ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
Instituto de Investigación Tecnológica (IIT)

PHD THESIS

INDOOR TOPOLOGICAL SLAM USING FRONTAL COMPUTER VISION

Author: Jaime Boal Martín-Larrauri
Supervisor: Prof. Dr. Álvaro Sánchez Miralles

MADRID
July 2014

This dissertation was typeset with LaTeX and compiled in TeXmaker using the MacTeX-2013 distribution. The font families used are Bitstream Charter, Utopia, Bookman, and Computer Modern. Unless otherwise noted, all figures were created by the author using Microsoft Visio®, Adobe Illustrator®, and MATLAB®.

CONSTANCIA REGISTRAL DEL TRIBUNAL DEL ACTO DE LA DEFENSA DE TESIS DOCTORAL

TÍTULO: Indoor Topological SLAM Using Frontal Computer Vision

AUTOR: Jaime Boal Martín-Larrauri

DIRECTOR: Prof. Dr. Álvaro Sánchez Miralles

TUTOR-PONENTE:

DEPARTAMENTO: Instituto de Investigación Tecnológica (IIT)

FACULTAD O ESCUELA: Escuela Técnica Superior de Ingeniería (ICAI)

Miembros del Tribunal Calificador:

PRESIDENTE: **Firma:**

VOCAL: **Firma:**

VOCAL: **Firma:**

VOCAL: **Firma:**

SECRETARIO: **Firma:**

Fecha de lectura:

Calificación:

To my parents

Acknowledgment

This dissertation represents the end of the road, the conclusion of an extremely important chapter of my life. I really look forward to discovering what awaits around the next corner. I may no longer be a student but, be assured, I will always remain an avid learner. During the four years I have spent working on this thesis I have had the opportunity of meeting and getting to know well many exceptional people who have accompanied me along the way and to whom I owe a great debt of gratitude.

First and foremost, I would like to thank Álvaro Sánchez Miralles, my supervisor, my colleague, my friend, for his great advice and support. Since we met seven years ago, we have gone through many things together and have created memories to last a lifetime. When I look back, I realize how much the young student who knocked your door for the first time has changed. With your guidance, I have become a better professional and, most importantly, a better person. No matter where life takes us, you will always be able to count on me for whatever you may need.

The Institute for Research in Technology (IIT) as institution and the group of talented people who make it possible deserve my deepest gratitude too. I cannot imagine anywhere better to develop a Ph.D. If I had to start over, I would definitely choose IIT. The working environment is simply unequalled and everyone is always willing to lend you a hand. Please, never forget what makes you different.

I would like to give special thanks to my fellow research assistants at the Intelligent Systems Area (ASI) with whom I have shared so many moments and who have been essential to make this endeavor so enjoyable. I cannot forget to mention my long-time friends Ismael and Alberto, with whom I embarked on this journey back in 2010. Nothing would have been the same without you two.

I am also grateful to all the people at the Autonomous Systems Laboratory (ASL) in ETH Zürich for having embraced me so warmly during my three-month stay, and to the friends I made at Maximilianeum for making me feel like at home.

During these four years I have also had the privilege of mentoring several students. I just hope that you learned from me as much as I did from and with you. In particular, I would like to mention Teresa, Carmen, Andrea, Daniel, Rodrigo, Francisco, and Pablo. I met you during those endless mockup building sessions, and now I have the honor of counting you as friends. Thank you for your support and for helping me forget about the thesis for a while.

Last but not least, I could not finish this acknowledgment without thanking my parents. You should be proud of yourselves, I made it this far because of you.

I am absolutely convinced that I have missed out many other people who would have deserved being cited. To all of you, wherever you are, thank you, merci, danke, gracias.

Contents

List of Figures

List of Tables

List of Algorithms

Acronyms

BRIEF	Binary Robust Independent Elementary Features
BRISK	Binary Robust Invariant Scalable Keypoints
CAT-SLAM	Continuous Appearance-based Trajectory SLAM
CenSurE	Center Surround Extremas
COLD	COsy Localization Database
CPE	Consistent Pose Estimation
DNA	Deoxyribonucleic acid
DP-FACT	Dirichlet Process Fast Adaptive Color Tags
EM	Expectation-Maximization
EVG	Extended Voronoi Graph
FAB-MAP	Fast Appearance-Based Mapping
FACT	Fast Adaptive Color Tags
FAST	Features from Accelerated Segment Test
FOV	Field Of View
FREAK	Fast Retina Keypoint
GNG	Growing Neural Gas
GTM	Graph Transformation Matching
GVG	Generalized Voronoi Graph
HCL	Hue Chroma Lightness
HMM	Hidden Markov Model
HSL	Hue Saturation Lightness
HSV	Hue Saturation Value
IDOL	Image Database for rObot Localization
IPJC	Incremental Posterior Joint Compatibility
JCBB	Joint Compatibility Branch and Bound
KLT	Kanade-Lucas-Tomasi
LPM	Local Perceptual Map
MAP	Maximum A Posteriori
MCMC	Markov-Chain Monte Carlo
MSER	Maximally Stable Extremal Regions
NLP	Natural Language Processing
NMS	Non-Maximum Suppression
ORB	Oriented FAST and Rotated BRIEF
PLISS	Place Labeling through Image Sequence Segmentation
POMDP	Partially Observable Markov Decision Process
PTM	Probabilistic Topological Map
RANSAC	RANdom SAmple Consensus
RBPF	Rao-Blackwellized Particle Filter

RGB	Red Green Blue
RGB-D	RGB Depth
SfM	Structure from Motion
SIFT	Scale Invariant Feature Transform
SLAM	Simultaneous Localization And Mapping
SSH	Spatial Semantic Hierarchy
SURF	Speeded-Up Robust Features
SUSurE	Speeded Up Surround Extrema
U-SURF	Upright SURF
VPACK	Visually Perceivable Adjacent Color histograms and Keypoints

Abstract

The last few years have seen a great leap forward towards autonomous mobile robots, and it is just a matter of time that they become a regular part of our lives. However, some fundamental problems need to be addressed before a robot can be assigned to any particular high-level application. One of these challenges is to provide the robot with the ability of localizing itself in a previously unvisited environment without having to supply it with a map of the area in advance. Therefore, when the robot is moved to a new area, it should incrementally build a map on its own and determine its position within it. This is known as the Simultaneous Localization And Mapping (SLAM) problem.

The map can be a very precise metric model or, alternatively, follow a topological approach that resembles human beings' more intuitive representation of space. We have an abstract notion of distance but are still able to determine where we are using vision to identify distinct places and the transitions among them. Hence, if we do not need to answer the question "Where am I?" with precision of millimeters and degrees, why should a robot? Moreover, in those applications in which a robot and human beings need to interact, the map should ideally be a common communication framework, so the more similar the map is to the way we structure spatial information, the easier interaction would be.

This thesis proposes a relatively computationally inexpensive solution to the SLAM problem inspired by human behavior. A forward-facing camera is the only sensor employed to make the system easily portable to a wide range of robotic platforms. By means of computer vision, the robot extracts a complementary collection of cues (vertical edges, color information, and keypoints) that focus both on the general characteristics of the scene and on the details, and employs a novel matching procedure that builds on concepts borrowed from the natural language processing field.

These features are then used to automatically identify qualitatively different locations that are susceptible of being considered a place in the map using an online topological segmentation algorithm based on the algebraic connectivity of graphs. Every time the robot arrives at a place, a Bayesian formulation is employed to decide if the robot is in a new or an already known location in order to update the topological map accordingly. As keeping track of all possibilities over time is computationally intractable, a particle filter is used to take only the most probable topologies into account.

Keywords: Computer vision · Feature detection and matching · Mobile robots · SLAM · Topological modeling of the environment

Resumen

En los últimos años se ha producido un gran avance hacia el desarrollo de robots móviles autónomos y es sólo cuestión de tiempo que se conviertan en un elemento habitual de nuestras vidas. Sin embargo, hay que abordar algunos problemas fundamentales antes de que un robot pueda desempeñar cualquier tarea de alto nivel. Uno de estos retos es el de dotar al robot con la habilidad de localizarse en un entorno previamente inexplorado, sin tener que proporcionarle el mapa de la zona con antelación. Así pues, cuando se coloque al robot en una nueva área, debería ser capaz de construir incrementalmente un mapa y determinar su posición dentro del mismo. A este problema se le denomina localización y mapeado simultáneos (SLAM, por sus siglas en inglés).

El mapa puede ser un modelo métrico muy preciso o, por el contrario, seguir una estrategia topológica que recuerda a la forma intuitiva en que los seres humanos representamos el espacio. Tenemos una noción abstracta de la distancia pero aun así somos capaces de determinar dónde estamos usando la visión para identificar lugares distintos y la relación que existe entre ellos. Por consiguiente, si no necesitamos responder a la pregunta "¿Dónde estoy?" con precisión de milímetros y grados, ¿por qué debería un robot? Más aún, en aquellas aplicaciones en las que el robot y los seres humanos tengan que interactuar, el mapa debería ser idealmente un medio de comunicación común, por lo que cuanto más se parezca a la forma en que estructuramos la información espacial, más fácil debería ser dicha interacción.

Esta tesis propone una solución al problema de la localización y mapeado simultáneos inspirado por el comportamiento humano. Se utiliza una cámara frontal como único sensor para hacer el sistema compatible con un amplio abanico de plataformas robóticas. Por medio de visión artificial, el robot extrae un conjunto de características complementarias (bordes verticales, información de color y puntos característicos) que se centran tanto en aspectos generales de la escena como en los detalles, y emplea un nuevo procedimiento para determinar la correspondencia entre características que se asienta en conceptos tomados del campo del procesamiento de lenguajes naturales.

Las citadas características se utilizan para identificar automáticamente lugares susceptibles de ser considerados una ubicacion en el mapa, mediante un algoritmo de segmentación basado en la conectividad algebraica de un grafo. Cada vez que el robot llega a un nuevo lugar, se emplea una formulación bayesiana para decidir si la ubicación es nueva o se encuentra en una ya conocida, con el fin de actualizar el mapa topológico en consecuencia. Como evaluar continuamente todas las posibles combinaciones es computacionalmente inviable, se usa un filtro de partículas para tener en cuenta sólo aquellas topologías más probables.

Palabras clave: Visión artificial · Detección y correspondencia de características · Robots móviles · SLAM · Modelado topológico del entorno

1 Introduction

Whatever you can do or dream, begin it.
Johann Wolfgang von Goethe
(1749–1832)

This first chapter introduces the rationale behind this thesis as well as its main objectives. In addition, it provides the reader with a general overview of the organization and the outline of the dissertation in order to make it easier to follow.

1.1. Motivation

The field of robotics has experienced a rapid growth over the past few decades following the desire to increase productivity and free human beings from dull, dangerous, or repetitive tasks. As a result, factories, assembly plants, and warehouses are currently full of industrial robots that manufacture almost every consumer product. However, in order to broaden their application to other areas and unleash their full potential, robots should be able to move freely through the environment. Nowadays, there are multiple examples of service robots that can autonomously perform tasks as different as cleaning the floor [Ulr+97], guiding tourists in a museum [Bur+99], behaving as a shop assistant [Gro+08], or even serving as a restaurant host [Bre+12]. What they all have in common is that they need to know where they are in space to behave autonomously and, to this end, they require a map of the environment, which can be either given by a human being or built by the robot itself.

The Simultaneous Localization And Mapping (SLAM) problem intends to provide a mobile robot with the ability of incrementally building a consistent map of a previously unvisited environment and, at the same time, localize itself within the map it is creating. This problem has received increasing attention lately, especially following a metric approach and associated to laser range scanners. However, there exist much less attempts of addressing this issue from a topological point of view (i.e., modeling the world as a collection of places connected based on their relative position) which enables to deal with the uncertainty more efficiently and results in more natural and compact representations that scale better with the size of the environment.

Moreover, according to the literature review carried out, which is included as a separate chapter (Chapter 2) due to its length, most of the latest publications on topological SLAM opt for computer vision as the primary sensory source because it has always been regarded as the ideal sensor technology. This can be easily explained by the fact that cameras combine a very powerful set of properties, like low weight and cost, wide availability, and the wealth of information they capture, to cite some of the most relevant. In addition, they resemble human beings' main perception system, which we employ to navigate in a mostly topological manner, enhanced with semantic information that allows us to interact with our surroundings. More specifically, the vast majority of the researchers choose omnidirectional cameras due to the fact that they provide rotational invariance and a 360º field of view from which a large amount of distinctive features can be extracted. The trade-off for this decision is that omnidirectional cameras have relatively complicated installation requirements making them more difficult to use in any type of robot, require an unwrapping step before doing any further processing, and are more expensive than their directional counterparts. Thus, this thesis proposes using forward-facing vision instead, which has been proven sufficient in natural life, as it is employed by a wide variety of different animal species, including human beings.

Environments can be roughly classified in three groups: structured and unstructured outdoor environments, and indoor environments—which are structured in most cases—, each of which have dissimilar characteristics. Currently, focus is being put on structured outdoor environments in the context of autonomous driving, which requires highly precise and accurate pose estimates, at least locally, to be able to safely handle unexpected traffic situations and avoid potential risks for the vehicle's occupants. Hence, metric approaches prevail. Nevertheless, topological SLAM does have an important role to play in this field, both in loop-closing, as part of a hybrid SLAM implementation, and in long-term path planning. By contrast, unstructured outdoor environments are challenging for topological approaches because many far apart locations tend to look alike. Note that even human beings often need metric information to orient themselves in the countryside. Finally, indoor environments are typically the more structured of them all and knowing the robot's global pose with high accuracy is generally not a strong requirement. Therefore, it is possibly the best suited kind of environment for topological techniques. In addition, at present a great amount of the commercial autonomous mobile robots are targeted at indoor applications (e.g., service or surveillance robots). These are the reasons why this thesis focuses on indoor environments.

1.2. Thesis objectives

This thesis aims at providing an alternative solution for the simultaneous localization and mapping problem indoors using a topological approach that is capable of dealing efficiently with uncertainty, and adopting forward-facing computer vision as the primary sensory source. It is born out of the desire to fill a series of gaps that have been identified in the literature:

- Metric solutions for the SLAM problem tend to be computationally demanding, owing to the fact that they require more precision, and usually need multi-sensor data fusion, whereas the topological paradigm is a more compact solution that scales better with the size of the environment and resembles human-intuitive map building techniques. These properties of topological maps are desirable for applications which require human-robot interacton, as human beings do not perceive the environment using coordinates in 3D space. Due to the absence of an outstanding, computationally inexpensive, topological

approach, this thesis concentrates on developing a *topological solution for the SLAM problem*.

- Most of the existing topological SLAM implementations, similarly to metric SLAM methods, rely on the fusion of multiple sensor technologies which are often chosen ad hoc for a particular robotic architecture (e.g., wheeled or legged robots). As a consequence, the developed methods are not easily portable to other entities. The election of a *forward-facing camera* as the only source of sensory information, enables the system to resemble human navigation strategies and it should therefore be applicable to most autonomous mobile robots. Monocular vision is employed throughout this thesis, but the algorithms presented are amenable to stereo cameras too, which enable to perform obstacle avoidance without the need for additional range sensors.
- For the system to be practical, it has to *perform in "real-time"*. The term real-time is in quotes to indicate that the robot should be capable of moving at a reasonable pace through the environment but that high speed is not a requirement. In the context of robotic image processing, the term real-time means that the processing time of each image should be lower than or equal to the image capture time. However, the frame rate and the robot's speed can be adjusted so the number of images and the place in which they are taken are almost the same, and there is more time between consecutive captures.

1.3. Dissertation outline

The dissertation consists of six chapters including this first introductory one. To begin with, in order to help putting the thesis in context, Chapter 2 provides an extensive review that covers most of the approaches to topological SLAM that can be found in the literature, but without going into unnecessary details. Algorithms and techniques are discussed with more depth as they become related to the ones proposed in the thesis.

Starting from Chapter 3, three chapters deal with the modules required to implement a full topological SLAM system. Given that it is an incremental process, each chapter builds up on the previous and, consequently, the results of Chapter 5 encompass the developments of the whole thesis. For this reason, a separate chapter focused on the results is not included. Notwithstanding, this does not mean that developments of each chapter are indivisible, as each module can be implemented independently using input data from algorithms different from the ones proposed in this thesis.

These three chapters (3, 4, and 5) are all set out in the same manner with the intention of resembling the form of the most common and familiar scientific communication format, a technical paper. Owing to this articulation of the information, the dissertation can be read from beginning to end or, alternatively, each chapter can be consulted individually, as they are self-contained. These core chapters comply with the following structure:

- First, after a brief introduction to situate the chapter in the context of the thesis, the related works are put forward and explained in more detail than in the initial literature review, highlighting the aspects that are key to the comprehension of the chapter developments.
- Second, the actual proposed algorithm is explained and justified. A theoretical background is provided, if necessary, along with practical implementation considerations.
- Subsequently, the results of the different experiments carried out to validate the methodology proposed in real indoor environments are set forth and discussed.

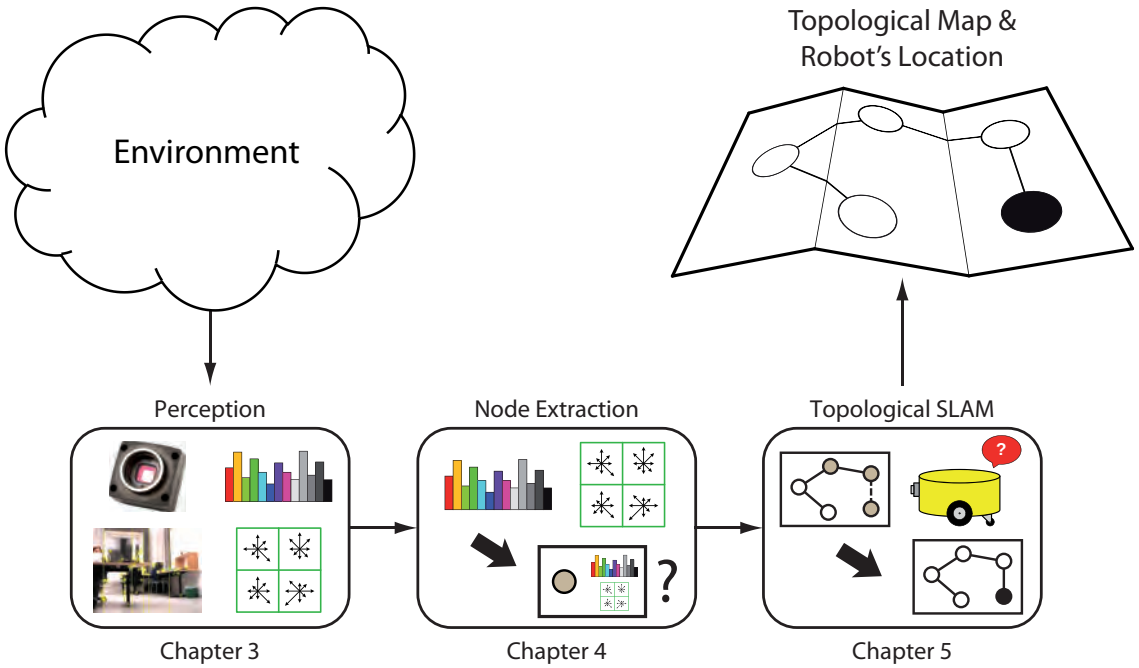


Figure 1.1. Modules in which the dissertation is divided. Chapter 3 concentrates on the extraction and description of lightweight and distinctive features from the environment that can be used for topological place identification. These features are used to obtain topological locations or nodes in Chapter 4. The last step involves updating the topological map and the robot's location by probabilistically deciding whether the robot has arrived at a new, previously unexplored, place or has reached an already visited node.

- Finally, the main conclusions that can be drawn from the results and the contributions made are summarized.

Figure 1.1 shows the modules that are treated in each of the three core chapters. In order to be able to perform any kind of task, a robot must sense and process information from its surroundings in the first place. Chapter 3 presents a collection of complementary features extracted from monocular images, coined VPACK (Visually Perceivable Adjacent Color histograms and Keypoints), that comprises vertical edges, color histograms, and a few fast-to-compute keypoint features, together with a matching procedure inspired by the field of natural language processing (NLP). In order to check if VPACK can be used for place recognition, the distinctiveness of this combination of features is validated by solving a localization problem without any additional information (e.g., odometry). The idea behind VPACK is to extract as much information as possible from images that cover a limited field of view within reasonable computing times. It is designed to bring together local features (keypoints) with semi-global information (color histograms) that can compensate the drawbacks of each other, and provide them, by means of the vertical lines extracted, with a qualitative ordering without having to perform complex geometric calculations. Note that the fact that vertical lines are used has to be taken into account when implementing the proposed algorithm in different robotic architectures (wheeled, legged, flying) because there are some contraints that the camera has to fulfill (i.e., the camera should have no roll and its focal plane should be parallel to the planes containing the vertical edges). As the method is intended for indoor environments, this is almost always true for wheeled robots, and can be achieved for the other types of platforms if they have an appropriate control system, as indoors perturbations, like wind, are usually not very significant.

Once the robot is able to tell apart different locations in the environment, the following step is to automatically determine, from the input visual stream, the underlying topological places that will constitute the nodes of the topological map. Chapter 4 introduces a clustering method based on the algebraic connectivity of graphs to obtain topological locations online that can be also used with descriptors of variable length (which is the case of VPACK), contrary to statistical techniques that require fixed length descriptors, as its only input are non-negative similarity measurements. Thus, it is an alternative to manually defining thresholds on similarities to determine when the robot has left and area and entered another.

Finally, every time a new node is identified, the robot has to update the topological map it is building by inserting its location within this map. The most common situation is that the robot is unsure about whether the node it currently is in corresponds to an already visited place or, by contrast, is a completely new location. Solving this uncertainty is precisely the goal of topological SLAM. Chapter 5 proposes a passive approach to topological SLAM (i.e., it is assumed that navigation commands cannot be issued and, thus, the robot's behavior cannot be controlled) that uses visual information as the single input. As a result, the method is rather independent of the robotic platform it is implemented in (e.g., legged, wheeled, flying). A Bayesian framework is employed to integrate appearance (VPACK) and adjacency information. By means of a particle filter, several possible topologies are tracked until the correct one can be determined.

