALUATION(I_r, I_q)
                                        ▷ I_r - Reference Image, l_q - Query Image
    likelihoods = [      ]
    for i = 1 to n do                   ▷ For each image i in I_r
        v_i =Euclidean_Distance(V_i, V_q)   ▷ V_i and V_q are VLAD descriptors.
        v_i =Spatial_Similarity(Z_i, Z_q, W, H, b) ▷ Z_i and Z_q are lists of quantized words. ▷
    W and H are width and height of images, respectively.   ▷ b is bin width for shift histograms.
        likelihoods = [likelihoods, v_i * s_i]
    end for
end procedure
```

In the second part, we pose the image level loop closure as a classification problem that uses four different similarity metrics. These similarity metrics are computed between the query image and each of the reference images and used as features for a Naives Bayes (KF09) classifier.

The structure of our Naive bayes classifier is illustrated in Figure 3.3. The variable L is the target random variable that takes on binary values 1 and 0 indicating the presence or absence of a loop closure respectively. The remaining random variables are called the observed random variables and are used as features for the classifier. The random variables associated with the four similarity metrics are:

- V is continuous and represents the similarity of an image pair using VLAD descriptors.

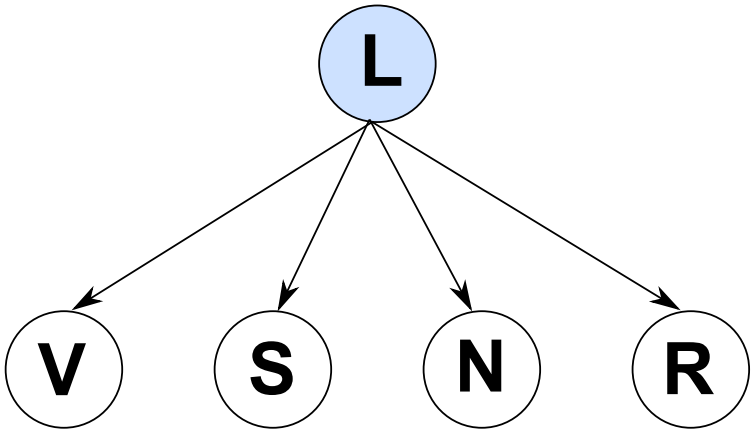


Figure 3.3: Graphical model of our Naive Bayes framework.

- S is continuous and represents the spatial similarity computed using local image descriptors on an image pair.
- N is discrete and imposes a constraint on the minimum number of matches required for a valid loop closure.
- R is continuous and represents the measure of similarities in local odometry.

The probability of a loop closure given the observed data is given by the conditional probability,

$$p\left(L_i|\mathbf{X}\right), \qquad L_i \in \{1, 0\} \tag{3.10}$$

Where, L_i indicates the binary state of loop closure presence or absence, and the observed random variables (similarity metrics) are represented by $\mathbf{X} = \{V, S, N, R\}$ for the simplicity of notation. Using bayes rule, we can rewrite equation 3.10 as

$$p\left(L_i|\mathbf{X}\right) = \frac{p\left(L_i\right) p\left(\mathbf{X}|L_i\right)}{p\left(\mathbf{X}\right)} \tag{3.11}$$

The denominator term is just used for normalization and hence we rewrite equation 3.11 as

$$p\left(L_i|\mathbf{X}\right) \propto p\left(L_i\right) p\left(\mathbf{X}|L_i\right) \tag{3.12}$$

The term $p\left(L_i\right)$ is called the prior probability and $p\left(\mathbf{X}|L_i\right)$ is called the likelihood of the observation variables. As can be seen from the graphical model in Figure 3.3 (and from naives assumption), the similarity metrics are conditionally independent of each other given the loop closure state variable L. As a result, we can write the likelihood term as the product of individual likelihoods as in equation 3.13.

$$p\left(\mathbf{X}|L_i\right) = p\left(V|L_i\right) p\left(S|L_i\right) p\left(N|L_i\right) p\left(R|L_i\right) \tag{3.13}$$

Finally, we say that a loop closure has occurred, if the conditional probability of loop closure exceeds the conditional probability of no loop closure (equation 3.14), and no loop closure otherwise.

$$p\left(L_1|\mathbf{X}\right) > p\left(L_0|\mathbf{X}\right) \tag{3.14}$$

To compute these conditional probabilities, we need to evaluate the prior probability of loop closure which we assume to be uniformly distributed ($p(L_0) = p(L_1) = 0.5$) and the four likelihood terms (equation 3.13). The likelihoods determine the likelihood of a particular similarity measure given the loop closure state and can be evaluated using Conditional Probability Distribution (CPD) functions. CPDs themselves are probability distributions that satisfy the axioms of probability and are governed by parameters learned from training dataset. We use gaussian CPDs to model all of our conditional distributions except $p(N|L_i)$.

Ground-truth $\mathcal{G}$ in the form of GPS readings at the time of image acquisition is used to classify image pairs as loop closure or non loop closure examples that are used as training data $T = \{T_1, T_0\}$.

$$T_1 = \{(I_j, I_k) \quad | \quad |j-k| > p \text{ and } g(I_j, I_k) \leq \alpha\} \tag{3.15}$$
$$T_0 = \{(I_j, I_k) \quad | \quad |j-k| > p \text{ and } g(I_j, I_k) > \alpha\} \tag{3.16}$$

T_1 and T_0 are the sets of positive and negative examples for loop closures respectively. The first condition $|j-k| > p$ ensures that the image pair should not be adjacent and should be separated by p frames. $g(I_j, I_k)$ is the distance between the locations of acquisition of the images computed from the GPS ground-truth $\mathcal{G}$, which should be less than or equal to α to be a loop closure.

The following subsections discuss each of the similarity measures and their CPDs.

3.4.1 VLAD similarity

This similarity measure $v = \|V_j - V_q\|$ is the euclidean distance between VLAD descriptors of a reference image I_j and a query image I_q. We use a Gaussian CPD to model the likelihood of VLAD similarity as,

$$p(V = v|L_1) = \mathcal{N}(v|\mu_{v1,T1}, \sigma^2_{v1,T1}) \tag{3.17}$$
$$p(V = v|L_0) = \mathcal{N}(v|\mu_{v0,T0}, \sigma^2_{v0,T0}) \tag{3.18}$$

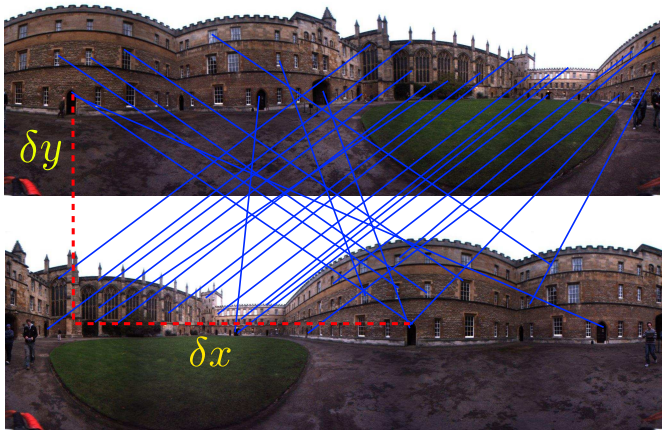
The parameter sets $\{\mu_{v1,T1}, \sigma^2_{v1,T1}\}$ and $\{\mu_{v0,T0}, \sigma^2_{v0,T0}\}$ are parameters of the Gaussian distributions that maximize the likelihood of loop closure and no loop closure events on the training sets T_1 and T_0 respectively. The maximum likelihood estimates for parameters of a Gaussian distribution have already been presented in equation 3.9.

3.4.2 Spatial Similarity

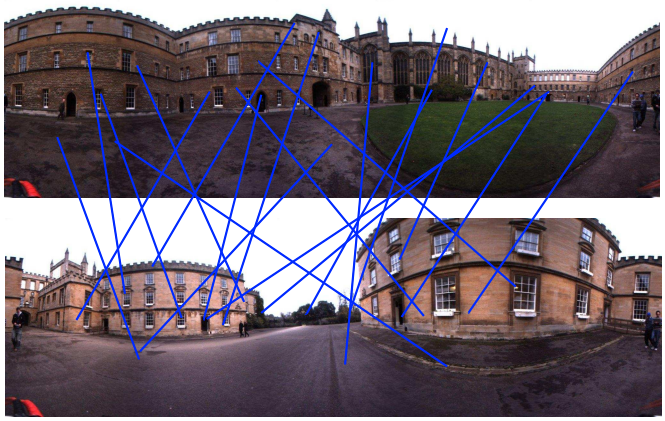
This section discusses the spatial similarity measure represented by the random variable S. We use spatial constraints to exploit the cyclic image structure of panoramic images, to compute similarity between two images. Panoramic images are obtained by unwrapping omnidirectional images or stitching individual images from LadyBug cameras. In this chapter, when we refer to omnidirectional images, we imply panoramic (unwrapped omnidirectional or LadyBug) images.

Due to the 360 degree field of view, omnidirectional image content remains invariant to in-place rotations and minor translations. Let us consider two omnidirectional images acquired at approximately same location but with different heading directions ; and the robot is assumed to move in locally planar environments. Since the omnidirectional images have a circular field of view, distances between different objects in an image are well preserved even under a change in heading direction and a slight translation. In other words, the spatial structure of the objects (also applies to local image features) does not change with an in-place rotation of the camera. Hence if two images are from the same place, all objects in the first image should be shifted by the same amount to take their positions in the second image. A collective zero shift in object/feature coordinates indicates that the images are acquired in the same place and same heading direction, while a collective non-zero shift indicates same place with different heading. In case of a non match different objects will have different shifts and there is no notion of a collective shift. This situation is illustrated in Figure 3.4 from which, one can infer that the feature shifts of the true loop closure follow a converging pattern with few outliers and those of the false loop closure look dispersed. This technique is particularly useful in discriminating true loop closures from false loop closures which arise due to perceptual aliasing.

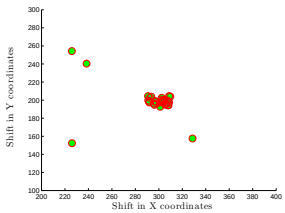
Let the set of features shifts between an image pair be $\mathbf{F} = \{(\delta_1^x, \delta_1^y), (\delta_2^x, \delta_2^y), \dots\}$. We use a bivariate gaussian distribution $\mathcal{N}_{\mathcal{F}}(\mu_{\mathbf{F}}, \mathbf{\Sigma_F})$ to model the distribution of the shifts. For a true loop closure, since the shifts will be concentrated around a small area the gaussian will be narrow and peaked, and a flat gaussian otherwise. For the distribution $\mathcal{N}_{\mathcal{F}}$ to accurately represent the shifts, outliers generated by a few possibly wrong matches have to be eliminated (see Figure 3.4c). Computationally expensive techniques like Robust Regression can be used to get rid of the outliers. However, to be able to quickly repeat this process for several image pairs, a simpler heuristic is employed. The heuristic works on a fair assumption that in case of a loop closure at least 50% of the feature matches are accurate. Therefore, we estimate a temporary mean μ_t on the shift points $\mathbf{F}$ and then consider the top 50% of the nearest shift points to μ_t to re-estimate the mean μ_F which is more accurate. Using this μ_F and the nearest shifts, we also compute the covariance matrix Σ_F.



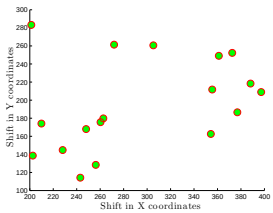
(a) True Match



(b) False Match



(c) True Match Feature Shifts



(d) False Match Feature Shifts

Figure 3.4: Feature Shift analysis of a true match and a false match. 3.4a shows features matched across a pair of images which are acquired in the same place. Matches are shown with blue lines. Shift in X and Y coordinates of a matched feature pair is demonstrated with a red dashed line. 3.4b illustrates a false match of a pair of images acquired at different places. 3.4c and 3.4d show the plots of matched feature shifts ($(\delta x, \delta y)$ in Figure 3.4a) corresponding to the true and false match cases respectively.

Now the task of obtaining a similarity measure by studying the structure of the distribution $\mathcal{N}_{\mathcal{F}}$ remains. To achieve this, we first assume an ideal normal distribution $\mathcal{N}_{\mathcal{I}}$ whose parameters $\{\mu_I, \Sigma_I\}$ represent (in pixels) shifts of an hypothetical image match corresponding to an accurate loop closure. As we shall see, the choice of mean μ_I is not important. However, we assume Σ_I to be a diagonal matrix $diag(d_{I1}, d_{I2})$ which defines the width of the gaussian in pixels. Generally modest widths such as $d_{I1} = d_{I2} = 5$ or $d_{I1} = d_{I2} = 10$ are sufficient and larger widths produce undesirable similarity measures.

To compute a similarity score, we compare the distributions $\mathcal{N}_{\mathcal{F}}$ obtained on an image pair with an ideal distribution $\mathcal{N}_{\mathcal{I}}$, using Jeffries-Matusita (JM) distance (Jef61). JM-distance extends Bhattacharya distance (Mah36) which calculates the distance between two distributions. JM-distance projects the Bhattacharya distance values to a bounded interval. JM-distance between $\mathcal{N}_{\mathcal{F}}$ and $\mathcal{N}_{\mathcal{I}}$ can be written as,

$$JM(\mathcal{N}_{\mathcal{F}}, \mathcal{N}_{\mathcal{I}}) = \sqrt{2\left(1 - e^{-B(\mathcal{N}_{\mathcal{F}}, \mathcal{N}_{\mathcal{I}})}\right)} \tag{3.19}$$

Where $B(\mathcal{N}_{\mathcal{F}}, \mathcal{N}_{\mathcal{I}})$ is the Bhattacharya distance between the distributions which can be given as,

$$B(\mathcal{N}_{\mathcal{F}}, \mathcal{N}_{\mathcal{I}}) = \frac{1}{8} M_{\Sigma}\left(\mu_F, \mu_I\right) + \frac{1}{4}\left(2\log|\Sigma| - \log|\Sigma_F| - \log|\Sigma_I|\right) \tag{3.20}$$

$$M_{\Sigma}\left(\mu_F, \mu_I\right) = (\mu_F - \mu_I)^T \Sigma^{-1} (\mu_F - \mu_I) \tag{3.21}$$

$$\Sigma = \frac{\Sigma_F + \Sigma_I}{2} \tag{3.22}$$

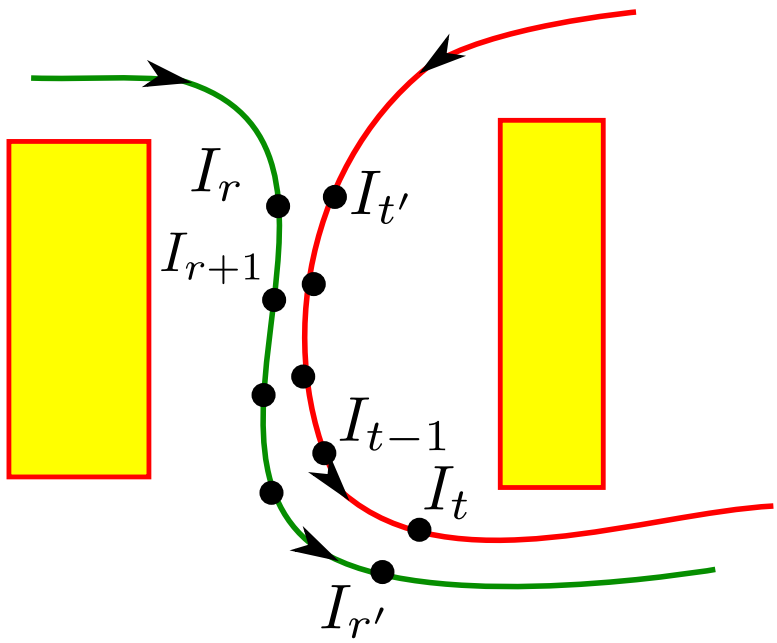
Since we are comparing the distribution of shifts with what would have been an ideal distribution on shifts, it requires us to assume a common mean. Substituting $\mu_F = \mu_I$ in equation 3.20 would leave us with the following expression (equation 3.23) for $B(\mathcal{N}_{\mathcal{F}}, \mathcal{N}_{\mathcal{I}})$ to be used in equation 3.19.

$$B(\mathcal{N}_{\mathcal{F}}, \mathcal{N}_{\mathcal{I}}) = \frac{1}{4}\left(2\log|\Sigma| - \log|\Sigma_F| - \log|\Sigma_I|\right) \tag{3.23}$$

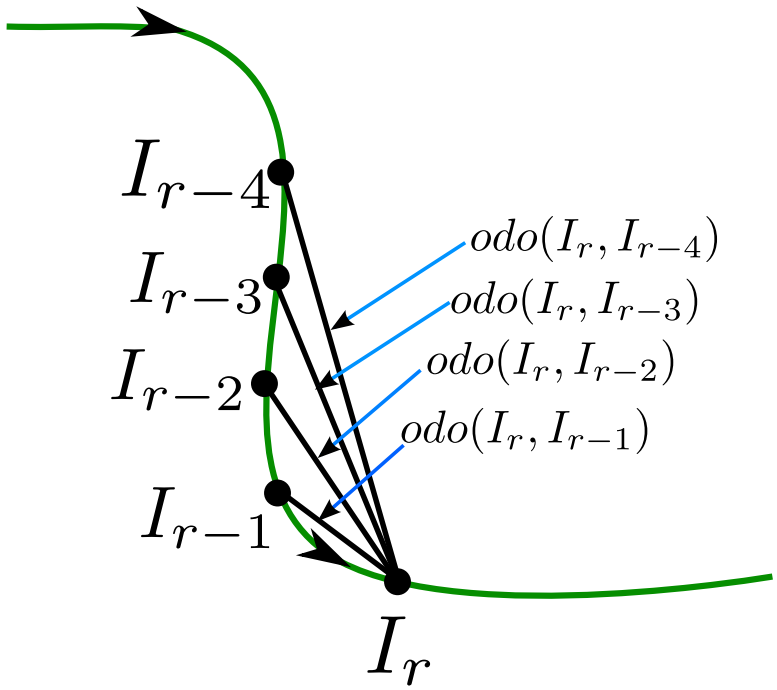
Thus we have our spatial similarity measure $s = JM(\mathcal{N}_{\mathcal{F}}, \mathcal{N}_{\mathcal{I}})$. To learn the conditional distribution $p(S|L_i)$, a gaussian CPD is used which allows to estimate the parameters that maximize the likelihood of the distributions $\mathcal{N}(s|\mu_{s1,T1}, \sigma^2_{s1,T1})$ and $\mathcal{N}(s|\mu_{s0,T0}, \sigma^2_{s0,T0})$ from the training data.

3.4.3 Similarity of Local Odometry

This similarity metric measures the correlation of local odometry information between the current and a previous traversal. While a robot is re-traversing a



(a)



(b)

Figure 3.5: Figure 3.5a shows a re-traversal. Green trajectory is the previous traversal and the red trajectory is the current traversal. Figure 3.5b shows the components of the trajectory structure evaluation function $O(.)$.

previously visited area causing a series of loop closures, the structure of the re-traversal trajectory should be the same as that of the previous visit. Figure 3.5 shows an example of a re-traversal (loop closure) which occurs over a period of time and several image match pairs supporting loop closure. Let I_{p-4} and I_{t-4} be the first reference and query image match pair causing a loop closure, and let I_t and I_p be the current loop closure image pair. The trajectories can be compared using the similarity metric given in equation 3.24.

$$r = min\left(\frac{O(I_p)}{O(I_t)}, \frac{O(I_t)}{O(I_p)}\right), \quad where \quad O(I_t) = \sum_{k=t}^{t-h} odo(I_t, I_k) \tag{3.24}$$

Where $odo(I_t, I_k)$ is the distance between image I_t to I_k, computed using pose and heading information obtained from the robot odometry. The parameter h indicates the length of the past trajectory (history) that is used for the similarity computation. A graphical representation of the function $O(.)$ with $h = 4$ can be seen in Figure 3.5. The min operator is used such that the similarity measures are always less than or equal to 1. This metric is used to measure the consistency of the evolution of a sequence of loop closures. In a way, this metric discourages false loop closures more than encouraging true loop closures. This similarity measure is used to evaluate $p(R|L_i)$ which is modelled with a gaussian CPD learned from training data similar to the previous two similarity measures.

3.4.4 Minimum Matches Constraint

This is a constraint rather than a similarity measure and it is not learned on training data. This constraint imposes that there be a minimum number of features matched between a pair of images to be considered a loop closure. We employ this constraint especially since the other similarity measures are independent of the number of feature matches and as a result produce false loop closures.

We make use of the CPD in equation 3.25 for a true loop closure (L_1) where θ_{min} is the minimum number of feature matches, θ_{suf} is the sufficient number of matches and n is the number of matches for the image pair being considered. This CPD assigns zero probability if the number of matches are less than minimum required and an increasingly linear probability thereafter, and finally a constant probability if the matches are above the sufficient level. This CPD is designed such that it follows both the probability rules $\Sigma_n p(N = n|L_1) = 1$ and $\forall_n \quad p(N =$

$n|L_1) \geq 0$.

$$\begin{aligned} & if \quad n \leq \theta_{min}, \quad p(N = n|L_1) = 0 \\ if \quad n > \theta_{min} \quad and \quad & n < \theta_{suf}, \quad p(N = n|L_1) = \frac{2(n - \theta_{min})}{\theta^2} \\ & if \quad n \geq \theta_{suf}, \quad p(N = n|L_1) = \frac{2}{\theta} \\ & \theta = \theta_{suf} - \theta_{min} \end{aligned} \tag{3.25}$$

Similarly the CPD for no loop closure is given by equation 3.26 which is essentially the inverse of the true loop closure CPD. Figure 3.6 illustrates both the CPDs graphically.

$$\begin{aligned} & if \quad n \leq \theta_{min}, \quad p(N = n|L_0) = \frac{2}{\theta} \\ if \quad n > \theta_{min} \quad and \quad & n < \theta_{suf}, \quad p(N = n|L_0) = \frac{2(\theta_{suf} - n)}{\theta^2} \\ & if \quad n < \theta_{suf}, \quad p(N = n|L_0) = 0 \end{aligned} \tag{3.26}$$

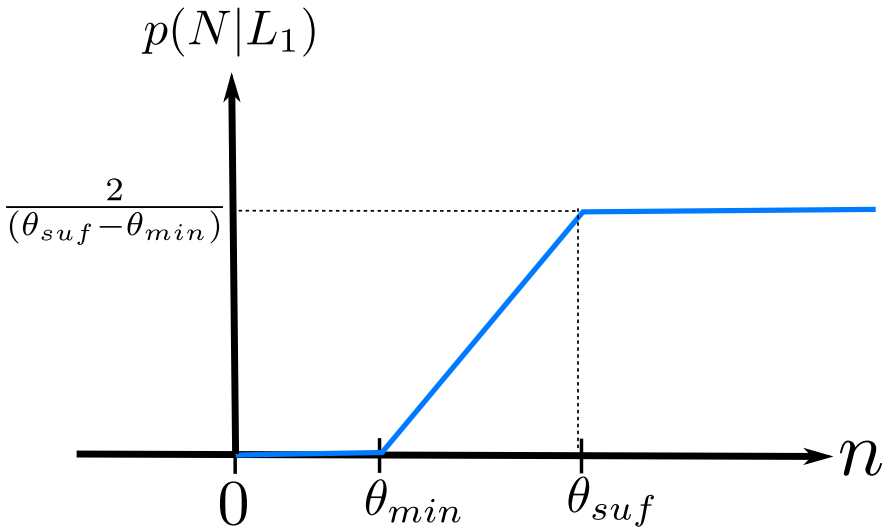
3.4.5 Post-Processing

Given an image pair to match for loop closure detection, we evaluate all the CPDs as discussed in the earlier sections and evaluate the conditional probabilities of loop closure. A loop closure is said to be detected if the conditional probabilities of loop closure exceeds that of no loop closure as in equation 3.14. When a loop closure occurs, the image is added to the corresponding node.

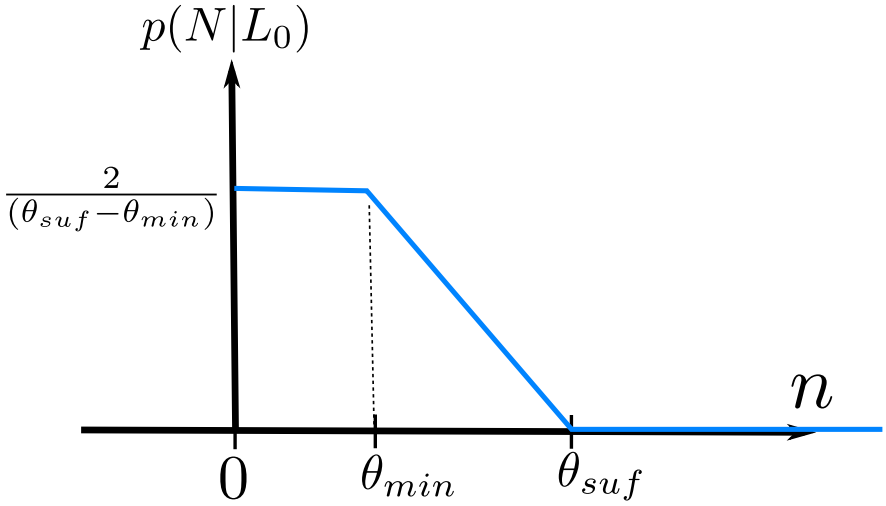
It is possible that multiple images surrounding an actual image might be tagged as positive loop closures for any given query image. In this case, we choose a single reference image whose conditional probability is the highest as the final loop closure result.

3.5 Experiments

Experimental validation of our approach has been performed on four publicly available datasets containing omnidirectional/panoramic images namely IPDS-



(a)



(b)

Figure 3.6: Graphical representation of CPDs for conditional probability of minimum matches constraint.

old[1], IPDS[2], NewCollege[3] and Bicocca(26b)[4]. Various details of these datasets can be found in Table 3.1a. The sequences of IPDS-old particularly consist of gray-scale omnidirectional images acquired in outdoor environments and can be quite challenging to any loop closure algorithm due to the poor image quality, whereas the IPDS and NewCollege sequences contain good quality colored images acquired in outdoot environments. Bicocca sequences are of average quality in comparison and contain indoor images. High quality ground-truth information based on Real-time Kinematic GPS (RTK-GPS) is available for IPDS-old and IPDS datsets, while NewCollege dataset provides poor quality ground-truth from a low cost GPS. As a result, the loop closure accuracy evaluation on NewCollege dataset needs manual inspection. Bicocca provides extended ground-truth readings obtained by laser scan matching that is more accurate and exhaustive than the included GPS ground-truth data.

Keeping the immense amount of the data in view, we uniformly sampled the sequences (as in Table 3.1b) for our use in experiments.

3.5.1 Features and Parameters

This section discusses the vocabulary tree learning and a few user provided parameters for our algorithms.

64−dimensional Upright SURF (USURF-64) are used as local image features. Training data to learn vocabulary trees is formed by randomly selecting 20% of images from each sequence; USURF features extracted on all the training images are used in learning the bag of words vocabularies for VLAD computation and spatial similarity evaluation. These two vocabularies are learned using a single vocabulary tree - the first level of the tree contains 128 nodes and each of these nodes are again split with a branching factor of 6 for 3 levels having a total of $128 * 6^3 = 27648$ leaf nodes. The vocabulary tree structure is depicted in Figure 3.7. Each USURF feature is quantized at two levels - one at the first level of the tree (forms a 128-word vocabulary) which is used for VLAD and the other at the leaf nodes (forms a 27648−word vocabulary) which is used for spatial similarity analysis. Full VLAD descriptors computed on all the training images are used to learn the PCA matrix P.

Most of the parameters used in this algorithm are learned automatically from the ground-truth data. However, a few parameters listed in table 3.2 are set manually. The first five parameters are related to vocabulary tree construction and VLAD, are selected by trial and error.

[1]http://hemanthk.me/wordpress/datasets/
[2]http://ipds.univ-bpclermont.fr/
[3]http://www.robots.ox.ac.uk/NewCollegeData/
[4]http://www.rawseeds.org/rs/datasets/view//6

Sequence	**#(Images)**	**Traj.**	**Vel.**	**FPS**
PAVIN (IPDS-old)	8002	1.3 km	2.3 m/sec	15 hz
Cezeaux (IPDS-old)	80913	15.4 km	2.5 m/sec	15 hz
PAVIN-Jonco (IPDS)	8092	2.3 km	2.12 m/sec	7.5 hz
Cezeaux-Sealiz (IPDS)	22999	7.8 km	2.5 m/sec	7.5 hz
NewCollege	7854	2.2 km	1.0 m/sec	3 hz
Bicocca	26337	1.1 km	0.5 m/sec	15 hz

(a) Datasets Description

Sequence	**FPS**	**#(Images)**
PAVIN (IPDS-old)	2 hz	1144
Cezeaux (IPDS-old)	2 hz	11571
PAVIN-Jonco (IPDS)	3.6 hz	3986
Cezeaux-Sealiz (IPDS)	3.6 hz	11498
NewCollege	1.5 hz	3977
Bicocca	2 hz	3763

(b) Data used for Experimentation.

Table 3.1: Datasets Description. (Traj.=Trajectory Length, Vel.=Average Acquisition Velocity, FPS=Images Frames acquired Per Second)

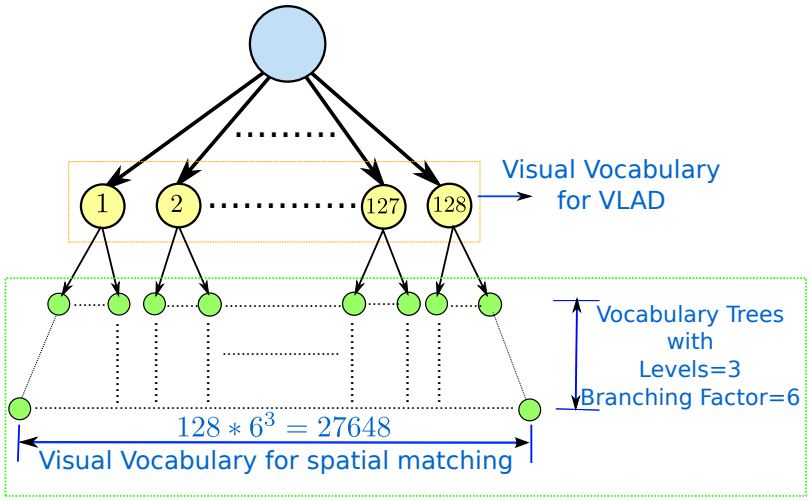


Figure 3.7: Two-level feature descriptor quantization structure. The first level quantizes descriptors using a float 128 word vocabulary while the second level of quantization further quantizes the descriptors using vocabulary tree structure.

The prior probability of a loop closure $p(L_i)$ varies for each frame and is difficult to figure out. For example if the robot just started to build a map, the prior probability of a loop closure is low. However, if the robot has mapped most of the environment but is trying to ensure full coverage, the prior on loop closure should be high. There is no simple way to estimate the prior and hence we choose to nullify its effect on the classification result by choosing a uniform prior.

The ideal covariances d_{I1} and d_{I2} for the covariance matrix Σ_I represent the width of the gaussian distribution $\mathcal{N}_\mathfrak{I}$ in pixels. The idea is that the covariance should make the gaussian narrow and peaked. Unless the width is too large the choice of these parameters does not really affect the classification performance as a CPD is learned on top the similarity score computed using these parameters. As a result the parameters of the CPD will be learned such that they are adjusted to the choice of the covariances d_{I1} and d_{I2}.

θ_{min} and θ_{suf} limit loop closure decisions with insufficient data. These values are chosen manually by observing false positives that are generated due to insufficient data.

Finally, p aims at eliminating positive loop closure examples that are generated by adjacent images acquired during the same traversal. The value is chosen such that at a given image acquisition frame-rate and the velocity of the robot, the separation of the frames is at least four times of the loop closure distance threshold α. In turn, α is chosen such that the accuracy is sufficient to eventually perform tasks such as SLAM. The parameter h that controls the length of historic odometry data to compare local trajectory structure should in general be small,

Parameter Name	Variable	Value
Bag of words vocabulary size for VLAD	k	128
USURF descriptor size	l	64
Full VLAD descriptor size	$k * l$	8192
PCA VLAD descriptor size		128
Vocabulary size for spatial similarity		27648
Prior probability of loop closure	$p(L_1) = p(L_0)$	0.5
Spatial similarity Ideal Covariances	$d_{I1} = d_{I2}$	5
Minimum feature matches	θ_{min}	10
Sufficient feature matches	θ_{suf}	30
Frame separation (Training Data)	p	50
Loop closure distance threshold	α	5 meters
Trejectory length for Local odometry	h	5

Table 3.2: Manually set Parameters

as larger values might induce odometry drift error and can lead to incorrect similarities.

3.5.2 Node Level Loop Closure

Sparsity and accuracy of node level loop closure algorithm are discussed in this section.

Sparsity is the measure of the number of nodes used to represent a topological map. Fewer nodes in a map indicate higher sparsity. However, higher sparsity does not imply map accuracy. In overly sparse maps, representative features of a node represent huge number of images and hence become excessively generalized rather than precisely representing a particular place of the environment. As a result, missing out true loop closure candidates during node level loop closure (NLLC) is likely. Weak NLLC may cause too many reference images to be considered for the image level loop closure (ILLC) drastically increasing the computational cost. This scenario leads to loss in precision of NLLC. Another likely scenario is that the true loop closure candidates may not even be in the reference image set, causing a loss in recall. Hence a map with optimal sparsity is one that ensures maximum possible recall rate in NLLC while maintaining good precision. Recall of NLLC is calculated as the number of correct nodes retrieved out of all correct nodes. The precision of NLLC is calculated as the ratio of the number of

Sequence	#(Nodes)	#(im.)/node	(Traj.)/node	Precision	Recall
PAVIN	64	17.8	20.3 m	29%	89%
Cezeaux	537	21.5	28.6 m	18%	78%
PAVIN-Jonco	105	37.5	21.9 m	41%	93%
Cezeaux-Sealiz	332	34.6	23.4 m	36%	87%
NewCollege	112	35.5	19.6 m	38%	92%
Bicocca	145	25.9	7.5 m	31%	84%

Table 3.3: Node Statistics with the preliminary framework. #(Nodes) - Number of nodes of the map built on the sequence. #(im.)/node - Average number of images represented by each node. (Traj.)/Node - Average trajectory length represented by each node.

Sequence	r_n	#(Nodes)	#(images)/node	(Traj.)/node
NewCollege	0.7	126	31.5	17.5 m
PAVIN	0.9	74	19.06	21.6 m
Cezeaux	0.9	572	20.22	26.2 m

Table 3.4: Node Statistics. #(Nodes) - Number of nodes of the map built on the sequence. #(images)/node - Average number of images represented by each node. (Traj.)/Node - Average trajectory length represented by each node.

reference images from the correct node (contains the loop closure image) to the total number of reference images retrieved through NLLC. As NLLC controls the reference images for ILLC, recall of ILLC is limited by the recall of NLLC.

For the first case in which we fix the parameters manually, node radius for PAVIN and Cezeaux sequences was set as $r_n = 0.9$ and that of Newcollege sequence as $r_n = 0.7$. The kernel width for node similarity computation is chosen to be $\sigma = 1.2 * n_r$, such that it gives a slight cushion to account for noise in node similarity evaluation. Table 3.3 shows various node statistics of the maps built on the three sequences. It can be observed that the average number of images represented per node is between 20 and 30.

The first three columns of Table 3.4 represent various indicators of sparsity obtained on the six sequences by NLLC for the second case. Besides bringing auotonomy and flexibility, learning the parameters instead of experimentally fixing them improved the sparsity of the algorithm around 10%. We can see that for Cezeaux sequence average trajectory length represented by each node stands highest at 28.6 meters. One reason for this is that Cezeaux contains many open

areas devoid of any obstacles. Bicocca on the other hand has the lowest trajectory per node value since it is an indoor environment where obstacles are closer to the camera and hence appearance changes across every few frames. The last two columns show the maximum recall obtained on each sequence and the corresponding precision.

The parameters of NLLC on all sequences had to be learned on parts of individual sequences. Our preliminary attempts to generalize parameter learning for all the sequences over common data lead us to the conclusion that the NLLC parameters strongly depend on the type of the environment and the quality of the images used. This conclusion was based on the poor recall rates for NLLC. All the six sequences have widely different characteristics: PAVIN has low quality images and has been acquired in a simulated city environment where roads and buildings are scaled down compared to a real environment ; Cezeaux contains low quality images but is acquired on a real environment under urban and suburban settings; PAVIN-old and Cezeaux-old resemble PAVIN and Cezeaux except for the high-quality of images; NewCollege data contains mostly suburban like environments and vegetation with high-quality images; Bicocca contains images of moderate quality acquired in indoor settings. Therefore NLLC parameters for each environment and camera have to be learned separately. To learn the NLLC parameters, we select k^* by comparing clusterings for 100 values of k as it is shown in the figure 3.2. These values are selected from the set $\mathbf{k} = \{k_1, k_2, \ldots, k_{100}\}$ where k_i is equal to $\lceil n(\mathbf{I})/i \rceil$ and $n(\mathbf{I})$ is the number of images in the sequence. For example, k_{50} for PAVIN sequence would be $\lceil 1144/50 \rceil = 22$.

3.5.3 Image Level Loop Closure

This section details the accuracies of image level loop closure (ILLC), factors affecting the accuracy and comparison with other approaches.

Accuracy of ILLC is measured by precision/recall. In context of ILLC, precision is the ratio of true loop closures to the total number of loop closure detected and recall is the ratio of number of loop closures detected to the total number of loop closures in the ground truth. Loss in precision may lead to the construction of spurious links between nodes in the topological graph resulting in a faulty map. Therefore we measure accuracy of ILLC as the maximum recall obtained with 100% precision.

For the preliminary case, the results are obtained by varying the similarity threshold T_s which controls the size of reference nodes (and therefore reference images) considered for image similarity analysis. As we can see in the figure 3.8 100% precision is only possible till 35% recall for the Newcollege sequence, 24% for the PAVIN sequence. A 100% precision was never reached on the Cezeaux sequence. The reason is the low resolution images and the significant illumination

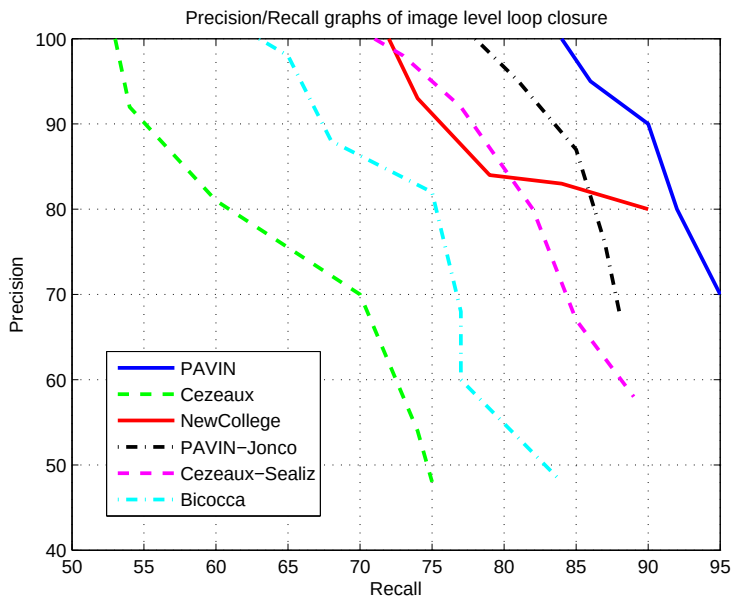
variation. Spatial constraints also helped the toughest Cezeaux sequence on which a full precision has been achieved with a 43% recall. The advantage of spatial constraints in increasing precision is demonstrated. It should be noted that no geometric verification has been applied to obtain the results.

More detailed analyses are done for the second framework. Figure 3.8a shows the precision recall graphs obtained on the six sequences. As we can see, the highest recall rates of 84% and 78% are achieved on PAVIN and PAVIN-Jonco sequences respectively. Cezeaux sequence has the lowest recall at 53%. Bicocca and NewCollege sequences exhibhit an interesting trend; although they have recall rates of 63% and 71%, they are not the best possible recalls rates. Figure 3.8b which shows precision recall graphs obtained using our algorithm without making use of the local odometry check (i.e.$\forall_r \quad p(R = r|L_0) = p(R = r|L_1) = 1$). These graphs show that Bicocca has a recall rate of 78% which is 15% more than the first case, and Newcollege sequence has its recall rate (75%) improved by 4%. For the remaining sequences, the recall rate is reduced: for PAVIN and Cezeaux by 4% each, PAVIN-Jonco by 8% and Cezeaux-Sealiz by 6%. The trajectories and the loop closures detected on each sequence are shown in Figures 3.9, 3.10, 3.11, 3.12, 3.13 and 3.14.

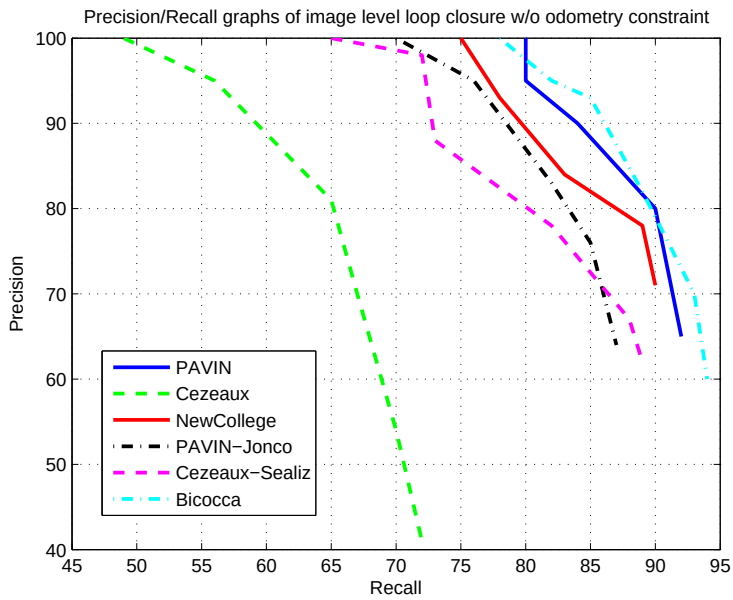
So the question is, what makes the NewCollege and Bicocca sequences different. The answer is that both of them are not acquired on well defined paths like roads. Due to the lack of well defined paths, a loop closure can occur at a location while the robot is navigating over a very different trajectory compared to that of the previous visit. As a result, the odometry check rejects this case as a mismatch leading to a true negative. Newcollege sequence has fewer situations like this whereas Bicoccoa sequence which is acquired indoors has more of them, which is indicated by their change in recall rates when odometry check is excluded. In conclusion, when applying this algorithm to indoor environments or places where there is no restriction on a path to take, it is best to exclude the odometry constraint.

We compared our accuracies to that of two other approaches: FABMAP 2.0[1](CN10a) and DLoopDetector[2](GLT11). For FABMAP 2.0, we have used the software provided by the authors and used the algorithm with default parameters except for those concerned with feature extraction. Precision/recall was obtained by varying the feature extraction threshold (varying the number of features per image) and the loop closure posterior threshold. Similarly for DLoopDetector, we used the author provided software with default settings and without fundamental matrix extraction. Feature extraction threshold, number of consistent matches (k) and similarity threshold (α) were varied to study precision/recall. Visual word vocabularies of the same size as ours constructed using 128 dimensional up-

[1]http://www.robots.ox.ac.uk/ mjc/Software.htm
[2]https://github.com/dorian3d/DLoopDetector



(a) Precision/Recall of ILLC algorithm



(b) Precision/Recall of ILLC algorithm without using odometry constraint

Figure 3.8: Precision Recall graphs of image level loop closure with and without odometry constraint.

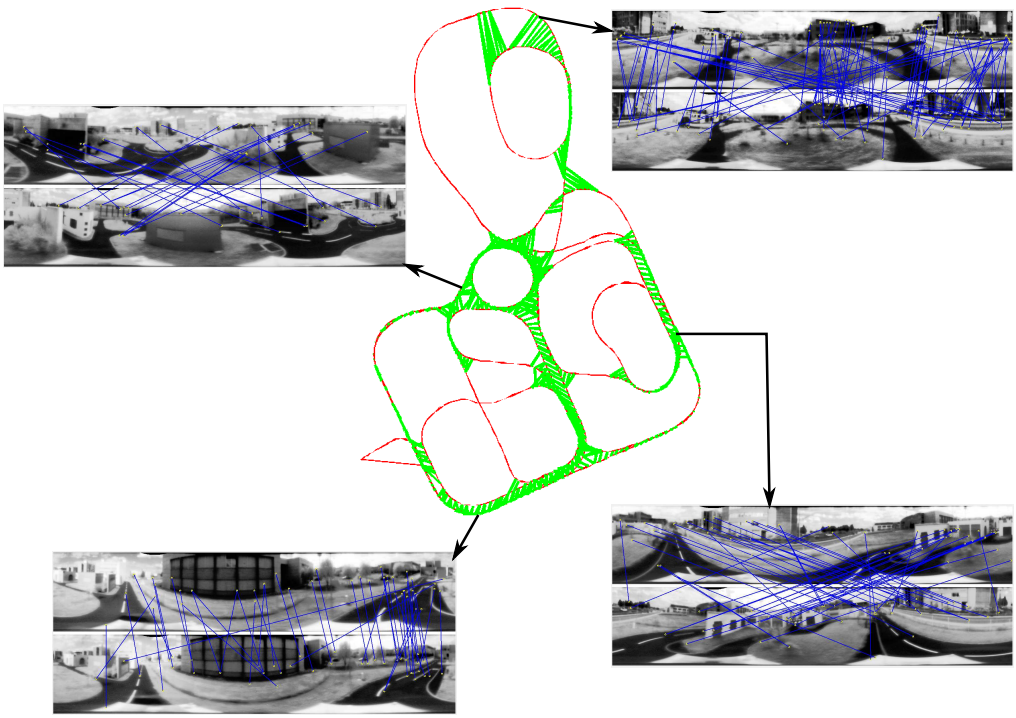


Figure 3.9: PAVIN (Trajectory in red, loop closures in green)

right SURF descriptors and BRIEF descriptors for FABMAP and DLoopDetector respectively. The comparison of accuracy of these approaches with ours can be seen in table 3.5. It can be seen that our approach achieves the best recall on all the sequences followed closely by DLoopDetector and finally by FABMAP. The recall rates of the two other approaches might have been improved if geometric verification of loop closure results was included but as we perform loop closure sans geometry check, this comparison ensures fairness.

3.5.4 Computational Time

Real-time operation is one of the vital elements of loop closure. Although our code is not optimized and we run it on a laptop with Intel Core i7 processor, we reach to process minimum 5 frame per second.