The last chapter summarizes the main conclusions and contributions of the thesis, including the publication record. In addition, future research directions that seem worth exploring to improve the results obtained, deal with open issues, or broaden the application scope of the solutions presented, are commented on.

Throughout the dissertation, three different image datasets are used to carry out experiments: two of them belong to publicly available databases (KTH-IDOL2 and COLD) and correspond to office environments; the third one involves a home environment and was specifically recorded for this thesis because a similar dataset that met the requirements of VPACK (see Section 3.2) could not be found. More details on the datasets are provided in Appendix A.

In order to test VPACK, the KTH-IDOL2 and the home environment datasets are employed because they are qualitatively different. In an office environment the structure is generally repetitive, and the furniture and the predominant colors tend to be similar everywhere, while rooms in a house usually show more diversity. By contrast, in Chapters 4 and 5 only the KTH-IDOL2 and the COLD dataset captured in Saarbrücken are used. The two main reasons are that they both have additional information like a ground-truth that is used to illustrate the performance of the algorithms presented and, more importantly, that as they are accessible on the Internet, anyone can compare own results with the ones obtained in this work. The fact that the Saarbrücken dataset is not tested in Chapter 3 is not an issue, because VPACK is used as image descriptor in all of the core chapters.

2

Literature Review

A capacity and taste for reading
gives access to whatever has already
been discovered by others.
Abraham Lincoln (1809–1865)

This chapter sets the scene for the thesis by providing an introductory overview to the field of topological simultaneous localization and mapping. The information is organized from the general to the more specific, starting with the definition of the main terms and concepts, and gradually progressing to the particular techniques that have been applied to topological SLAM in the literature in terms of feature detection, node extraction, map matching, and map fusion. The content of this chapter is an updated version of the journal article [Boa+14b].

2.1. Introduction

Mobile robotics' ultimate aim is to develop fully autonomous entities capable of performing rather complicated tasks, without the need for human intervention, during extended periods of time. Over the past three decades, this objective has constantly faced harsh difficulties which have hindered progress. The most recurrent issues in the literature, which are yet to be completely resolved, are stated below.

A mobile robot must be able to navigate through the environment in order to achieve its goals. According to Leonard and Durrant-Whyte [LDW91a], this general problem can be summarized in three questions: "Where am I?," "Where am I going?," and "How should I get there?" The first question addresses the *localization problem*, which intends to estimate the robot's pose (i.e., location and orientation) using data gathered by distinct sensors and knowledge of previous locations. However, the presence of noisy sensor measurements makes this problem harder than it may seem at first sight. The precision with which this problem is solved decisively affects the answer to the other two questions, as it is necessary to localize oneself in the environment to safely interact with it, decide what the following step should be, and how to accomplish it.

During the localization process, a robot must resort to some kind of reference system, in other words, it requires a map. The extensive research survey carried out by Thrun [Thr02] collects the main open issues concerning *robotic mapping*, which are succinctly presented henceforth. Currently, there are robust methods for mapping structured, static, and bounded environments, whereas mapping unstructured, dynamic, or large-scale unknown environments remains largely an unsolved problem.

According to Thrun [Thr02], the robotic mapping problem is "that of acquiring a spatial model of a robot's environment." To this end, robots must be equipped with sensors that enable them to perceive the outside world. Once again, sensor errors and range limitations pose a great difficulty.

The first challenge in robotic mapping develops from the measurement noise. Usually, this issue can be overcome if the noise is statistically independent, as it can be canceled out performing enough measurements. Unfortunately, this does not always occur in robotic mapping because, whenever incremental sensors (e.g., encoders) are used, errors in navigation control accumulate progressively and condition the way in which subsequent measurements are interpreted. As a result, if a robot cannot rely on the layout of the environment whatever it infers about its surroundings is plagued by systematic, correlated errors. Leonard and Durrant-Whyte [LDW91b] state the *correlation problem* as follows:

> If a mobile robot uses an observation of an imprecisely known target to update its position, the resulting vehicle position estimate becomes correlated with the feature location estimate. Likewise, correlations are introduced if an observation taken from an imprecisely known position is used to update the location estimate of a feature in the map.

The second difficulty of the robot mapping problem derives from the amount and complexity of the features required to describe the objects that are being mapped, because the computational burden grows exponentially as the map becomes more detailed. Obviously, it is absolutely different to restrict to the description of corridors, intersections, and doors, than to build a 3D visual map.

A third, and perhaps the hardest, issue is the *correspondence problem*, which attempts to determine if sensor measurements taken at different times correspond to the same physical entity. A specific instance of this problem occurs when returning to an already visited area, because the robot has to realize that it has arrived at a previously mapped location. This is known as the *loop-closing problem*. Another particular case is the so-called *first location problem* or *kidnapped robot problem* [Koe+06], which occurs when a robot is placed in an unknown position of an environment for which it has a map.

Fourth, the vast majority of environments are dynamic. Doh *et al.* [Doh+09] further classify the concept of dynamic environments in *temporary* dynamics, which are instantaneous changes that can be discarded by consecutive sensor measurements (e.g., moving objects like walking people), and *semi-permanent* dynamics or *scene variability* [KB02], which are changes that persist for a prolonged period of time. This second type of dynamics makes the correspondence problem even more difficult to solve, as it provides another manner in which apparently inconsistent sensor measurements can be interpreted. Suppose a robot perceives a closed door that was previously modeled as open. This observation may be explained by two equally plausible hypotheses: either the door position has changed, or the robot is in error about its current location. At present, there are almost no mapping algorithms capable of coping with

this difficulty. On the contrary, most approaches assume a static world and, as a consequence of this simplification, anything that moves apart from the robot is regarded as noise. In fact, the majority of the experimental tests in the literature are carried out in rather controlled environments and never mention how to deal with these troublesome dynamics. Doh *et al.* [Doh+09] are an exception to this trend due to the fact they take door position changes into consideration.

Finally, robots must navigate through the environment while mapping on account of sensor range limitations. The operation of generating navigation commands with the aim of building a map is known as *robotic exploration*. Although the commands issued during the exploration of the environment provide relevant information about the locations at which different sensor measurements were obtained, motion is also subject to errors (e.g., wheel slippage). Therefore, these controls alone are insufficient to determine a robot's pose.

2.2. The SLAM problem

As mentioned by Thrun [Thr02], the localization and mapping problems are often tackled together in the literature. Essentially, both problems are uncertain and, when trying to solve them individually, the other introduces systematic error. By contrast, estimating both at the same time makes the measurement and control noises independent. Notice, nevertheless, that robot mapping is like the *chicken and egg problem*: "A robot needs to know its position to build a map, and it requires a map in order to determine its position [Yam+98]."

The immediate question inferred from this idea is if it is possible for a mobile robot to be placed at an unknown location in an unknown environment and, despite this, incrementally build a consistent map of the environment using local information while simultaneously determining its location within this map. This is known as the *simultaneous localization and mapping (SLAM) problem* [DWB06; BDW06]. During more than a decade, a solution to this issue has been regarded as a key milestone in the pursuit for truly autonomous robots. At present, it can be safely asserted that the SLAM problem has been solved in different manners, at least, from a theoretical point of view. Notwithstanding, substantial issues remain open concerning the implementation of these SLAM solutions.

The majority of the problems that researchers are currently facing are those of computational nature [BDW06]. In order to overcome the correspondence problem, each location in the environment must be unequivocally distinguishable from all the rest. This implies gathering either plenty of similar features or a more restricted number with richer information in every place analyzed. In any case, the computational burden rapidly increases to intractable levels in large environments. Therefore, most approaches make a trade-off between computation time and precision or global distinctiveness, that is, they either limit the number of locations considered or reduce the number of features analyzed in each place.

2.3. Types of maps

So far, mapping has been referred to as a whole. However, there exist several types of maps which require diverse data acquisition techniques and present different associated problems. In general, maps can be divided into the four groups listed below:

- *Metric maps* represent the environment as a set of object or obstacle coordinates with the aid of raw data and geometric features (e.g., lines, edges). Although localization

and mapping with this approach can be very accurate and result in very high precision representations of the environment, the required data volume grows at a much higher rate than the size of the region being mapped and, therefore, involves complicated calculations [LDW91b; FM03].

- Conversely, *topological maps* model the environment as a graph. They are based on the discretization of the continuous world into a finite set of places or locations (nodes)[1] connected according to their relative position in the environment. These maps provide a compact representation, since only distinctive places within the environment are encoded. Consequently, they are much less computationally demanding, as there is no need for a precise localization, and navigation commands follow naturally from the graph. Nevertheless, the main problem of this method is *perceptual aliasing*, in other words, that there is always a risk that two distinct locations appear identical to the robot's sensors [FM03; CN01; KW94].

- *Hybrid maps* are a combination of the previous two which intend to compensate the drawbacks of both approaches when applied alone. On the one hand, reduce the computational burden of metric maps and, on the other hand, increase topological distinctiveness. To this end, they use a global topological map to move between places, and rely on a metric representation in bounded local spaces for precise navigation [Bla+08; Nie+04; Ziv+05]. It is important to bear in mind that these maps are often referred to in the literature as *hierarchical*. However, this term should be avoided, as it can be easily confused with topological graph representations which involve several abstraction processes (i.e., create an *atlas* with progressively detailed sub-maps) [Lis+03].

- Finally, *semantic maps* contain, in addition to spatial information about the environment, assignments of the mapped features to entities of known classes. This means that they hold data on objects, functionalities, events, or relations in the robot's environment whose knowledge permits a high-level goal-directed behavior, enables reasoning, and helps to resolve location ambiguities [NH08].

According to the previous definitions, maps can be sorted in increasing level of abstraction in metric, hybrid, topological, and semantic (Figure 2.1).

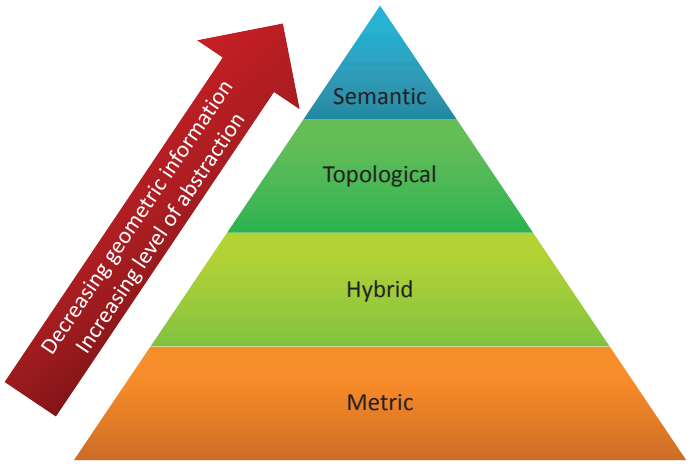


Figure 2.1. Level of abstraction hierarchy for maps.

[1]The terms *place*, *location*, and *node*, are treated as synonyms throughout the thesis. They refer to a position in space, either a single point or a region, that is easily distinguishable based on one or several characteristics that can be perceived using the sensors available in the robot. They can be represented by one or more sensory inputs.

2.4. Why choose a topological approach?

In principle, two classical opposite approaches exist to address the SLAM problem. The first one models the environment using a metric map, enabling an accurate estimation of the robot's position. It provides a dense representation of the environment, which has large storage requirements, and is particularly well suited to precise trajectory planning.

In the second approach, the environment is segmented into distinctive places using a topological map, which relies on a higher level of representation than metric mapping, making symbolic goal-directed planning and navigation possible. It also provides a more compact representation that is more in accordance with the size of the environment [Ang+08b] in spite of requiring more complex sensory information which often implies more processing. The largely cited papers by Kuipers and Levitt [KL88] and Kuipers and Byun [KB91] can be regarded as the seminal work which triggered a paradigm shift from a metric to a topological approach in robotic map building. Contrary to previous developments, which extracted the geometry of the environment from sensor measurements and then inferred the topology from it (see the work by Chatila and Laumond [CL85], for instance), they proposed constructing a topological description based on simple control strategies in the first place, and incorporating local metric information in each of the identified nodes afterwards.

Albeit, considering that metric maps are more accurate and that a hybrid approach helps to overcome storage problems, why should purely topological maps be used? To begin with, topological navigation is a behavior employed by a variety of different animal species, including human beings. We do not need to answer the question "Where am I?" in millimeters and degrees in order to safely move through the environment [Bro90]. On the contrary, rather than navigating using coordinates, we have an abstract notion of distance but are still able to recognize where we are in space [Ram+05]. Moreover, Brooks [Bro85] supports the belief that topological maps are a means of coping with uncertainty in mobile robot navigation. The absence of metric and geometric information, which is replaced by notions of proximity and order, eliminates dead-reckoning error issues, which no longer accumulate. As high precision is not a requirement, the topological approach is more robust than the metric because a consistent map can still be built even if the sensors' measurement uncertainty is large. By contrast, this situation is much more troublesome in the context of metric mapping.

In conclusion, topological representation resembles human intuitive navigation system, which has been proven to deal efficiently with uncertainty, and results in a straightforward map from which path planning follows naturally.

2.5. Topological SLAM

2.5.1. Breaking up the problem

Several stages are required to implement a topological SLAM algorithm. First of all, the sensor technologies that are going to be used to perceive landmarks in the environment have to be chosen. Once this decision is made, the following step is to determine which feature detection and description algorithms are going to be applied. At this point, two different approaches exist: either treat each sampling step as a different location or, alternatively, attempt to reduce the sequence of observations into a set of meaningful topological nodes. This block is enclosed by a discontinuous line in Figure 2.2 to indicate that it is optional.

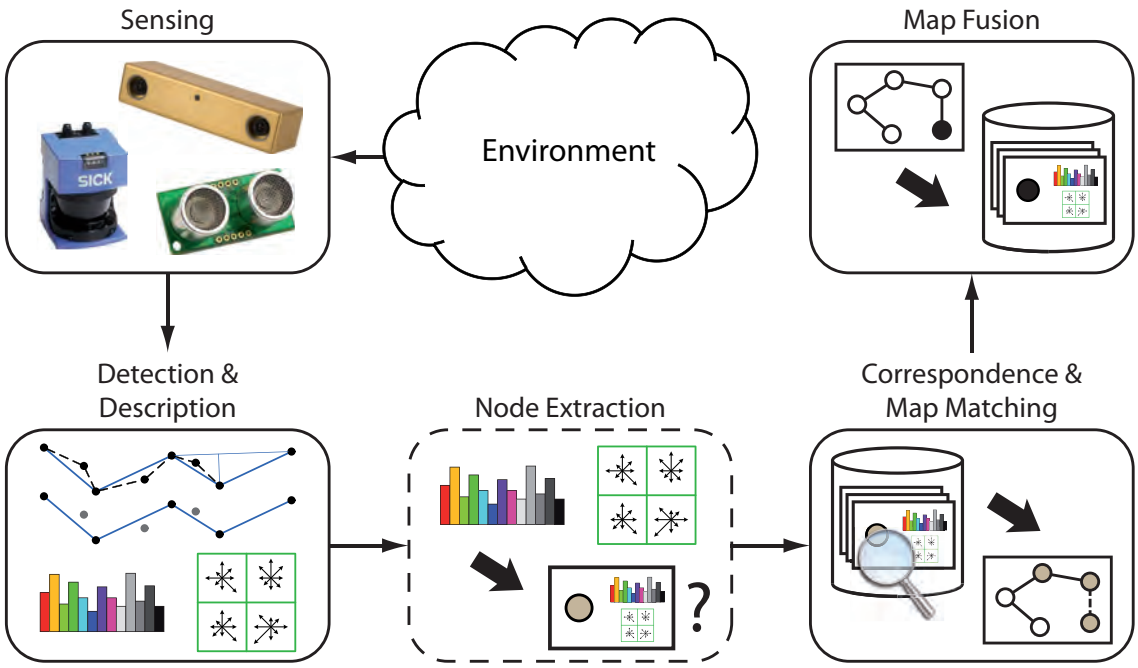


Figure 2.2. Topological SLAM overview. The robot acquires sensory information from one or several sources; selected features are detected and encoded; optionally, topological nodes are extracted; the current location is then compared with a database of previously visited places resulting in a belief state (i.e., the robot could be in several locations with different probabilities); finally, once the uncertainty has been resolved, either a new node is added to the database or the information of an existing one is updated.

In any of the cases, every time whatever is considered a node is extracted, the features gathered in that location must be compared with the stored nodes. Due to the fact that it is almost impossible to extract exactly the same features when revisiting a place, and that several locations may look alike, the most common situation is that the robot is uncertain about its position after performing this comparison. This is depicted in Figure 2.2 with gray nodes. The robot may either be in various known positions or, alternatively, have reached a new node (illustrated by a discontinuous line).

Consequently, the robot is forced to keep record of the probability of being in each node until the uncertainty is somehow resolved. At this point, both the robot's location and the map become simultaneously unambiguous. Should it happen to be no match, the system must label the current location as a new place. Otherwise, data should be added to the current node definition in order to update its appearance and enhance its distinctiveness for future revisiting.

2.5.2. Sensing

A mobile robot must start by sensing its surroundings to acquire information that allows it to determine its position before undertaking any other task. Table 2.1 collects the different sensor technologies that have been applied to extract topological data from the environment over the past two decades. Papers that build on earlier work are grouped together. As stated by Ranganathan and Dellaert [RD08], laser range scanners are currently *de facto* standard in robotics, due to their ability to provide precise depth estimates and form dense point clouds which resemble the scene structure, although substantial research is being carried out on computer vision due to recent progress in image processing, and because cameras are typically

Table 2.1. Sensors used in the literature to implement topological mapping systems

	Range		Odometry		Cameras			
Reference	Sonar	Laser	Encoder	Compass	Monocular	Omnidirectional	Stereo	RGB-D
Kuipers & Byun [KB91]	✓			✓				
Kortenkamp & Weymouth [KW94]	✓				✓			
Owen & Nehmzow [ON98]	✓			✓				
Gutmann & Konolige [GK99]		✓	✓					
Hafner [Haf00]				✓		✓		
Ulrich & Nourbakhsh [UN00]						✓		
Choset & Nagatani [CN01]	✓	✓	✓					
Tomatis *et al.* [Tom+02]		✓	✓					
Gross *et al.* [Gro+03], Koenig *et al.* [Koe+08]			✓			✓		
Anguelov *et al.* [Ang+04]		✓				✓		
Kuipers *et al.* [Kui+04]	✓	✓						
Modayil *et al.* [Mod+04]		✓	✓					
Andreasson *et al.* [And+05]			✓			✓		
Goedemé *et al.* [Goe+05; Goe+07; GG08]						✓		
Stachniss *et al.* [Sta+05]		✓				✓		
Tapus & Siegwart [TS05; Tap05]		✓	✓			✓		
Zivkovic *et al.* [Ziv+05]						✓		
Fraundorfer *et al.* [Fra+07]					✓			
Vasudevan *et al.* [Vas+07]		✓	✓				✓	
Angeli *et al.* [Ang+08a; Ang+08b; Ang+08c]					✓			
Cummins & Newman [CN08; CN11]					✓			
Nüchter & Hertzberg [NH08]		✓						
Ranganathan & Dellaert [RD08; RD11]		✓	✓			✓[a]		
Sabatta *et al.* [Sab08; Sab+10]					✓			
Doh *et al.* [Doh+09]		✓	✓					
M. Liu *et al.* [Liu+09; LS12; LS14]						✓		
Tully *et al.* [Tul+09]	✓		✓					
Werner *et al.* [Wer+09a; Wer+09b; Wer+12]	✓					✓		
Chang *et al.* [Cha+10]					✓			
Lui & Jarvis [LJ10]					✓	✓	✓	
Romero & Cazorla [RC10; RC12]						✓		
Maddern *et al.* [Mad+11; Mad+12a]			✓			✓		
Cadena *et al.* [Cad+12]							✓	
Fernández-Moral *et al.* [FM+13]								✓
Labbé & Michaud [LM13]					✓			
Y. Liu & Zhang [LZ13]						✓[a]		
TOTAL	7	12	12	3	8	16	3	1

[a] Multi-camera rig

less expensive and provide more distinctive features, which is fundamental for topological SLAM [Pir+03; LJ10].

The use of visual data as the primary source of information in SLAM systems has not had time to converge to generally efficient and robust solutions yet, hence leaving much room for experimentation and improvement. Notwithstanding, albeit sensing the world through a camera lens can be less accurate than laser range sensing, the richness of the information encoded has already proved to be sufficient to obtain reliable estimates of camera motion and scene structure. However, it is important to point out that the vast majority of the articles reviewed in Table 2.1 that use computer vision opt for omnidirectional cameras (Figure 2.3). This can be easily explained by the fact that omnidirectional cameras are the only ones that guarantee *rotational invariance* (i.e., no matter the heading direction a robot has in a given location, the image captured is always the same) and cover a 360º field of view that enables to extract a large amount of informative features.

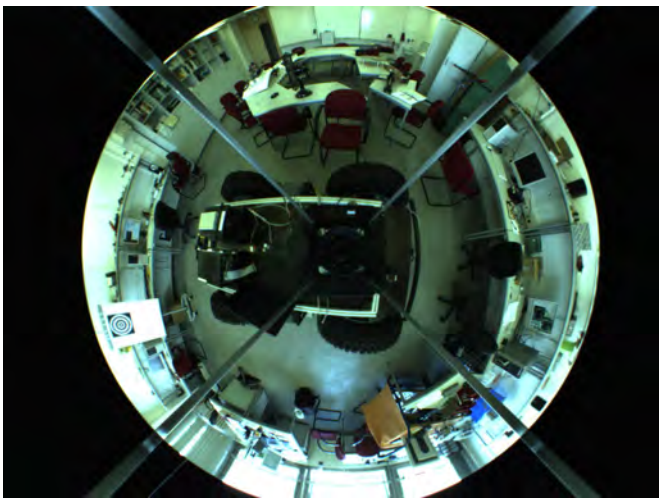


Figure 2.3. Example of an image captured with an omnidirectional camera (© User:Sgeureka / Wikimedia Commons / CC-BY-SA-3.0 / GFDL).

Finally, it is worth noting that even RGB-D sensors are still not widely used, they are bound to become popular in the years to come because they provide reasonably accurate depth estimates, in addition to color images, for a much lower price than the cheapest of laser range scanners.