There are five major modules in our algorithm: local feature extraction (SURF), local feature quantization, VLAD Extraction, Node similarity analysis and image similarity analysis. Map built on Cezeaux sequence is used in computational time analysis since it is the longest sequence of the three. The average and the maximum computation time of each module is given in the table 3.6. SURF feature extraction takes 120 milliseconds on average and this is the most time consuming

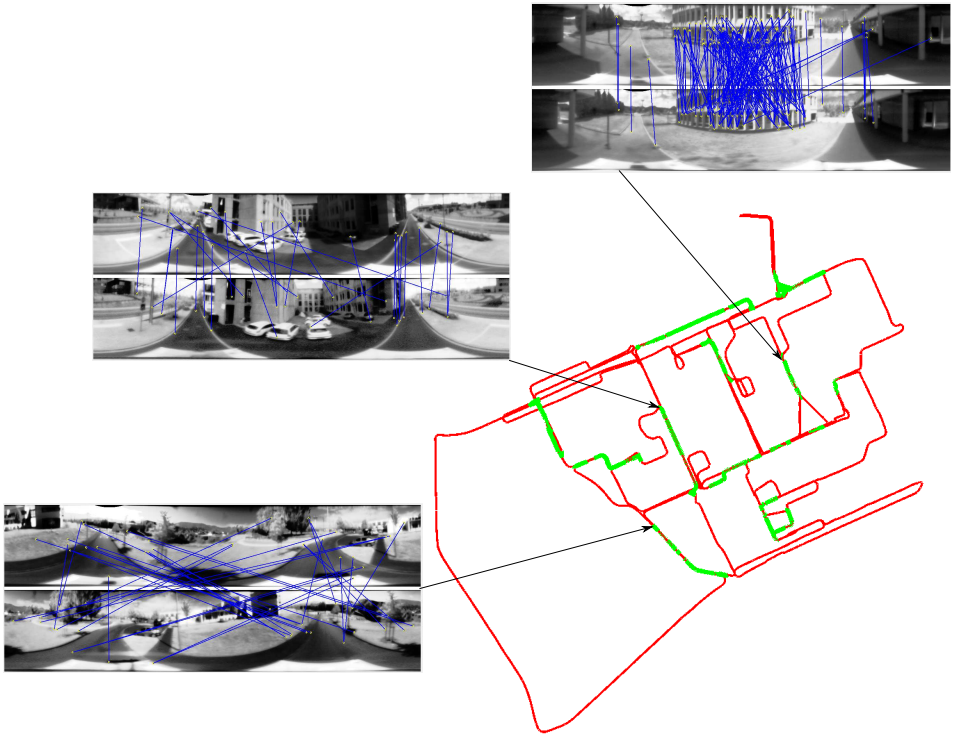


Figure 3.10: Cezeaux (Trajectory in red, loop closures in green)

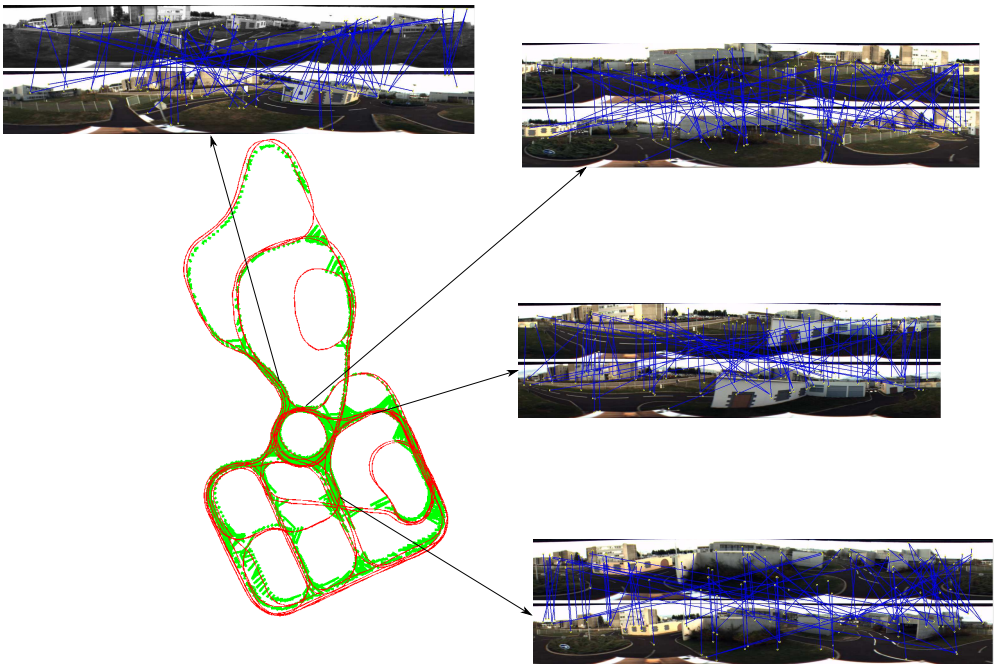


Figure 3.11: PAVIN-Jonco (Trajectory in red, loop closures in green)

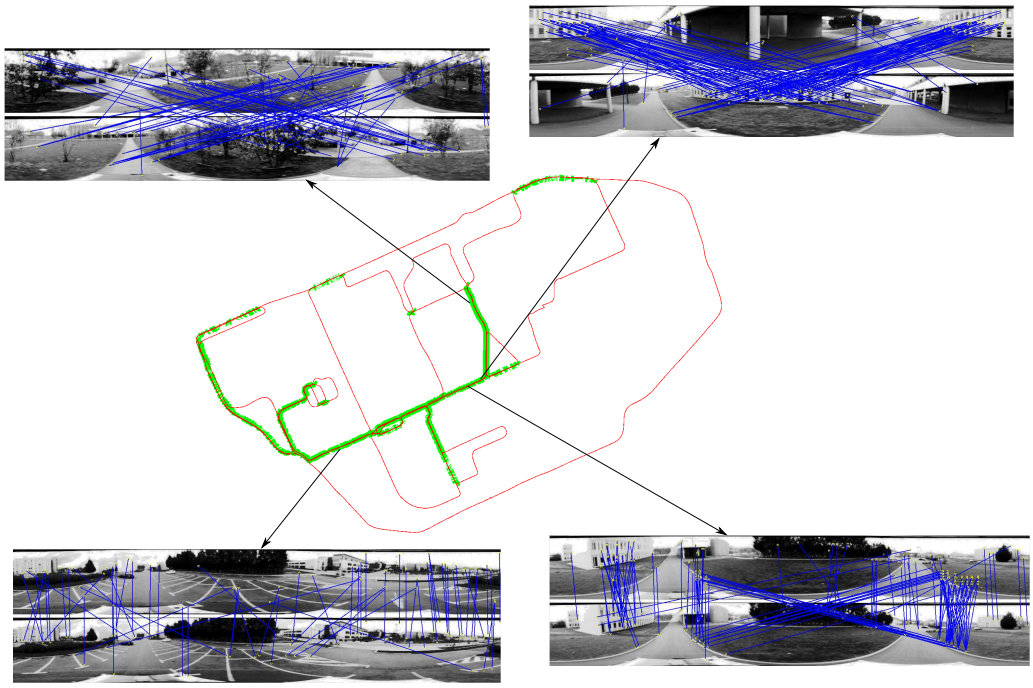


Figure 3.12: Cezeaux-Sealiz (Trajectory in red, loop closures in green)

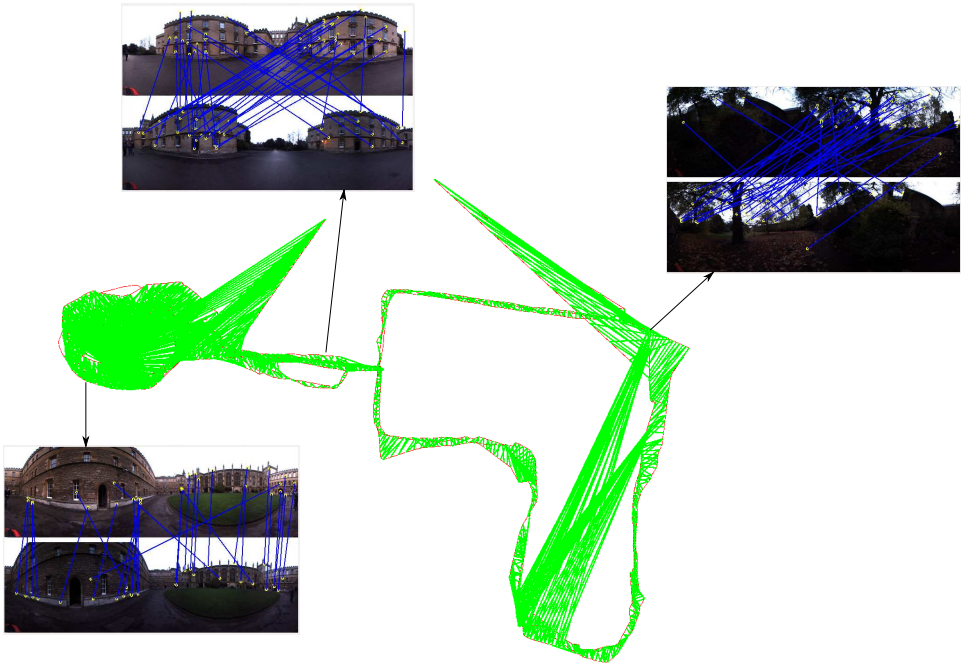


Figure 3.13: NewCollege (Trajectory in red, loop closures in green)

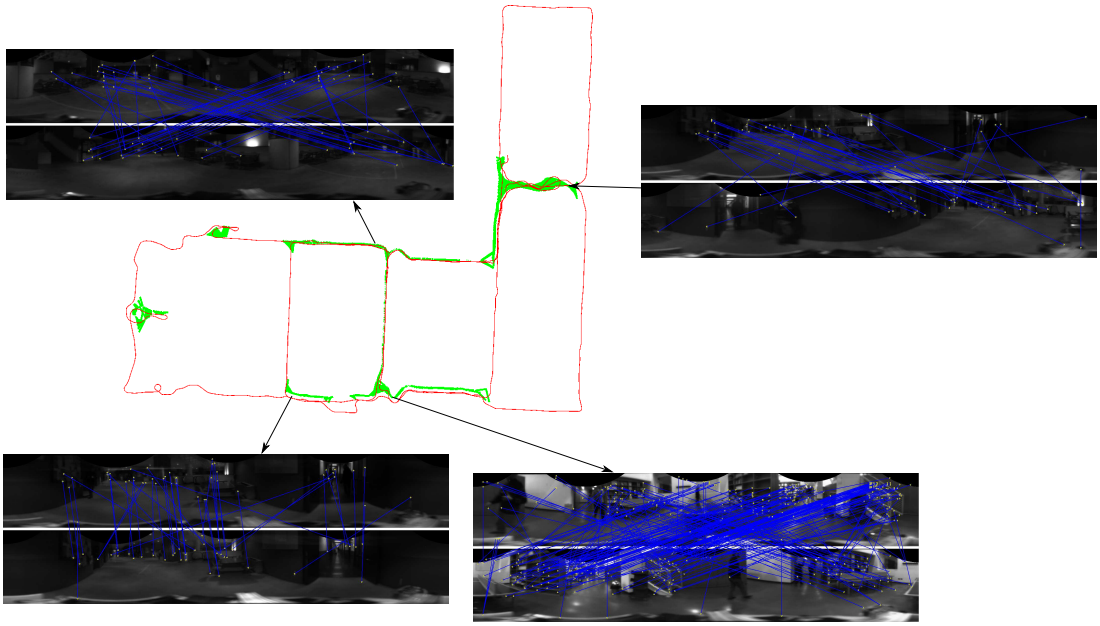


Figure 3.14: Bicocca (Trajectory in red, loop closures in green)

Sequence	Our Approach	FABMAP 2.0	DLoopDetector
PAVIN	84%	45%	61%
Cezeaux	53%	19%	31%
PAVIN-Jonco	78%	60%	65%
Cezeaux-Sealiz	71%	48%	58%
NewCollege	75%	43%	65%
Bicocca	78%	55%	61%

Table 3.5: Recall values on six sequences. The recalls of our approach listed above are the best values obtained of the two variants of ILLC that are discussed.

Process	Mean time (ms.)	Max. time (ms.)
SURF	120	132
Quantization	30	98
VLAD	38	66
Node Similarity	10	34
Image Similarity	6	14

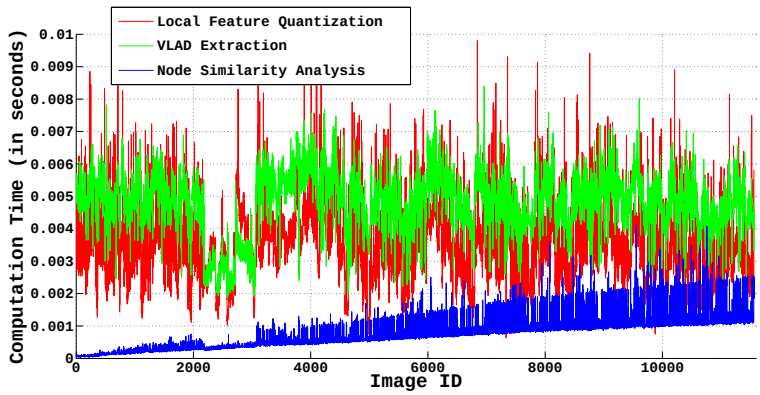
Table 3.6: The computational time of each step of our algorithm is given for the longest test sequence Cezeaux. Considering that the code is not thoroughly optimized and the code runs on a laptop with Intel Core I7 processor.

of the five modules. Feature quantization, VLAD extraction and node similarity analysis are performed within a few milliseconds as can be seen in Figure 3.15a. Figure 3.15b shows the image similarity analysis time along with the per frame processing time without local feature extraction. We can see that the per-frame processing time (excluding feature extraction) just reaches 40 milliseconds at maximum with 11571 images in the map. Almost 70% of the computation time is taken up by the local feature extraction. Including feature extraction, it takes a maximum of around 160 milliseconds per frame providing the capability of processing at least 5 frames per second. Such fast runtimes can even facilitate online map building (building the map during acquisition) on the datasets considered. However, (GLT11) reports runtimes that are three times faster in the worst case and eight times faster in the average case. This approach gains its power by using binary descriptors.

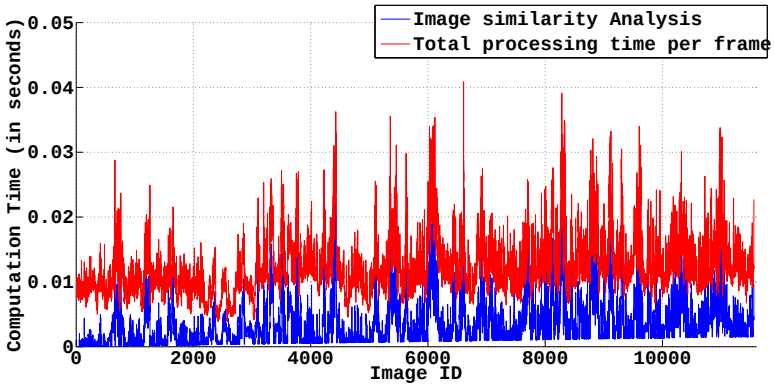
The changed framework by modeling the loop closure detection as classification problem neither improved nor decreased the computational performance since we had to perform similar operations as in our preliminary framework (KUM13). The difference in computational times comes up in the offline step while learning the NLLC parameters. Since we have to generate different clusterings of the data for various values of k to evaluate k^*, it takes almost a day per sequence to perform all the clusterings using a python script.

3.6 Conclusion and Future Work

In the present work, we proposed a hierarchical appearance based loop closure detection framework. The process of map building and the two phases of loop closure have been discussed in detail. First, we manually tune the parameters and then we change the algorithm to classification framework by proposing auto-



(a)



(b)

Figure 3.15: Run-times for various modules of the loop closure algorithm. Figure 3.15a plots run-times for the feature quantization time, VLAD extraction time and Node similarity analysis. Figure 3.15b shows the run-times of image similarity analysis and the total time taken to process each image frame.

matic parameter learning algorithms. Experimental results obtained on publicly available sequences have been reported and analyzed. The present approach has been compared with two state of the art approaches and also with the preliminary version. Experimental results demonstrating the sparsity, accuracy and computational time efficiency achieved by using are presented. It has been shown to be better in performing loop closure without geometric verification step.

Large scale operability was also shown to be possible with more than 10000 images in the sequences. The code of our algorithm will be made publicly available soon.

Although the results seem satisfactory, there is still room for improvement in three aspects. The first aspect is that of parameter learning for NLLC, for which we perform 100 different clusterings on each sequence. This process is very time consuming and can take several hours. Therefore, the first possible improvement is in computation time of paramater learning process of NLLLC. The second improvement is to use an improved VLAD with Fisher encoding (JPD+12) which is shown to offer better accuracy than plain VLAD. The third area is to replace SURF descriptors with binary descriptors like ORB (RRKB11) or BRIEF (CLO+12) which are faster to compute and storage efficient; VLAD/Fisher encoding should be adapted to binary descriptors by using Hamming distances. Therefore, we envision to address these problems in our future work.

4

Hybrid Mapping

This chapter discusses development of hierarchical mapping framework and proposes two strategies. The first strategy utilizes our hierarchical loop closure algorithm as it is presented in the previous chapter jointly with 3D reconstruction algorithm. Therefore, it is called as hybrid topo-metric map. The second upgrades this combination by adding semantic information into it. In other words, the semantic information allows us to tag the environment and also to decide automatically between metric and topological model. Therefore, it is called as hybrid semantic map.

Our topo-metric map achieves to combine metric and topological information by focusing on separability of maps and hierarchy. It proposes a hierarchical map representation which uses our image sequence partitioning (ISP) technique. The hierarchical map built can be understood as a topological map with nodes corresponding to certain regions in the environment. Each node in turn is made up of a set of images acquired in that region. These maps are further augmented with metric information at those nodes which correspond to image sub sequences acquired while the robot is revisiting the already mapped areas. Metrical information becomes invaluable during autonomous robot navigation through these places which contains junctions and turns. Actually, the goal is to obtain better computational efficiency than pure metrical mapping techniques and better accuracy as well as usability for robot guidance and navigation compared to the topological mapping. Hence we call the resulting maps hybrid since they primarily contain topological information and metrical information at places that are important for navigation. Experimental results obtained on sequences acquired in an outdoor environment are provided to demonstrate our approach.

The second framework exploits metric, topological and semantic information. As it is for the first framework, a topological map is built on an input image sequence by using a sequence partitioning technique such that each node is rep-

resented by a set of images acquired in a particular region of the environment. At the same time, structure from motion based metric reconstruction is also computed over the images of each region. A Conditional Random Field based classification on the metrical information is used to semantically label the local robot path (road in our case) as straight, curved or junctions.

The main difference is coming from the fact that the first concentrates only on how to represent the spatial structure and it misses these important concepts such as junctions, straight roads and turns. Actually, it is an important missing point for traditional robot maps. For example, a metric map may represent the structure of a road but it does not pinpoint whether this road is straight, bent, turn or a junction. Moreover, it does not even mark that the given structure is a road. It is also similar in topological map. It gives the connection information between two nodes but it does not indicate the type of the connection if it is through a junction turn or a straight road.

In fact, we call this extra information as semantic and the maps which are integrating this into the traditional robotic maps as hybrid semantic maps. Thus, we propose to use the same strategy for a large scale robot map and we propose a multi-layer approach in which each layer corresponds to different level of abstraction and represents the environment as precise as it is required. Experimental results obtained on KITTI odometry sequences acquired in challenging urban environments are provided to demonstrate our approach.

The remainder of this chapter is organized as follows. Section 4.1 introduces the first framework and explanation of the each step of the algorithm is given in following sections. Then section 4.2 explains the second framework in details. Our experiments for the both frameworks are detailed in section 4.3.1. Finally, we argue the nature of our research and give a summary of our contributions in the section 4.4.

4.1 Hybrid Topo-Metric Framework

As it is shown in the flowchart figure 4.1, our mapping algorithm consists of two main blocks as Image Sequence Partitioning (ISP) and 3D reconstruction modules. It starts with local image feature extraction for each newly acquired image. Any of the current local feature detectors such as SIFT, SURF, etc. can be utilized at local image extraction step. While extracted local image features are used for data association between related images for 3D reconstruction modules, they are quantized into visual words by using a kd-vocabulary tree (PCI+08a) for ISP.

The Image sequence partitioning (ISP) module, which is the modified version of our hierarchical loop closure detection algorithm as it is explained in the pre-

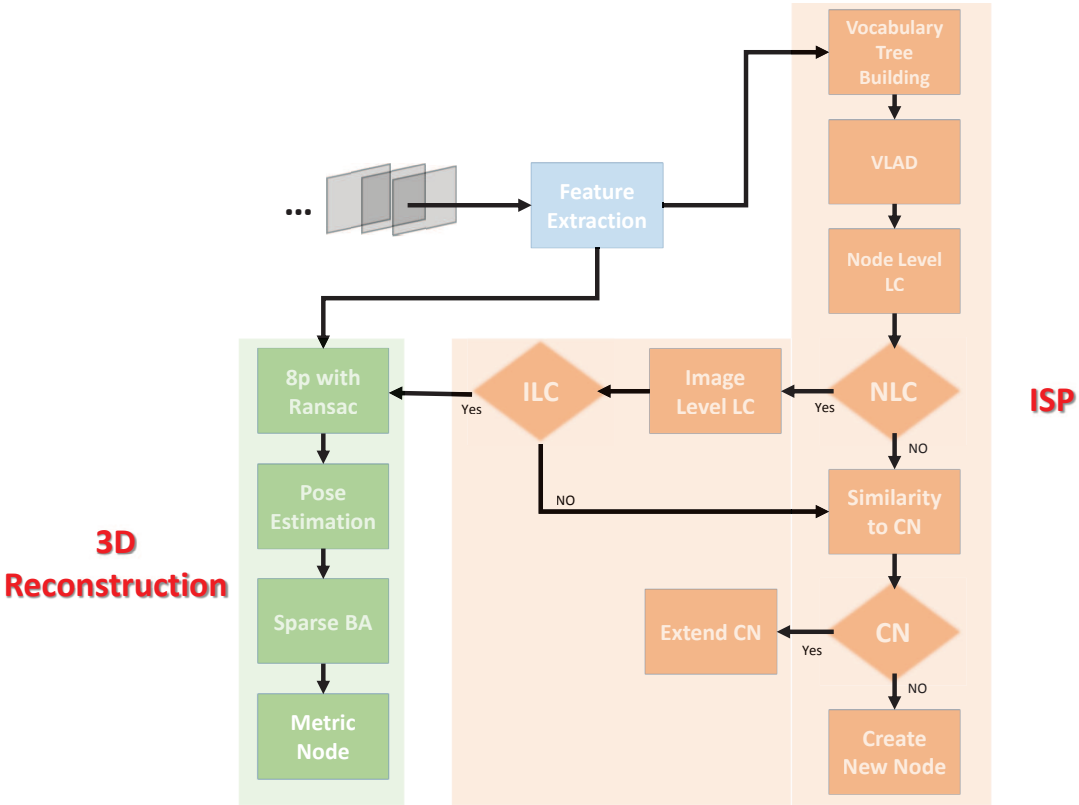


Figure 4.1: A global modular view of our hybrid topo-metric mapping framework. NLC and ILS stand for nood level loop closure and image level loop closure detection respectively. CN stands for current node.

vious chapter, is to classify image sequence in a graphical structure that consists of nodes and edges. Given each acquired frame, ISP realizes its function in three steps .

- The first step is to search for loop closures at node level and then at image level.
- If there is no loop closure, the second step is to compare the given frame with the current node and to expand the current node.
- If it is not similar to the current node either, the third step is to create a new node.

Following this strategy, a topological structure of the environment is captured hierarchically. Instead of checking the similarity between the current frame and each reference frame in the map, the node level loop closure downsizes the search space by constructing a subset of candidate nodes for image level loop closure step. The node level loop closure detection process is implemented using VLAD (Vector of Locally Aggregated Descriptors) (JDSP10a). For the image level loop closure detection, the combination of VLAD, local image descriptors and 3D information is used.

Finally, we construct a globally topological map which consists of two different types of nodes. In other words, the environment is modeled in metric or appearance based fashion under a global graphical model. The following sections discuss each of the above discussed modules in detail.

4.1.1 Image Sequence Partitioning Module

The first step of our mapping algorithm is to classify image sequence in a graphical structure that consists of nodes and edges. This strategy is adopted from the image sequence partitioning (ISP) part that is used in our previous chapter on hierarchical loop closure detection and an overview with the modifications is given here for the sake of completeness.

ISP has three main functions. The first one is to detect loop closures at node level and then at image level. If loop closure detection fails, the second function of ISP is to create a new node or expand an existing node. The last function of ISP module is to create a new node after the first two are eliminated respectively.

It starts with feature extraction for each frame of the video stream and then extracted features are quantized into visual words by using a kd-vocabulary tree (PCI+08a). Afterwards, a centroid is calculated for each feature corresponding to the word. Meanwhile, the vector difference between the feature descriptor

and the corresponding centroid are computed and accumulated for each quantized descriptor as quantization residues. These quantization residues of all the descriptors assigned to each word are summed up and stored as column vectors of the matrix d. Therefore, the matrix d will have k (vocabulary size) columns each corresponding to a word in the vocabulary and l rows indicating the feature dimension. A full VLAD descriptor (JDSP10a) is given by augmenting d matrix which needs to be projected to a lower dimensional space. For practicality, the full VLAD is compressed to a 128 dimensional V descriptor by using PCA projection matrix P. V descriptors are used to check the similarity between the nodes in the map and the current image which covers node level loop closure and deciding on expanding the current node or creating a new node.

If there is no loop closure detected, the second case of ISP is activated. The same similarity measure with the node level loop closure detection is used between the query image and the current node. The query image is added to the current node if they are similar. In case of dissimilarity between them, the final case of ISP is run to create a new node and an edge which represents a relative pose between the last image of the previous node and the query image.

4.1.2 Loop Closure Detection

The first function of image sequence partitioning module is to detect loop closures. Loop closure is the problem of detecting if the robot is revisiting a previously explored area of the environment. Loop closure detection is carried out at node level and image level respectively.