2.5.3. Detection and description

This section concentrates on the detection methods found in the literature to extract landmarks from sensor readings. According to Stankiewicz and Kalia [SK07], the use of landmarks implicitly assumes three properties: persistence, saliency, and informativeness. To begin with, a landmark should be *persistent*, that is, the features should still be present when the robot returns to the location anytime in the future. Furthermore, it ought to be *perceptually salient*, which means that the landmark must be easily detectable and identifiable. Finally, a landmark needs to be *informative*. In other words, it should provide evidence about the robot's pose or the action it should take when observing it.

Following with the reasoning by Stankiewicz and Kalia [SK07], there exist two different types of landmarks: *structural* and *object* landmarks. The former are defined as geometric features that can serve as cues, like intersections, entrances —named *gateways* by Kortenkamp and

Weymouth [KW94]—, or corners [Tom+02; Tap05]. The latter are objects in the environment which are independent of its structure, such as signs. These are often identified using computer vision by means of interest points or regions. From these definitions, it is intuitively obvious that object landmarks typically provide more information regarding spatial coordinates, as two intersections look alike but a poster on a wall is probably unique. Unfortunately, it is more than somewhat unlikely to find a single type of cue that combines all of the previous properties.

Table 2.2 shows the numerous feature extraction methods employed in the references presented in Table 2.1, which are briefly introduced below. For those techniques that are common knowledge in the field, only references to surveys or seminal papers are put forward. Emphasis is put on the less generalized techniques.

2.5.3.1. Geometric features and gateways

At early stages, due to the fact that the only widespread sensor technology was sonar, feature detection reduced to what has been called geometric features in Table 2.2 (i.e., distances to different obstacles which allow to identify simple topological landmarks such as corners or dead ends) and *gateways*, which are an extension of the previous to detect openings. With the rise of laser range scanners, these approaches became more precise owing to the acquisition of dense point clouds. A list of some geometric features that can be extracted from laser readings (e.g., the average distance between two consecutive beams or the perimeter of the area covered by a scan) is provided by Martínez-Mozos *et al.* [MM+05].

2.5.3.2. Lines and planes

Human-made environments are full of vertical and horizontal lines and, therefore, constitute an invaluable source of topological information. Line and plane extraction techniques are usually employed in conjunction with laser range scanners. There exist many approaches for line extraction, some of which are compared by Nguyen *et al.* [Ngu+05]. As far as topological feature detection is concerned, the Douglas-Peucker algorithm [Tap05] (also known as *split-and-merge*), EM (Expectation-Maximization) applied to line fitting [Pfi+03], the Hough transform [FP03], and RANSAC (RANdom SAmple Consensus) [FB81] have been employed. The latter is a general algorithm for model adjustment in the presence of many data outliers which has further applications; for instance, Nüchter and Hertzberg [NH08] adopt this technique for plane extraction. Fernández-Moral *et al.* [FM+13] use a region growing technique to obtain planar patches instead [HB13]. These planes are then described by their normal vector, centroid, area...

2.5.3.3. Color and intensity histograms

With the introduction of computer vision techniques, simple methods like color or intensity histograms were initially applied. Hafner [Haf00] employed the intensities of vertically averaged and smoothed panoramas, whereas Ulrich and Nourbakhsh [UN00] extracted histograms in the RGB and HSL color spaces from omnidirectional images. However, it was soon widely accepted that the information obtained from histograms was not sufficiently distinctive and reliable—they can be potentially identical for two images with different content, and are very sensitive to illumination changes—to use them as a sole characteristic detector. Thus, this approach has now become a part of, or a complement for, other more consistent and informative methods. For instance, Tapus and Siegwart [TS05] and Liu and Siegwart [LS14] use color patches as part of a collection of features (see Section 2.5.3.10). Still, Werner *et al.* [Wer+09a] have

Table 2.2. Feature extraction techniques for topological navigation grouped according to sensor technologies: range sensors (i.e., sonar and laser) and cameras. Visual features are further classified into scene-based, edges, attention-based, keypoints, and affine covariant regions.

	Range sensors				Cameras									
					Scene			Edges		Att.	Kpts.		Regs.	
Reference	Geometric features	Gateways	Lines	Planes	Color/Intensity histograms	Haar-like features	Gist	Vertical edges	Invariant column segments	Saliency	SIFT	SURF	Harris-affine + SIFT desc.	MSER + SIFT descriptor
Kuipers & Byun [KB91]	✓													
Kortenkamp & Weymouth [KW94]		✓						✓						
Owen & Nehmzow [ON98]	✓													
Gutmann & Konolige [GK99]	✓													
Hafner [Haf00]					✓									
Ulrich & Nourbakhsh [UN00]					✓									
Choset & Nagatani [CN01]	✓													
Tomatis *et al.* [Tom+02]	✓	✓												
Gross *et al.* [Gro+03], Koenig *et al.* [Koe+08]					✓									
Anguelov *et al.* [Ang+04]		✓	✓		✓									
Kuipers *et al.* [Kui+04]		✓												
Modayil *et al.* [Mod+04]	✓													
Andreasson *et al.* [And+05]											✓			
Goedemé *et al.* [Goe+05; Goe+07]									✓		✓			
Stachniss *et al.* [Sta+05]	✓					✓								
Tapus & Siegwart [TS05; Tap05]	✓		✓		✓			✓						
Zivkovic *et al.* [Ziv+05]											✓			
Fraundorfer *et al.* [Fra+07]														✓
Vasudevan *et al.* [Vas+07]		✓	✓								✓			
Angeli *et al.* [Ang+08a; Ang+08b; Ang+08c]											✓			
Cummins & Newman [CN08; CN11]												✓		
Nüchter & Hertzberg [NH08]		✓		✓		✓								
Ranganathan & Dellaert [RD08; RD11]	✓										✓		✓	✓
Sabatta *et al.* [Sab08; Sab+10]											✓			
Doh *et al.* [Doh+09]	✓	✓	✓											
M. Liu *et al.* [Liu+09; LS12; LS14]					✓			✓						
Tully *et al.* [Tul+09]	✓													
Werner *et al.* [Wer+09a; Wer+09b; Wer+12]	✓				✓									
Chang *et al.* [Cha+10]							✓			✓				
Lui & Jarvis [LJ10]						✓						✓		
Romero & Cazorla [RC10; RC12]														✓
Maddern *et al.* [Mad+11; Mad+12a]												✓		
Cadena *et al.* [Cad+12]	✓											✓		
Fernández-Moral *et al.* [FM+13]	✓			✓										
Labbé & Michaud [LM13]												✓		
Y. Liu & Zhang [LZ13]							✓							
TOTAL	14	7	4	2	7	3	2	3	1	1	7	5	1	3

recently employed color histograms as the only feature in a low-demanding topological SLAM system.

2.5.3.4. Haar-like features

Haar-like features are inspired by Haar wavelets and were initially developed by Viola and Jones [VJ01] for object detection. Similarly to the Fourier transform, which is used to decompose complex signals into a series of sine waves, Haar wavelets are applied to obtain a summation of simpler images which can be used to extract a discriminative and robust to occlusions and light changes signature, although rotation variant.

Haar-like features compute the intensity difference between adjacent rectangles arranged in diverse configurations (Figure 2.4). Each Haar feature is used to train a weak classifier with AdaBoost [FS97]. These are then combined to produce a strong classifier capable of detecting different objects.

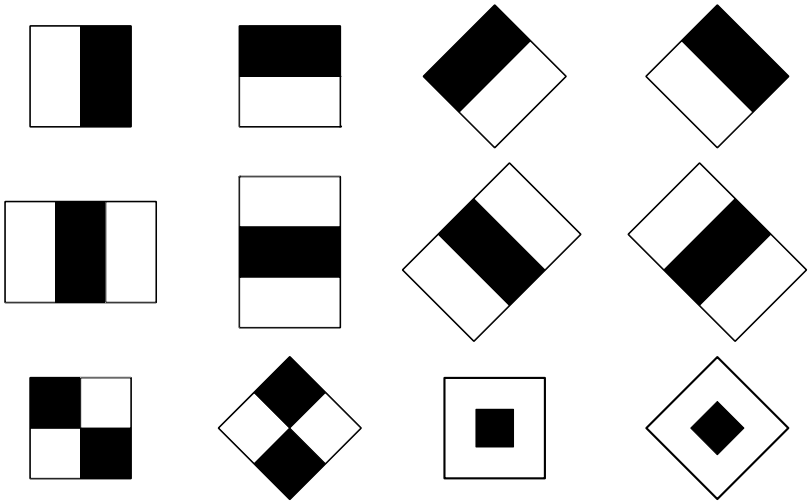


Figure 2.4. Sample Haar features. Intensities in the white areas are treated as positive; the black areas count as negative.

Both Stachniss *et al.* [Sta+05] and Nüchter and Hertzberg [NH08] use an extended set of Haar-like features proposed by Lienhart *et al.* [Lie+03] to recognize objects in the image. Conversely, Lui and Jarvis [LJ10] use a feature extraction method based directly on the standard 2D Haar wavelet decomposition proposed by Jacobs *et al.* [Jac+95] that was adapted for mobile robotics by Ho and Jarvis [HJ08].

2.5.3.5. Gist

Another holistic approach to feature detection is Gist [SI05; SI07], which captures simple characteristics of an image such as orientation, color, and intensity applying center-surround differences (i.e., the image value of a pixel or group of pixels is compared to its outer surroundings) across image scales. The average of each type of feature in every scale is then computed for 4x4 grid subregions of the image and saved as a feature vector. If many different types of features and scales are used, the combined dimension of the vectors can be quite large. Hence, a dimension reduction is performed by applying Principal Component Analysis (PCA) [Jol05] followed by Independent Component Analysis (ICA) [HO00] to obtain the final descriptor.

2.5.3.6. Edges

Edges are used to obtain outlines in the context of computer vision. In particular, Tapus and Siegwart [TS05] and Liu and Siegwart [LS14] utilize the Sobel operator as an intermediate step to obtain segments of vertical edges (see Section 2.5.3.10), whereas Goedemé *et al.* [Goe+04] employs this operator to apply the so-called *invariant column segments* method, which is not an edge detector strictly speaking but a specialization of the affine invariant regions that are commented below (Section 2.5.3.9). These column segments are described using eleven different invariant properties: a *geometric invariant* based on the segment lengths and their distance to the horizon; three *color invariants*, one for each RGB channel; and seven *intensity invariants* that characterize the intensity profile along the segment. For further reference, a comparison of several edge detectors can be found in Ziou and Tabbone [ZT98] and in Maini and Aggarwal [MA09].

2.5.3.7. Attention-based detectors

These detectors are based on the concept of *saliency* and inspired by the primate visual system. According to Itti *et al.* [Itt+98], salient regions are areas of an image that capture viewers' attention at a first glance. Starting from several feature representations, named conspicuity maps, that encode information from simple features such as color, center-surround differences (i.e., bright centers surrounded by a dark area or vice versa) and local orientation information, a *saliency map* is obtained by normalizing the values of every conspicuity map independently to a fixed range $[0, M]$, multiplying each map by a factor $(M - \bar{m})^2$, where $\bar{m}$ is the mean value of all the local maxima without considering the global maximum, and combining them together using a weighted average. This algorithm promotes maps with a few strong peaks, which correspond to eye-catching areas of the image (e.g., a red sign in a forest background), and naturally discards very homogeneous ones. This approach was developed by the same research group as Gist (Section 2.5.3.5), which explains why the same initial features are used in both methods and why they usually appear together in the literature.

2.5.3.8. Keypoints

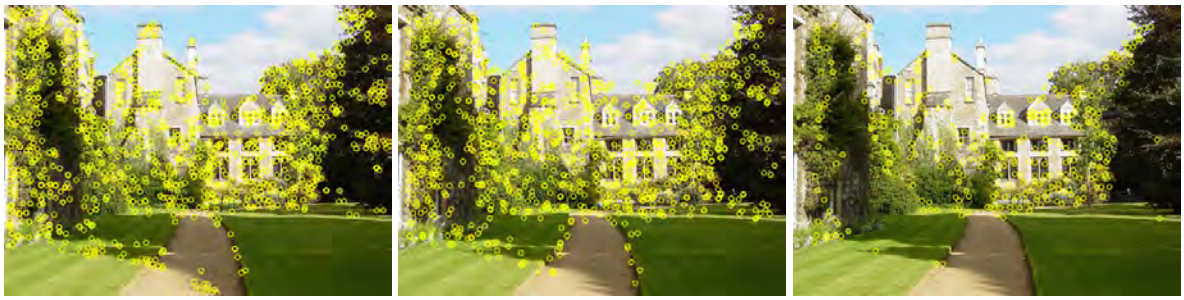
In the context of computer vision, interest points are features well-localized in the image that provide rich local information and exhibit a stable behavior across substantial variations in the illumination conditions, affine distortion, and viewpoint changes, which allows to detect them repeatedly. These keypoints are described in terms of local properties, such as image gradients, computed at different image scales.

The most pre-eminent keypoint detection and description algorithm is SIFT (Scale Invariant Feature Transform) [Low99; Low04; Se+05], which is the standard for vision-based topological SLAM. Later on, Bay *et al.* [Bay+08] developed SURF (Speeded-Up Robust Features) with the aim of reducing the computational burden of SIFT. This fact makes it a better candidate for real-time applications and explains why it has been employed in the most recent publications that opt for keypoints [CN08; Mad+11; Cad+12; LM13].

Apart from these two globally used keypoint detectors, there exist many other alternatives. It is worth mentioning CenSurE (Center Surround Extrema) [Agr+08] which is significantly faster than the previous two methods, but slightly more sensitive to rotation. Its most common implementation is referred to as Star, because the shape of the detector resembles that geometric figure. SUSurE (Speeded Up Surround Extrema) [EMC09] is an interest point detector and

descriptor based on CenSurE capable of executing two to three times faster with only a slight loss in repeatability.

Lately, a lot of attention is being paid to developing efficient keypoint detectors and descriptors that can be employed in real-time applications and with limited computation power. This has given rise to binary descriptors that are much faster to compute and match at the expense of a loss in distinctiveness. Some examples include BRIEF [Cal+10], BRISK [Leu+11], and FREAK [Ala+12]. A comparison of these and other binary descriptors can be found in [Fer+13].



(a) (b) (c)

Figure 2.5. Sample SIFT (a), SURF (b), and Star (c) keypoints extracted from an image of the *Oxford buildings dataset* [Phi+07]. The features were obtained using the default parameters for the three algorithms in the OpenCV library [Bra00].

2.5.3.9. Affine covariant region detectors

Affine covariant region detectors emerged with the idea of extracting features from images that were robust to perspective transformations. It is unclear which is the best among them, as they are often complementary and well suited for extracting regions with different properties. Mikolajczyk *et al.* [Mik+06] carried out a survey comparing the most common detectors, among which Harris-affine and MSER (Maximally Stable Extremal Regions) can be found. These features are then encoded using keypoint descriptors such as SIFT or SURF.

It is also interesting to point out that Romero and Cazorla [RC10] run the JSEG segmentation algorithm [DM01] prior to applying MSER described with SIFT with the aim of grouping features according to the image region to which they belong and produce a graph with them.

2.5.3.10. Fingerprint of places

Once set forth the most common feature extraction methods, it is clear that they all have advantages and disadvantages which make them suitable for specific applications. As has been seen, in the pursuit of a more generally applicable method, some authors like Chang *et al.* [Cha+10], and Liu and Zhang [LZ13] have tried to employ methods that combine several types of features such as Gist (Section 2.5.3.5) and Saliency (Section 2.5.3.7).

Another interesting approach to feature combination has its origin in the paper by Lamon *et al.* [Lam+01] where the term *fingerprint of places* was coined to refer to a circular list of complementary simple features (color patches and vertical edges), obtained from omnidirectional images, whose order matches their relative position around the robot. This idea led to the publication of a series of pieces of work that further developed on the concept of fingerprint. Of special relevance is that of Tapus and Siegwart [TS05] where, thanks to the

information provided by two 180º laser range scanners, corners and empty areas (i.e., when there are more than 20º of free space between two features) are additionally detected.

More recently, Liu *et al.* [Liu+09] proposed a much simpler fingerprint procedure, exclusively based on panoramic images, which extracts vertical edges under the belief that the prevailing lines naturally segment a structured environment into meaningful areas, and uses the distance among those lines and the mean U-V chrominance of the defined regions as a lightweight descriptor called FACT (Fast Adaptive Color Tags), which was later granted with statistical meaning and renamed DP-FACT for Dirichlet Process FACT [LS12; LS14].

2.5.4. Node extraction

As aforementioned, there exist two diverging approaches to topological representation of the environment. The difference between them is that while one treats the sensor readings acquired at each discrete time step as a node, the other attempts to group similar sensory inputs together to reduce the dimensionality of the resulting map. In the second case, feature matching is first performed locally to detect when the robot has arrived at a different topological place, which can have been previously visited or not. Only then, the features of that place are compared to the rest of the encoded locations. Feature matching is discussed in Section 2.5.5.

When range sensors were the primary sensory source, the majority of the solutions employed geometric properties of the environment. For example, Choset and Nagatani [CN01] use *Voronoi nodes*, locations equidistant to three obstacles in a 2D planar map, found with sonar readings. In a corridor-like environment, these nodes correspond to junctions and dead-ends. In addition, Kuipers *et al.* [Kui+04] assimilate *gateways* (e.g., doorways, corridor beginnings and endings) to nodes.

With the rise of visual perception systems, new techniques were developed. Tapus and Siegwart [TS05] identify a new node whenever the similarity between the last two fingerprints of places falls below an experimentally defined *threshold*. Romero and Cazorla [RC12] follow a similar approach. The difference is that the current observation is compared to the descriptor of the latest identified node rather than to the previous observation. The main drawback of this solution is that the number of nodes identified decisively depends on the thresholds, which need to be carefully adjusted.

Instead of using similarities straight away, Ranganathan and Dellaert [RD09] apply the concept of Bayesian surprise introduced by Itti and Baldi [IB05] to identify topological locations. Mainly based on the concept of saliency, it states that relevant stimuli represent statistical outliers or, in other words, sudden or unexpected changes in the environment [RD08; IB05]. This method can be implemented for different sensor technologies, predominantly laser and cameras, and applied to several elementary features such as color, intensity, orientation, or motion. It is based on building a model of the current location to predict the next observation, and looking for a large deviation between this prediction and the actual measurement. In a later work, Ranganathan [Ran10; Ran12] uses a Bayesian change-point detection algorithm [AM07] as part of a place labeling method. Alternatively, Liu and Siegwart [LS12; LS14] adopt a Dirichlet process mixture model for labeling.

Finally, Chapoulie *et al.* [Cha+13] construct two multivariate normal distributions from the last N observations and apply a hypothesis test based on the Neyman-Pearson lemma [NP33] to determine if the first and the second halves of the observations correspond to the same topological location.

2.5.5. Correspondence and map matching

After detecting the distinguishing features in the environment, the subsequent step in traditional metric SLAM implementations is to track the features detected between two consecutive sensor samples. The distance between equal features is then used to compute how much the robot has moved and, if there is an encoder available, both measurements are merged to minimize errors. Afterwards, according to the movement, the current location in the map is calculated.

By contrast, in pure topological SLAM systems, correspondence and map matching are the same. In general, there is no need to know how much the robot has moved, but only to identify if it has returned to an already visited place. Thus, it forces to repeatedly solve a loop-closing problem because correspondence is computed among the already encoded nodes instead of with the previous sample [Wer+09b].

It is important to remember that it is almost impossible to obtain two identical samples because of measurement noise, changes in the environment and, in addition, because when revisiting a place the robot performs the measurements in a slightly different location or with another orientation. For this reasons, correspondence and map matching are usually carried out by means of *dissimilarity measurements*, like the Mahalanobis [Sab08; GG08], Euclidean [And+05; GG08] and χ^2 distances [Fil07], or the Jeffrey divergence [UN00], whereas Tapus and Siegwart [TS05] employ a modified version of the *global alignment* algorithm, proposed by Needleman and Wunsch [NW70] to compare DNA sequences, which takes the uncertainty of the detected features into consideration. The latter opted for this approach, which accounts for an average 83.82% of correct classifications in indoor and outdoor environments, after comparing it with Bayesian programming and a hybrid technique which merges the global alignment with uncertainty and Bayesian programming methods.

Moreover, in the context of visual topological SLAM Angeli *et al.* [Ang+08b] and Romero and Cazorla [RC10] utilize the *relative position* of the local features within the images as a matching criterion. However, while the former uses RANSAC to ensure that geometric constraints are met [Nis04], the latter applies the Graph Transformation Matching (GTM) algorithm by Aguilar *et al.* [Agu+09]. In addition, Li and Olson [LO12] proposes the Incremental Posterior Joint Compatibility (IPJC) test to match constellations of features together rather than considering them individually. Although its formulation is equivalent to the well know Joint Compatibility Branch and Bound (JCBB) test [NT01], it is faster and more accurate, and performs better on non-linear problems.

Finally, as map matching becomes more demanding when the mapped area grows, some authors like Goedemé *et al.* [Goe+04] or Romero and Cazorla [RC10] propose applying *space-partitioning techniques* like *k*d-trees in order to optimize the search and comparison processes. In addition, other researchers like Fraundorfer *et al.* [Fra+07], Angeli *et al.* [Ang+08c; Ang+08a; Ang+08b] and Cummins and Newman [CN08] propose building *bag-of-words* models [SZ03; Csu+04; NS06; NZ06], from SIFT [Low04] and SURF [Bay+08] features respectively, to enable fast matching. A bag-of-words model consists in quantizing features into a set of discrete values or "words". The output is therefore a histogram where each bin corresponds to a different word. For matching purposes, each keypoint identified in the current image is first assigned to a word and is the resulting histogram what is compared across nodes, as matching histograms is much faster than finding the similarity between hundreds of individual features. The vocabulary (i.e., the collection of words) is usually built offline from a dataset recorded in a similar environment, although Filliat [Fil07] proposes a method to construct it online.

2.5.6. Map fusion: Dealing with loop-closing uncertainty

The final stage in topological SLAM involves updating the map. If the current location does not correspond to any node known in advance, then the robot is in an unexplored area and, therefore, if the measurements meet the requirements to be considered a distinctive place, it should be added to the map. A more complex situation occurs when there is a positive match. Remember that for topological SLAM one of the most awkward problems is perceptual aliasing, and suppose that for map matching only sensory information is used. Consequently, there may be several nodes in the map that coincide with the measurements. Notwithstanding, this by no means signify that it is an already visited place. This section concentrates on the different manners in which loop-closing uncertainty in topological maps has been tackled in the literature.

2.5.6.1. The consistent pose estimation paradigm

Some of the early developments on map fusion are inspired by the concept of *consistent pose estimation* (CPE) introduced by Lu and Milios [LM97], which attempts to globally optimize the recorded set of poses based on how well neighboring sensor scans match. Gutmann and Konolige [GK99] presented the local registration/global correlation algorithm that is based on building local metric maps (named *local patches*) from the last few measurements in order keep the accumulated odometric error low and ensure topological correctness. The global metric map is then incrementally updated by comparing the topological structure of the latest patch with older portions of the map. A high match score with low ambiguity and variance indicates a loop closure. The experiments, carried out with robots equipped with laser sensors and encoders in four different environments of up to 80 by 25 meters, yield fairly good metric maps under the assumption that local patches are accurate enough. Later on, Konolige [Kon04] presented an efficient algorithm for multiple-loop maps that allows to extend the CPE method to map much larger areas (i.e., around 10^5 distinct locations).

2.5.6.2. Spatial semantic hierarchy

The *spatial semantic hierarchy* (SSH) is a model of knowledge for large-space introduced by Kuipers [Kui00]. It involves four qualitative and quantitative representations. At the *control* level, the agent continuously seeks *distinctive states* with a combination of trajectory-following and hill-climbing strategies. The *causal* level abstracts this pattern of behavior into a discrete model described in terms of states, sensory views, actions, and the causal relations among them. The *topological* level introduces the concepts of places, paths, and regions, and links them through turn and travel actions in order to explain the regularities observed among views in the control level. Finally, the *metrical* level represents a global geometric map of the environment in a single frame of reference. This framework was subsequently formalized using non-monotonic logic by Remolina and Kuipers [RK04].

Kuipers *et al.* [Kui+04] extended the basic SSH with *local perceptual maps* (LPMs), a bounded occupancy grid. In this work, they identify *gateways* in corridors as the locations where the distance between the medial axis edge and the obstacles is a local minimum close to a larger maximum. However, they believe that other alternatives are possible. In addition, they include *path fragments* associated to the gateways. This information, along with travel control laws, is employed to obtain a *local topology* of a place in terms of distinctive states and directed paths.

In order to produce the global topological map, a tree whose nodes are topological map-distinctive state pairs is maintained and pruned over time by matching local topologies, and LPMs if necessary. Instead of pruning, Johnson and Kuipers [JK12] proposed expanding only the most probable hypothesis to ensure that you can always backtrack in case of error and find the correct map. Further developments of this research line include improvements to loop-closing with the incorporation of the planarity constraint [SK04], and the construction of accurate global metric maps from the topological skeleton obtained [Mod+04].

2.5.6.3. Partially observable Markov decision processes

Partially Observable Markov Decision Processes (POMDPs) [Cas+96; KS96; Kae+98] have also been employed for topological map building due to their ability to determine the navigation policy that the robot should follow in order to reduce uncertainty. Tomatis *et al.* [Tom+02] and Tapus and Siegwart [TS05] extended POMDPs to perform multi-hypothesis tracking and determine a pose distribution. However, as computing an optimal policy is intractable in large environments, Tomatis *et al.* [Tom+02] suggested using the *most likely state* criterion to choose the following action, whereas Tapus and Siegwart [TS05] opted for another heuristic, the entropy of the current location probability distribution, to decide the control commands. In the latter case, whenever the entropy falls below an experimentally determined threshold, the robot's location is assumed certain and the map is updated accordingly, either by adding a new node or by merging the latest fingerprint information with the node representative.

Loop closures are also identified by means of the POMDP. Whenever the robot returns to a previously visited place, the probability of that location should split in two. One hypothesis would correspond to a new location and the other to a node already present in the map. If both divergent peaks evolve similarly over time, a loop closure is assumed [Tap05].

2.5.6.4. Probabilistic topological maps

A Bayesian inference framework has also been explored for topological mapping. Ranganathan and Dellaert coined the term *probabilistic topological map* (PTM), a sample-based representation that estimates the posterior distribution over all the possible topologies that can be built given a set of sensor measurements [RD04; Ran08]. Due to the fact that this is a problem of a combinatorial nature that rapidly becomes computationally intractable, they proposed approximating the solution by drawing samples from the distribution using Markov-Chain Monte Carlo sampling (MCMC) [RD04; Ran+06]. In principle, this technique is applicable to any landmark detection scheme as long as the landmark detection algorithm does not provide false negatives (i.e., the robot's sensors do not fail to recognize landmarks).

Afterwards, they presented Rao-Blackwellized Particle Filters (RBPFs) [Dou+00b; Mon+02] as an alternative to MCMC sampling for PTMs [RD06a; Ran08; RD11]. Particle filters are yet another Monte Carlo localization technique used to probabilistically estimate the state of a system under noisy measurement conditions. This technique permits incremental inference in the space of topologies, conversely to MCMC which is a batch algorithm, and can therefore be computed in real time. In order to overcome the samples degeneracy problem over time [Dou+00a], that can lead to convergence issues, they suggest integrating odometric data to draw more likely particles with higher probability. However, the selection of the appropriate number of particles still remains an open issue, as particle filtering inherently has the risk of disposing the correct map. Koenig *et al.* [Koe+08] also employ a RBPF in which each particle incrementally constructs its own graph of the environment using color histograms and

odometry information. Local graphs are compared with the global graph to determine the best matches and, simultaneously, the resampling weights for each particle. Other example of the application of particle filters, in this case the regular version [Dou+01], is that of Andreasson *et al.* [And+05].

The main advantage of PTMs is that all decisions are reversible and the algorithm is therefore capable of recovering from incorrect loop closures. In the end, only a small set of similar topologies have non-negligible probabilities. The experiments conducted suggest that, if the environment is unambiguous (i.e., it does not have symmetries), the ground-truth topology is assigned a much higher posterior probability mass than the other alternatives.

2.5.6.5. Voronoi graphs and neighboring information

Choset and Nagatani [CN01] represent the environment by means of a Generalized Voronoi Graph (GVG). A GVG is the set of points equidistant to n obstacles in n dimensions. When used in the plane, it reduces to the set of points equidistant to two (or more) obstacles, and define a roadmap of the robot's free space. Voronoi meet points, which are locations equidistant to $n + 1$ obstacles, are used as natural landmarks because they provide topologically meaningful information that can be extracted online (e.g., junctions, dead-ends...). The main problem with Voronoi nodes is that they are very sensitive to changes in the configuration of the environment. If non-structural obstacles are moved, Voronoi vertices may appear or vanish.

In order to achieve SLAM, the robot follows simple control commands looking for these nodes in the environment. Loop-closing is carried out by comparing the subgraph built from the latest observed nodes to the already encoded map. Ambiguity is resolved by following a candidate path and ruling out inconsistent matches based on the new visited places. This method as is assumes that the robot is equipped with infinite range sonar sensors, and is only suitable for static and planar environments with plenty of obstacles. Based on this idea, Beeson *et al.* [Bee+05] introduced Extended Voronoi Graphs (EVGs) to address the problems of GVGs derived from limited sensory horizons by means of local perceptual maps (Section 2.5.6.2).

The research path initiated by Werner *et al.* [Wer+08a; Wer+08b] is also remarkable. They apply Bayesian inference to obtain a topological map in ambiguous environments that explains the set of observations without the need for motion knowledge. The method is based on guaranteeing consistency between the local neighboring information extracted from the latest n observations and the constructed map while keeping the number of topological vertices as low as possible, following the Occam's razor principle. Topological places, where captures are acquired, are identified by means of a GVG using sonar readings. The algorithm assumes that there exist some prior information about the connectivity but not about the number of distinct locations in the environment.

Initially, a sequential Monte-Carlo technique was employed to maintain a series of candidate maps [Wer+09a], which was later replaced by a particle filter [Wer+09b]. In order to be able to recover from incorrect loop closures, Tully *et al.* [Tul+09] introduced a multi-hypothesis approach based on a tree expansion algorithm specifically conceived for edge-ordered graphs [Dud+93], as well as a series of pruning rules to keep the number of hypothesis under control. Recently, Tao *et al.* [Tao+11] discussed the benefits of Saturated Generalized Voronoi Graphs (S-GVG), that employ a wall-following behavior to navigate within sensor range limits, and performed SLAM using a similar hypothesis tree. Finally, Werner *et al.* [Wer+12] suggested applying *stochastic local search* to produce the topological map.

Before concluding this section, it is worth mentioning the work by Doh *et al.* [Doh+09], who deal with semi-permanent dynamics induced by door opening and closing. They classify GVG nodes in invariant (i.e., junctions, corners and ends of corridors) and variant (i.e., doors). Nodes are told apart using the areas between two local minima of a sensor scan (which can be used to identify doors), and looking for a vanishing point from a range scan or in an image (for invariant nodes).

2.5.6.6. Appearance-based topological SLAM

Appearance-based SLAM attempts to infer topological maps based only on visual information. They discard employing odometric data because it is prone to cumulative errors, especially on slippery surfaces. Most early approaches rely on SIFT keypoints extracted from omnidirectional images. Some examples include the work by Zivkovic *et al.* [Ziv+05], who solve the map building process using graph-cuts, and Goedemé *et al.* [Goe+07], who resort to Dempster-Shafer theory of evidence [Dem67] for loop-closing. Unfortunately, these solutions require offline computation.

Later on, Fraundorfer *et al.* [Fra+07] presented a real-time framework based on the *bag-of-words* paradigm [Csu+04], where images are quantized in terms of unordered elementary features taken from an offline-built dictionary. Loop-closing is identified by visual word comparison following a voting scheme. Romero and Cazorla [RC10; RC12] take a similar approach but without the need for a dictionary. They build graphs from homogeneous regions using MSER features described with SIFT and use the GTM algorithm for matching. They then compare the graphs from newly acquired images with the latest visited topological node representative. If the matching score is below a threshold, it is then compared—using another threshold—with the rest of the encoded vertices in order to identify loop closures. If no match is found, a new node is added to the map. The main inconvenient of this algorithm is that it is extremely sensitive to the two thresholds. The value of these parameters has a decisive impact on the final topology obtained.

Angeli *et al.* [Ang+08c; Ang+08a; Ang+08b] proposed a method that builds the vocabulary online, following the procedure developed by Filliat [Fil07]. The problem of loop-closing is addressed following a Bayesian approach. The probability of transition between locations is modeled using a sum of Gaussians to assign higher probability to adjacent states, whereas the correspondence likelihood is computed by means of voting using the *tf-idf* coefficient [SZ03].

Furthermore, Fast Appearance-Based Mapping (FAB-MAP), which is a Bayesian framework for navigation and mapping exclusively based on appearance information developed by Cummins and Newman as a solution to loop closure detection [CN07; CN08; Cum09; CN10b], has attracted a great deal of attention. It relies on a vocabulary model constructed offline from the clustering of SURF features extracted from a large collection of independent images. The words obtained are then organized using a Chow-Liu tree [CL68] to capture the dependencies among them (e.g., car wheels and car doors are likely to appear together) and build a generative model. This vocabulary model is used to approximate the partition function in the Bayesian formulation, which provides a natural probabilistic measure of when an observation should be labeled as a new location. The experiments conducted outdoors suggest that it performs well in repetitive environments and is fast enough for online loop-closing. The fact that it requires offline training is an inconvenient, although tests carried out indoors with the bag-of-

words model built for outdoor environments produce surprisingly good results according to the authors.

Some improvements have been introduced to the original algorithm since its presentation. First, speed was increased by more than 25 times, with only a slight degradation in accuracy, thanks to the usage of concentration inequalities to reduce the number of hypothesis considered [CN10a]. The formulation of the algorithm was also modified to operate on very large environments (over trajectories of around 1,000 km) [CN11]. Finally, Paul and Newman [PN10] incorporated the spatial arrangement of visual words to improve distinctiveness. Recently, Johns and Yang proposed methods to deal with short-term [JY13b] and long-term dynamics [JY13a] without having to store several different images for the same location.

Continuous Appearance-based Trajectory SLAM (CAT-SLAM) [Mad+11] incorporates odometry, following the approach of FastSLAM [Mon+02], to appearance-based SLAM using FAB-MAP. The current location is modeled as a probability distribution over a trajectory and appearance is treated as a continuous variable. The evaluation of the distribution is carried out using a RBPF. Compared to FAB-MAP, it identifies three times as many loop closures at 100% precision (i.e., with no false positives). By contrast, FAB-MAP is capable of recognizing places when approached from a different direction, whereas CAT-SLAM cannot because it relies on odometric information. Enhancements to computational and memory storage requirements, like pruning those nodes in the trajectory that are locally uninformative once a preset maximum number of nodes is reached, were subsequently introduced to allow continuous operation on much larger environments [Mad+12a; Mad+12b].

In the line of using odometry for appearance-based topological SLAM, Lui and Jarvis [LJ10] implement a correction algorithm for loop closure detection that relies on *visual odometry*. They employ the Kanade-Lucas-Tomasi (KLT) features [LK81; TK91; ST94] to estimate the distance traveled and column image comparison using the Sum of Absolute Differences (SAD)—also known as Sum of Absolute Errors (SAE)—for the front 180º field of view (FOV) of the robot to estimate the bearing. These are then used to reduce the matches retrieved from the database, using a Haar wavelet-based signature, utilizing the relaxation algorithm proposed by Duckett *et al.* [Duc+00]. The current location is then told apart by means of SURF [Bay+08]. This system has been proven effective in indoor and semi-outdoor environments. However, its main drawback lies in the complexity of the robot infrastructure which includes an omnidirectional stereovision system and a web camera to perform visual odometry, as well as a stereo camera for obstacle avoidance.

2.5.6.7. The final stage: Updating the map

Finally, once the uncertainty has been resolved, the new information gathered should be incorporated to the map for future reference. The most common solution is to store the new features as an alternative representation of the node. However, in the long run this ends requiring too much storage space and unnecessarily multiplying the comparisons required for map matching. For this reason, some authors suggest, on the one hand, removing any unobserved nodes, features, and relations or, better, implementing a gradual "forgetting" process which could take into account changes in the environment (e.g., an open door appears closed when revisiting a place) [Vas+07]. On the other hand, Kuipers and Beeson [KB02], Tapus and Siegwart [TS05], and Liu and Siegwart [LS14] propose creating a mean node representative with a view to reducing the impact of scene variability. However, this approach can lead to representative degradation in the long term.

2.6. Concluding remarks

This chapter has introduced the topological SLAM problem, along with the different steps that need to be fulfilled and numerous techniques that could be chosen to implement them, for the reader to acquire a general overview of the field.

In sensing, after a decade of predominance of laser range scanners we are apparently undergoing a paradigm shift and most researchers are abandoning laser sensors, which are too expensive bearing in mind that topological approaches do not take full advantage of their precision and which have already been thoroughly studied and extensively employed, for cameras, probably owing to the fact that these are much cheaper and enable to extract plenty of different features that can help push topological applications forward. The majority of the works reviewed focus on omnidirectional imaging because they provide rotational invariance and cover a 360º field of view. By contrast, much less publications deal with frontal computer vision, either monocular or stereo, despite resembling human beings' visual system and being much easier to install in any mobile entity.

As far as detection using computer vision is concerned, currently there seem to be two main open research lines. On the one side, many authors rely on keypoints, which have proved to perform well, especially in combination with bag-of-words models. Still, as SIFT and SURF are the most employed keypoint features, the robots must have a reasonable amount of computation power on board. Thus, they are commonly used in outdoor robots that have to map large environments, although not exclusively. On the other hand, good results have also been achieved with a wisely chosen collection of simple complementary features (e.g., color, vertical edges...). This approach is usually preferred in indoor applications and should be a better alternative for smaller, computationally limited service robots.

As mentioned, topological node extraction is not indispensable but may be of interest for robotic platforms that run on low specification hardware, as loop-closing will only need to be performed when a new node is reached. This resembles human behavior, as we only attempt to localize ourselves when we arrive at a new place; meanwhile, we concentrate on tasks such as short-term path planning and obstacle avoidance. Two fundamental alternatives exist for node detection: building a discrete multinomial model that allows to apply statistical procedures or employing thresholds directly on similarity measurements to determine when the current set of features perceived no longer matches the previous node identified.

In map matching and fusion, the probabilistic approach appears to be the most consolidated solution, because it allows to keep track of several hypotheses and recover from incorrect loop closures. However, in spite of the topological representation being less computationally demanding, constantly solving a loop-closing problem can be cumbersome in large maps as the robot can simultaneously believe to be in many distinct locations, which results in having to deal with a huge pose distribution that multiplies the calculations required. For this reason, most of the approaches resort to odometry in one way or another to reduce the list of possible candidates.

With respect to the literature, this thesis is located as indicated in Table 2.3. A camera is used as the primary sensor and, from the available options, a monocular one was chosen. For each of the remaining modules in topological SLAM, only the most relevant techniques that can be applied in conjunction with computer vision are shown. For detection and description, the two major alternatives, keypoints and fingerprints, are combined with the aim of making the most of their strengths and compensating for their weaknesses. In node extraction, a hybrid

approach that is generally applicable, like threshold-based methods, but that builds a model of the previous observations is used. Regarding correspondence and map matching, features are compared by means of dissimilarity measurements taking the relative position of the features into account with a method inspired by the natural language processing field. Finally, in order to be able to evaluate and keep track of several possible topologies over time, instead of having to make a decision every time step, a particle filter is employed.

Table 2.3. The thesis with respect to the literature. The techniques employed appear in bold type.

Module	Technique			
Sensor (Camera)	**Monocular**	Omnidirectional	Stereo	RGB-D
Detection & Description	**Keypoints**		**Fingerprints**	
Node extraction	Threshold-based	**Hybrid**		Model-based
Correspondence & Map matching	Dissimilarity measurements	**Dissimilarity measurements with relative position**		Bag-of-words
Map fusion	POMDPs	MAP Bayesian formulation		**Particle filters**

3

Visually Perceivable Adjacent Color Histograms and Keypoints

All our knowledge has
its origins in our perceptions.
Leonardo da Vinci (1452–1519)

The first task a robot must accomplish is to perceive its surroundings to convert sensor readings into meaningful information that it can employ to localize itself in the environment. Cameras have always been regarded as the ideal sensing technology for topological feature extraction and several methods that use this sensor have been proposed in the literature. However, they are either time-consuming, require additional sensors, or are very sensitive to perceptual aliasing, especially if used in conjunction with directional cameras, which do not cover a 360º field of view.

At the sight of these limitations, this chapter presents a fast-to-compute collection of features extracted from monocular images, and a matching procedure for location identification in structured indoor environments inspired by the natural language processing field. Although only dominant vertical edges, color histograms, and a reduced number of keypoints are employed, the matching framework introduced allows to incorporate almost any other type of feature. The results of the experiments carried out in home and office environments suggest that the proposed method could be used for real-time topological scene recognition even if the environment changes moderately over time. A summarized version of this chapter can be found in the journal article [Boa+14a].

3.1. Introduction

The initial step in any topological SLAM implementation is to define what is going to be considered a landmark in the environment and choose the convenient sensing technologies to perceive them or, conversely, select one or several type of sensors and determine which cues can be extracted from the data they provide. Surprisingly, although an inappropriate decision at this stage complicates the subsequent steps of the algorithm, it is often disregarded and makes

it even more difficult to overcome the *perceptual aliasing* problem (i.e., two totally distinct locations appear identical to the robot's sensors), not to mention the added complexity for the already challenging *correspondence problem* (i.e., attempt to determine if sensor measurements taken at different times correspond to the same physical location).

By means of vision, human beings identify relevant or distinctive aspects of the environment that are used as landmarks for localization. Hence, why a robot equipped with a camera should not be able to do exactly the same? Most of the developments in visual topological SLAM are based on techniques inherited from the object detection world, due to the overwhelming popularity of the successful *keypoint* (and affine covariant region) detectors which provide highly distinctive, persistent, and robust features [Boo+07; Sab08]. Notwithstanding, their high computational burden, especially with large images, leaves very little time for other tasks like map fusion, motion planning, or obstacle avoidance in robots running on low specification hardware, because these techniques were originally designed to run in batch, as opposed to real-time applications, precisely one of the essential requirements in robotics. A widespread attempt to overcome this issue is to use bag-of-words models [SZ03; Csu+04], commonly employed in natural language processing and information retrieval, to speed up the feature matching process. These models consist in categorizing the space of features into a set of representative "words", called *vocabulary* or *dictionary*, using a quantization technique. Each cluster center obtained represents an individual "word". Features in an image are thus reduced to a histogram of word counts that is easy to compare with some distance measure [Cha07].

The quality of the vocabulary has a direct impact on the performance of the model because it has to be rich enough to allow the robot to easily distinguish between locations. Therefore, choosing an appropriate training dataset that is similar to the environment the robot will navigate through is a critical task. Finding this dataset is usually not problematic for structured outdoor environments because there are huge repositories of images available on the Internet like, for instance, Google Street View [Wwwb]. By contrast, it is much more difficult to obtain image datasets from indoor environments, which are inherently more diverse. In the latter case, the best alternative is to manually record images of the environment the robot is going to move in, with the consequent loss of generality, and use them to build the dictionary. This solution definitely works, but for a SLAM application it is somewhat like "cheating" because we are providing the robot with information of the environment in advance. Bearing this in mind, Filliat [Fil07] proposes a method to build the vocabulary online that relies on an incremental nearest neighbor classifier. For every new feature, if the closest "word" in the dictionary is farther than a threshold, an additional "word" is incorporated to the vocabulary. In exchange for being online, the resulting dictionary is sensitive to noise and to the feature extraction and processing order, which does not happen with batch methods like k-means clustering [Mac67].

Furthermore, bag-of-words models produce holistic representations because the whole image is reduced to a histogram of visual word counts. Hence, no geometric information is preserved (i.e., the relative position of the features in the image is lost). A solution to this issue is to apply the *spatial pyramid* framework [Laz+06] that consists in dividing the image into a fine grid and extracting a bag-of-words histogram in each cell. The main inconvenient of this approach is that the segmentation is completely arbitrary. Some examples of the usage of bag-of-words models include the publications by Angeli *et al.* [Ang+08b] who adopt standard SIFT features [Low04] together with hue and value histograms in a HSV color space, Fraundorfer *et al.* [Fra+07] who use MSER features [Mat+02] described with SIFT, and Cummins and Newman [CN08], where SURF features [Bay+08] are employed for scene recognition.