4.1.2.1 Node Level Loop Closure Detection

Measuring the similarity between each incoming frame and the reference nodes is called as node level loop closure check. This step is using the same algorithm with the node level loop closure detection step of the second framework given in the previous chapter although the application is different. Therefore, we shortly give the mathematical formulas without repeating the detailed explanation.

Each incoming frame is expressed with V vectors obtained by downsizing the standard global image descriptor VLAD (JDSP10a). While the images are represented by VLAD vectors, the nodes which consist in the set of visually similar images are modeled with ellipsoids whose axes are given by a multivariate Gaussian distribution over V^{N_i} of its member images I^{N_i}. That allows us to represent each node N_i by the mean vector μ^{N_i} and the covariance matrix $\Sigma_d^{N_i}$ and to be able to measure the similarity between the images and nodes as a distance variable. The distance between the VLAD vector V_q of current image I_q and the

distribution which represent the node can be calculated by using Mahalanobis distance such that

$$\Delta_{V_q}^{N_i} = \left(V_q - \mu^{N_i}\right)^T \Sigma_d^{N_i{}^{-1}} \left(V_q - \mu^{N_i}\right) \tag{4.1}$$

where $\Delta_{V_q}^{N_i}$ is the Mahalanobis distance. Current image I_q is considered as similar to a node N_i if the similarity condition $|\Delta_{V_q}^{N_i} - \mu_{ns}^{\Delta}| \leq 3\sigma_{ns}^{\Delta}$ where $\mu_{ns}^{\Delta}|$ and σ_{ns}^{Δ} are the mean and standard deviation is satisfied. The mean and standard deviation $\mu_{ns}^{\Delta}, \sigma_{ns}^{\Delta}$ are learned on training data which consists of image sequence, associated GPS readings and VLAD descriptors.

If node level loop closure turns positive with a group of candidate nodes $\mathbf{N}^*$, a reference set $\mathbf{I}^R$ is constructed with the member images of these nodes for the image level loop closure detection.

4.1.2.2 Image Level Loop Closure Detection

Given that node level loop closure turns positive with a set of candidate nodes $\mathbf{N}^*$, all the member images of these nodes constructs a reference image set $\mathbf{I}^R$ for image level loop closure detection step. This step aims to find the closest frame to particular query image I_q in the given reference set $\mathbf{I}^R$ by measuring the similarities image by image. The image level loop closure problem is defined as a classification problem as it is defined in the section 3.4 of previous chapter. Unfortunately, some of the similarity metrics presented in that framework are highly dependent on the physical character of omni-directional images. Moreover, some of them use the data obtained from odometry sensor as a secondary sensor. However, we use perspective camera images as an only sensor data which forces us to define new similarity metrics instead of the ones used in the section 3.4 for this problem. Fortunately, our hybrid model brings us an advantage which is the estimated relative poses as well as 3D points and facilitates the process of defining new similarity measures. Therefore, we present modified and new similarity metrics which are used as features for our Naive Bayes classifier (KF09) here.

- R is a discrete random variable and is computed based on matching statistics between the query and the reference image. The detailed explanation is given in subsection 4.1.2.3.

- V is a continuous random variable and defined as the euclidean distance between the query image VLAD descriptor and the reference image descriptor. The detailed explanation is given in subsection 4.1.2.4.

- G is a continuous random variable and is computed based on rotation and translation estimation between the query and the reference image. The detailed explanation is given in subsection 4.1.2.5.
- l is a continuous random variable and is computed based on reconstruction error resulted between the query and reference image. The detailed explanation is given in subsection 4.1.2.6.

The loop closure event is represented with a binary random variable L and the probability of loop closure is calculated by

$$p\left(L_i|\mathbf{X}\right), \qquad L_i \in \{1, 0\} \tag{4.2}$$

where $\mathbf{X} = \{V, S, N, R\}$ is the observed random variables set listed above. Equation 4.2 can be reformulated by using Bayes rule

$$p\left(L_i|\mathbf{X}\right) = \frac{p\left(L_i\right) p\left(\mathbf{X}|L_i\right)}{p\left(\mathbf{X}\right)} \tag{4.3}$$

where $p\left(\mathbf{X}\right)$ is normalization factor, $p\left(L_i\right)$ is the prior probability and $p\left(\mathbf{X}|L_i\right)$ is the likelihood function. The successful loop closure event is marked if the condition $p\left(L_1|\mathbf{X}\right) > p\left(L_0|\mathbf{X}\right)$ is satisfied. To check this condition, the equation 4.2 is rewritten by eliminating the denominator and using the conditional independence of each feature;

$$p\left(L_i|\mathbf{X}\right) \propto p\left(L_i\right) p\left(V|L_i\right) p\left(S|L_i\right) p\left(N|L_i\right) p\left(R|L_i\right) \tag{4.4}$$

Where the prior probability $p\left(L_i\right)$ is modeled with uniform distribution. The individual likelihood terms are learned empirically. A training data set which consists of images and their synchronous GPS readings is constructed for this aim. The GPS distance between non-adjacent image pairs is used to classify them as loop closure or non loop closure samples which construct the ground truth data set.

When the image level loop closure condition is also satisfied, the metric node construction process starts with the reference image set and query image. The local image features which are extracted for V descriptor are called as keypoints and used for this step. The first key frame I_1 for reconstruction is always the first query image of the loop closure set. The keypoints extracted in the query image are matched by tracking them into the member images of the reference image set. Given key frames and matched keypoints, the camera poses and 3D points are computed in two different types. For the first key frame triplets $\{I_1, I_2, I_3\}$ which consist of two images from reference set and query image, the method by Nister (Nis04) is utilized to build a baseline construction because we only have 2D feature matches. It consists of computing the essential matrix between the

first and last frames of the triplet using a sample of 8 point correspondences with a RANSAC. The best hypothesis is chosen by computing the re-projection error over the 3 views for all the matched interest points and keeping the one with the higher number of inlier matches. These 3D points estimated through the baseline are used to estimate the pose of upcoming loop closure image matches as well as go backward in reference image dataset until the image I_{r-n} in which there are not sufficient number of matches with I_r (reference image) and I_q (current image) for estimating its pose.

At the end of this process, key frames are represented by relative poses:

$$P_i^j \in R^{4\times 4} \tag{4.5}$$

where P_i^j is the camera projection matrix and i, j shows reference and current key frames respectively. 3D points are also represented with its visibility history

$$X_i^{k_i} \in R^4 \tag{4.6}$$

where $X_i^{k_i}$ is the ith 3D point's position defined in the reference key frame and k_i is the set of key frames which observe this point. By using the relative key frame representation, the 2D measurement of 3D points are given as

$$x_i^j = P_k^j X_i^k \tag{4.7}$$

where x_i^j is the re-projection of ith 3D point on the jth key frame. At each metric partition, there is exactly one reference key-frame. This reconstruction process continues until the detected loop closure set is completed. The last step is a sparse bundle adjustment for optimizing the camera poses and the 3D points by using the following cost function:

$$\mathbf{C} = \min \sum_{i=1}^{n_l} \sum_{j=1}^{n_i} \epsilon_{ij}^2 \tag{4.8}$$

where $\epsilon_{ij} = d(p_{ij}, x_i^j P_i^j X_i^j)$ is the Euclidean distance between the observed image point i and its estimated projection on keyframe j and I_{ij} is the binary variable which shows if the i-th image point is seen by j-th keyframe. Using LAPACK, a sparse bundle adjustment algorithm (MLD+09) is implemented and used for each metrical part independently.

4.1.2.3 Statistical Matching Constraint

We use a minimum matches constraint which eliminates the obvious false loop closures before going into calculation of the similarity measures. Although we

check the similarity between images by using global image descriptor VLAD, we employ also matching of local image features for essential matrix estimation and 3D reconstruction. Therefore, defining this constraint does not bring us extra computational cost. In fact, it accelerates the process. We define a ratio based on the extracted and matched local image features between reference and query image,

$$r = \frac{\#M_{qr}}{min(\#F_q, \#F_r)} \tag{4.9}$$

where $\#M_{qr}$ is the number of matched features between reference and query images and $\#F_q, \#F_r$ are the numbers of extracted features in the query and reference images respectively. Using this ratio we define a conditional probability function

$$\begin{cases} p(R = r|L_1) = 0 & \text{when } r \leq R_{min} \\ p(R = r|L_1) = \dfrac{2(r - R_{min})}{R_{dif}^2} & \text{when } r > R_{min} \quad and \quad r < R_{suf} \\ p(R = r|L_1) = \dfrac{2}{R_{dif}} & \text{when } r > R_{suf} \end{cases} \tag{4.10}$$

where R_{min} and R_{suf} are minimum and sufficient ratio values for loop closure event and R_{dif} is the absolute difference between them. Given that loop is closed, it is a simple conditional probability density CPD which gives zero probability if the calculated ratio by equation 4.9 is less than the minimum value and it is modeled with a linear increasing function between minimum and sufficient ratio values. if the calculated ratio is more than the sufficient value, it gives constant probability. The same function is also applied for the no loop closure samples by inverting it.

4.1.2.4 VLAD similarity

VLAD similarity is calculated based on the euclidean distance v between the descriptors of given images. The likelihood of this similarity is modeled with a Gaussian conditional distribution.

$$p(V = v|L_1) = \mathcal{N}(v|\mu_{v1,T1}, \sigma^2_{v1,T1}) \tag{4.11}$$

$$p(V = v|L_0) = \mathcal{N}(v|\mu_{v0,T0}, \sigma^2_{v0,T0}) \tag{4.12}$$

Where T_1 and T_0 are the training sets of positive and negative samples for loop closures respectively. The parameters $\{\mu_{v1,T1}, \sigma^2_{v1,T1}\}$ and $\{\mu_{v0,T0}, \sigma^2_{v0,T0}\}$

which maximize the likelihood functions on the positive and negative samples of training sets are calculated as,

$$\mu_{v,T} = \frac{1}{n}\sum_{i=1}^{n} v_i \qquad \sigma_{v,T} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(v_i - \mu_{v,T})^2} \tag{4.13}$$

4.1.2.5 Geometric similarity

VLAD descriptor based likelihood alone does not guarantee to find the closest image due to the limitation of 2D appearance information. Therefore, essential matrix estimation step which consequently gives relative rotation and translation between given images is added. Although estimated translation vector is up to scale, combination of rotation matrix and translation is still invaluable to eliminate the multiple matches to one reference image and to find the closest image to the reference.

A variable g_i is defined as $g_i = R_{iq}$ for the ith reference image and query image. The likelihood based on this geometric orientation variable is modeled with a Gaussian conditional distribution.

$$p(G = g|L_1) = \mathcal{N}(g|\mu_{g1,T1}, \sigma^2_{g1,T1}) \tag{4.14}$$

$$p(G = g|L_0) = \mathcal{N}(g|\mu_{g0,T0}, \sigma^2_{g0,T0}) \tag{4.15}$$

$$\mathbf{I^R} = \{\mathbf{I^{N_k}} : \forall_k N_k \in \mathbf{N}^*\} \tag{4.16}$$

The parameters $\{\mu_{v1,T1}, \sigma^2_{v1,T1}\}$ and $\{\mu_{v0,T0}, \sigma^2_{v0,T0}\}$ which maximize the likelihood functions on the positive and negative samples of training sets.The calculation of maximum likelihood estimates for parameters of a Gaussian distribution is given in equation 4.13 and the same strategy is followed here too.

4.1.2.6 Correlation Between Trajectories

Unidirectional loop closure means that a robot is passing through the same area in same direction again. Therefore, the two reconstruction obtained from this area should be consistent with each other. Using this fact, we aim to measure the similarity between the two reconstruction obtained by the current and previous pass from the same place. Assume that I_{r-8} and I_{q-8} be the first reference and query images of loop closure area. The average lateral distance between the cameras and the 3D points above the camera height is used to measure the similarity between two trajectories after fixing the scale between two reconstruction results. We divide the segment into 20 histogram bins along the lateral axis x

of the camera. Each bin of the histogram represents an equally divided lateral distance interval. The frequency of each bin l gives the number of 3D points fall into this interval. We normalize the histogram by dividing it by the total number of samples of the histogram. The normalized histogram frequencies are used to calculate conditional probability distribution as it is done in VLAD similarity and Geometric similarity.

4.1.3 Adding a new node

The third case of ISP module is to create a new node after the first two are eliminated respectively. In other words, if node level loop closure turns an empty set or if the conditional probabilities of no loop closure event exceed that of loop closure, then the query image is compared with the current node N_c. In case of being dissimilar with also current node, a new node is created with that particular query image as well as a connecting edge between the new node and the previous current node is also constructed.

The rotation matrix and up to scale translation vector are extracted from the estimated pose matrix $P_i^j \in R^{4\times4}$

$$P_i^j = K \times_i R^j \left[I -_i \breve{T}^j \right] \tag{4.17}$$

where K is the internal camera parameters matrix, ${}_iR^j$ is the rotation matrix and ${}_i\breve{T}^j$ is the relative translation vector between the related images. This information is used for defining the edges between nodes.

4.1.4 Parameters

This subsection discusses the parameters used for hybrid topo−metric mapping. Table 4.1 lists them all which have been introduced for this framework. Our algorithm starts with 64−dimensional Upright SURF (USURF-64) (BTG08) extraction for each frame of the video stream. 20% of images from each test sequence are selected randomly to construct a training data for learning vocabulary tree. The vocabulary tree has 128 nodes each of which has branching factor of 6 and contains 3 levels. In other words, it has $128 * 6^3 = 27648$ leaf nodes. VLAD vectors for each image are computed based on this learning process. PCA projection matrix P which is used to compress the full VLAD vector to a 128 dimensional V descriptor vector is also learned by using this training data.

During this learning process, the conditional probability distributions are also computed. There are only few parameters which are determined by the user based on experiments. P which is the minimum number of images between loop closure pair is the first parameter given by the user and fixed to 100. Another parameter

Parameter Name	Variable	Value
Bag of words vocabulary size for VLAD	k	128
USURF descriptor size	l	64
Full VLAD descriptor size	$k * l$	8192
PCA VLAD descriptor size		128
Vocabulary size for spatial similarity		27648
Prior Loop Closure Probability		0.5
Minimum number of images for loop closure pair		100
Minimum feature matching ratio loop closures	R_{min}	0.4
Sufficient feature matching ratio loop closures	R_{suf}	0.75

Table 4.1: Parameters

is the prior probability of a loop closure which is used to calculate overall image level conditional loop closure probability. Although it can be modeled in different ways such as increasing with the number of nodes in the map, we choose to model it with uniform distribution due to the simplicity. Minimum R_{min} and sufficient ratio R_{suf} parameters used for the calculation of statistical matching constrained are another example to the parameters selected based on experiments.

4.2 Hybrid Map Based on Road Semantics

As it is shown in the flowchart figure 4.2, our mapping algorithm consists of four main block. Given a new frame, we start with the 3D reconstruction part. We extract interest points on the new frame and match them with those of the previous frame by searching in a predefined window around each interest point. Based on the matching score, key frames of the image sequence are selected. Then, camera poses and 3D locations of interest points belonging to the key frames are computed. This is followed by sparse bundle adjustment for optimizing the poses and 3D points. We limit this 3D reconstruction process to a dynamically defined number of key frames over robot path. It should be noted that our aim is not to obtain a global metric map. However, we retain local metric information at certain areas for navigation purposes and for road path classification.

The next step is robot path classification which mainly comprises of detecting junction and geometric path orientation. The junction detection step considers 3D points and camera poses from 3D reconstruction step as its inputs and then searches for empty spaces along the left side of the robot path. On occurrence of

an empty space, which means that robot is probably passing through a junction, the key frames at that area are chosen for further processing for road/curb border detection.

Road/curb border detection uses modified version of Kim's lane detection algorithm (Kim08) whose input is 2D interest points extracted on the key frames acquired over the empty areas. Using the output of this algorithm, the existence or non-existence of a junction over the empty space is decided. Nonetheless, if this step does not give positive results or even if there is no empty space detected, the key frames are considered for geometric path orientation check such as straight or curvy roads. Finally, either the current node is extended if the path orientation is consistent with the current node's path orientation or a new node is created based on the changed orientation.

If the classification results in a junction detection, the corresponding key frames are labeled as junction area. Extracted 2D features of those key frames are quantized into visual words by using a visual vocabulary tree learned on a training junction data set. Then a node level and image level loop closure check is performed as it is explained in our loop closure detection chapter 3 with a design difference which means loop closure search area is limited within the junction nodes instead of all map. For the sake of completeness, node level loop closure downsizes the search space by constructing a subset of candidate nodes for image level loop closure step. Moreover, an empty set can be also produced which shows that newly acquired image belongs to the current junction node.

Finally, we construct two different types of nodes which models the environment in metric or appearance based fashion and give us three essential semantic features for an outdoor environment. The following sections discuss each of the above discussed modules in detail.

4.2.1 Feature Extraction and 3D Point Cloud Generation

It starts with USURF (BTG08) extraction for each frame of the video stream acquired by a monocular calibrated camera. The input frames are divided into a fixed number of grids and only one feature per grid is extracted so that we could speed up feature extraction process and increase the performance of matching as well as 3D reconstruction process. Once the features are extracted, they are matched within a fixed search region around the interest points from the previous frame. When the ratio between the number of matched features and the total number of the extracted features of the current frame is below a threshold, current frame is considered as a new key frame.

$$\frac{\#(d_i \cap d_c)}{\#(d_i \cup d_c)} \leq \tau \tag{4.18}$$

Where d_i and d_c represent the extracted descriptors of ith keyframe and the current frame respectively. By selecting the key frames from the reference video stream, we guarantee that there is enough camera motion between two frames for 3D reconstruction and the sparsity of our mapping algorithm is improved since we only consider key frames instead of every incoming frame.

The rest of the process shares the same formulation with the last step of image level loop closure detection module 4.1.2.2 of our topo-metric framework. Therefore, we are not repeating it here one more time.

4.2.2 Robot Path Classification

The goal of this step is to classify each incoming key-frame as intersections or non-intersections. For this, we use the 3D point cloud $\mathcal{P}_{3D}$ built over the n immediately preceding frames. The point cloud is expressed in the coordinate system $\mathcal{F}$ corresponding to the camera pose of the latest key-frame.

In order to simplify the explanations, we consider only the left side however the same can be done for the right side either. Since we assume that the vehicle is always traversing on the right sight of the road the monocular images only capture the environment on the left side of vehicle. Hence we use only that part of the local environment in classification.

The local environment (w.r.t the latest key-frame) falling on the left side of the robot's path is used for the classification task. The local environment is divided into a set $\mathbf{S}$ of n segments of uniform-length and parallel to the ground plane as shown in figure 4.3. The label of each segment $s_i \in S$ is represented by a categorical random variable l_i which can take a value from predefined binary label set $B = \{Junction, Non-junction\}$. The state of the environment as a whole can be represented by a set of random variables $\mathbf{l} = \{l_1, l_2, \ldots, l_n\}$.

Given the input $X = \{\mathcal{P}_{3D}, \mathcal{D}_{2D}\}$, we need to find a label set $\mathbf{l}$ that best describes the environment and maximizes a posterior distribution $P(\mathbf{l}/X)$. In order to avoid generatively modeling the complex dependencies between input data and labels, we utilize Conditional Random Fields (CRF) to directly learn the posterior distribution $P(\mathbf{l}/X)$. We estimate the maximum a posterior (MAP) labeling $\mathbf{l}^*$ given by;

$$\mathbf{l}^* = \text{argmax}_{\mathbf{l}\in\mathbf{L}} P(\mathbf{l}/X) = \text{argmin}_{\mathbf{l}\in\mathbf{L}} E(\mathbf{l}) \tag{4.19}$$

where $E(\mathbf{l})$ is an energy function which consists of unary and pairwise potentials, $\mathbf{L}$ is the set of all possible (2^n) labelings. The energy function $E(\mathbf{l})$ is described as

$$E(\mathbf{l}) = \sum_{i\in\{n\}} \psi_i(l_i) + \sum_{(i,j)\in\mathfrak{N}} \psi_{ij}(l_i, l_j) \tag{4.20}$$

where $\mathfrak{N}$, $\psi_i(.)$ and $\psi_{ij}(.,.)$ are the set of neighboring segments, unary potential functions and pairwise potential functions respectively. Before we give detailed information about the unary and pairwise potential functions which are used in the path classification, a brief description of CRF is given in the following section.

4.2.2.1 Conditional Random Fields

Conditional random field is a popular probabilistic method which combines the ability of graphical models to compactly model multivariate data with the ability of classification methods to perform prediction using large sets of input features. CRFs have been used in many areas such as natural language processing, computer vision, and bioinformatics.

Given a particular observation sequence x, conditional models define a conditional probability $p(Y \mid x)$ over label sequences. Conditional models are utilized to find the best label set for an observation sequence x which maximizes the conditional probability. Based on this conditional modeling, CRFs are a form of undirected graphical model for labeling and segmenting sequential data. Basically they construct a single log-linear distribution over label sequences given a particular observation sequence.

The probability of a particular label sequence y given observation sequence x is first introduced as a normalized product of potential functions by Lafferty et al. (LMP01). Each of them has the form of

$$\exp(\sum_j \lambda_j t_j(y_{i-1}, y_i, x, i) + \sum_k \mu_k s_k(y_i, x, i)) \tag{4.21}$$

where $t_j(y_{i-1}, y_i, x, i)$ is a transition feature function of the entire observation sequence x and the labels at positions i and $i-1$ as well as $s_k(y_i, x, i)$ is a state feature function of the ith label and the observation sequence. Meanwhile, λ_j and μ_k are parameters which are learned from training data.

The easiest way of defining feature functions which needs to represent the characteristic of the empirical distribution of the training is to use real-valued features $b(x, i)$ of the observation.

Each feature function takes on the value of one of these real-valued observation features $b(x, i)$ depending on the value of the current state in the case of a state function or previous and current states in the case of a transition function. Therefore, it is guaranteed that all feature functions are real-valued and they can be written as

$$F_j(y, x) = \sum_{i=1}^{n} f_j(y_{i-1}, y_i, x, i) \tag{4.22}$$

where $f_j(y_{i-1}, y_i, x, i)$ can be a transition or a state function. Using this representation, the conditional probability of a label sequence y given an observation sequence x is given as;

$$p(y \mid x, \lambda) = \frac{1}{Z(x)} \exp(\sum_j \lambda_j F_j(y, x)) \tag{4.23}$$

Where $Z(x)$ is a normalization factor. Here, estimating probability distributions from a set of training data is achieved by using maximum entropy. Entropy of a probability distribution is defined as a measure of uncertainty. Therefore, maximum value obtained if the distribution is close to the uniform.

Under the assumption of training data are independently and identically distributed, the products of $p(y \mid x, \lambda)$ over all training data can be written as the likelihood.

Maximum likelihood training is done by maximizing the the log-likelihood function. In the case of a CRF, the log-likelihood function is defined as

$$L(\lambda) = \sum_k \left[\log(\frac{1}{Z(x)}) + \sum_j \lambda_j F_j(y, x) \right] \tag{4.24}$$

Which is a concave function. Therefore, it guarantees the convergence to the global maximum.

4.2.2.2 Feature Functions

This section discusses the feature functions which construct unary and pairwise potentials for the path classification algorithm. The used feature functions are listed in the table 4.2 and they are classified regarding if they are used in the calculation of unary or pairwise potentials.

For unary potentials, we choose a set of exponential functions which express the empirical distribution of the training data and gives the cost of assigning a particular label to each segment

$$\psi_u = e^{-\sum_i (w_i F_i(X))} \tag{4.25}$$

where $F_i(X)$ is the feature functions which map the measurements to the feature space. On the other hand, the pairwise potentials represent the transition between sectors. Given the sectors, it can be seen as a measure of how neighbor segments i and j should interact with each other by considering spatial smoothness.

$$\psi = e^{-\sum_{ij} (O(F_i - F_j) w_{ij})} \tag{4.26}$$

Feature Function	Pairwise or Unary
Height	Unary
Relative Density	Pairwise
Multi Height Density	Pairwise
Tracking factor	Unary
Lateral Distance	Unary
Relative Width	Unary
Surface Estimation	Pairwise
Road Border Estimation	Unary

Table 4.2: The proposed feature functions are listed above and it is also marked that they are used in calculation of unary or pairwise potential.

Where $O(F_i - F_j)$ gives the relative change ratio between the same feature functions of two neighboring segments. In fact, it basically measures the percentage of the change in pairwise feature functions.

Height: The weighted average height of all the points in a segment are considered. This feature F_H is used as a part of unary potential computation and is given as,

$$F_H = \frac{1}{N_{S_i}} \sum\nolimits_{p \in S_i} e^{-|p_h - C_h|} p_h \tag{4.27}$$

Where N_{S_i} is the number of points in the segment, p_h is the height of each point in the segment and C_h is the average camera height of the trajectory. Using exponential weights, we emphasize the points around camera height and eliminate the ones coming from ground plane and so high objects such as sky. The average of these heights is used as the feature. Height features plays an important role in detecting free spaces as they contain very few objects above camera level.

Relative Density: Generally buildings, trees, parked cars have a certain density while roads, sky or free spaces have really low density. We observe that the number of 3D points decrease while getting close to a free space. This is used in unary potentials and is computed as,

$$F_D = N_{S_i} \tag{4.28}$$

The absolute difference in the number of 3D points between neighboring segments are used in pairwise potential calculation.

Multi Height Density: Each segment is divided into 12 histogram bins along the vertical axis y of the camera. Each bin of the histogram represents an equally divided height interval. The frequency of each bin j gives the number of

3D points fall into this interval. We normalize the histogram by dividing it by the total number of samples of the histogram. The normalized histogram frequencies are used in unary potentials. Percentage of decrease or increase in the number of points between each corresponding histogram bins of neighboring segments are used as pairwise potential calculation.

Tracking factor: We check the visibility frequency of the 3D points. We measure the number of frames in which each 3D point is tracked. This helps in eliminating the points which are produced by moving objects. For instance, while the 3D points coming from stable objects such as facade of surrounding buildings are tracked for a long time, the ones coming from moving objects such as other cars, buses etc, can only be seen in couple of frames. Tracking factor for each segment is expressed as a histogram. Each bin of the histogram represents the number of frames in which a feature has been observed. The frequency of each bin k represents the number of features that have been observed k times. We use a histogram of 7 bins with the last bin representing 8 or more observations. We normalize the histogram by dividing it by the total number of samples of the histogram. The normalized histogram features are used in unary potential calculation.

Lateral Distance to the camera trajectory: This feature is based on the lateral distance between camera trajectory and 3D objects in a segment. We divide each segment into 20 histogram bins along the lateral axis x of the camera. Each bin of the histogram represents an equally divided lateral distance interval. The frequency of each bin l gives the number of 3D points which fall into this interval. We normalize the histogram by dividing it by the total number of samples of the histogram. The normalized histogram frequencies are used in unary potentials.

Relative width: This feature gives the number of empty segments with connection information. We mark a segment as empty if the number of 3D points inside it is less than 60% of the average number of 3D points in all segments or the number falls more than 40% between neighboring segments. The feature function gets 0 if the segment is nonempty. It takes 1 or a value equal to the number of connected empty segments if the segment is empty. Using this feature, narrow empty spaces between buildings are eliminated from the real road junctions. It is used in unary potential calculations.

Surface Estimation: Although the 3D point cloud, which is estimated from a front moving monocular camera, is sparse and gives us upto scale depth estimation, an approximate local surface can be calculated by using relative depths. Such as, the 2d projections of 3D points are used and 2D Delaunay triangulation method (She96) is performed. The relative angle of dominant normal vector for each segment is used as unary potential. The dot products of normal vectors between neighboring segments are used in pairwise potential.

Road Border Estimation: Using the 3D point cloud, a rough 2D segmentation of the road area in front of the car is estimated in the key-frames and saved as image patches. These image patches are given to the lane detection algorithm (Kim08) which is modified according to our purpose. In other words, the classifier that is used for line detection in their algorithm is changed to detect curbs and road borders. The angle between the camera trajectory and the detected road border is calculated for each key-frame and the ones in which the difference is more than 30 degree are marked as 1 while the others are marked as 0. This feature added to our classifier by counting the ones and zeros in each segment. For each 3D point $X_j^{k_i}$ in each segment, k_i $i = 1, \ldots, n$ gives us the set of key-frames in which it is observed. A value which shows the points coming from possible junctions is calculated as

$$\vartheta_j = \frac{1}{|k_i|} \sum X_j^{k_i} \tag{4.29}$$

Where $X_j^{k_i}$ can have 1 or 0 based on the feature. We sum all ϑ_j values for each segment and divide it to the number of points inside the segment. Using this feature, we give high weights to the points which can be seen only inside the junctions while giving small weights to the rest.

4.2.2.3 Parameter Estimation & Inference

The values of potential functions are given by a linear combination of individual features. These weights are considered as the parameters of CRF. We maximize the log-likelihood of the conditional probability along with an l1-norm based regularization term (Goo03). Addition of the l1-norm makes the log-likelihood a strictly concave function and hence the parameter estimation reduces to finding the local optimum which is also the global optimum (global maximum). The solution is found using the limited memory BFGS (Broyden–Fletcher–Goldfarb–Shanno) algorithm (AG07) which is a second degree approach approximates and compactly stores the hessian matrix to search through the parameter space.

The inference problem involves finding the maximum a posteriori (MAP) label vector $\mathbf{l}^*$ that maximizes the conditional probability $P(\mathbf{l}|X)$. We use the generalized belief propagation (l-GBP) (YFW05) algorithm to find the MAP solution.

4.2.3 Loop Closure Detection

Loop closure is the problem of detecting if the robot is revisiting a previously explored area of the environment. By definition, a robot must turn to be able to close a loop and it can only be done by passing through junctions in modern road network structure. Using this observation, we divide the environment into

discrete places as it is described in section 4.2 and each place can be revisited only by passing through a connecting junction or junctions in our map. Therefore, loops can only be closed at junctions and we limit the search area into the nodes created at junctions.

Each time robot enters a junction, loop closure search is also activated and incoming keyframe is placed into the best among the three possibilities in topological level;

- it can correspond to a part which was previously visited,
- it can belong to the current topological node,
- it can lead the algorithm to create a new topological node.

For this aim, we utilize our hierarchical loop closure detection algorithm at node level and at image level, respectively as it is seen in Figure 4.2. Moreover, it is modified by limiting the search space with junctions and by adding metric validation step to increase the precision rate. The current section gives the details of the loop closure processes.

Extracted local image features of each key-frame are quantized into visual words by using a kd-vocabulary tree (PCI+08b) learned on a training dataset. The visual words are utilized for node and image level loop closure operations. To achieve an optimized loop closure performance, indexing technique is used to store visual words in the memory. Indexing is mainly used in text retrieval applications and inverted file (MR98) is the most common data structure to perform this task. Each key-frame is represented as a list of visual words and each word has an associated inverted file that gives a list of references to the key-frames in which the word is observed. In fact, inverted files gives us which key-frames observe that particular word and how many times. In this work, we utilize an extended version of inverted file which is compatible with our hierarchical loop closure detection by indexing image as well as node memberships of visual words respectively and is giving key-point locations of the visual words. This is named as Spatial Hierarchical Inverted files Figure 4.4 and answers three main questions:

- In which nodes did a particular visual word occur?
- In which key-frames of a particular node did that particular visual word occur?
- what are the key-point locations of a particular visual word in an image belonging to a particular node?

With each newly acquired keyframe in junction, a similarity criterion is calculated;

$$\begin{aligned} p(I_t = i|F_t, \mathbf{M}_{t-1}) &= \frac{p(F_t|I_t = i, \mathbf{M}_{t-1})p(I_t = i|\mathbf{M}_{t-1})}{p(F_t|\mathbf{M}_{t-1})} \\ &= \eta p(F_t|I_t = i, \mathbf{M}_{t-1})p(I_t = i|\mathbf{M}_{t-1}) \\ &= \eta p(F_t|I_t = i, \mathbf{M}_{t-1}) \\ &\sum_j p(I_t = i|I_{t-1} = j)p(I_{t-1} = j|\mathbf{M}_{t-1}) \end{aligned} \tag{4.30}$$

Where I_t is the image frame acquired at time t, F_t is its feature measurement and $\mathbf{M_{t-1}}$ is the existing map. By using recursive Bayesian filter, posterior probability of current frame can be written as multiplication of the likelihood term $p(F_t|I_t, \mathbf{M_{t-1}})$ and sum of $p(I_t|I_{t-1})$ the transition prior and $p(I_{t-1}|\mathbf{M_{t-1}})$ the posterior probability from the previous time step over the two neighboring hypotheses.

This similarity criterion calculated by equation 4.30 is used to place the query image into the best among the three possibilities based on the most distinct matches chosen by using nearest neighbor ratio (Low04b).

The first case corresponds to the loop closure detection which is carried out in two steps named as node level and image level. The second case corresponds to adding new frame into current node by measuring similarity between the current node and a newly acquired image. The third case corresponds to creating a new node with an edge which gives the relative rotation and translation information with the last node.

Node level loop closure downsizes the search space by constructing a subset of candidate nodes for image level loop closure step. Moreover, an empty set can be also produced which shows that newly acquired image either belongs to current node or belongs to a new node. Given the node level occurrence information which is retrieved from the associated spatial HIFs for each visual word of the query image $w_i \in \mathbf{W_q}$, the likelihood for node similarity is calculated by using Term Inverse Document frequency (AFDM08).

In case node level loop closure turns positive with a group of candidate nodes $\mathbf{N}^*$, a reference set $\mathbf{I}^R$ is constructed with the member images of these nodes.

$$\mathbf{I^R} = \{\mathbf{I^{N_k}} : \forall_k N_k \in \mathbf{N}^*\} \tag{4.31}$$

Where, $\mathbf{I}^{N_k}$ is the set of images belonging to the node N_k. For each visual word of the query image, similarity scores based on sub-linearly scaled term frequency and inverse document frequency STF-IDF are computed as in equation 4.32.

The main difference between the standard TF-IDF used at node level and STF-IDF is that the first one gives more importance to the repeating visual words. However, common visual words between reference and query image are more important for image level loop closure detection even if they occur only once on the corresponding images. Likelihood values are obtained by smoothing and normalizing these similarity scores. A posterior probability is calculated by using the recursive Bayesian filter given by equation 4.30 and images which have higher posterior probability than 0.8 are selected as winning images.

$$STF - IDF(w_i, I_k) = (1 + \log(n(w_i, I_k))). \log(\frac{n(I)}{n(w_i, I)}) \tag{4.32}$$

Although this hierarchical policy gives refined loop closure results, it does not guarantee that there is only one image which has a likelihood value above the threshold for each step and loop closures are detected at the closest robot locations. Indeed, it contains certain level of noise and temporal inconsistencies such as multiple matching to the same frame or they can be mapped backwardly at some points although we know robot always move forward direction. Therefore, an epi-polar geometry verification with RANSAC step is added. In fact, this step is extended to pose estimation with 3D reconstruction for the region in which robot enters and exits the loop closure.

4.2.4 Parameters

It starts with 64−dimensional Upright SURF (USURF−64) (BTG08) extraction for each frame of the video stream acquired by a monocular calibrated camera. The input frames are split into grids with each of them 7×4 windows and only one feature per grid is extracted so that we could speed up feature extraction process and increase the performance of matching as well as 3D reconstruction process. Once the features are extracted, they are matched within a fixed search region 50×50 around the interest points from the previous frame. When the ratio between the number of matched features and the total number of the extracted features of the current frame is below a threshold $\tau = 0.8$, current frame is considered as a new key frame. The average observed distance between keyframes is around 2 meters. The key-frames are used for 3D reconstruction step and there are approximately 400 new 3D points are constructed with each new key-frame. Here, our aim is not to get the best construction results. In fact we use this extracted metric information for path classification step and for the areas where a robot needs metric accuracy to navigate. Therefore, we build it within maximum 80 meter long distances.

The loop closure detection consists of 2 steps such as training and testing. Although, we use 3 sequences for testing, we build the training data by randomly se-

lecting 10% of images from all 22 sequences of KITTI Odometry dataset. USURF features extracted on all the training images are used in learning of bag of words vocabularies. We used a vocabulary tree with branching factor ($k = 6$) and levels ($l = 6$). During the topological map building, there are two parameters given as node similarity cut-off θ_N and image similarity cut-off θ_I. The best results are obtained by setting these two parameters to 0.5 and 0.7 respectively. More detailed explanation of the parameters related with loop closure and topological map building can be found in our previous papers (KUM13, KM14). The number of uniform length segments n for the path classification is chosen as 20. The rest of the classification parameters related with CRF are learned at training step.

For the training, we manually select junction and non junction urban areas areas from odometry dataset sequences. Each of them consists of 80 frames and is given to our 3D reconstruction algorithm . Resulting 3D point clouds are divided into 20 uniformly length segment and each segment is labeled manually in order to obtain ground truth. The feature functions are computed as well as normalized for the segments. The normalized results are given to the parameter estimation algorithm.

4.3 Experiments

PAVIN sequences which are part of the Institute Pascal data sets (IPDS) and a part of the popular KITTI odometry dataset are used for experiments.

The PAVIN sequences are acquired by a car-like robotic platform named VIPALAB on which there are a dozen of sensors. These sensors are an odometry system, an Inertial Measurement Unit (IMU) which is supported by RTK GPS, a front looking Lidar and several cameras. An on−board computer unit with a Middleware software allows us to collect data from its sensor suit. The platform has a control panel which allows us to drive the vehicle manually as well as autonomously by using the on board computer system for data collection and experiments. More information about the IPDS are available on the website [1]. For the experiments with PAVIN datasets, we use images captured by a fish eye camera with 185 degree horizontal field of view which is mounted in the front upper part of the Vipalab. The resolution is 1280×960 pixel and the test sequences are recorded at 15 fps. The snapshots from a Pavin sequence is shown in the Figure 4.5. We make use of perceptive images from PAVIN which contains an artificial urban environment with 600 meters trajectory length and the RTK−GPS is used to obtain the ground truth information.

The PAVIN sequences are utilized to test the first framework. Due to the miniaturized scale of PAVIN, it is not suitable for testing the second framework.

[1]The IP datasets website is: http://ipds.univ-bpclermont.fr/

High quality ground-truth information based on Real-time Kinematic GPS (RTK-GPS) is available for Pavin sequences of IPDS and the selected sequences of KITTI dataset. To demonstrate the performance of the proposed mapping strategy, smaller subsets are extracted from the sequences which originally contain a huge number of frames and then are used as input of the mapping algorithm. The mapping performance is shown based on loop closure accuracy and sparsity due to the global topological character of our map. To give complete idea about our strategy, some trajectories estimated for metric nodes are also demonstrated.

On the other hand KITTI dataset is acquired in large urban areas. For testing the second framework, we use a part of the popular KITTI odometry dataset (GLSU13) which contains various urban roads and junctions. The dataset consist of 22 sequences covering the trajectory of 39.2 km and contains 1241×376 undistorted images as shown in the figure 4.6. Out of these we use 3 sequences as they contain loop closures in urban areas spanning around 8 km of trajectory.

Our main assumption is that there is a direct relationship between junctions and the places where loops are closed. However, there is not a ground-truth which shows the locations of loop closures and junctions in this dataset. Hence, we prepare manually ground truth analysis of loop closures with the number of junctions that they contain in each selected sequences and show it in the tables 4.3, 4.4, 4.5. There are 9 loop closures and each of them contains at least one junction. Only exception to this observation is that if the sequence starts right after a junction, then it might be missed. We deal with this kind of problem by extending the detected junctions areas and combining the relatively close junctions under a common junction place in our algorithm.

Table 4.3: Ground truth transcript of loop closure intervals of the Sequence 00

Loop Closure Intervals			# of Junc.
0-100	$\Longleftrightarrow$	4452-4533	1
118-196	$\Longleftrightarrow$	1570-1640	1
386-412	$\Longleftrightarrow$	2443-2460	1
386-941	$\Longleftrightarrow$	3392-3844	5
2349-2462	$\Longleftrightarrow$	3292-3418	1
Total # of Frames : 4541			

Table 4.4: Ground truth transcript of loop closure intervals of the Sequence 05

Loop Closure Intervals	# of Junc.
31-121 $\Longleftrightarrow$ 2430-2512	1
565-787 $\Longleftrightarrow$ 1324-1530	1
819-885 $\Longleftrightarrow$ 2581-2627	1
Total # of Frames : 2761	

Table 4.5: Ground truth transcript of loop closure intervals of the Sequence 07

Loop Closure Intervals	# of Junc.
0-13 $\Longleftrightarrow$ 1060-1067	1
Total # of Frames : 1101	

4.3.1 Results

4.3.1.1 Hybrid Topo-Metric Framework

The performance of the first framework is demonstrated by using small subsets that are extracted from the PAVIN sequences. The first performance measure for a globally topological map is the balance between sparsity and accuracy. Sparsity is a parameter which is inversely proportional with the number of nodes in the map. Although higher sparsity is a preferable feature, the accuracy of the map is an another important issue which limits the sparsity. In our case extremely growing number of images per node causes us to miss true loop closures at node level loop closure detection. Therefore, we take the maximum sparsity which supplies us maximum recall rate in node level. The table 4.6 shows the sparsity of our first frame work on PAVIN sequences. Using hybrid strategy which means that the metric model is used at loop closures increases the average number of images under these nodes and therefore results with higher sparsity.

On the other hand, global accuracy of the first framework is given by precision recall curves based on loop closure detection. Figure 4.7 illustrates the precision-recall of the loop closure decisions achieved by the first framework. More detailed results on this part and detailed analysis on finding optimum recall and precision rate was given in the previous chapter therefore we give it here as a performance measure.

Sequence	#(Nodes)	#(images)/node	(Traj.)/node
PAVIN	68	17.6	21.15 m
PAVIN Jonco	221	19.72	22.2 m

Table 4.6: Node Statistics. #(Nodes) - Number of nodes of the map built on the sequence. #(images)/node - Average number of images represented by each node. (Traj.)/Node - Average trajectory length represented by each node.

Table 4.7: Structure of the topo-metric map for PAVIN.

The number of the frames	1196
The number of the nodes in the map	68
The number of the frames per node	17.6
The number of the metric nodes in the map	24

Topo-metric structure of the map is given in the table 4.7 for the first sequence and in the table 4.8 for the second sequence.

These tables show various node statistics of the map built on the PAVIN sequences. For the first sequence, there are 1196 images under 68 nodes in the map while there are 4378 images under 221 nodes in the map of the second sequence. Metric reconstruction is carried out separately for 24 nodes and 70 nodes in the maps respectively. Keeping the number of the metric nodes small compared to the topological nodes is important in the sense of computational efficiency as it is shown in the table 4.9.

Figures 4.8 and 4.9 show the global mapping results and also gives a detailed view for two specific parts of the map such as the left turn and roundabout. At the global view, the map is separated into metric and topological parts which are shown as green and red respectively. 3D reconstruction is carried out at the red places independently and the rest of the map is constructed based on topological model.

Table 4.8: Structure of the topo-metric map for PAVIN-Jonco.

The number of the frames	4378
The number of the nodes in the map	221
The number of the frames per node	19.72
The number of the metric nodes in the map	70

Table 4.9: Computational Time per frame in milliseconds.

Pose Estimation with SBA	430 ms
Similarity analysis	126 ms

4.3.1.2 Hybrid Semantic Framework

The features calculated from the 3D point cloud, camera poses and 2D features are given to the CRF module. The figure 4.10 gives the general framework of this step for three different junction models considered in this work. The first level of the classification starts with key frame selection and then 3D reconstruction is obtained. Based on this 3D point cloud and 2D features, feature functions are calculated and they are given to the CRF module.

The figures 4.14 and 4.15 show the junction detection results for each sequence on the trajectories of the real driving traces in the maps taken as screen-shots of Google Maps. Except of the right turns that robot passes without turning through, the other junction areas are detected correctly and shown on the figure with yellow boxes. However, there is still a chance to detect right turns in the case of multiple passes as it happens for the 15th junction of the Sequence OO shown in the figure 4.14.It is a right turn according to robot's direction and is not detected at the first pass of the robot due to the fact that we limit the path classification step with the left sight of the robot trajectory. At the second pass from this junction, the robot turns right through it and so it is detected without being able to associate it with its location at the first pass shown in detail in the figure 4.11. Using the loop closure detection which happens in the succeeding left turn, this right turn is also associated with the first pass and placed into our map correctly. In fact, this kind of places which contains close junctions are combined and saved as high resolution metric model with the help of multiple passes and loop closure detection.

Another issue we observe comes from the lack of detecting bidirectional loop closures which means traveling over a same path again in a different direction due to the fact that monocular images can not provide a visual perception in all direction. For instance the robot crosses the junction 4 in the first sequence 4.14 two times but with different directions which are perpendicular to each other. Therefore, it is shown as two different junctions in our map.

The addition of junction detection in our mapping algorithm definitely enhances the loop closure detection performance which is an important criteria for obtaining a concrete map. This is shown by using precision recall curves calculated for the sequence 00 and 05. The sequence 07 contains only one loop therefore it is not considered for the calculation of precision and recall curves although it is representative for urban junction detection case. Figure 4.13 demonstrates

the trajectory plots of the sequences with detected loop closures highlighted. Our method is compared with FABMAP (CN08a) as it is shown in figure 4.12. FABMAP results are calculated by utilizing OpenFABMAP code (GMW+11) with a standard configuration. It is trained with the same training set that we use for our loop closure algorithm. By changing the node similarity cut-off θ_N, the image similarity cut-off θ_I and the size of detected junction areas, we obtain precision recall pairs in the figure 4.12 for our algorithm.

Although precision recall curves have a strong meaning from the image retrieval perspective, it is not sufficient enough to evaluate performance of a loop closure detection from mapping perspective. Missing a loop completely or finding false loops have catastrophic results on resulting map. In other words, focusing on recall rates which means of matching each frame in each loop can be cause of creating false connections between impossible places or missing a part of environment in a map. Therefore we keep false positive and false negative detection of loop closure at zero instead of trying to increase recall rates at the frame level in our algorithm as it is shown in table 4.10.

Table 4.10: Comparison of ground truth number and detected number of loop closures are given. First row shows the total number of loop closures in overall dataset. Then the following rows give the values for each sequence separately. As it is seen our algorithm detects the loops without any false positives.

	Ground truth Loop Closure #	Detected Loop Closure #
Total	9	9
Seq. 00:	5	5
Seq. 05:	3	3
Seq. 07:	1	1

4.4 Conclusion

Two hybrid mapping models are proposed in this chapter. They share the global topological structure which organizes images into places and represents them as nodes. While the first framework is supplying only metric information at limited areas under this global structure, the hybridization of the second framework is more developed with the help of road semantics. Moreover, the second extends the global maps not only with metric but also with semantic information. Therefore,

the first framework can be seen as a preliminary work while the second is the complete version.

Topo-metric framework introduces a new way to combine metric and topological information in a common map for large environments. It exploits an efficient representation built from our appearance based hierarchical loop closure detection strategy that allows instant loop closing. Hierarchical loop closure algorithm with ISP module is augmented with geometric edges and metric partitions. We use metric partitions for the places visited multiple times and connect these areas with topological mapping strategy. It is tested on the sequences which are captured in an outdoor environment.

In the second framework, we have proposed another new and advanced hybrid mapping approach which integrates spatial and semantic information for obtaining a scalable and navigable representation of large urban environments. The proposed hybrid map has had the ability to choose automatically between representing the environment only with topological model or a topo-metric model that would not be possible without exploiting the semantic knowledge. Among the different high level semantic concepts available in urban environment, we have illustrated the possibility of extracting road junctions.This provides important advantages such as acquiring well-established hybrid map, improving loop closure detection, minimizing the usage of computationally expensive metric model without loosing the accuracy for navigation task, increasing scalibility capacity and achieving a suitable model for map matching as well as merging algorithms.

Real world sequences have been used to test our system with certain assumptions. A C++ code of our algorithm will be made publicly available soon. Due to the limited field of view offered by monocular vision, we have restricted our classification step with the left side of the robot path. However, these can be improved by using multi-camera setups which propose bigger field of view at sides and make the depth estimation more precise and rich for extracting the semantic information better.

In the future work we plan to test our algorithm with multi camera setup which can lift the necessity of traversing the places again in the same direction in order to associate them with already mapped areas. We will also focus on further investigation of the suitability of our map to facilitate map matching and merging with available global maps.