An alternative approach has its origin in an article by Lamon *et al.* [Lam+01], where the term *fingerprint* of places was coined to refer to a circular list of complementary simple features (color patches and vertical edges), obtained from omnidirectional images, whose order matches their relative position around the robot. This idea led to the publication of a series of papers that further developed the concept of fingerprint. Of special relevance is the work of Tapus and Siegwart [TS05] where, thanks to the information provided by two 180º laser range scanners, corners and empty areas are additionally detected. They employ a modified version of the *global alignment* algorithm [NW70], used to compare DNA sequences, capable of dealing with uncertainty to obtain a matching probability of the features. Unfortunately, this approach requires multiple and expensive sensors.

More recently, Liu *et al.* [Liu+09] proposed a much simpler fingerprint procedure, exclusively based on panoramic images, that extracts vertical edges under the belief that the prevailing lines naturally segment a structured environment into meaningful areas, and encode the distance among those edges and the mean U-V chrominance of the defined regions—in a LUV color space—in a lightweight descriptor called FACT, which was later granted with statistical meaning and renamed DP-FACT [LS12; LS14]. Although using the U-V chrominance is an interesting option due to the fact that the difference between colors can be computed applying the Euclidean distance, the average approach always has the risk that two completely different regions result in a very similar value. Moreover, for frontal computer vision, where only a limited field of view is available, this method heavily suffers from perceptual aliasing because the features that can be extracted are too weak and reduced in number.

As can be observed, color is often used as a complement, but there exist some approaches worth mentioning that use it as the single source of information. Ulrich and Nourbakhsh [UN00] build six n-bin histograms from the whole image, one for each of the channels in the RGB and HSL color spaces, whereas Werner *et al.* [Wer+09a] construct a single 3D histogram from RGB tuples. Color is definitely a rich source of visual information but is extremely sensitive to illumination changes, especially in the RGB color space where the color and illumination components are not independent.

It is important to emphasize that the vast majority of the aforementioned methods employ an omnidirectional camera. This is easily explained by the fact that omnidirectional cameras are the only ones which guarantee *rotational invariance* (i.e., no matter what orientation a robot has in a given location, the image captured is always the same). This is a very desirable property but these cameras have the disadvantages that they are more complex to install, their image quality is not as good compared to a directional camera and, conversely to other type of cameras like stereo or RGB-D, additional sensors are often required for navigation in order to estimate the distance to obstacles. Furthermore, all the methods presented above have either one or several of the following drawbacks. They require plenty of different, and often costly, sensors, are sensitive to perceptual aliasing, require offline training, or are computationally expensive to an extent that makes it fairly difficult to use them for real-time applications in robots with limited computational resources.

Bearing these shortcomings in mind, this chapter proposes a lightweight vision-only monocular feature extraction procedure based on the notion of *fingerprint* and a matching algorithm adapted from the natural language processing world, aimed at topological localization—or, in general, at topological SLAM—for structured indoor environments. Both feature extraction and matching have been designed to ensure a reasonable computing time that guarantees that the proposed procedure can be employed in real-time applications. The

usage of monocular images allows this method to be applied to any type of robot, as opposed to omnidirectional cameras that require a more complicated installation. Moreover, it is compatible with stereo vision that permits to obtain depth estimates for obstacle avoidance with a single camera.

The rest of the chapter is structured as follows. First, Section 3.2 details the necessary steps to extract the proposed collection of features, followed by an explanation of the matching technique employed to find correspondences between fingerprints in Section 3.3. Subsequently, Section 3.4 comments on the results obtained for different image sets and, finally, the most relevant conclusions are drawn in Section 3.5.

3.2. Proposed fingerprint

A monocular camera has been chosen as the unique sensor to extract a collection of complementary features to derive a fingerprint, taking into account the three properties stated by Stankiewicz and Kalia [SK07]: saliency, persistence, and informativeness that are explained in detail in Section 2.5.3. First, the image is segmented into different subregions by extracting structural vertical edges, which are persistent and salient but fairly uninformative. As a consequence, the subsequent features can be computed in parallel in each of the resulting subimages and, moreover, they are granted with spatial meaning (i.e. they become ordered) based on the image content rather than by an arbitrary grid as with the spatial pyramid framework.

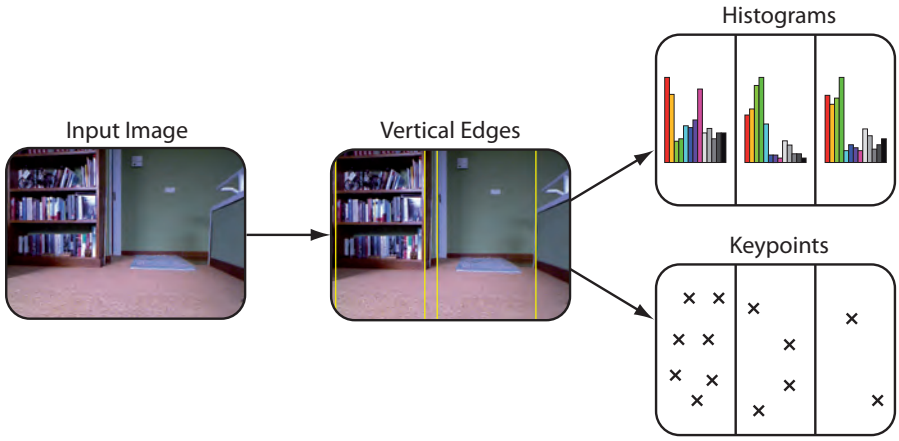


Figure 3.1. Fingerprint generation process. Vertical edges are extracted to split the image into several subregions. Then, color histograms and keypoints are computed for each of the subimages.

Any type of feature can be computed in these subregions, but a combination of color histograms and keypoints is proposed in this thesis (Figure 3.1). These features compensate for each other's drawbacks as histograms operate on a global basis—or semi-global, because they are computed in subimages—whereas keypoints are local. For instance, keypoints have trouble with homogeneous regions that can be told apart using color information. This fingerprint has been coined VPACK (Visually Perceivable Adjacent Color histograms and Keypoints) to convey the idea of a "visual pack", a collection of visual features wrapped together. The details regarding the feature extraction process are put forward in the following sections.

3.2.1. Vertical edges

Like in the work by Liu *et al.* [Liu+09], the hypothesis that vertical edges naturally divide structured environments into informatively distinct regions is supported in this thesis. However,

note that this assertion turns out to be valid only if the lens distortion of the camera is corrected, its focal plane is parallel to the planes containing the vertical edges—a small amount of pitch may be allowed—and the roll angle is null (see Figure 3.2). For structured indoor environments this happens to be generally easy to achieve as the floor is normally perpendicular to building walls.

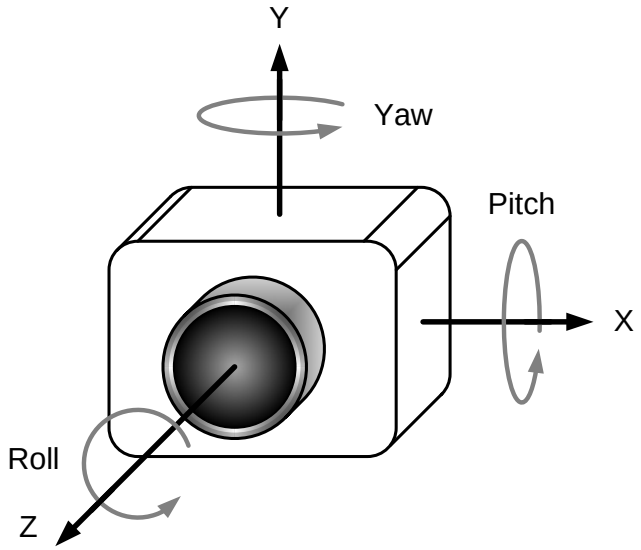


Figure 3.2. Camera rotation angles.

With a view to taking advantage of this segmentation property, dominant edges are first extracted from the image. A flowchart of the process is shown in Figure 3.3. The initial step is to convert the input image to grayscale and compute the *Sobel operator* over the x direction to enhance the vertical responses (Figure 3.4b). Mathematically speaking, the Sobel operator estimates the gradient of the image intensity I by convolving it with two 3x3 kernels or masks that approximate the horizontal G_x and vertical G_y derivatives (3.1). The gradient's magnitude and orientation at each point can then be calculated from this information as indicated in (3.2) [GW08]. Nevertheless, for vertical edge detection only the kernel along the x direction needs to be taken into consideration.

$$G_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} * I \qquad G_y = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} * I \tag{3.1}$$

$$|G| = \sqrt{G_x^2 + G_y^2} \qquad \Theta = \arctan\left(\frac{G_y}{G_x}\right) \tag{3.2}$$

Afterwards, in order to compensate for a small amount of roll in the camera or for a slight inclination of the floor, a 3 by 1 *horizontal morphological dilation* is performed (Figure 3.4c). This allows to rejoin accidentally cut vertical edges. It may be argued that after this step it is necessary to apply a median filter to remove the additional noise introduced. However, on closer examination one notes that it is dispensable at the sight of the next operation.

A *vertical morphological opening* (i.e., an erosion followed by a dilation) is then carried out to remove weak responses, which have been defined as those segments shorter than an experimentally adjusted threshold length (Figure 3.4d). Given that the edges extracted are being used as a segmentation characteristic of the environment, the recommended threshold is

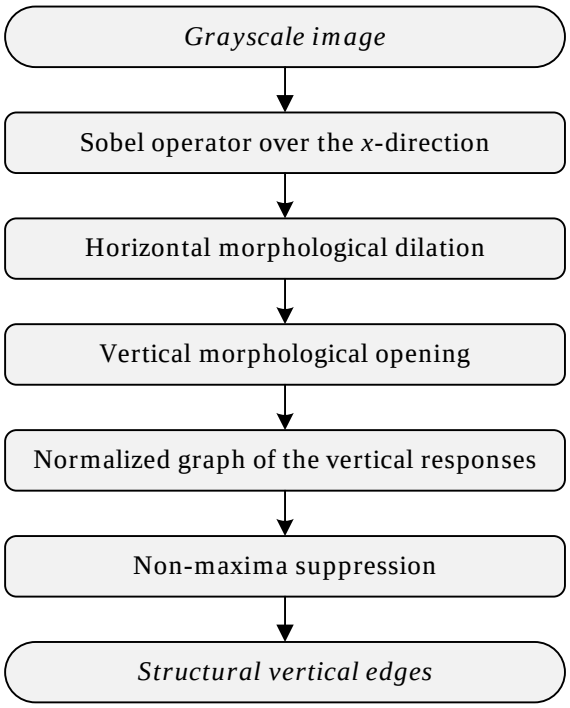


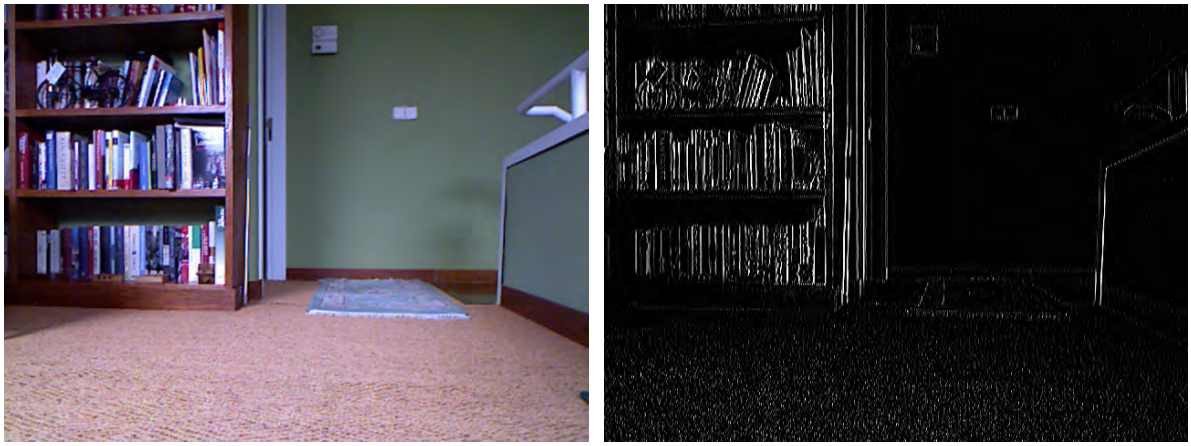
Figure 3.3. Vertical edge extraction flowchart.

rather high, around one-fifth of the image height, with the aim of capturing structural lines, like wall corners or furniture edges, while staying immune to noise coming from less permanent sources such as, for instance, books on a shelf. Furthermore, this operation disposes of the noise added in the previous step.

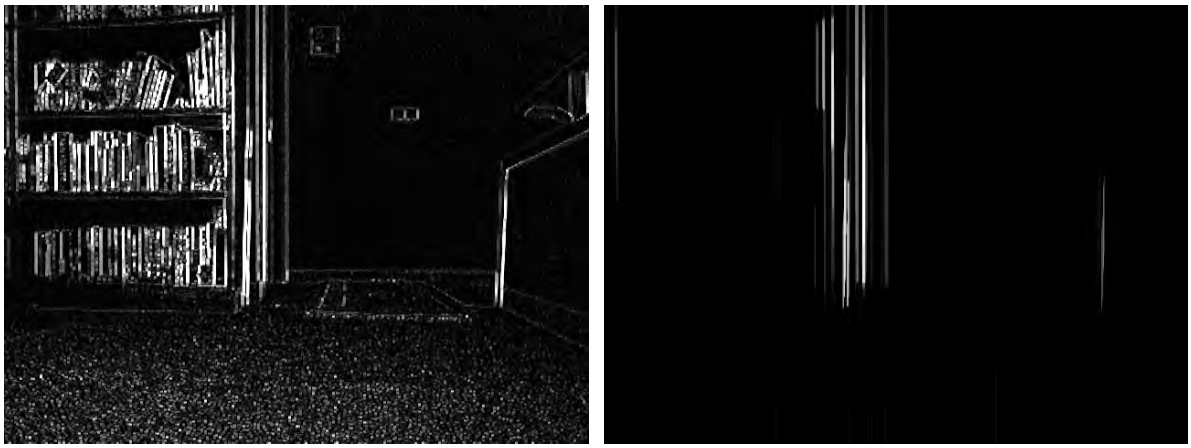
Afterwards, a *normalized graph of the overall vertical response* of each pixel column is computed (Figure 3.4e). The objective is twofold: to reduce the dimensionality of the problem and to identify the predominant vertical edges. Instead of filtering out those values lower than the mean plus a standard deviation, as suggested by Tapus [Tap05], the values higher than 5 times the mean are preserved. The output happens to be almost identical in the typical cases. However, the threshold proposed overcomes a minor of pitfall the mean plus standard deviation approach. Imagine a rare case where an image does not contain any outstanding vertical edges. Therefore, all the values are bound to be close to the mean, and the standard deviation is expected to be low. Consequently, any noisy value that departs marginally from this threshold would be classified as a line. By contrast, with the proposed method no vertical edges would be identified.

Notice that this last stage is completely different from applying a more restrictive condition to the opening operator. In fact, it is a way of preventing the removal of partially occluded structural lines. Let's consider someone resting his hand against a door frame. If the opening threshold were larger, the two segments in which the frame is divided will probably be suppressed. By contrast, with the proposed solution, they would end up counting for the same vertical response.

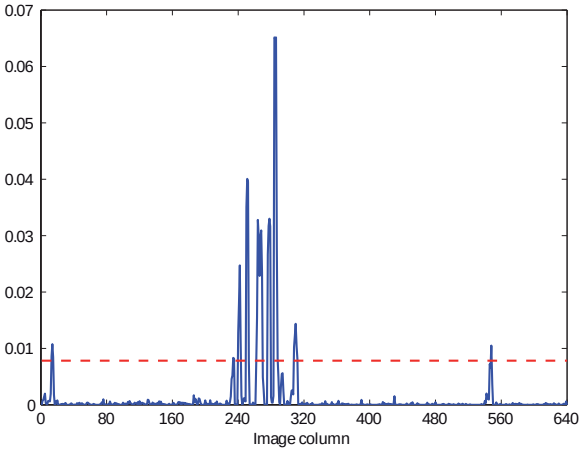
Finally, a *non-maximum suppression* (NMS) algorithm is applied to establish a minimum separation between lines (Figure 3.4f). As the objective of the extracted lines is not to serve directly as visual cues but to divide the image into different regions to permit a more local computation of features, it seems clear that these regions must possess a minimum width (e.g., 5% of the image width). To this end, an adapted version of the efficient

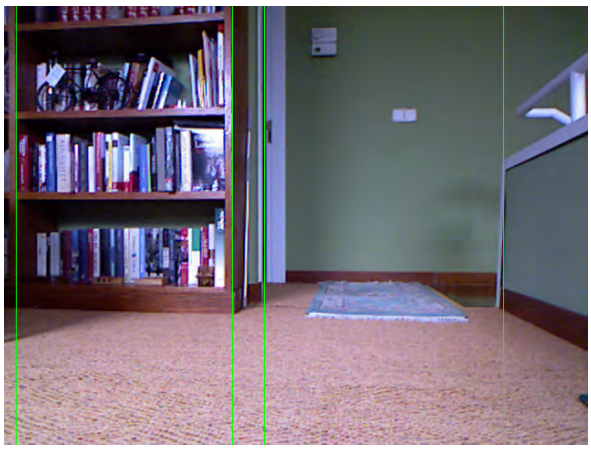


(a) (b)



(c) (d)





(e) (f)

Figure 3.4. During edge extraction, the input color image (a) is converted to grayscale in order to apply the Sobel operator over the x direction (b). A horizontal dilation is then performed to enhance vertical responses (c), followed by a vertical morphological opening to remove short edge segments (d). Subsequently, a normalized histogram of the edge responses is computed and the values lower than a given threshold are filtered out (e). Finally, non-maximum suppression is applied to preserve the strongest response over a fixed neighborhood. The extracted vertical edges are drawn on the input image for illustrative purposes in (f).

Algorithm 3.1. Revised 1D non-maximum suppression for a $(2n+1)$ neighborhood (Adapted from Neubeck and van Gool [NG06])

Input : A 1D input sequence H of length L, $H = \{H[0], \ldots, H[L-1]\}$.
Output : List of *maxima* indices.
Define : n as the half neighborhood width.
i as the index of the current maximum candidate.
j as the upper neighbor of i.
k as the upper neighbor of j.
chkpt is the lowest index that still needs to be checked.
pmax holds the maximum value of subsequences $H[l], \ldots, H[r]$ as:
$pmax[idx] = \max\{H[idx], H[idx+1], \ldots, H[r-1], H[r]\}$.

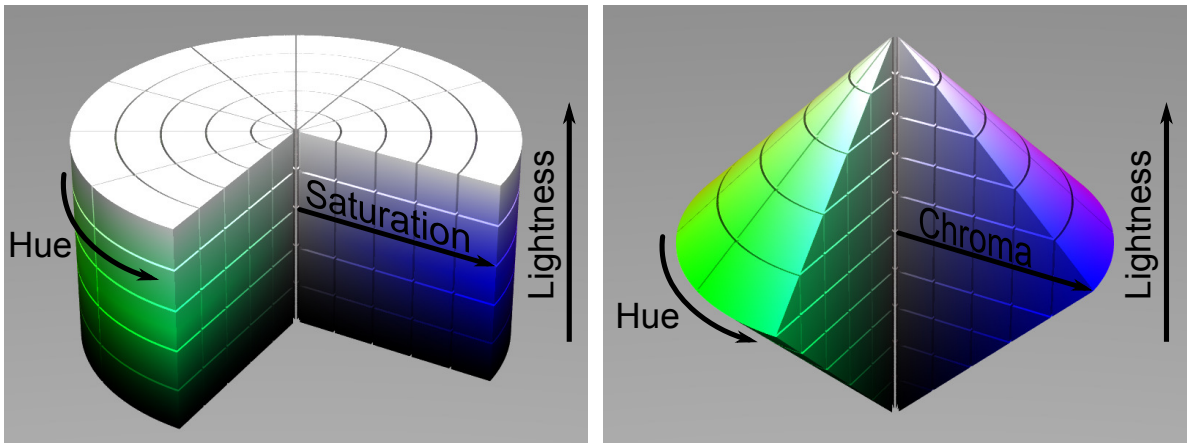
```
i ← ComputePartialMax(0, n − 1);
chkpt ← 0;
pmax[] ← ∅;
maxima[] ← ∅;
while i < L − 2n do
    j ← ComputePartialMax(i, i + n);
    k ← ComputePartialMax(i + n + 1, j + n);
    if i = j or H[j] > H[k] then
        if (chkpt ≤ j − n or H[j] ≥ pmax[chkpt]) and (i = j − n or H[j] ≥ pmax[j − n]) then
            maxima.append(j);
        if i < j then
            chkpt ← i + n + 1;
        i ← j + n + 1;
    else
        i ← k;
        chkpt ← j + n + 1;
        while i < L − 1 do
            j ← ComputePartialMax(min(chkpt, L − 1), min(i + n, L − 1));
            if H[i] > H[j] then
                maxima.append(i);
                i ← i + n + 1;
                break;
            else
                chkpt ← i + n + 1;
                i ← j;
        end
    end
end
return maxima;

ComputePartialMax(from, to)
    pmax[to] ← H[to];
    best ← to;
    while to > from do
        to ← to − 1;
        if H[to] ≤ H[best] then
            pmax[to] ← H[best];
        else
            pmax[to] ← H[to];
            best ← to;
    end
    return best;
```

1D ($2n+1$) neighborhood NMS algorithm developed by Neubeck and van Gool [NG06] is applied (Algorithm 3.1). The modification introduced makes the algorithm consider maxima close to the image borders.

Before concluding this section, it is important to remark the main assets and limitations this type of feature presents. On the side of the advantages, apart from the fact that in indoor environments vertical edges are present almost everywhere (e.g., corners, door frames, shelves...), one should not forget to mention *rotational invariance*. Assuming the aforesaid camera position constraints are met, no matter the orientation the robot has, vertical edges always appear the same, exception made of occlusions as, under certain viewpoints, they disappear behind other objects. A drawback derives from the usage of edges as segmentation lines, since a minimum distance between edges is forced in order to obtain relevant information in each of the intervals. Hence, as the robot moves closer to a group of edges, their distances in the image increase and additional vertical edges are likely to be identified.