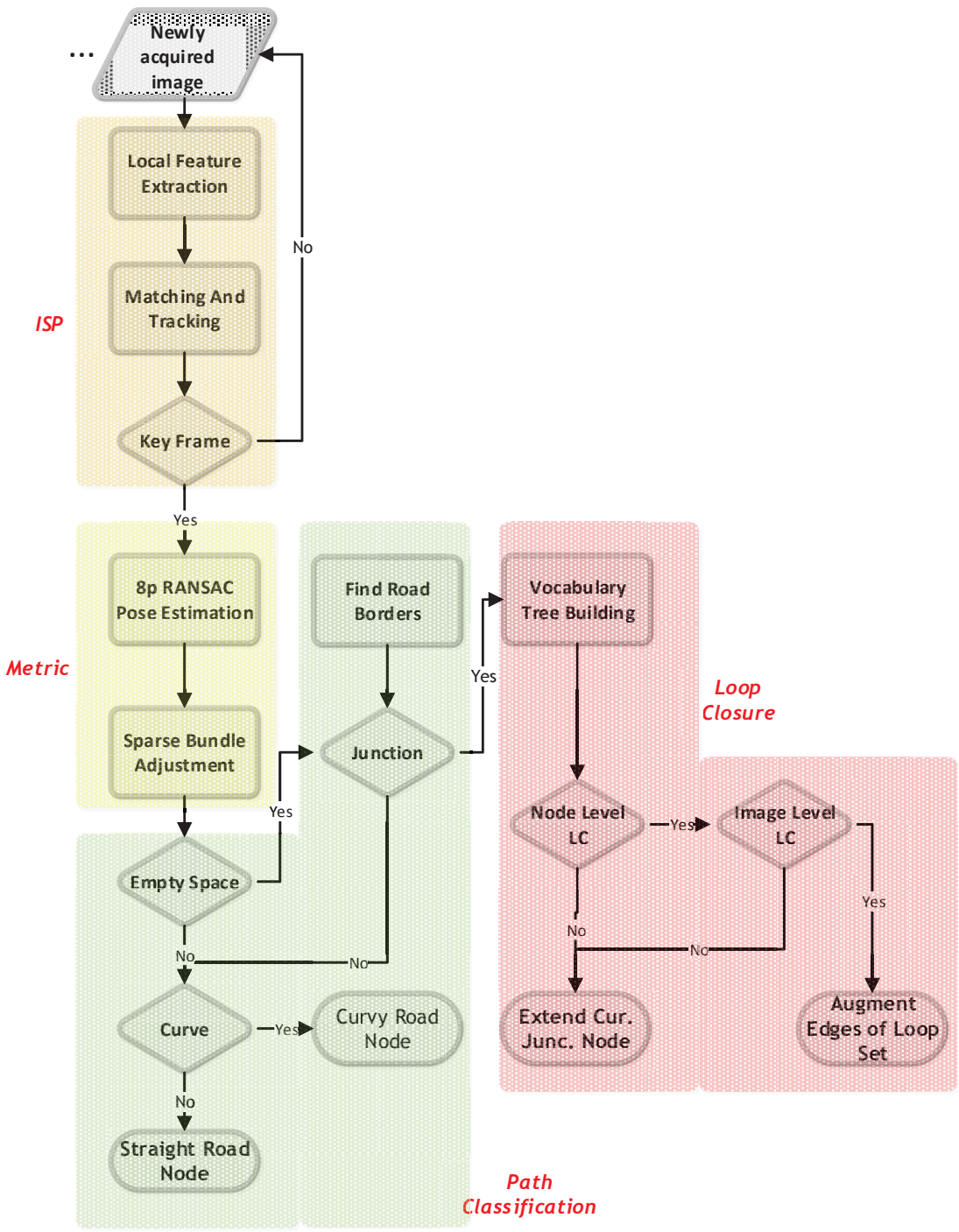


Figure 4.2: Flow chart. There are four main blocks given as image sequence partitioning 'ISP', local 3D reconstruction 'metric', path classification and loop closure detection.

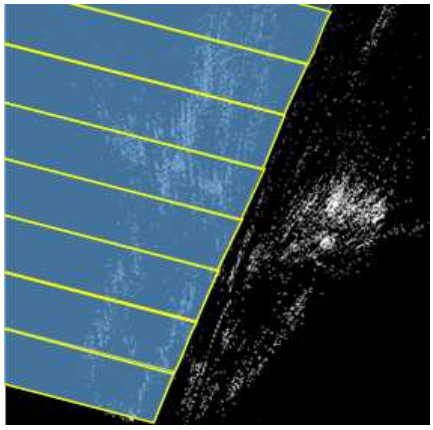


Figure 4.3: Toy example illustrating how the environment (point cloud) is divided into a set $\mathbf{S}$ of $n = 8$ segments of uniform-length and parallel to the ground plane.

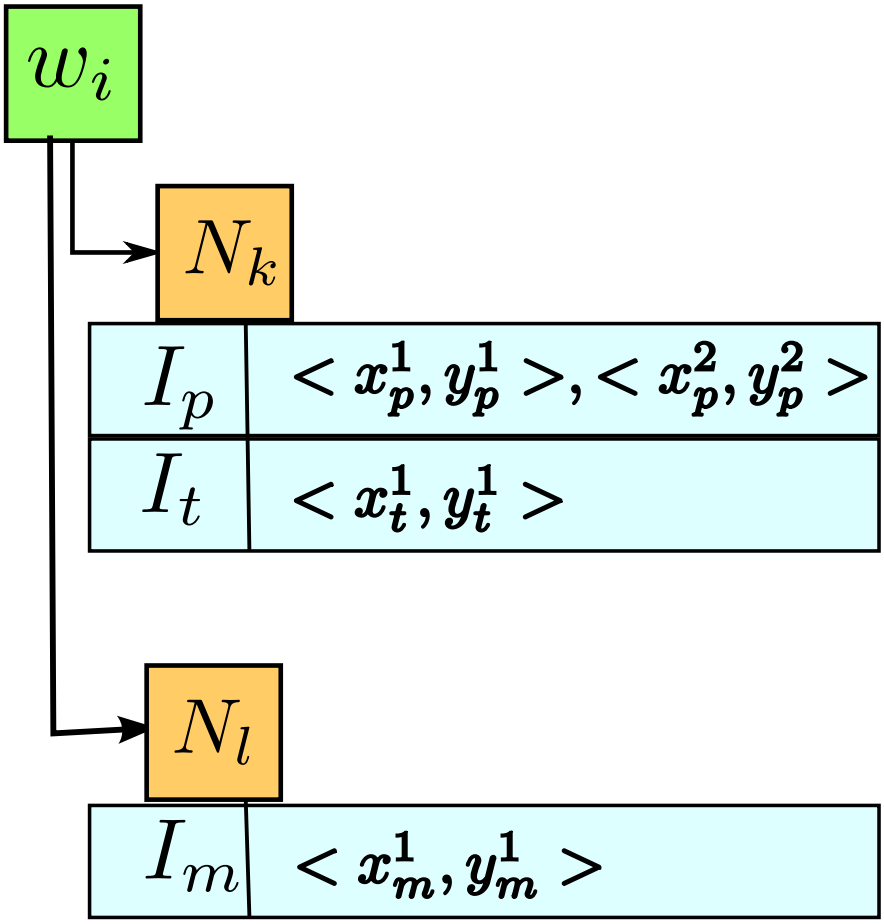
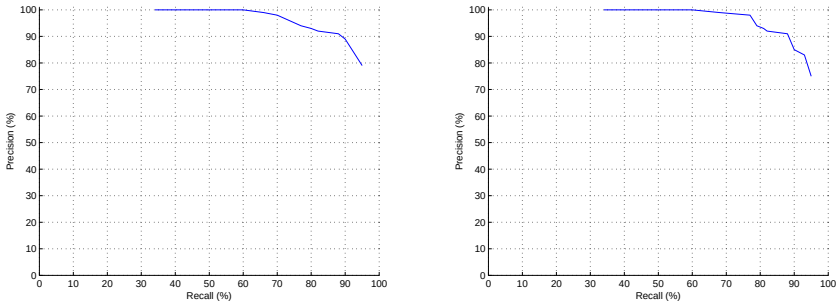


Figure 4.4: Spatial Hierarchical Inverted File models node N_j and image I_t membership information of visual words w_i as well as gives the key-points coordinates which constructs that particular word. In this toy example, visual word w_i is observed at the pixel coordinates x_p^1, y_p^1 of the image I_p as well as at the pixel coordinates x_t^1, y_t^1 of the image I_t under the node N_k and at the pixel coordinates x_m^1, y_m^1 of the image I_m under the node N_l.

Figure 4.5: Snapshots from Pavin dataset.

Figure 4.6: Example frames from selected sequences are shown.



(a) Precision-Recall for the PAVIN.

(b) Precision-Recall for the PAVIN Jonco.

Figure 4.7: Precision-Recall graphs on the reference datasets.

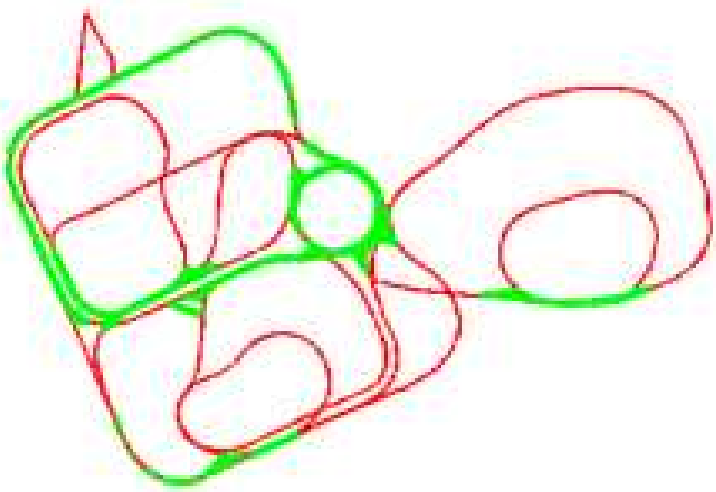
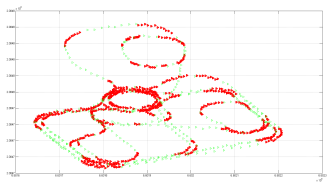
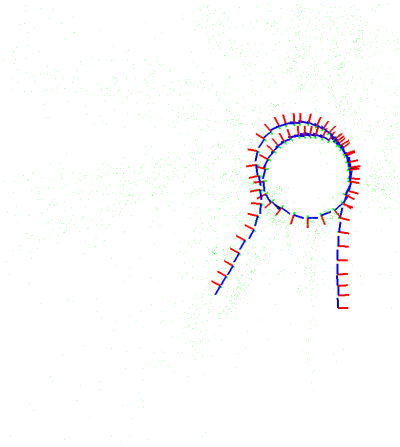


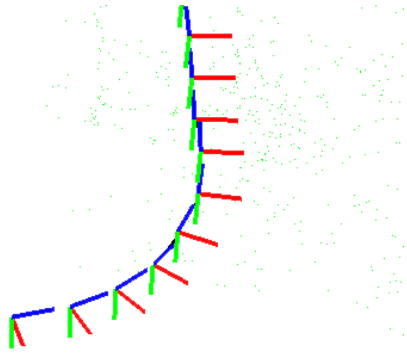
Figure 4.8: The overall map of first PAVIN sequence. The green areas represent the loop closure and therefore the metric nodes.



(a) Global map



(b) Reconstruction of Roundabout.



(c) Reconstruction of Road bend.

Figure 4.9: The experimental results. The first figure shows the overall map. The green areas represent the topological parts and the red areas represent the metric parts. The second figure zooms into the roundabout while robot is turning around more than one tour and the third zooms into the area while robot is driving into a bend.

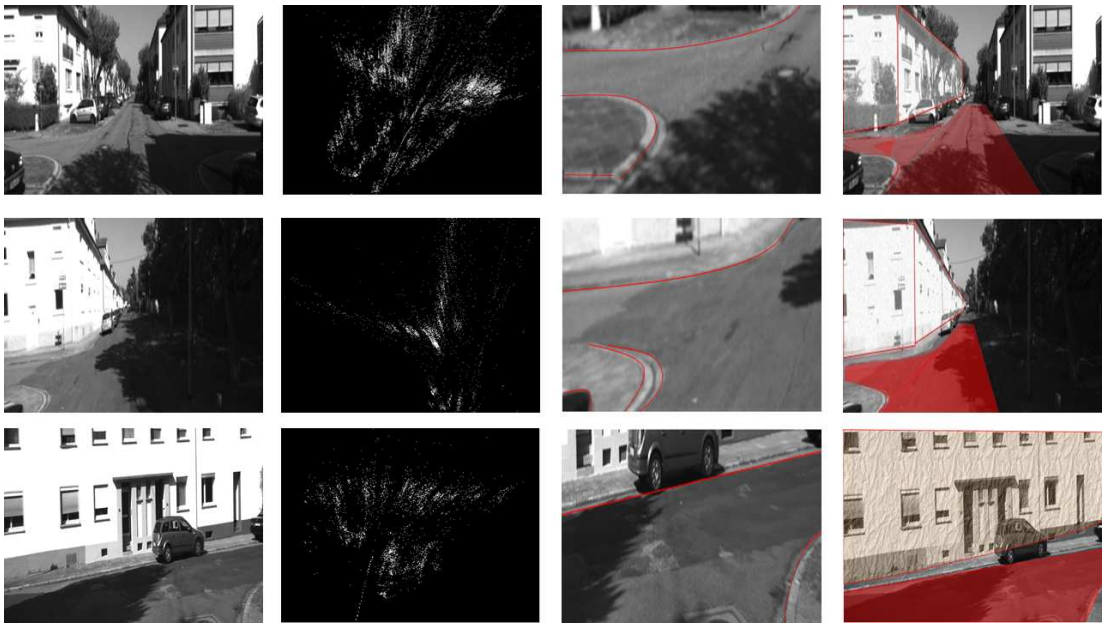


Figure 4.10: Simplified level of path classification step. The first column shows the three different type of junctions from the sequences. The second column shows the 3D point cloud extracted while robot is traversing these junctions. The third column shows the dominant line segments relative to the robot trajectory coming from the road surface. The last column shows the result of our CRF model in response of the feature functions which are using 3D and 2D available information.

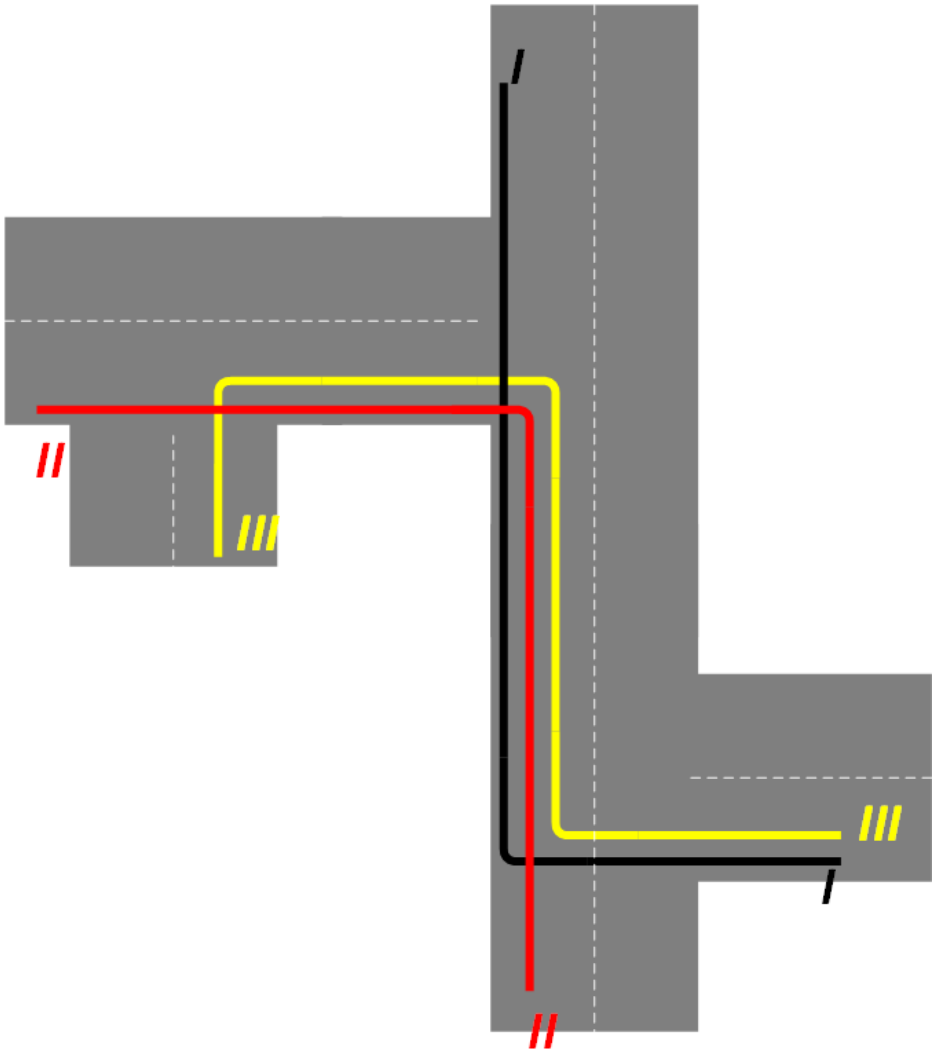


Figure 4.11: It shows that we miss the two right turns at the first passes. However, using the loop closure detections between first and second passes as well as second and third passes, these right turns are also added to our map.

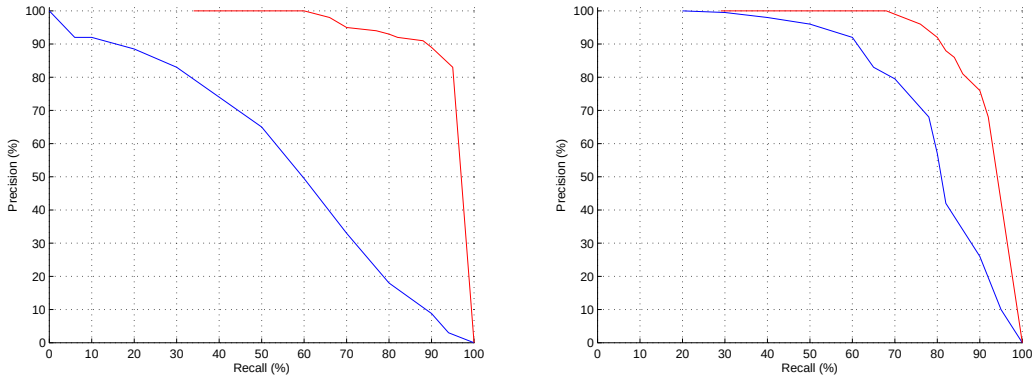


Figure 4.12: Precision Recall curve for the sequence 00 and 05 respectively. The red curve shows the performance of our algorithm while the blue is showing the perfomance of FABMAP.

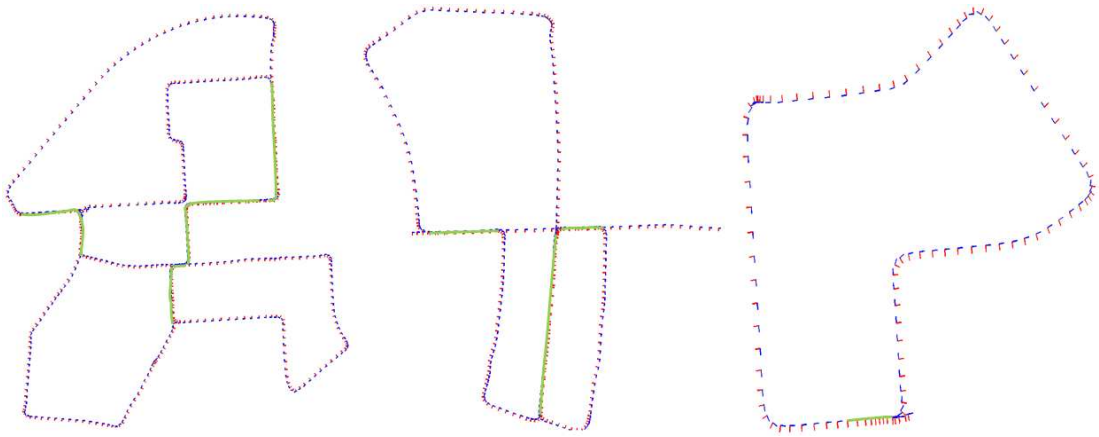


Figure 4.13: Dataset sequence trajectories plotted in blue. The detected loop closure regions are shown in green.

Figure 4.14: The results of path classification are shown on the real trajectory of sequence 00. There are 25 different junctions and some of them are visited by robot multiple times. Yellow squares show the detected junction and the red one shows the missed junction at which robot traverse through a straight road with a narrow right turn.

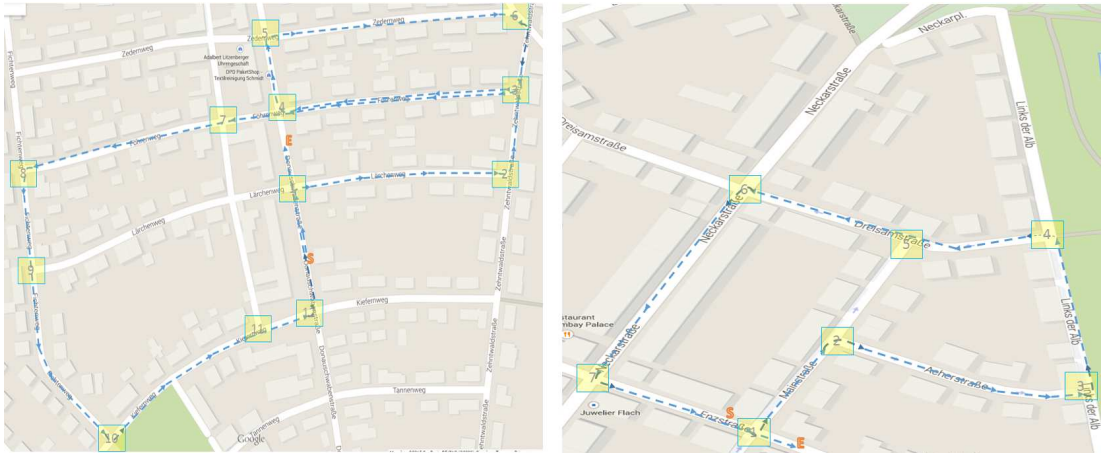


Figure 4.15: The results of path classification are shown on the real trajectory of sequence 05 on the left and sequence 07 on the right. There are 12 and 7 different junctions in each sequence respectively. Yellow squares show the detected junction and there are no missed junctions in these sequences.

5

Conclusion

To conclude, we will first give a summary of what is covered in this thesis. Then, we will outline possible directions for future work in this area.

5.1 Summary of the Thesis

In this thesis vision based hybrid map building in outdoors is proposed. Multiple branches from the computer vision field is brought together for developing systems capable of visual hybrid SLAM. Local and global feature extraction, descriptors, image matching, image retrieval, structure from motion, image understanding and labeling can be counted among these relevant branches of research.

Our system addresses some of the problems encountered in previous approaches. We address the ability to detect previously seen places in an efficient and robust manner, consistency, accuracy, and scalability when mapping out large environments. Each of these contributions makes the final map more useful in outdoors for applications such as autonomous driving and mobile robot navigation.

In Chapter 2, the literature on vision based SLAM are reviewed. We classify the field based on the model used for representing the environment seen by the system. Particularly,the methods are investigated under metric SLAM, topological SLAM, semantic SLAM and hybrid SLAM. We also highlighted the algorithms utilized in our approach which have been developed by others in the field.

In Chapter 3, we proposed a hierarchical topological loop closure detection framework. The process of map building and the two phases of loop closure have been discussed in detail. First, we have manually tuned the parameters of loop closure detection algorithm and then an automatic parameter learning based algorithm has been proposed. Experimental results obtained on six publicly available sequences have been reported and analyzed. The present approach has

been compared with two state of the art approaches and has been shown to be better in performing loop closure without geometric verification step.

In Chapter 4 we proposed two hybrid mapping models. In fact, they can be seen as steps of developing a complete hybrid map. They share the global topological structure which organizes images into places and represents them as nodes. The loop closure algorithm which is proposed in chapter 2 is utilized for this aim. While the first framework is supplying only metric information at limited areas under this global structure, the hybridization of the second framework is more developed with the help of road semantics. Moreover, the second extends the global maps not only with metric but also with semantic information. Therefore,the first framework can be seen as a preliminary work while the second is the complete version.

Adding metrical information between places is beneficial in robotic tasks like navigation and planning as well as facilitation of human interaction with the map. For example, topological maps cannot be understood by humans directly without the introduction of at least simple metrics like directionality of nodes. In this regard,our topo-metric framework introduces a new way to combine metric and topological information in a common map for large environments. It exploits an efficient representation built from our appearance based hierarchical loop closure detection strategy that allows instant loop closing.

Hierarchical loop closure algorithm with ISP module is augmented with geometric edges which gives directionality of nodes and metric partitions. Metric partitions can be understood as a place metrically represented using a 3D point cloud. We use metric partitions for the places visited multiple times and connect these areas with topological mapping strategy. Therefore; metric information encoding is not expensive regarding to the computational complexity and memory consumption since only certain parts of the map are locally consistent and global consistency is not demanded at all. It is tested on the sequences which are captured in an outdoor environment.

In the second framework, we have proposed another new and advanced hybrid mapping approach which integrates spatial and semantic information for obtaining a scalable and navigable representation of large urban environments. The proposed hybrid map has had the ability to choose automatically between representing the environment only with topological model or a topo-metric model that would not be possible without exploiting the semantic knowledge. Among the different high level semantic concepts available in urban environment, we have illustrated the possibility of extracting road junctions. Using the junction information of the environment, we can force loop closures over the whole trajectory between two junctions, given that a few loop closures were initially detected and the robot is moving in the same direction as the previous traversal. This provides important advantages such as acquiring well-established hybrid map, improving

loop closure detection, minimizing the usage of computationally expensive metric model without loosing the accuracy for navigation task, increasing scalability capacity and achieving a suitable model for map matching as well as merging algorithms. Real world sequences have been used to test our system with certain assumptions.

5.2 Future Work

The current hybrid mapping framework can be extended in several ways to be useful for robot navigation, path planning and human robot interaction.

Although the loop closure results presented in this thesis show improvement in the capability of loop closure detection approaches, there is still room for improvement in three aspects. The first aspect is that of parameter learning for NLLC, for which we perform 100 different clustering on each sequence. This process is very time consuming and can take several hours. The second improvement is to use an improved VLAD with Fisher encoding (JPD+12) which is shown to offer better accuracy than plain VLAD. The third area is to replace SURF descriptors with binary descriptors like ORB (RRKB11) or BRIEF (CLO+12) which are faster to compute and more storage efficient; VLAD/Fisher encoding should be adapted to binary descriptors by using Hamming distances. Therefore, we envision to address these problems in our future work.

The results present a significant improvement in the capability of robot mapping approach compared to earlier approaches. However, there is a lot of work remains to make these systems suitable for using in autonomous driving applications. Regarding to the physical system setup, we plan to test our algorithm with multi camera setup which can lift the necessity of traversing the places again in the same direction in order to associate them with already mapped areas. We believe that it can also improve the ability of detecting different types of intersections more robustly. We will also focus on further investigation of the suitability of our map to facilitate map matching and merging with global available maps. This will decrease the dependency on loops for correcting the maps and global maps will enhance the local robot map with its available rich semantic information. In fact, map merging can facilitate the exchange of knowledge between robots and humans independently of their location and can result in large scale hybrid mapping in terms of cities and countries.

Given that hybrid mapping approaches are complex systems which are the combination of several models and algorithms, the performance evaluation of these is ambiguous. However, each module which constructs the whole system together can be separately assessed. For example, the recall and the precision rates are used to show the loop closure performance. Meanwhile, the performance

of a metric map is evaluated by comparing it with ground truth trajectory or for indoor environments reconstructed metric map can be compared with CAD models of the buildings. A challenge for the upcoming research is to obtain a common evaluation metric which allows to compare those hybrid mapping techniques.

References

[ACS02] J. Andrade Cetto and Alberto Sanfeliu. Concurrent map building and localization with landmark validation. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 2, pages 693–696 vol.2, 2002. 17

[AD03] A. Ansar and K. Daniilidis. Linear pose estimation from points or lines. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(5):578–589, May 2003. 21

[AD09] Roy Anati and Kostas Daniilidis. Constructing topological maps using markov random fields and loop-closure detection. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 37–45. Neural Information Processing Systems, 2009. 27

[ADMF08] A. Angeli, S. Doncieux, J.-A. Meyer, and D. Filliat. Real-time visual loop-closure detection. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 1842–1847, May 2008. 33

[ADMF09] Adrien Angeli, Stéphane Doncieux, Jean-Arcady Meyer, and David Filliat. Visual topological slam and global localization. In *ICRA*, pages 4300–4305, 2009. 11, 33

[AFDM08] Adrien Angeli, David Filliat, Stéphane Doncieux, and Jean-Arcady Meyer. Fast and incremental method for loop-closure detection usingbags of visual words. *IEEE Transactions on Robotics*, 24(5):1027–1037, 2008. 11, 33, 99

[AG07] Galen Andrew and Jianfeng Gao. Scalable training of l1-regularized log-linear models. In *Proceedings of the 24th International Conference on Machine Learning*, pages 33–40, New York, NY, USA, 2007. ACM. 97

[AJK11] Supreeth Achar, C.V. Jawahar, and K.M. Krishna. Large scale visual localization in urban environments. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 5642–5648, May 2011. 32

[AR13] Charu C. Aggarwal and Chandan K. Reddy. *Data Clustering: Algorithms and Applications*. Chapman & Hall/CRC, 1st edition, 2013. 52

[BFMG08] J.-L. Blanco, J.-A. Fernandez-Madrigal, and J. Gonzalez. Toward a unified bayesian approach to hybrid metric–topological slam. *Robotics, IEEE Transactions on*, 24(2):259–270, 2008. 41

[BGJFM09] José-Luis Blanco, Javier González-Jiménez, and Juan-Antonio Fernández-Madrigal. Subjective local maps for hybrid metric-topological slam. *Robotics and Autonomous Systems*, 57(1):64–74, 2009. 41

[BHK12] Hernán Badino, Daniel F. Huber, and Takeo Kanade. Real-time topometric localization. In *ICRA*, pages 1635–1642, 2012. 41

[BNL+03] Michael Bosse, Paul Newman, John Leonard, Martin Soika, Wendelin Feiten, and Seth Teller. An atlas framework for scalable mapping. In *in IEEE International Conference on Robotics and Automation*, pages 1899–1906, 2003. 41

[BNLT04] Michael Bosse, Paul Newman, John Leonard, and Seth Teller. Simultaneous localization and map building in large-scale cyclic environments using the atlas framework. *The International Journal of Robotics Research*, 23(12):1113–1139, 2004. 22

[BPVT05] D.M. Bradley, R. Patel, N. Vandapel, and S.M. Thayer. Real-time image-based topological localization in large outdoor environments. In *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 3670–3677, Aug 2005. 25

[BRSM+09] Radu Bogdan Rusu, Aravind Sundaresan, Benoit Morisset, Kris Hauser, Motilal Agrawal, Jean-Claude Latombe, and Michael Beetz. Leaving flatland: Efficient real-time three-dimensional perception and motion planning. *Journal of Field Robotics*, 26(10):841–862, 2009. 18

[BSD+12] J.-C. Bazin, Yongduek Seo, C. Demonceaux, P. Vasseur, K. Ikeuchi, Inso Kweon, and M. Pollefeys. Globally optimal line clustering and vanishing point estimation in manhattan world. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 638–645, June 2012. 21

[BTG08] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. *Computer Vision and Image Understanding (CVIU)*, 110(3):346–359, 2008. 47, 89, 91, 100

[BTZK07] O. Booij, B. Terwijn, Z. Zivkovic, and B. Krose. Navigation using an appearance based topological map. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 3927–3932, April 2007. 41

[BXD+13] J. Bordes, P Xu, P. Davoine, P. Zhao, and T. Denoeux. Information fusion and evidential grammars for object class segmentation. In *nternational Conference on Intelligent Robots and Systems, 5th Workshop on Planning, Perception and Navigation for Intelligent Vehicles*, 2013. 39

[BZK09] O. Booij, Z. Zivkovic, and B. Kröse. Efficient data association for view based slam using connected dominating sets. *Robot. Auton. Syst.*, 57(12):1225–1234, December 2009. 41

[CAD11] G. Carrera, A. Angeli, and A.J. Davison. Slam-based automatic extrinsic calibration of a multi-camera rig. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 2652–2659, May 2011. 19

[CAHA07] Weiss Christian, Masselli Andreas, Tamimi Hashem, and Zell Andreas. Fast outdoor robot localization using integral invariants. In *The 5th International Conference on Computer Vision Systems*, 2007. 25

[CCVR12] T.A. Ciarfuglia, G. Costante, P. Valigi, and E. Ricci. A discriminative approach for appearance based loop closing. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 3837–3843, Oct 2012. 32

[CDM08] J. Civera, A.J. Davison, and J. Montiel. Inverse depth parametrization for monocular slam. *Robotics, IEEE Transactions on*, 24(5):932–945, Oct 2008. 23

[CDR+07] L. Clemente, A. Davison, I. Reid, J. Neira, and J. Tardós. Mapping large loops with a single hand-held camera. In *Proceedings of Robotics: Science and Systems*, Atlanta, GA, USA, June 2007. 23

[CGLR+10] C. Cadena, D. Galvez-Lopez, F. Ramos, J.D. Tardos, and J. Neira. Robust place recognition with stereo cameras. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 5182–5189, 2010. 32

[CGLR+11] J. Civera, D. Galvez-Lopez, L. Riazuelo, J.D. Tardos, and J.M.M. Montiel. Towards semantic slam using a monocular camera. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 1277–1284, Sept 2011. 37

[CL68] C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *Information Theory, IEEE Transactions on*, 14(3):462–467, 1968. 28

[CL85] R. Chatila and J. Laumond. Position referencing and consistent world modeling for mobile robots. In *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, volume 2, pages 138–145, Mar 1985. 17

[CLO+12] Michael Calonder, Vincent Lepetit, Mustafa Ozuysal, Tomasz Trzcinski, Christoph Strecha, and Pascal Fua. Brief: Computing a local binary descriptor very fast. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(7):1281–1298, July 2012. 25, 78, 121

[CN08a] M. Cummins and P. Newman. FAB-MAP: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, 27(6):647–665, 2008. 106

[CN08b] Mark Cummins and Paul Newman. FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. *The International Journal of Robotics Research*, 27(6):647–665, 2008. 11, 30

[CN09] Mark Cummins and Paul Newman. Highly scalable appearance-only SLAM - FAB-MAP 2.0. In *Proceedings of Robotics: Science and Systems*, Seattle, USA, June 2009. 11, 46

[CN10a] M. Cummins and P. Newman. Appearance-only SLAM at Large Scale with FAB-MAP 2.0. *The International Journal of Robotics Research*, November 2010. 70

[CN10b] Mark Cummins and Paul Newman. Appearance-only SLAM at large scale with FAB-MAP 2.0. *The International Journal of Robotics Research*, 2010. 11, 30

[CNT01] J.A. Castellanos, J. Neira, and J.D. Tardos. Multisensor fusion for simultaneous localization and map building. *Robotics and Automation, IEEE Transactions on*, 17(6):908–914, Dec 2001. 18

[Col10] M Collett. How desert ants use a visual landmark for guidance along a habitual route. *Proceedings of The National Academy of Sciences PNAS*, 107 (25):11638–11643, 2010. 23

[CRF11] A. Chapoulie, Patrick Rives, and D. Filliat. A spherical representation for efficient visual loop closing. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 335–342, Nov 2011. 34

[CRF12] Alexandre Chapoulie, Patrick Rives, and David Filliat. Topological segmentation of indoors outdoors sequences of spherical views. In *IROS*, pages 4288–4295, 2012. 26

[CRF13] A. Chapoulie, P. Rives, and D. Filliat. Appearance-based segmentation of indoors/outdoors sequences of spherical views. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 1946–1951, Nov 2013. 26

[CSCN11] C. Case, B. Suresh, A. Coates, and A.Y. Ng. Autonomous sign reading for semantic mapping. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3297–3303, May 2011. 37

[Dav03] A.J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *International Conference on Computer Vision, ICCV*, Nice, France, October 2003. 17, 19, 22, 23

[DCD11] Feras Dayoub, Grzegorz Cielniak, and Tom Duckett. A sparse hybrid map for vision-guided mobile robots. In Achim J. Lilienthal, editor, *ECMR*, pages 213–218. Learning Systems Lab, AASS, Örebro University, 2011. 41

[DRMS07] A.J. Davison, I.D. Reid, N.D. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(6):1052–1067, June 2007. 19

[DWB06] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *Robotics & Automation Magazine, IEEE*, 13(2):99–110, 2006. ix, 18

[ED06] E. Eade and Tom Drummond. Scalable monocular slam. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 469–476, June 2006. 23

[FEN07] F. Fraundorfer, C. Engels, and D. Nistér. Topological mapping, localization and navigation using image collections. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 3872–3877. IEEE, 2007. 30

[Fil07] D. Filliat. A visual bag of words method for interactive qualitative localization and mapping. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 3921–3926, 2007. 33

[FRJ+12] Yue Feng, Jinchang Ren, Jianmin Jiang, Martin Halvey, and Joemon M. Jose. Effective venue image retrieval using robust feature extraction and model constrained matching for mobile robot localization. *Mach. Vis. Appl.*, 23(5):1011–1027, 2012. 37

[GCCMC08] A.P. Gee, D. Chekhlov, A. Calway, and W. Mayol-Cuevas. Discovering higher level structure in visual slam. *Robotics, IEEE Transactions on*, 24(5):980–990, Oct 2008. 22

[GFO15] Emilio Garcia-Fidalgo and Alberto Ortiz. Vision-based topological mapping and localization methods. *Robot. Auton. Syst.*, 64(C):1–20, February 2015. 28

[GLSU13] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013. 102

[GLT11] D. Galvez-Lopez and J.D. Tardos. Real-time loop detection with bags of binary words. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 51–58, 2011. 11, 32, 46, 70, 76

[GMMW10] Arren Glover, Will Maddern, Michael Milford, and Gordon Wyeth. FAB-MAP + RatSLAM: Appearance-based SLAM for Multiple Times of Day. In *ICRA*, Anchorage, USA, 2010. 30

[GMW+11] A. Glover, W. Maddern, M. Warren, S. Reid, M. Milford, and G. Wyeth. Openfabmap: An open source toolbox for appearance-based loop closure detection. In *The International Conference on Robotics and Automation*, St Paul, Minnesota, 2011. IEEE. 106

[GN01] J.E. Guivant and E.M. Nebot. Optimization of the simultaneous localization and map-building algorithm for real-time implementation. *Robotics and Automation, IEEE Transactions on*, 17(3):242–257, 2001. 22

[GNTVG07] Toon Goedemé, Marnix Nuttin, Tinne Tuytelaars, and Luc Van Gool. Omnidirectional vision based topological navigation. *International Journal of Computer Vision*, 74(3):219–236, 2007. 34

[Goo03] Joshua Goodman. Exponential priors for maximum entropy models. In *In Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 305–312, 2003. 97

[GTV+05] T. Goedemé, T. Tuytelaars, G. Vanacker, M. Nuttin, L. Van Gool, and L. Van Gool. Feature based omnidirectional sparse visual path following. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1806–1811, Edmonton, Canada, August 2005. 33

[GWAH13] M. Gunther, T. Wiemann, S. Albrecht, and J. Hertzberg. Building semantic object maps from sparse and noisy 3d data. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 2228–2233, Nov 2013. 37

[Har97] Richard I. Hartley. Lines and points in three views and the trifocal tensor. *Int. J. Comput. Vision*, 22(2):125–140, March 1997. 21

[HBH+11] Albert S. Huang, Abraham Bachrach, Peter Henry, Michael Krainin, Daniel Maturana, Dieter Fox, and Nicholas Roy. Visual odometry and mapping for autonomous flight using an rgb-d camera. In *Int. Symposium on Robotics Research (ISRR)*, Flagstaff, Arizona, USA, Aug. 2011. 19

[HW79] J. A. Hartigan and M. A. Wong. A K-means clustering algorithm, applied statistics. *Applied Statistics*, 28:100–108, 1979. 51

[HZM06] Xuming He, Richard S. Zemel, and Volodymyr Mnih. Topological map learning from outdoor image sequences. *J. Field Robotics*, 23(11-12):1091–1104, 2006. 27

[JDSP10a] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *IEEE Conference on Computer Vision & Pattern Recognition*, pages 3304–3311, San Francisco, USA, 2010. 45, 47, 49, 82, 83

[JDSP10b] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *IEEE Conference on Computer Vision & Pattern Recognition*, pages 3304–3311, jun 2010. 34

[Jef61] H. Jeffreys. *Theory of Probability*. Oxford, Oxford, England, third edition, 1961. 59

[JPD+12] H. Jegou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid. Aggregating local image descriptors into compact codes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(9):1704–1716, Sept 2012. 78, 121

[JY11] Edward Johns and Guang-Zhong Yang. Global localization in a dense continuous topological map. In *ICRA*, pages 1032–1037. IEEE, 2011. 27

[JY13a] E. Johns and Guang-Zhong Yang. Dynamic scene models for incremental, long-term, appearance-based localisation. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 2731–2736, May 2013. 32

[JY13b] E. Johns and Guang-Zhong Yang. Feature co-occurrence maps: Appearance-based localisation throughout the day. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 3212–3218, May 2013. 32

[KA08] K. Konolige and M. Agrawal. Frameslam: From bundle adjustment to real-time visual mapping. *Robotics, IEEE Transactions on*, 24(5):1066–1077, Oct 2008. 19, 21

[KBC+10] Kurt Konolige, James Bowman, J.D. Chen, Patrick Mihelich, Michael Calonder, Vincent Lepetit, and Pascal Fua. View-based maps. *Int. J. Rob. Res.*, 29(8):941–957, July 2010. 19, 21, 30

[KCG13] Ioannis Kostavelis, Konstantinos Charalampous, and Antonios Gasteratos. Online spatiotemporal-coherent semantic maps for advanced robot navigation. In *International Conference on Intelligent Robots and Systems, 5th Workshop on Planning, Perception and Navigation for Intelligent Vehicles, IEEE*, 2013. 40

[KD10] Michael Kaess and Frank Dellaert. Probabilistic structure matching for visual slam with a multi-camera rig. *Comput. Vis. Image Underst.*, 114(2):286–296, February 2010. 19

[KF09] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009. 54, 84

[KG13] Ioannis Kostavelis and Antonios Gasteratos. Learning spatially semantic representations for cognitive robot navigation. *Robot. Auton. Syst.*, 61(12):1460–1475, December 2013. 37, 39

[KG15] Ioannis Kostavelis and Antonios Gasteratos. Semantic mapping for mobile robotics tasks: A survey. *Robot. Auton. Syst.*, 66(4):86–103, 2015. x, 40

[KGBN12] I. Kostavelis, A. Gasteratos, E. Boukas, and L. Nalpantidis. Learning the terrain and planning a collision-free trajectory for indoor post-disaster environments. In *Safety, Security, and Rescue Robotics (SSRR), 2012 IEEE International Symposium on*, pages 1–6, Nov 2012. 36

[Kim08] ZuWhan Kim. Robust lane detection and tracking in challenging scenarios. *Intelligent Transportation Systems, IEEE Transactions on*, 9(1):16–26, March 2008. 91, 97

[KK10] A.K. Krishnan and K.M. Krishna. A visual exploration algorithm using semantic cues that constructs image based hybrid maps. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 1316–1321, Oct 2010. 39

[KKG09] Jens Kessler, Alexander Koenig, and Horst-Michael Gross. An improved sensor model on appearance based slam. In Rüdiger Dillmann, Jürgen Beyerer, Christoph Stiller, Johann Marius Zöllner, and Tobias Gindele, editors, *AMS*, Informatik Aktuell, pages 153–160. Springer, 2009. 28

[KLY05] Jana Kosecká, Fayin Li, and Xiaolong Yang. Global localization and relative positioning based on scale-invariant keypoints. *Robotics and Autonomous Systems*, 52(1):27–38, 2005. 25, 27

[KM07] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, ISMAR

'07, pages 1–10, Washington, DC, USA, 2007. IEEE Computer Society. 19, 21

[KM14] Hemanth Korrapati and Youcef Mezouar. Vision-based sparse topological mapping. *Robotics and Autonomous Systems*, 62(9):1259 – 1270, 2014. Intelligent Autonomous Systems. 101

[KMHS03] H. Katsura, J. Miura, M. Hild, and Y. Shirai. A view-based outdoor navigation using object recognition robust to changes of weather and seasons. In *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 3, pages 2974–2979 vol.3, Oct 2003. 39

[KNG12] Ioannis Kostavelis, Lazaros Nalpantidis, and Antonios Gasteratos. Collision risk assessment for autonomous robots by offline traversability learning. *Robot. Auton. Syst.*, 60(11):1367–1376, November 2012. 37

[KTH10] Aram Kawewong, Sirinart Tangruamsub, and Osamu Hasegawa. Position-invariant robust features for long-term recognition of dynamic outdoor scenes. *IEICE Transactions*, 93-D(9):2587–2601, 2010. 27

[KTTH11] Aram Kawewong, Noppharit Tongprasit, Sirinart Tangruamsub, and Osamu Hasegawa. Online and incremental appearance-based slam in highly dynamic environments. *Int. J. Rob. Res.*, 30(1):33–55, January 2011. 27

[KUM13] H. Korrapati, F. Uzer, and Y. Mezouar. Hierarchical visual mapping with omnidirectional images. In *Intelligent Robots and Systems, 2013. IROS 2013. IEEE/RSJ International Conference on*, 2013. 34, 41, 76, 101

[KYS13] Dong Wook Ko, Chuho Yi, and Il Hong Suh. Semantic mapping and navigation: A bayesian approach. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 2630–2636, Nov 2013. 39

[LBJL07] T. Lemaire, C. Berger, I.K. Jung, and S. Lacroix. Vision-based slam: Stereo and monocular approaches. *International Journal of Computer Vision*, 74(3):343 – 364, 2007. 18

[LBKS13] Henning Lategahn, Johannes Beck, Bernd Kitt, and Christoph Stiller. How to learn an illumination robust image feature for

place recognition. In *Intelligent Vehicles Symposium*, pages 285–291. IEEE, 2013. 26

[LCN12] Y. Latif, C. Cadena, and J. Neira. Realizing, reversing, recovering: Incremental robust loop closing over time using the irrr algorithm. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 4211–4217, Oct 2012. 32

[LDVP14] Dieu Sang Ly, Cédric Demonceaux, Pascal Vasseur, and Claude Pégard. Extrinsic calibration of heterogeneous cameras by line images. *Mach. Vision Appl.*, 25(6):1601–1614, August 2014. 21

[LDW91] J.J. Leonard and H.F. Durrant Whyte. Mobile robot localization by tracking geometric beacons. *Robotics and Automation, IEEE Transactions on*, 7(3):376–382, Jun 1991. 17

[LFP12] Jongwoo Lim, Jan-Michael Frahm, and Marc Pollefeys. Online environment mapping using metric-topological maps. *I. J. Robotic Res.*, 31(12):1394–1408, 2012. 42

[LICM11] Ouiddad Labbani-Igbida, Cyril Charron, and El Mustapha Mouaddib. Haar invariant signatures and spatial recognition using omnidirectional visual information only. *Auton. Robots*, 30(3):333–349, April 2011. 26

[LK06] Fayin Li and J. Kosecka. Probabilistic location recognition using reduced feature set. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 3405–3410, 2006. 27

[LLY13] Huei-Yung Lin, Yu-Hsiang Lin, and Jia-Wei Yao. Scene change detection and topological map construction using omnidirectional image sequences. In *Proceedings of the 13. IAPR International Conference on Machine Vision Applications, MVA 2013, Kyoto, Japan, May 20-23, 2013*, pages 57–60, 2013. 34

[LM11] M. Labbe and F. Michaud. Memory management for real-time appearance-based loop closure detection. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 1271–1276, Sept 2011. 33

[LMP01] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and

labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. 93

[LMS+05] B. Lisien, D. Morales, D. Silver, G. Kantor, I. Rekleitis, and H. Choset. The hierarchical atlas. *Robotics, IEEE Transactions on*, 21(3):473–481, June 2005. 41

[LNJS01] Pierre Lamon, Illah R. Nourbakhsh, Björn Jensen, and Roland Siegwart. Deriving and matching image fingerprint sequences for mobile robot localization. In *ICRA*, pages 1609–1614. IEEE, 2001. 26

[Low04a] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004. 27

[Low04b] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004. 47, 99

[LSC13] Maohai Li, Han Wang 0001, Lining Sun, and Zesu Cai. Robust omnidirectional mobile robot topological navigation system using omnidirectional vision. *Eng. Appl. of AI*, 26(8):1942–1952, 2013. 28