3.2.2. Color histograms

It is undeniable that color is a very informative visual source of information. However, capturing color with a camera is more challenging than it seems at first sight because it is very sensitive to illumination changes. For this reason, a color space like HSV or HSL which is supposed to split the color (hue) from the brightness information appears to be a reasonable starting point for color extraction [Lam+01; TC92; UN00]. Between the two aforementioned color spaces, HSL (Figure 3.5a) has been chosen over HSV because it is symmetrical to lightness and darkness [Bol+09].



(a) **(b)**

Figure 3.5. Hue-Saturation-Lightness or HSL (a) and Hue-Chroma-Lightness or HCL (b) color models. (*Source:* Figure created using the ShapeGrid macro by Michael Horvath [Hor08]).

In spite of the color component being separated from saturation and lightness in the HSL color space, these components should not be overlooked in color extraction as hue is meaningless if colors are too bright, dark, or desaturated—they appear to be white, black, or gray, respectively—, and can consequently lead to misclassification if they are not taken into account. Following this reasoning, color can be separated into *chromatic* and *achromatic* regions depending on the saturation and luminance components. Tseng and Chang [TC92] put forward different thresholds to define the achromatic area. Very dark and bright pixels are filtered out using only lightness to discriminate, whereas saturation is also used for mid-range

intensity values. However, the values proposed have been proven excessively restrictive and the fact that the color information is richer near the mid-plane of the HSL cylinder, as shown in Figure 3.5a, forces to apply different saturation thresholds depending on the value of the lightness channel.

An alternative and more intuitive approach is to employ the Hue-Chroma-Lightness (HCL) bicone model instead (Figure 3.5b), where the saturation component is replaced by a combination of lightness and saturation known as *chroma*. In this case, it is sufficient to remove the central cylinder (using the chroma channel) to tell the color information apart. The pixels whose chroma values are lower or equal to 12.5% are classified as gray.

Moving to the actual algorithm (Figure 3.6), for each of the subimages defined by the vertical edges extracted in Section 3.2.1, chromatic pixels are first identified, using the chroma threshold proposed above, with the aim of building eight-bin hue histograms, as eight distinct values have proved to be enough in [Tap05] to encode color information. Nonetheless, building histograms directly, adding up the pixels that correspond to each bin, has an important drawback. It would not be uncommon to find locations where many pixels lie along the border between two adjacent bins and that in two consecutive images those pixels fall into different bins, giving rise to completely dissimilar histograms.

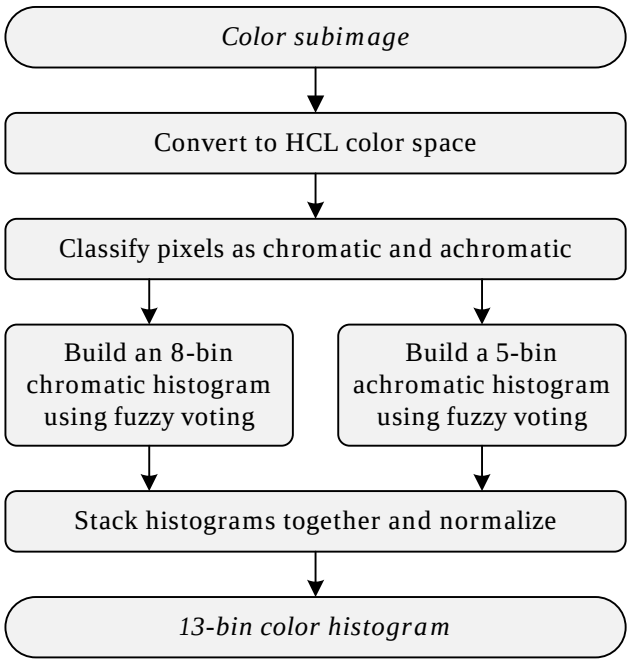


Figure 3.6. Color histogram extraction flowchart.

In order to address this issue, a *fuzzy voting scheme* with triangular membership functions, inspired by the one first proposed in Lamon *et al.* [Lam+01], has been implemented (Figure 3.7). Rather than counting for a single bin, each pixel belongs to two histogram columns simultaneously, in an inversely proportional manner to its distance to both bin centers. Note that as the hue component is circular, the histogram ends are contiguous. To perform this computation, a histogram with a large number of bins is obtained first (e.g., 256 columns, which match the number of quantization levels that can be obtained with 8 bits). The value of every bin in the chromatic histogram is obtained by performing the scalar product of the pixels involved (e.g., 64 bins in the case of 256 hue divisions and a desired eight-bin output histogram) with the triangular mask.

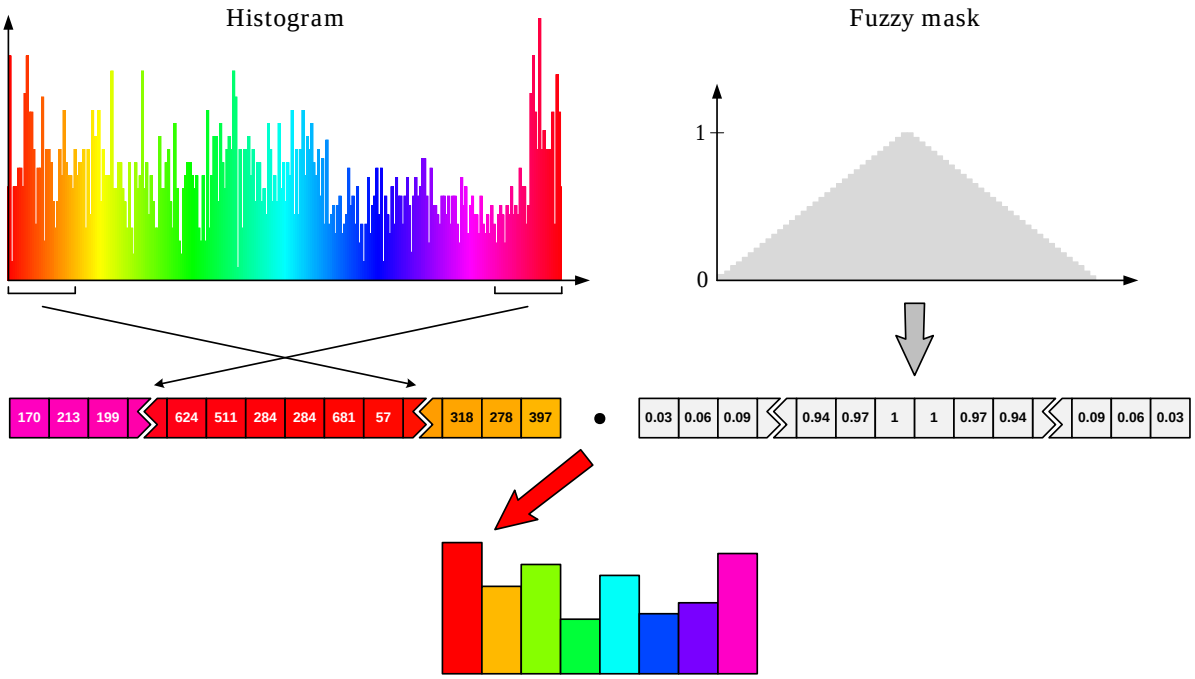


Figure 3.7. Example of the fuzzy voting process to compute the red bin of the chromatic histogram. First, the columns involved are picked from the large histogram. In the case of red, the order has to be reversed so consecutive hue values appear together. The scalar product with a triangular fuzzy mask is then computed to obtain the unnormalized value of the red bin. This process is repeated for each color.

However, there exist locations where white, gray, black are the prevailing colors, and therefore the hue component is not sufficiently meaningful. For this reason, the pixels from the achromatic region are used to obtain a five-bin histogram to accommodate different tones of gray (i.e., white, light gray, medium gray, dark gray, and black) following a procedure that is analogous to the one applied to the chromatic area except for the fact that low and high values are not mixed when performing the voting. If due to the illumination conditions of the environment the images are prone to over- or under-exposure, the apexes of the HCL bicone can be removed in order to mitigate this issue. The trade-off for this decision is that pure black and white would no longer be identified.

Finally, both the eight-bin color and the five-bin grayscale histograms are stacked together and then normalized to obtain a single 13-bin histogram.

3.2.3. Keypoints

Given that keypoints have been proven effective for rather difficult object recognition tasks, it seems appropriate to include this type of visual cue in the fingerprint. There exist many different alternatives and the choice of the keypoint detector and descriptor depends on many factors like the type of environment, the computing power available, and the size of the images analyzed, among others. In Section 3.4 the performance of the proposed method is analyzed for three qualitatively distinct keypoints: SIFT [Low04], which is robust but slow; Star features—a variant of CenSurE [Agr+08]—described with upright SURF [Bay+08], which are faster but less robust; and ORB [Rub+11] that is much faster due to the fact that it employs the high-speed detector FAST [RD06b; Ros+10] and a less distinctive binary descriptor like BRIEF [Cal+10].

In any case, no matter which type of keypoints is chosen, no more than 100 robust keypoints, distributed among the different subimages, are extracted in total. These keypoints are obtained by iteratively adjusting the response threshold. However, if for any reason more points are

still present after a reduced number of iterations, they are all preserved. Note that this is a totally opposite approach to that of the image recognition field, where several hundreds or even thousands of features are extracted from each image in order to be able to perform matching with relatively low uncertainty.

3.2.4. Final descriptor

The resulting fingerprint, schematically depicted in Figure 3.8, consists of two sorted sets of $n+1$ elements, where n is the number of structural vertical edges identified in the image. One set contains 13-bin histograms (eight bins represent color and the other five model grayscale values) whereas the n elements of the second set are collections of keypoints extracted from each of the subimages defined by the vertical lines. The steps required to build the fingerprint are summarized in Algorithm 3.2.

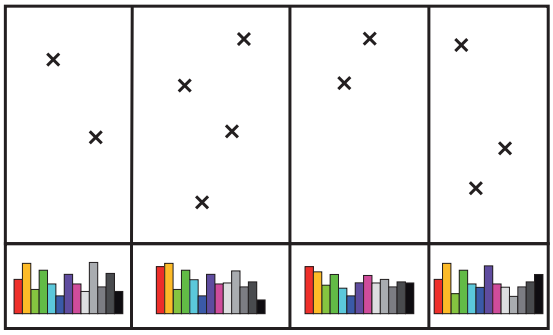


Figure 3.8. Final VPACK descriptor. From every region defined by structural vertical edges, keypoints and color histograms (which include both chromatic and achromatic pixels) are obtained.

Algorithm 3.2. Fingerprint generation

Identify structural vertical edges to split the capture in subimages (Section 3.2.1).

1: Compute the Sobel operator over the x direction.
2: Perform a 3 by 1 morphological dilation.
3: Apply a vertical morphological opening of size 1/5 of the image height.
4: Sum each column and compute a normalized graph of the vertical response.
5: Filter out those values lower than 5 times the mean.
6: Apply a NMS algorithm to ensure a minimum separation between lines (e.g., 5% of the image width).

foreach subimage **do**

Compute a 13-bin histogram that encodes color and grayscale information (Section 3.2.2).

1: Using the HCL bicone model, classify pixels as chromatic (chroma $>$ 12.5%) or achromatic (chroma $\leq$ 12.5%).
2: Build an 8-bin color histogram and a 5-bin grayscale histogram using a fuzzy voting scheme.
3: Stack both histograms together and normalize.

Extract and describe a reduced amount of robust keypoints (Section 3.2.3).

end

3.3. *n*-gram matching

Once the features of an image have been extracted and described, they have to be compared with those previously gathered in order to evaluate their similarity and decide on the current location. This section introduces a matching procedure based on n-grams, with a view to taking adjacency between features into account. The term n-gram is employed in natural language

processing (NLP) to refer to a subsequence of *n* consecutive elements (i.e., letters or words) within a larger sequence [JM09]. When *n* is equal to one, they are called "unigrams"; for *n* equal to two, "bigrams"; and three-element grams are referred to as "trigrams". For instance, from a letter perspective, in the famous sentence by Isaac Asimov known as the First Law of Robotics [Asi42], the unigram *m* appears five times; the bigram *ma*, three times; and the trigram *may*, only once.

> A robot may not injure a human being or, through inaction, allow a human being to come to harm.

However, this concept can be easily extended to different contexts. In this chapter, each of the subimages defined by the structural vertical edges extracted in Section 3.2.1 is considered a "letter". As usual in NLP, only *n*-grams of up to 3 items are evaluated.

3.3.1. From features to *n*-grams

In order to be able to apply the *n*-gram framework, corresponding "letters" between the query and the reference images must be first identified. To begin with, individual keypoints and histograms are matched.

The similarity between two *histograms* is determined using the normalized version (i.e., the range of possible values is scaled to the [0,1] interval) of the commonly used *Hellinger distance* (3.3) [Cha07; LeC86], which has been chosen because it satisfies the metric axioms[1] (i.e., non-negativity, reflexivity, symmetry, and triangle inequality) [Com+03]. This metric is sometimes confused in the literature with the *Bhattacharyya distance* because both make use of the *Bhattacharyya coefficient* (3.4)—also known as *Hellinger affinity*—, which is nothing more than the sum of the geometric means of each i-bin pair. The actual Bhattacharyya distance is similar but violates the triangle inequality property [Kai67]. For every histogram in the reference image R, only the best match in the query image Q is kept as long as its normalized Hellinger distance is no larger than 0.3. This threshold has been experimentally adjusted.

$$d_H(Q,R) = \sqrt{1-\rho(Q,R)} \tag{3.3}$$

$$\rho(Q,R) = \sum_{i=1}^{N} \sqrt{Q_i \cdot R_i}\ , \text{ where } \sum_{i=1}^{N} Q_i = \sum_{i=1}^{N} R_i = 1 \tag{3.4}$$

By contrast, the best match for each query *keypoint* is retrieved from the keypoints of all the reference images using the Euclidean distance, which is the one generally used in the literature for these type of features, except for the case of binary descriptors (like the ones in ORB), where the Hamming distance [Ham50] is employed instead.

Once individual corresponding features have been identified, the *matching score* between "letters" can be computed in the following manner. Recall that each "letter" is represented by a single histogram and several keypoints. For histograms, the matching score is calculated as $1 - d_H(Q,R)$, whereas for keypoints it is the ratio between the number of keypoints matched and the total number of keypoints in the reference "letter".

[1] Non-negativity: $d(x,y) \geq 0$
Reflexivity: $d(x,y) = 0 \Leftrightarrow x = y$
Symmetry: $d(x,y) = d(y,x)$
Triangle inequality: $d(x,z) \leq d(x,y) + d(y,z)$

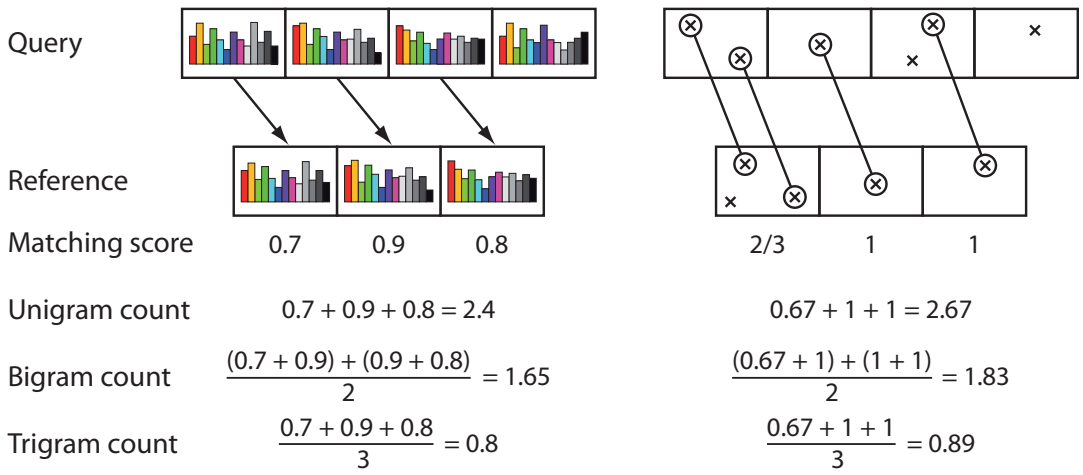


Figure 3.9. Worked example of the *n*-gram counts computation. The image depicts a query and a reference fingerprint. Lines and arrows indicate corresponding histograms and keypoints. The matching score for individual subimages is 1 minus the Hellinger distance for histograms and the number of matched keypoints over the total number of keypoints in the reference subimage. Three corresponding unigrams, two bigrams, and one trigram have been identified for both types of features. According to the numbers of the example, this means, for instance, that 1.65 effective histogram bigrams out of two possible have been matched.

Finally, the number of unigrams, bigrams, and trigrams that have been matched are computed for each type of feature separately (i.e., three *n*-gram counts for histograms and three for keypoints are obtained). This implies looking for *n* consecutive subimages both in the query and representative images that correspond to each other. Two subimages are considered to match if they have at least one feature in common. *n*-gram counts are then computed as the sum of the average matching scores of its "letters". See Figure 3.9 for an example.

3.3.2. Matching algorithm

With the identified *n*-grams, unigram, bigram, and trigram models are built separately for color histograms and keypoints. The process, which is analogous for both types of features, is explained below (Figure 3.10).

For each of the *n*-gram models, the probability of being at a previously visited location is estimated employing a naive Bayes approach with Laplace smoothing [RN10], to avoid having zero probabilities that are problematic in the case of error. Let $n \in \{1, \dots, N\}$ be the different models, L_k denote one of the K encoded locations and O stand for the features in the query image. The probability of being at location L_k given O for *n*-gram model n can be therefore computed as

$$p_n\left(L_k|O\right) = \frac{x_{n,k} + \alpha_n}{\sum_{k=1}^{K} x_{n,k} + \alpha_n \cdot K}, \qquad \forall n \tag{3.5}$$

where $x_{n,k}$ is the number of grams of size n that have been matched in L_k (i.e., unigram, bigram, and trigram counts obtained in Section 3.3.1) and $\alpha_n > 0$ is the smoothing parameter, which can be regarded as a measure of the confidence in the observations; the more observations, the less it affects the probability. If an add-one smoothing is chosen (i.e., $\alpha_n = 1$), one assumes that every seen or unseen event occurred once more than it did in the data and, as a consequence, it moves a lot of probability mass from seen to unseen events. In practice, much lower values are used, mainly to prevent zero probabilities. For this particular application, $\alpha_n = 0.01\ \forall n$ was

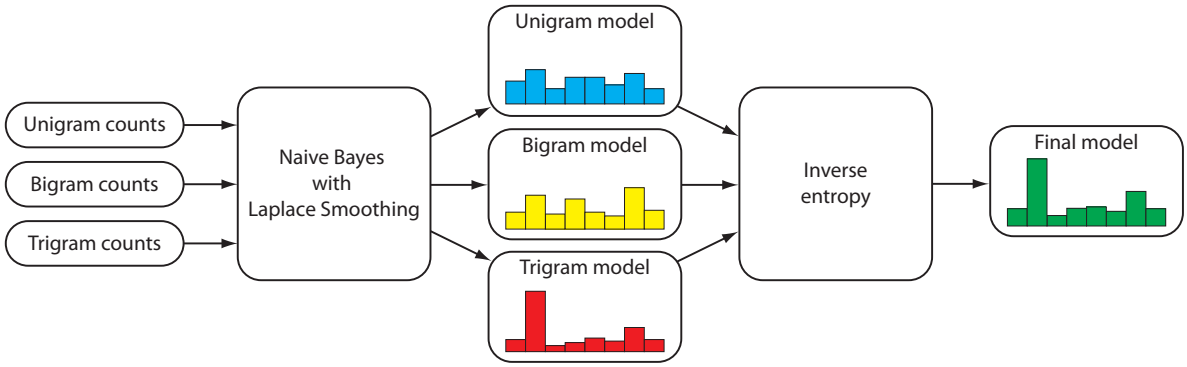


Figure 3.10. *n*-gram matching steps. The unigram, bigram, and trigram counts are employed to build three discrete probability distributions for the current location using Naive Bayes with Laplace smoothing. For localization, these distributions are then combined and weighed using inverse entropy, which gives more importance to the one that has more information (i.e., is more confident about its prediction) to obtain the probabilities of being in each encoded location. As can be observed, the green output model, which has been built using the values of the *n*-gram models depicted, resembles the red trigram model, but takes into consideration the contribution of the yellow bigram model to the penultimate bin.

selected. Nonetheless, the value of α_n need not necessarily be the same for low- and high-order models (e.g., unigram and trigram models respectively). The algorithm is not very sensitive to these parameters as long as α_n is much smaller than one.

After this process, there are three different discrete probability distributions for the current location that have to be combined somehow. Two distinct approaches exist in NLP: interpolation and back-off. The main difference between them is that the former considers all models whereas the latter relies solely on the most complex (trigrams in this case) and only 'backs-off' to lower-order *n*-grams if there is no evidence in the higher-order model (i.e., no trigrams have been observed) [JM09]. For the particular task of scene recognition, back-off is too optimistic as, contrary to NLP, observations are noisy and one cannot blindly trust the most complex model. For this reason, an interpolation approach has been chosen.

The problem that arises is how to determine the weighting coefficients of each of the models in the absence of training data to adjust them. If VPACK is to be used for localization, the solution that can be adopted is to estimate the coefficients every time, using the concept of inverse entropy [MD+07]. In information theory, *entropy* is a measurement of the uncertainty of a random variable and its value is larger the less peaked (more uncertain) the probability distribution is. As the inverse entropy works in the opposite manner, larger weights are assigned to those models that are more confident about the current location. By contrast, if only the similarity between two images needs to be determined, the weights have to be manually set.

The entropy H_n of a probability distribution is defined as

$$H_n = \sum_{k=1}^{K} -p_n\left(L_k|O\right) \cdot \log\left(p_n\left(L_k|O\right)\right), \qquad \forall n \tag{3.6}$$

The weight ω_n for each model can then be computed with the following expression

$$\omega_n = \frac{\frac{1}{H_n}}{\sum_{n=1}^{N} \frac{1}{H_n}}, \qquad \forall n \tag{3.7}$$

The resulting probability distribution is calculated as

$$p\left(L_k|O\right) = \sum_{n=1}^{N} \omega_n \cdot p_n\left(L_k|O\right), \qquad \forall k \tag{3.8}$$

This interpolation procedure based on inverse entropy helps to deal with those situations where high-order *n*-grams do not exist. For instance, if no trigrams are found for any reference image, either because the query image is not similar enough or because the reference images cannot be segmented in at least three subimages, the resulting trigram model will be a discrete uniform distribution. As it is the maximum entropy probability distribution, its weight will be small compared to other lower-order models, assuming that they have enough information.

The aforementioned algorithm is computed twice in order to obtain two different probability distributions, one according to the color histograms and another to the keypoints. These features can be considered independent, as histograms are global features—or semi-global in this case—while keypoints are local, so both models are simply multiplied and normalized to compute the final probabilities. As a result, if both distributions agree on the current location, they reinforce each other and provide a significantly higher matching probability. To sum up, the steps taken to perform feature matching are succinctly put forward in Algorithm 3.3.

Algorithm 3.3. *n*-gram matching procedure

for histograms and keypoints independently **do**
Obtain matching *n*-grams in the reference images used to represent different locations.
The procedure for histograms and keypoints is summarized below in steps 1 through 3:

if histograms **then**
foreach representative image **do**
1: For every histogram in the representative image, find the best match in the query image using the Hellinger distance (3.3). Keep those that satisfy $d_H(Q,R) \leq 0.3$.
2: Compute the score for each match as $1 - d_H(Q,R)$.
end
else if keypoints **then**
1: Find the best match for each keypoint in the query image among the keypoints of all representative images using the Euclidean distance (or the Hamming distance for binary descriptors). A threshold on the distance can be optionally applied.
2: Matching score $= \dfrac{\#\text{ matched keypoints}}{\#\text{ keypoints in the reference subimage}}$
end

3: Compute *n*-gram counts for each location $x_{n,k}$ using the matching scores (Figure 3.9).

Build *n*-gram models using naive Bayes with Laplace smoothing from the *n*-gram counts (3.5).
$\Rightarrow$ **Output:** *n* histogram and *n* keypoint models (for unigrams, bigrams, and trigrams if $n = 3$).

Employ inverse entropy to combine the *n*-gram models:
1: Compute the entropy of every *n*-gram model (3.6).
2: Calculate the weighting factor of each model using inverse entropy (3.7) .
3: Obtain the weighted sum of the three models (3.8).
$\Rightarrow$ **Output:** One histogram and one keypoint model.
end

Multiply the histogram and keypoint probability distributions to obtain the final probabilities.

3.4. Results and discussion

The proposed fingerprint and the matching algorithm, programmed in C++ using the OpenCV library [Bra00], have been tested in two qualitatively different environments: an office and a house. The former corresponds to the publicly available KTH-IDOL2 database [Luo+07], whereas the latter was custom made. See Appendix A for more details on the datasets.

In both experiments, the environment was first modeled by manually selecting a few representative images, taken at different coordinates and with different orientations, from each of the different a priori defined locations. For instance, in Figure 3.13 the kitchen is encoded using five captures. These images were acquired from different datasets than those used for testing.

Every query image was then compared to all of the representative images but, as the robot can only sense one image at a time, only the representative that provided the highest matching score for each location was considered to compute the probabilities. As mentioned in Section 3.2.3, SIFT [Low04], Star features [Agr+08] described with upright SURF [Bay+08], and ORB [Rub+11] have been tested as keypoint detectors and descriptors.

3.4.1. Office environment: KTH-IDOL2 database

In the KTH-IDOL2 database, there exist four different image collections for night illumination conditions acquired using the PowerBot Dumbo robot. They will be referred to as *Dumbo night 1* to *Dumbo night 4* from now on. The four datasets are different from each other because there are people walking around, and objects being used and moved. From *Dumbo night 1*, twenty

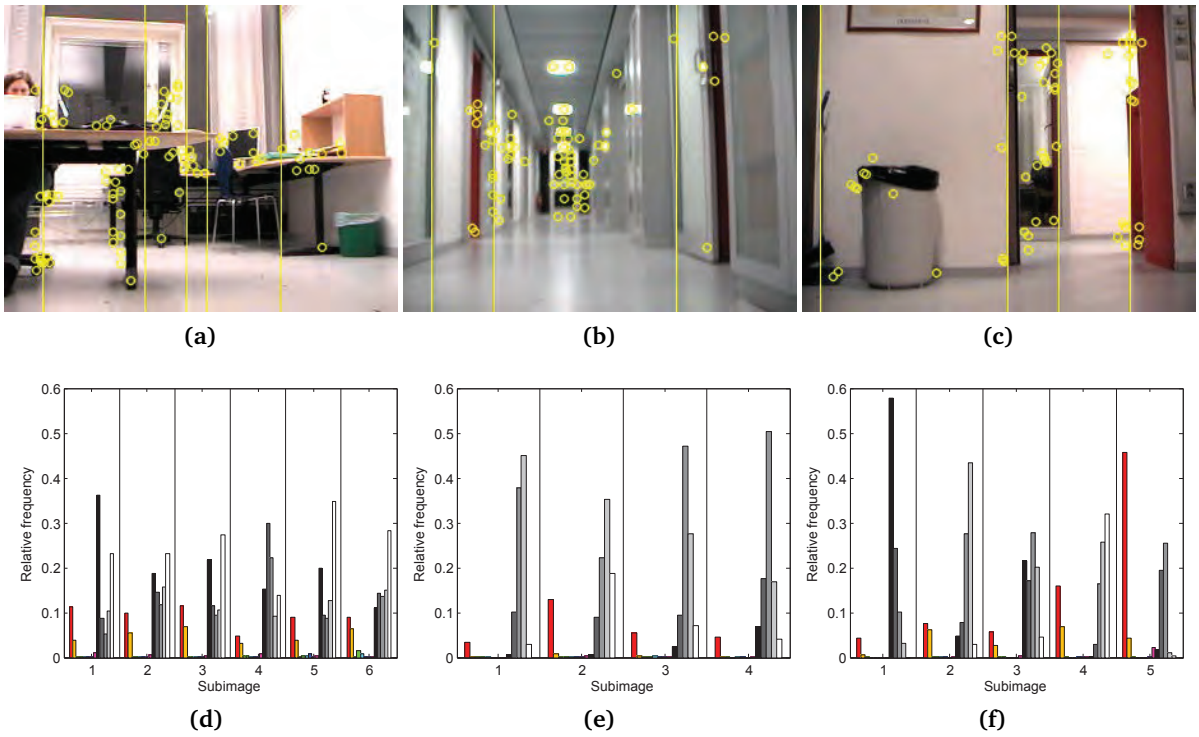


Figure 3.11. Sample representative images of the office for two people (a), the corridor (b), and the kitchen (c) extracted from the *Dumbo night 1* dataset with the identified vertical lines and Star keypoints superimposed. The corresponding color histograms (d)–(f) are presented underneath. In each of the histograms, the first eight bins represent the color components, whereas the last five are grayscale values.

two reference images were selected to model the five locations present in the environment: printer area, corridor, room for two people, room for one person, and kitchen. Some sample images are shown in Figure 3.11. *Dumbo night 2* and *Dumbo night 3* were used for testing. The *Dumbo night 2* dataset consists of 952 images and was acquired the same day as *Dumbo night 1*, whereas the 1034 images of *Dumbo night 3* were recorded four months later. As a consequence, there are noticeable differences between the latter and the reference images.

Table 3.1 presents the precision (i.e., percentage of correctly classified instances in the retrieved results) for each type of keypoint with different probability thresholds. The threshold is defined as the minimum probability a predicted location must have to be considered valid. For instance, in the case of the first row of *Dumbo night 2* using SIFT, this means that 86.28% of the 736 query images whose predicted location had a probability equal to or higher than 0.5 were correctly assigned to the location (corridor, kitchen...) they belong to. According to these results, which might be slightly inaccurate because it is difficult to define the ground-truth border between locations, it seems clear that the method performs well for all types of keypoints and that it is relatively robust to changes that may occur in the environment over time, especially if a high probability threshold is chosen (e.g., above 0.7). Nevertheless, SIFT is mildly ahead of Star and ORB, and correctly identifies more images within the sequence.

Table 3.1. Results for *Dumbo night 2* and *Dumbo night 3* datasets for the VPACK descriptor with different types of keypoints and thresholds. Both the number of images that are above the threshold and the precision are shown.

	DUMBO NIGHT 2		
	Threshold	Images	Precision
SIFT	0.5	736	86.28%
	0.6	626	91.85%
	0.7	538	93.87%
	0.8	435	96.55%
	0.9	327	99.08%
Star	0.5	707	81.33%
	0.6	578	86.33%
	0.7	452	89.82%
	0.8	326	93.56%
	0.9	206	97.09%
ORB	0.5	656	74.70%
	0.6	532	80.08%
	0.7	429	86.01%
	0.8	345	93.04%
	0.9	239	97.91%

	DUMBO NIGHT 3		
	Threshold	Images	Precision
SIFT	0.5	768	74.87%
	0.6	616	80.52%
	0.7	492	86.18%
	0.8	332	93.98%
	0.9	197	97.93%
Star	0.5	753	68.13%
	0.6	584	72.60%
	0.7	416	79.09%
	0.8	271	86.35%
	0.9	145	93.79%
ORB	0.5	692	65.03%
	0.6	526	69.58%
	0.7	373	79.09%
	0.8	235	86.38%
	0.9	139	93.84%

In order to present the results in a more visual and intuitive manner, Figure 3.12 shows an example of the most likely state prediction for each capture in *Dumbo night 2* using SIFT and Star features with a probability threshold of 0.5. Overall, the different locations are correctly determined, and the majority of the mistakes (identified as dots with a different color than its neighbors) occur near unsure locations like door openings. Some other errors are due to an incorrect choice of the representative images that do not cover the environment perfectly and to temporary dynamics like the cyan area that is shown between the printer area and the corridor, which is caused by a moving person.

In addition, a detailed analysis of the predictions obtained for *Dumbo night 2* with a threshold of 0.8 is shown in Figure 3.13. The sequence of visited places can be distinguished no matter which keypoint descriptor is used. The robot starts in the printer area, moves into

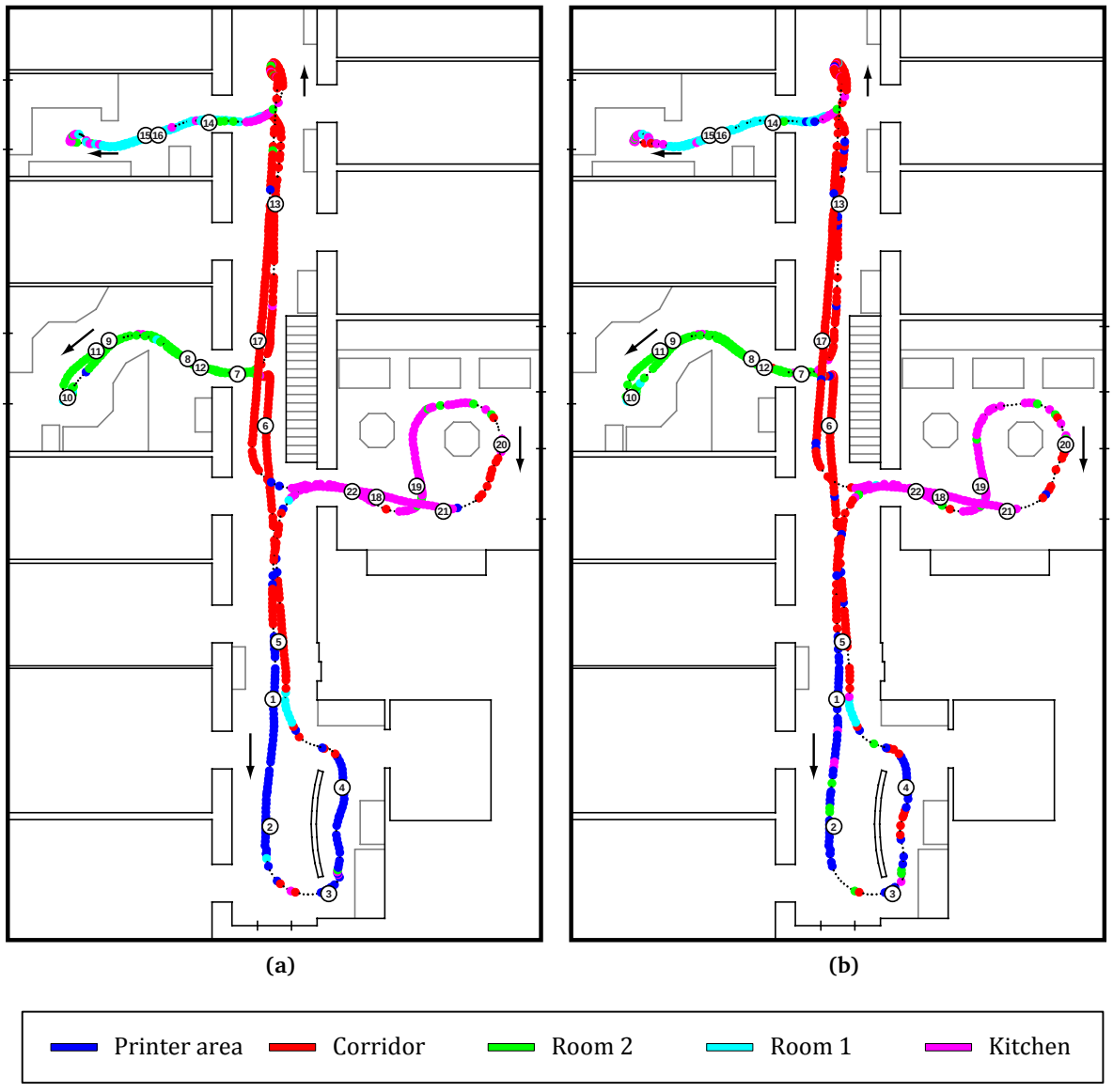


Figure 3.12. Place classification results using SIFT (a) and Star described with upright SURF (b) in *Dumbo night 2* with a threshold of 0.5. The circled numbers depict the approximate position where the reference images were acquired, as well as the order in which they were recorded. The black dots indicate locations the robot is unsure about for this threshold.

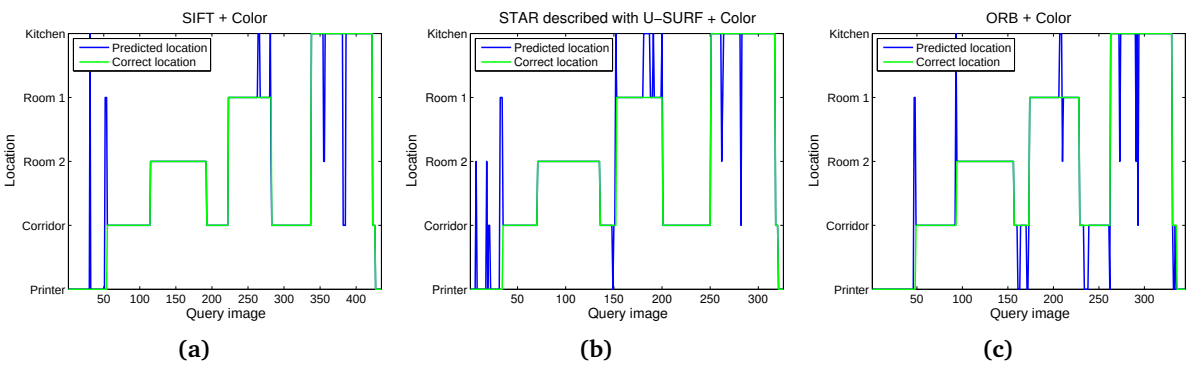


Figure 3.13. Comparison of the predicted and the ground-truth location with a threshold of 0.8 using SIFT (a), Star features described with upright SURF (b), and ORB (c) in *Dumbo night 2*.

the corridor, explores the room for two people, goes back to the corridor, enters the room for a single person, returns to the corridor, goes into the kitchen and ends in the printer area after going through the corridor briefly. Note that most of the errors occur near the transition areas (steps of the green line) which, as has been mentioned, are difficult to encode correctly. Some of them could be easily removed using a mode filter.

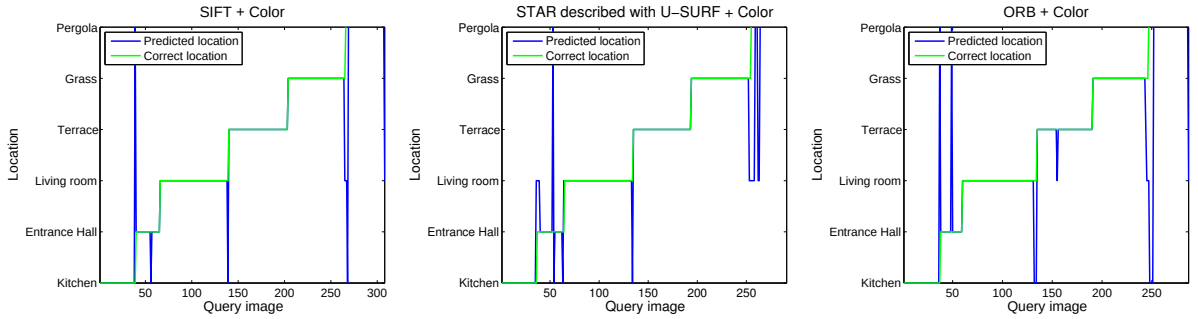
3.4.2. Home environment

For the home environment test, twelve images were used to represent six locations: kitchen, entrance hall, living room, terrace, grass area, and pergola. The results for the test dataset, which contains 373 captures, are shown in Table 3.2. In this case, the percentage of correctly classified instances is even better than in the KTH-IDOL2 database and, as can be observed in Figure 3.14, the errors concentrate, once again, near the transitions.

Table 3.2. Localization results for the home environment dataset with different probability thresholds for SIFT, Star described with U-SURF, and ORB keypoints; and with color histograms only.

	HOME ENVIRONMENT		
	Threshold	Images	Precision
SIFT	0.5	341	94.13%
	0.6	328	96.34%
	0.7	308	97.40%
	0.8	290	98.62%
	0.9	255	99.61%
Star	0.5	343	89.50%
	0.6	316	92.72%
	0.7	291	94.16%
	0.8	267	95.51%
	0.9	271	98.62%
ORB	0.5	335	89.25%
	0.6	312	91.99%
	0.7	287	94.77%
	0.8	264	96.21%
	0.9	209	99.52%

	HOME ENV. WITHOUT KEYPOINTS		
	Threshold	Images	Precision
Color	0.5	315	89.84%
	0.6	290	91.72%
	0.7	267	91.76%
	0.8	191	96.33%
	0.9	137	97.08%



(a) **(b)** **(c)**

Figure 3.14. Predicted versus actual location with a probability threshold of 0.7 in the *Home environment* using SIFT (a), Star described with U-SURF (b), and ORB (c).

This increase in the performance is most certainly due to the fact that, conversely to an office, where all walls tend to be of the same color and rooms normally look alike, color provides much more distinctive information in a house. In order to verify this assumption, the results of running VPACK without keypoints (i.e., extracting only color histograms between vertical edges)

have also been computed. The correctly classified instances account for similar percentages than when used in combination with keypoints, although the amount of images above the different thresholds is reduced. By contrast, if the same test is conducted in the office environment, the outcome is poor because white and gray are the prevailing colors everywhere. All these results suggest that keypoints and color histograms reinforce each other indeed.

3.4.3. Computing times

In general, the performance using SIFT seems to be somewhat better than with Star and ORB, but its main drawback is its high computational burden, which can be a problem for robots running on low-specification hardware. The average computation times for feature extraction and matching with all the representative images in the two environments tested, carried out on a 2nd generation Intel® Core™ i5 CPU at 1.6 GHz, are presented in Table 3.3. The significant difference between them is explained by the image resolutions; whereas in KTH-IDOL2 is of 309x240 pixels, in the home environment it rises to 640x480. If there is not much computational power available, either Star or ORB are fairly good alternatives to make VPACK lighter. Furthermore, in line with the classification results, using relatively small images could be another alternative to keep computing time under control.

Table 3.3. Average computing times for VPACK on both datasets.

	COMPUTATION TIMES	
	Office	Home
SIFT	399 ms	792 ms
Star	174 ms	217 ms
ORB	70 ms	134 ms

3.5. Conclusion

The vision-only fingerprint based on vertical edges, color histograms, and a few robust keypoints, along with the *n*-gram based matching algorithm presented in this chapter have been proven effective for topological scene recognition in structured indoor environments. The results of the experiments suggest that the method is fairly robust to small changes that may occur in the environment over time.

The combination of complementary features that operate on different scales permits to employ weaker and faster descriptors in order to keep computing times under control without compromising precision to a great extent. Feature extraction speed could be significantly improved if the algorithm is parallelized by taking advantage of the fact that vertical edges split the image into independent regions. At the same time, this segmentation property enables to order the features identified based on the appearance of the scene rather than by means of an arbitrary grid like in the spatial pyramid framework. Even though it has been designed for and tested with monocular images, this method is directly applicable to unwrapped panoramas as is. Moreover, the matching framework allows to extract almost any other type of feature from the subimages.

On the side of the disadvantages stands the fact that the resulting descriptor is of variable length—it depends on the number of vertical edges identified—, which can pose some difficulties on later steps of the topological SLAM implementation, as will be seen, and that matching keypoints individually, even if they are not many as in this case, can become computationally

expensive in the long run. In order to make the matching process easier and faster, an online bag-of-words method similar to the one proposed by Filliat [Fil07] could be implemented for the keypoint features. However, other incremental clustering algorithms that are less dependent on data noise and on the feature extraction and processing order, like the ones based on Growing Neural Gas (GNG) [GR+12; Bou+13], could be explored. In addition, even though effort has been put on mitigating the impact of illumination changes, when large variations with respect to the representative images exist, color is no longer informative. A possible solution would be to keep two or three representations of the environment (for artificial, and low- and high-intensity natural illumination). Finally, as has been seen, a way of automatically determining where to acquire representative images that really provide information about the environment is also required. This last issue is tackled in Chapter 4.