[LSP+09] Ming Liu, D. Scaramuzza, C. Pradalier, R. Siegwart, and Qijun Chen. Scene recognition with omnidirectional vision for topological map using lightweight adaptive descriptors. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 116–121, 2009. 26

[LSS13] B. Le Saux and M. Sanfourche. Rapid semantic mapping: Learn environment classifiers on the fly. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 3725–3730, Nov 2013. 39, 40

[LZ12] Yancg Liu and Hong Zhang. Visual loop closure detection with a compact image descriptor. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 1051–1056, 2012. 26

[LZLS13] Jin Han Lee, Guoxuan Zhang, Jongwoo Lim, and Il Hong Suh. Place recognition using straight lines for vision-based slam. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 3799–3806, May 2013. 32

[Mah36] Prasanta Chandra Mahalanobis. On the generalized distance in statistics. *Proceedings of the National Institute of Sciences (Calcutta)*, 2:49–55, 1936. 50, 59

[MASS13] Andras Majdik, Yves Albers-Schoenberg, and Davide Scaramuzza. Mav urban localization from google street view data. In *IROS*, pages 3979–3986. IEEE, 2013. 28

[MB06] O.M. Mozos and W. Burgard. Supervised learning of topological maps using semantic information extracted from range data. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 2772–2777, Oct 2006. 39

[MCKG10] AC Murillo, P. Campos, J. Kosecka, and J. Guerrero. Gist vocabularies in omnidirectional images for appearance based mapping and localization. *10th IEEE workshop on omnidirectional vision, camera networks and non-classical cameras,(OMNIVIS), held with robotics, science and systems*, 2010. 25

[MGGR+12] A.C. Murillo, D. Gutierrez-Gomez, A. Rituerto, L. Puig, and J.J. Guerrero. Wearable omnidirectional vision system for personal localization and guidance. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 8–14, June 2012. 40

[MGLLC11] A. Majdik, D. Galvez-Lopez, G. Lazea, and J.A. Castellanos. Adaptive appearance based loop-closing in heterogeneous environments. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 1256–1263, 2011. 32

[MGS07] Ana Cris Murillo, JJ Guerrero, and C Sagues. Surf features for efficient robot localization with omnidirectional images. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 3901–3907. IEEE, 2007. 42

[MLD+09] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. Generic and real-time structure from motion using local bundle adjustment. *Image and Vision Computing*, 27(8):1178–1193, 2009. 21, 86

[MLIM11] P. Merveilleux, O. Labbani-Igbida, and E.-M. Mouaddib. Real-time free space detection and navigation using omnidirectional vision and

parametric and geometric active contours. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 6312–6317, May 2011. 26

[MLIM12] R. Marie, O. Labbani-Igbida, and E.M. Mouaddib. Invariant signatures for omnidirectional visual place recognition and robot localization in unknown environments. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 2537–2540, Nov 2012. 26

[MLIM14] R. Marie, O. Labbani-Igbida, and E.M. Mouaddib. Scale space and free space topology analysis for omnidirectional images. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 4451–4456, May 2014. 26

[MMKH12] Oscar Martinez Mozos, Hitoshi Mizutani, Ryo Kurazume, and Tsutomu Hasegawa. Categorization of indoor places using the kinect sensor. *Sensors*, 12(5):6695, 2012. 37

[MMW11] W. Maddern, M. Milford, and Gordon Wyeth. Continuous appearance-based trajectory slam. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3595–3600, May 2011. 30

[MMW12] Will Maddern, Michael Milford, and Gordon Wyeth. Catslam: Probabilistic localisation and mapping using a continuous appearance-based trajectory. *Int. J. Rob. Res.*, 31(4):429–451, April 2012. 30

[MR98] Justin Zobel Alistair Moffat and Kotagiri Ramamohanarao. Inverted files versus signature files for text indexing. *ACM Trans. Database Syst.*, 23(4):453–490, 1998. 98

[MRS08] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008. 28

[MSC+09] C. Mei, G. Sibley, M. Cummins, P. Newman, and I. Reid. A constant time efficient stereo slam system. In *BMVC*, 2009. 19

[MSG+07] A. C. Murillo, C. Sagüés, J. J. Guerrero, T. Goedemé, T. Tuytelaars, and L. Van Gool. From omnidirectional images to hierarchical localization. *Robot. Auton. Syst.*, 55(5):372–382, May 2007. 42

[MTJ+07] Oscar M Mozos, Rudolph Triebel, Patric Jensfelt, Axel Rottmann, and Wolfram Burgard. Supervised semantic labeling of places using information extracted from laser and vision sensor data. *Robotics and Autonomous Systems Journal*, 55(5):391–402, May 2007. 40

[MTKW02] Michael Montemerlo, Sebastian Thrun, Daphne Koller, and Ben Wegbreit. Fastslam: a factored solution to the simultaneous localization and mapping problem. In *Eighteenth national conference on Artificial intelligence*, pages 593–598, Menlo Park, CA, USA, 2002. American Association for Artificial Intelligence. 22

[MTRW03] Michael Montemerlo, Sebastian Thrun, Daphne Roller, and Ben Wegbreit. Fastslam 2.0: an improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *Proceedings of the 18th international joint conference on Artificial intelligence*, IJCAI'03, pages 1151–1156, San Francisco, CA, USA, 2003. Morgan Kaufmann Publishers Inc. 22

[MW08] M.J. Milford and G.F. Wyeth. Mapping a suburb with a single camera using a biologically inspired slam system. *Robotics, IEEE Transactions on*, 24(5):1038–1053, Oct 2008. 23

[MW12] Michael Milford and Gordon Fraser Wyeth. Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In *ICRA*, pages 1643–1649. IEEE, 2012. 26

[MWP04] M.J. Milford, G.F. Wyeth, and D. Prasser. Ratslam: a hippocampal model for simultaneous localization and mapping. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, volume 1, pages 403–408 Vol.1, April 2004. 23

[MYH11] H. Morioka, S. Yi, and O. Hasegawa. Vision-based mobile robot's slam and navigation in crowded environments. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 3998–4005, Sept 2011. 27

[NCH06] P. Newman, D. Cole, and K. Ho. Outdoor slam using visual appearance and laser ranging. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 1180–1187, 2006. 17

[NG09] T. Nicosevici and R. Garcia. On-line visual vocabularies for robot navigation and mapping. In *Intelligent Robots and Systems, 2009.*

IROS 2009. IEEE/RSJ International Conference on, pages 205–212, Oct 2009. 33

[NG12] T. Nicosevici and R. Garcia. Automatic visual bag-of-words for online robot navigation and mapping. *Robotics, IEEE Transactions on*, 28(4):886–898, Aug 2012. 33

[NGRTC10] C. Nieto-Granda, J.G. Rogers, A.J.B. Trevor, and H.I. Christensen. Semantic map partitioning in indoor environments using regional analysis. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 1451–1456, Oct 2010. 37

[Nis04] D. Nister. An efficient solution to the five-point relative pose problem. *Transactions on Pattern Analysis and Machine Intelligence*, 26(6):756–770, June 2004. 19, 21, 85

[NLHS07] Andreas Nüchter, Kai Lingemann, Joachim Hertzberg, and Hartmut Surmann. 6d slam—3d mapping outdoor environments: Research articles. *J. Field Robot.*, 24(8-9):699–722, August 2007. 17

[NRG+04] C.W. Nielsen, B. Ricks, M.A. Goodrich, D. Bruemmer, D. Few, and M. Few. Snapshots for semantic maps. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 3, pages 2853–2858 vol.3, Oct 2004. 36

[NS06] David Nister and Henrik Stewenius. Scalable recognition with a vocabulary tree. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, CVPR '06, pages 2161–2168, Washington, DC, USA, 2006. IEEE Computer Society. x, 28, 30, 31

[NWSS11] Gabriel Nützi, Stephan Weiss, Davide Scaramuzza, and Roland Siegwart. Fusion of imu and vision for absolute scale estimation in monocular slam. *J. Intell. Robotics Syst.*, 61(1-4):287–299, January 2011. 18

[OMSM03] Clark F. Olson, Larry Matthies, Marcel Schoppers, and Mark W. Maimone. Rover navigation using stereo ego-motion. *Robotics and Autonomous Systems*, 43(4):215–229, 2003. 19

[OT01] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *Int. J. Comput. Vision*, 42(3):145–175, May 2001. 25

[OT+06] Aude Oliva, Antonio Torralba, et al. Building the gist of a scene: The role of global image features in recognition. *Progress in brain research*, 155:23, 2006. 25

[PCI+08a] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, June 2008. 49, 80, 82

[PCI+08b] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008. 98

[PGV+04] M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. Visual modeling with a hand-held camera. *International Journal of Computer Vision*, 59(3):207–232, 2004. 20

[PJ11] Andrzej Pronobis and Patric Jensfelt. Understanding the real world: Combining objects, appearance, geometry and topology for semantic mapping. Technical Report TRITA-CSC-CV 2011:1 CVAP319, KTH Royal Institute of Technology, CVAP/CAS, Stockholm, Sweden, May 2011. 37

[PJ12] A. Pronobis and P. Jensfelt. Large-scale semantic mapping and reasoning with heterogeneous modalities. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 3515–3522, May 2012. x, 35, 37

[PMCJ10] Andrzej Pronobis, Oscar M. Mozos, Barbara Caputo, and Patric Jensfelt. Multi-modal semantic place classification. *The International Journal of Robotics Research (IJRR), Special Issue on Robotic Vision*, 29(2-3):298–320, February 2010. 37

[PN10] R. Paul and P. Newman. Fab-map 3d: Topological mapping with spatial and visual appearance. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2649–2656, May 2010. 32

[PPTN08] Lina M. Paz, P. Pinies, J.D. Tardos, and J. Neira. Large-scale 6-dof slam with stereo-in-hand. *Robotics, IEEE Transactions on*, 24(5):946–957, Oct 2008. 19, 22

[PT08] P. Pinies and J.D. Tardos. Large-scale slam building conditionally independent local maps: Application to monocular vision. *Robotics, IEEE Transactions on*, 24(5):1094–1106, Oct 2008. 19, 22

[PTRN12] R. Paul, R. Triebel, D. Rus, and P. Newman. Semantic categorization of outdoor scenes with uncertainty estimates using multi-class gaussian process classification. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 2404–2410, Oct 2012. 39

[RC10] Anna Romero and Miguel Cazorla. Topological slam using omnidirectional images: Merging feature detectors and graph-matching. In Jacques Blanc-Talon, Don Bone, Wilfried Philips, Dan Popescu, and Paul Scheunders, editors, *Advanced Concepts for Intelligent Vision Systems*, volume 6474 of *Lecture Notes in Computer Science*, pages 464–475. Springer Berlin Heidelberg, 2010. 28

[RC12] Anna Romero and Miguel Cazorla. Topological visual mapping in robotics. *Cognitive Processing*, 13(1):305–308, 2012. 28

[RD06] Ananth Ranganathan and Frank Dellaert. A rao-blackwellized particle filter for topological mapping. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 810–817. IEEE, 2006. 32

[RD11] Ananth Ranganathan and Frank Dellaert. Online probabilistic topological mapping. *I. J. Robotic Res.*, 30(6):755–771, 2011. 32

[RL11] A. Ranganathan and Jongwoo Lim. Visual place categorization in maps. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 3982–3989, Sept 2011. 37

[RRKB11] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *International Conference on Computer Vision*, Barcelona, 11/2011 2011. 78, 121

[RRTN08] David Ribas, Pere Ridao, Juan Domingo Tardós, and José Neira. Underwater slam in man-made structured environments. *J. Field Robot.*, 25(11-12):898–921, November 2008. 17

[RTMF08] Bryan C. Russell, Antonio Torralba, Kevin P. Murphy, and William T. Freeman. Labelme: A database and web-based tool for image annotation. *Int. J. Comput. Vision*, 77(1-3):157–173, May 2008. 37

[RZL+03] Paul E. Rybski, Franziska Zacharias, Jean-François Lett, Osama Masoud, Maria L. Gini, and Nikolaos Papanikolopoulos. Using visual features to build topological maps of indoor environments. In *ICRA*, pages 850–855. IEEE, 2003. 27

[SBS07] G. Schindler, M. Brown, and R. Szeliski. City-scale location recognition. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–7, 2007. 32

[SGST13] S. Sengupta, E. Greveson, A. Shahrokni, and P.H.S. Torr. Urban 3d semantic modelling using stereo vision. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 580–585, May 2013. x, 38, 39

[She96] Jonathan Richard Shewchuk. Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. In Ming C. Lin and Dinesh Manocha, editors, *Applied Computational Geometry: Towards Geometric Engineering*, volume 1148 of *Lecture Notes in Computer Science*, pages 203–222. Springer-Verlag, May 1996. From the First ACM Workshop on Applied Computational Geometry. 96

[SI09] C. Siagian and L. Itti. Biologically inspired mobile robot vision localization. *Robotics, IEEE Transactions on*, 25(4):861–873, Aug 2009. 34

[SK10] Gautam Singh and Jana Kosecká. Visual loop closing using gist descriptors in manhattan world. In *in Omnidirectional Robot Vision workshop, held with IEEE ICRA*, 2010. 25

[SK12] G. Singh and J. Kosecka. Acquiring semantics induced topology in urban environments. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 3509–3514, May 2012. 39

[SLA07] M. Saedan, Chee Wang Lim, and V.M.H. Ang. Appearance-based slam with map loop closing using an omnidirectional camera. In *Advanced intelligent mechatronics, 2007 IEEE/ASME international conference on*, pages 1–6, Sept 2007. 28

[SLL02] Stephen Se, David G. Lowe, and James J. Little. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *I. J. Robotic Res.*, 21(8):735–760, 2002. 19

[SLL05] S. Se, D.G. Lowe, and J.J. Little. Vision-based global localization and mapping for mobile robots. *Robotics, IEEE Transactions on*, 21(3):364–375, June 2005. 17

[SMD10a] H. Strasdat, J. M. M. Montiel, and A. Davison. Scale drift-aware large scale monocular slam. In *Proceedings of Robotics: Science and Systems*, Zaragoza, Spain, June 2010. 19

[SMD10b] H. Strasdat, J.M.M. Montiel, and A.J. Davison. Real-time monocular slam: Why filter? In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2657–2664, May 2010. 21

[SP11] N. Sunderhauf and P. Protzel. Brief-gist - closing the loop by simple means. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 1234–1241, 2011. 26

[SRDC09] Sinisa Segvic, Anthony Remazeilles, Albert Diosi, and François Chaumette. A mapping and localization framework for scalable appearance-based navigation. *Computer Vision and Image Understanding*, 113(2):172–187, 2009. 41

[SS08] D. Scaramuzza and R. Siegwart. Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles. *Robotics, IEEE Transactions on*, 24(5):1015–1026, Oct 2008. 19

[SSC87] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, volume 4, pages 850–850, Mar 1987. 17

[SSC90] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In Ingemar J. Cox and Gordon T. Wilfong, editors, *Autonomous robot vehicles*, chapter Estimating uncertain spatial relationships in robotics, pages 167–193. Springer-Verlag New York, Inc., New York, NY, USA, 1990. 22

[SSLT12] S. Sengupta, P. Sturgess, L. Ladicky, and P.H.S. Torr. Automatic dense visual semantic mapping from street-level imagery. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 857–862, Oct 2012. 39

[SZ03] Josef Sivic and Andrew Zisserman. Video google: Efficient visual search of videos. In *Toward Category-Level Object Recognition*, pages 127–144, 2003. 28, 30, 47

[TBF98] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. A probabilistic approach to concurrent mapping and localization for mobile robots. *Auton. Robots*, 5(3-4):253–271, July 1998. 17

[TBF05] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005. 18, 22

[TKC12] Stephen T Tully , George A Kantor, and Howie Choset. A unified bayesian framework for global localization and slam in hybrid metric/topological maps. *International Journal of Robotics Research (IJRR)*, 31(3):271–288, March 2012. 41

[TKCW09] S. Tully, G. Kantor, H. Choset, and F. Werner. A multi-hypothesis topological slam approach for loop closing on edge-ordered graphs. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 4943–4948, Oct 2009. 41

[TKH11] N. Tongprasit, A. Kawewong, and O. Hasegawa. Pirf-nav 2: Speeded-up online and incremental appearance-based slam in an indoor environment. In *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*, pages 145–152, Jan 2011. 27

[TM08] T. Tuytelaars and K. Mikolajczyk. *Local Invariant Feature Detectors: A Survey*. Now Publishers, Incorporated, 2008. 27

[TMHF00] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment – a modern synthesis. In *Vision Algorithms: Theory and Practice*, volume 1883 of *Lecture Notes in Computer Science*, pages 298–372, 2000. 20

[TMM+07] S. Tully, Hyungpil Moon, D. Morales, G. Kantor, and H. Choset. Hybrid localization using the hierarchical atlas. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 2857–2864, Oct 2007. 41

[TNNL02] Juan D. Tardós, José Neira, Paul M. Newman, and John J. Leonard. Robust mapping and localization in indoor environments using sonar data. *I. J. Robotic Res.*, 21(4):311–330, 2002. 17

[TS05] Adriana Tapus and Roland Siegwart. Incremental robot mapping with fingerprints of places. In *IROS*, pages 2429–2434, 2005. 26

[UKR+14] F. Uzer, H. Korrapati, E. Royer, Y. Mezouar, and S. Lee. Vision based hybrid map building for mobile robot navigation. In *Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS)*, 2014. 41

[UN00] I. Ulrich and I. Nourbakhsh. Appearance-based place recognition for topological localization. In *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, volume 2, pages 1023–1029 vol.2, 2000. 25

[VDL07] Christoffer Valgren, Tom Duckett, and Achim J. Lilienthal. Incremental spectral clustering and its application to topological mapping. In *ICRA*, pages 4283–4288, 2007. 27

[VGNS07] Shrihari Vasudevan, Stefan Gächter, Viet Nguyen, and Roland Siegwart. Cognitive maps for mobile robots-an object based approach. *Robot. Auton. Syst.*, 55(5):359–371, May 2007. 37

[VL10] Christoffer Valgren and Achim J. Lilienthal. Sift, surf and seasons: Appearance-based long-term localization in outdoor environments. *Robot. Auton. Syst.*, 58(2):149–156, February 2010. 27

[VLD06] Christoffer Valgren, Achim J. Lilienthal, and Tom Duckett. Incremental topological mapping using omnidirectional vision. In *IROS*, pages 3441–3447. IEEE, 2006. 27

[VMS+09] P. Viswanathan, D. Meger, T. Southey, J.J. Little, and A. Mackworth. Automated spatial-semantic modeling with applications to place labeling and informed search. In *Computer and Robot Vision, 2009. CRV '09. Canadian Conference on*, pages 284–291, May 2009. 39

[VS08] Shrihari Vasudevan and Roland Siegwart. Bayesian space conceptualization and place classification for semantic maps in mobile robotics. *Robot. Auton. Syst.*, 56(6):522–537, June 2008. 39

[WCZ05] Junqiu Wang, R. Cipolla, and Hongbin Zha. Vision-based global localization using a visual vocabulary. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 4230–4235, 2005. 30

[WL10] Min-Liang Wang and Huei-Yung Lin. A hull census transform for scene change detection and recognition towards topological map building. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 548–553, Oct 2010. 34

[WMS09] Felix Werner, Frederic Maire, and Joaquin Sitte. Topological slam using fast vision techniques. In Jong-Hwan Kim, ShuzhiSam

Ge, Prahlad Vadakkepat, Norbert Jesse, Abdullah Al Manum, Sadasivan Puthusserypady K, Ulrich Rückert, Joaquin Sitte, Ulf Witkowski, Ryohei Nakatsu, Thomas Braunl, Jacky Baltes, John Anderson, Ching-Chang Wong, Igor Verner, and David Ahlgren, editors, *Advances in Robotics*, volume 5744 of *Lecture Notes in Computer Science*, pages 187–196. Springer Berlin Heidelberg, 2009. 25

[WMZ07] Christian Weiss, Andreas Masselli, and Andreas Zell. Fast vision-based localization for outdoor robots using a combination of global image features. In *6th Symposium on Intelligent Autonomous Vehicles (IAV 2007)*, Toulouse, France, September 2007. 34

[WTMZ07] Christian Weiss, Hashem Tamimi, Andreas Masselli, and Andreas Zell. A hybrid approach for vision-based outdoor robot localization using global and local image features. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2007)*, pages 1047–1052, San Diego, CA, USA, 2007. 34

[WY12] Junqiu Wang and Y. Yagi. Robust location recognition based on efficient feature integration. In *Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on*, pages 97–101, Dec 2012. 34

[WZC06] Junqiu Wang, Hongbin Zha, and R. Cipolla. Efficient topological localization using orientation adjacency coherence histograms. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 2, pages 271–274, 2006. 25, 34

[YFW05] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Constructing free energy approximations and generalized belief propagation algorithms. *IEEE Trans. Inf. Theor.*, 51(7):2282–2312, 2005. 97

[ZBK05] Zoran Zivkovic, Bram Bakker, and Ben J. A. Kröse. Hierarchical map building using visual landmarks and geometric constraints. In *IROS*, pages 2480–2485, 2005. 40

[Zha00] Zhengyou Zhang. A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1330–1334, Nov 2000. 19

Resumé

Dans cette thèse, nous présentons une nouvelle méthode de cartographie visuelle hybride qui exploite des informations métriques, topolgiques et sémantiques. Notre but est de réduire le coût calculatoire par rapport à des techniques de cartographie purement métriques. Comparé à de la cartographie topologiques, nous voulons plus de précision ainsi que la possibilité d'utiliser la carte pour le guidage de robots. Cette méthode hybride de construction de carte comprend deux étapes. La première étape peut être vue comme une carte topo-métrique avec des nœuds correspondants à certaines régions de l'environnement. Ces cartes sont ensuite complétées avec des données métriques aux nœuds correspondant à des sous-séquences d'images acquises quand le robot revenait dans des zones préalablement visitées. La deuxième étape augmente ce modèle en ajoutant des informations sémantiques. Une classification est effectuée sur la base des informations métriques en utilisant des champs de Markov conditionnels (CRF) pour donner un label sémantique à la trajectoire locale du robot (la route dans notre cas) qui peut être "doit", "virage" ou "intersection". L'information métrique des secteurs de route en virage ou en intersection est conservée alors que la métrique des lignes droites est effacée de la carte finale. La fermeture de boucle n'est réalisée que dans les intersections ce qui accroît l'efficacité du calcul et la précision de la carte. En intégrant tous ces nouveaux algorithmes, cette méthode hybride est robuste et peut être étendue à des environnements de grande taille. Elle peut être utilisée pour la navigation d'un robot mobile ou d'un véhicule autonome en environnement urbain. Nous présentons des résultats expérimentaux obtenus sur des jeux de données publics acquis en milieu urbain pour démontrer l'efficacité de l'approche proposée.

MOTS CLES— Cartographie Visuelle Hybride, SLAM, Fermeture de Boucle.

Abstract

In this thesis, a novel vision based hybrid mapping framework which exploits metric, topological and semantic information is presented. We aim to obtain better computational efficiency than pure metrical mapping techniques, better accuracy as well as usability for robot guidance compared to the topological mapping.

A crucial step of any mapping system is the loop closure detection which is the ability of knowing if the robot is revisiting a previously mapped area. Therefore, we first propose a hierarchical loop closure detection framework which also constructs the global topological structure of our hybrid map. Using this loop closure detection module, a hybrid mapping framework is proposed in two step. The first step can be understood as a topo-metric map with nodes corresponding to certain regions in the environment. Each node in turn is made up of a set of images acquired in that region. These maps are further augmented with metric information at those nodes which correspond to image sub-sequences acquired while the robot is revisiting the previously mapped area. The second step augments this model by using road semantics. A Conditional Random Field based classification on the metric reconstruction is used to semantically label the local robot path (road in our case) as straight, curved or junctions. Metric information of regions with curved roads and junctions is retained while that of other regions is discarded in the final map. Loop closure is performed only on junctions thereby increasing the efficiency and also accuracy of the map. By incorporating all of these new algorithms, the hybrid framework presented can perform as a robust, scalable SLAM approach, or act as a main part of a navigation tool which could be used on a mobile robot or an autonomous car in outdoor urban environments. Experimental results obtained on public datasets acquired in challenging urban environments are provided to demonstrate our approach.

KEYWORDS— Hybrid Mapping, SLAM, Loop Closure.