4

Segmentation of Topological Places

A place for everything,
everything in its place.
Benjamin Franklin (1706–1790)

After perceiving its surroundings, in standard topological SLAM approaches a robot needs to automatically segment the environment into meaningful and distinct locations that will constitute the nodes of the topological map. This chapter presents an algorithm to extract robust places online from image sequences based on the algebraic connectivity of graphs or Fiedler value, which provides an insight into how well connected several consecutive observations are.

The main contribution of the proposed method is that it is a theoretically supported alternative to manually tuning thresholds on similarities, which is a difficult task and environment dependent. Thresholds are the usual solution for variable length descriptors like VPACK because applying more sophisticated statistical techniques is complicated. As the algorithm presented only requires non-negative similarities as input, it can accommodate any type of feature detector and matching procedure. The method has been validated in two different office environments using exclusively visual information. Two distinct types of features, a bag-of-words model built from SIFT keypoints and VPACK, are employed to demonstrate that the method can be applied to both fixed and variable length descriptors with similar results. A paper based on this chapter has been submitted for publication and, at the time of writing, is under review [BSM].

4.1. Introduction

Topological maps model the environment as a graph, i.e. a set of distinct discrete locations or nodes connected by edges that indicate adjacency. However, determining when to label a place as a node in the graph is a complicated problem that has been tackled in several manners in the literature. Most of the early solutions involved geometric properties of the

environment, and were therefore tied to specific sensing technologies (i.e., range sensors and laser scanners). For instance, Choset and Nagatani [CN01] employ range sensor measurements to build a generalized Voronoi graph (GVG) whose *meet points* are assimilated to topological places (Figure 4.1). In a 2D planar map, like the one they assume, a Voronoi meet point is a location equidistant to three obstacles, which should correspond to junctions and dead-ends. In practice, in cluttered environments many spurious nodes are identified and the sparsity of the topological map is lost. Beeson *et al.* [Bee+05] overcome this problem through graph pruning, but another issue still persists: if non-structural obstacles are moved, the detected Voronoi vertices change.

Another geometric-based approach uses *gateways* as topological nodes. According to the publication by Kortenkamp and Weymouth [KW94], a gateway is not only distinctive, but also relevant in the sense that it opens up a new area for the robot to explore. Within this framework, it is worth mentioning the work by Kuipers *et al.* [Kui+04]. They rely on range sensors to perform a mid-line navigation strategy with a view to finding places where the distance to the lateral obstacles is a local minimum close to a larger maximum. These locations mainly correspond to doorways, and corridor beginning and ending points.

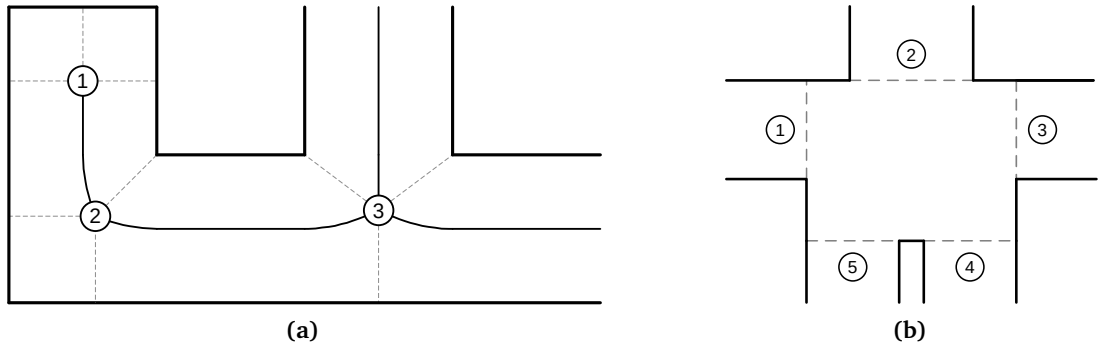


Figure 4.1. Example of Voronoi meet points (a) and gateways (b) used as topological places in corridor environments.

With the incorporation of cameras as the primary source of information, the aforementioned geometric-driven algorithms, which became no longer applicable, were replaced with thresholds on similarity measurements. Tapus and Siegwart [TS05] identify a new node whenever the similarity between the last two fingerprints of places, built from laser and visual inputs, falls below an experimentally defined threshold. Angeli *et al.* [Ang+08b] and Romero and Cazorla [RC12] follow a similar approach using only computer vision. The descriptor from the current capture is compared to that of the last identified node and, if the similarity is lower than a given threshold, it is assumed that the robot has left the previous area and arrived at another place, which can have been previously visited or not. Resolving this uncertainty is out of the scope of node extraction and, thus, of this chapter. Although sensor independent, these methods need to be tuned for each specific environment, as an incorrect choice of the threshold parameters can result in too dense or too sparse topological maps.

Rather than using similarity directly, Ranganathan and Dellaert [RD09] apply the concept of *Bayesian surprise* introduced by Itti and Baldi [IB05] to identify topological locations. With this method, a place is defined whenever a sudden or unexpected change in the environment occurs. It is based on building a Dirichlet-multinomial distribution or Multivariate Polya model of the current location and looking for a large deviation in the KL-divergence [KL51] between the prior and posterior distributions. A Dirichlet prior is often used with Bayesian approaches because

it provides higher probability to those events that have been observed frequently in the past. The method was successfully tested with both laser range scans and appearance measurements using a bag-of-words model. In a later work, Ranganathan [Ran10; Ran12] uses a Bayesian change-point detection algorithm [AM07] together with a bag-of-words model as part of a place labeling method named PLISS (Place Labeling through Image Sequence Segmentation).

Another related approach is DP-FACT [LS12; LS14], which employs a Dirichlet process mixture model to combine several types of features. In this case, two multinomial distributions are built through the discretization of the distance between consecutive vertical edges and the mean value of the U-V chrominance between these edges. The objective is to estimate the probability that the current observation belongs to any of the already encoded nodes or corresponds to a previously unvisited place. Therefore, not only node extraction but also scene recognition are simultaneously carried out. The computation of the probabilities relies on a Dirichlet prior and on the χ^2 test [Gag10], which is used to obtain the similarity between the current observations and each place's geometric and color models.

Last but not least, Chapoulie *et al.* [Cha+13] employ spherical harmonics [Bül02], which are similar to the 2D Fourier transform but defined on a spherical surface, to build two multivariate normal distributions from the data of a fixed size sliding window. With these distributions, adjusted with the first and second halves of the window, a hypothesis test based on the Neyman-Pearson lemma [Tse+06] is performed to identify transition points between dissimilar topological locations. The robot is considered to have reached a new place when the null hypothesis, which states that the parameters of both distributions are the same, has to be rejected.

As a conclusion, there exist two different approaches to sensor-independent topological location identification. One is based on building a model that allows to apply different statistical algorithms. This path has been proven effective in the literature, but it is not always possible or appropriate. For instance, whenever vision is used, the usual solution is to construct a bag-of-words model. However, for indoor environments it is not easy to find a training set to extract the vocabulary from—unless the same environment is traversed first in order to collect this training set—, and the performance of this method is thus compromised.

The second is a more general approach that uses thresholds on similarity measurements, where no prior model needs to be assumed or built, but the problem with thresholds is that they are difficult to adjust correctly. By contrast, this method is applicable even to model-based feature representations. For example, Angeli *et al.* [Ang+08b] evaluate the similarity between two bag-of-words histograms using the *tf-idf* coefficient [SZ03] within a voting scheme, and employ a threshold to decide if the robot has reached a new place.

This chapter proposes an online method based on the algebraic connectivity of graphs to address those situations in which only similarities between sensor readings are available without using thresholds directly. This scenario mostly occurs as a result of the application of fairly complex features and matching procedures with a view to reducing localization uncertainty. This is the case of VPACK (Chapter 3). Nevertheless, the method is applicable to any type of feature as long as a non-negative similarity measure can be defined. The algebraic connectivity of graphs is very closely related to spectral clustering, which has been used in robotics for topological segmentation in conjunction with range sensors [Liu+11]. In fact, the computation of the algebraic connectivity constitutes the first part of the spectral clustering algorithm.

The rest of the chapter is organized as follows. First, Section 4.2 provides a brief theoretical background on the algebraic connectivity of graphs and explains the actual topological node identification algorithm proposed. Subsequently, Section 4.3 presents the results obtained using a bag-of-words model built from standard SIFT features [Low04] and with VPACK as feature descriptors in two different office environments. Finally, the main conclusions are put forward in Section 4.4.

4.2. Node extraction using the algebraic connectivity

4.2.1. Theoretical background

The *Laplacian matrix*, the discrete analogue of the Laplace operator, exhibits a series of interesting properties in the context of graph theory that have been successfully exploited for dimensionality reduction (e.g., Laplacian eigenmaps [BN03]) and clustering applications (e.g., spectral clustering [Ng+02; vLu07]). One of these properties is related to its second-smallest eigenvalue, known as *algebraic connectivity* or Fiedler value, whose magnitude provides an insight into how well connected a graph is [Fie73; dAb07]. The eigenvector associated to this eigenvalue is referred to as Fiedler vector, and can be used to approximate the sparsest cut [Chu97; ST06] (i.e., partition a graph into two disjoint sets removing as few edges as possible [Cha08]). The steps required to compute the algebraic connectivity are put forward below and illustrated with a simple worked example.

In the case of an undirected graph, the Laplacian matrix is built from a symmetric *affinity or adjacency matrix* $A \in \mathbb{R}^{n \times n}$ that encodes the pairwise connectivity between graph nodes and a diagonal *degree matrix* $D \in \mathbb{R}^{n \times n}$ that indicates the number of edges that emanate from any given node. The elements a_{ij} and d_{ij} of these two matrices are computed as shown in (4.1).

$$a_{ij} = \begin{cases} 1 & \text{if } i \text{ and } j \text{ are linked} \\ 0 & \text{otherwise} \end{cases} \qquad d_{ij} = \begin{cases} \sum_{j=1}^{n} a_{ij} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

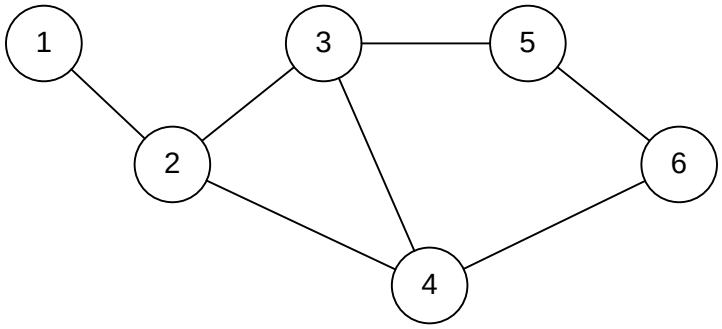


Figure 4.2. Sample graph used to explain the computation of the algebraic connectivity.

For the graph in Figure 4.2, the resulting affinity and degree matrices are:

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix} \qquad D = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{pmatrix} \tag{4.2}$$

There exist multiple definitions of the Laplacian matrix in the literature, each of which has different properties [Chu97]. The most commonly used are the unnormalized Laplacian L, the symmetric normalized Laplacian L_{sym}, and the random walk normalized Laplacian L_{rw}, which owes its name to the fact that it is the transition matrix of the standard random walk. In the equations below, I stands for the identity matrix.

$$L = D - A \tag{4.3}$$

$$L_{sym} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} A D^{-1/2} \tag{4.4}$$

$$L_{rw} = D^{-1} L = I - D^{-1} A \tag{4.5}$$

A normalized Laplacian is used throughout this chapter as suggested in [vLu07]. The symmetric Laplacian has been chosen, although similar results can be obtained with the random-walk Laplacian. L_{sym} is positive semi-definite and therefore its eigenvalues λ are always real and greater or equal to zero. In addition, this particular Laplacian matrix verifies that $\lambda \leq 2$.

The eigenvalues of the example graph using L_{sym} are $\lambda = \{0, 0.45, 1, 1, 1.71, 1.83\}$ and, consequently, the algebraic connectivity $\lambda_2 = 0.45$. The higher this value, the better connected the graph is. For instance, if an edge is added between nodes 1 and 3, the algebraic connectivity rises to $\lambda_2 = 0.51$. For this Laplacian matrix, the value of λ_2 for a fully connected graph (i.e., each node is linked to every other node) tends asymptotically to 1 as the number of nodes n grows (4.6).

$$\lambda_{2_{\max}} = \frac{n}{n-1} \tag{4.6}$$

As stated by [vLu07], the concepts and algorithms presented in this section can be extended to any arbitrary type of data if each data point is treated as a node and the connectivity between nodes is replaced by a non-negative similarity measure to indicate the strength of the link.

4.2.2. Change-point detection algorithm

The problem of recognizing topological places in image sequences is very closely related to clustering, which is one of the applications of the Laplacian matrix as mentioned. The objective is to identify when new images no longer correspond to the scene that was being observed before. At this point, a new cluster should be defined.

In order to identify these change-points, the algebraic connectivity is used. As the robot moves towards a different topological location, the algebraic connectivity tends to decrease because the latest images start to be dissimilar to the first captures acquired. In terms of graph theory, the new images (nodes) are poorly connected to the initial ones. However, one would expect that the images of the new location form a strongly connected group that makes the algebraic connectivity rise again. This increment can be used as an indicator that the robot has arrived at a different place.

Unfortunately, if the starting location is large compared to the destination, the increment can be marginal and, therefore, difficult to detect. In addition, a large location implies maintaining a big affinity matrix, which involves many pairwise comparison calculations that, depending on the similarity measure, can be computationally demanding. For these reasons, likewise Chapoulie *et al.* [Cha+13], a sliding window that only takes into account the last n images is proposed. This way, the Fiedler value will clearly increase as more images from the new place enter the window. In order to adjust this parameter, the camera's frame rate and the robot's

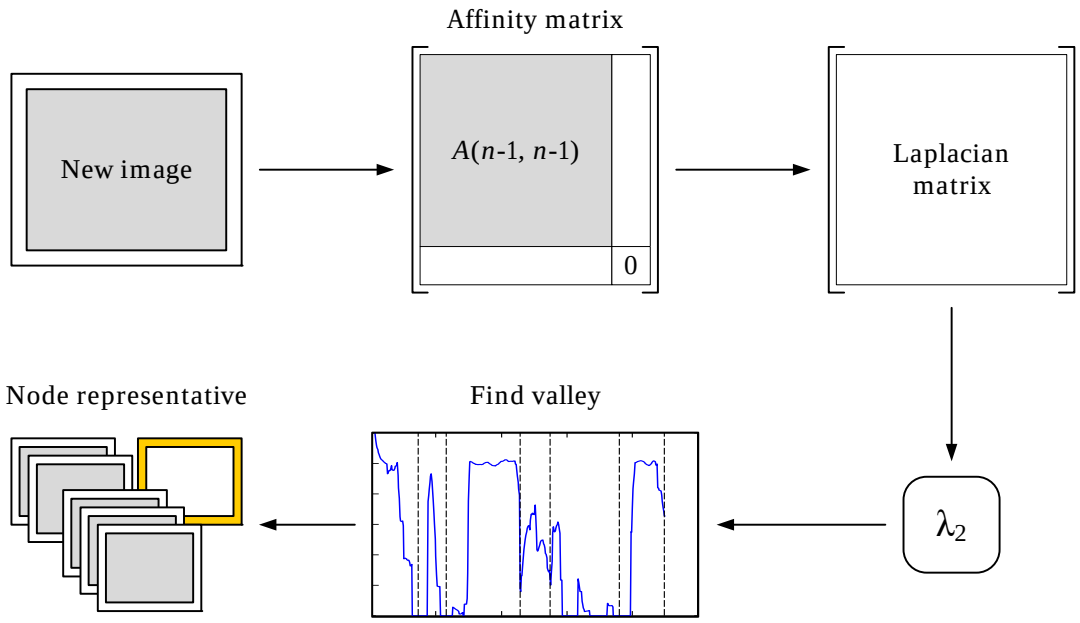


Figure 4.3. Topological node extraction procedure.

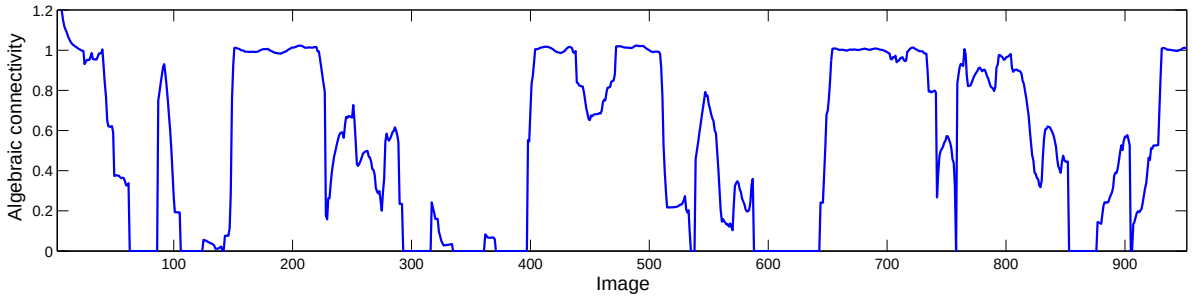


Figure 4.4. Sample time series representation of the Fiedler value obtained using a bag-of-words model.

speed must be taken into account. The lower the frame rate and the higher the speed, the smaller the sliding window should be.

The steps required to perform change-point detection for a new image are illustrated in Figure 4.3. To begin with, the affinity matrix is updated with the pairwise similarities between the current capture and the previous n-1 images. As it has to be symmetric, if the similarity measure employed does not exhibit this property, the mean of both pairwise similarities is employed. The affinity matrix is then used to compute the Laplacian matrix, from which the Fiedler value λ_2 is calculated.

If the evolution of λ_2 over time is plotted (Figure 4.4), it can be clearly seen that there exist different groups of strongly connected captures. In order to divide this time series into different clusters, several alternatives exist. At the sight of Figure 4.4, a possibility would be to look for a significant instantaneous rise in the value of λ_2 or even split it when it reaches zero. However, depending on the type of similarity measure employed, these conditions may not be met, especially the second one as will be seen in Section 4.3. Therefore, one of the many algorithms for peak and valley detection is used instead. A pseudocode is provided in Algorithm 4.1. The approach consists in finding valleys in between two peaks and segment the time series in those points. Two parameters are required to adjust the behavior of the algorithm: γ provides a lower bound for peaks (i.e., peaks below this value are ignored) and δ indicates the minimum difference in the value of the algebraic connectivity between consecutive peaks

Algorithm 4.1. Online valley detection function (Adapted from the algorithm by Eli Billauer [Bil12])

Define : $\gamma \in [0, \lambda_{2_{\max}}]$ as the minimum value for a peak to be considered and $\delta \in [0, \gamma]$ as the minimum absolute difference between consecutive peaks and valleys.
Input : The last algebraic connectivity λ_2 computed.
Output : Index of the frame where a valley was detected; 0 otherwise.

```
// Global variables
currentFrameIdx ← 1; // Incremented by the capturing module
lookForMaximum ← true;
maxValue ← −∞;
minValue ← ∞;
minIdx ← 0;

findValley(λ2)
    valleyIdx ← 0;

    // Check if the current algebraic connectivity is an extremum
    if λ2 > maxValue then
        maxValue ← λ2;
    else if λ2 < minValue then
        minValue ← λ2;
        minIdx ← currentFrameIdx;
    end

    // Look for peaks and valleys
    if lookForMaximum then
        if currentValue < maxValue- δ and maxValue ≥ γ then    // Peak found
            lookForMaximum ← false;
            minValue ← λ2;
        end
    else
        if currentValue > minValue + δ then    // Valley found
            lookForMaximum ← true;
            maxValue ← λ2;
            valleyIdx ← minIdx;
        end
    end

    return valleyIdx;
```

and valleys to declare an extremum. The first parameter allows to ignore image sequences that are not strongly connected and can therefore be easily missed in future traversals, whereas the second is used to filter out measurement noise (e.g., a person that briefly passes in front of the robot's camera). γ is restricted between 0 and the maximum value of the algebraic connectivity $\lambda_{2_{\max}}$, whereas δ ranges between 0 and γ. As both thresholds are related, attention should be paid to prevent assigning incompatible values.

4.2.3. Node representative selection

Once all the captures that form a cluster have been identified, it is necessary to choose a representative image. Conversely to some authors who use the features that correspond to the change-points to describe the topological places [Ran12; Cha+13], the image that best describes the cluster that is created with the partition (i.e., the frame that is more similar to the rest of the images that form the cluster) is used instead. This decision is supported by the work of Stankiewicz and Kalia [SK07], which states that whatever feature is used as a landmark should be persistent, salient, and informative. In other words, it must not disappear with time, it should be easy to detect, and it ought to provide meaningful evidence about the robot's location.

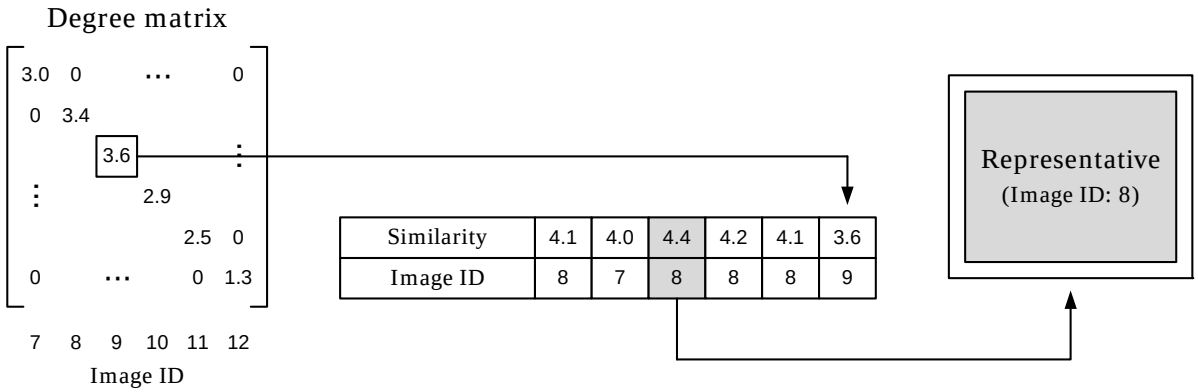


Figure 4.5. Node representative selection process. From every degree matrix, the image that is the most similar to the rest of the captures analyzed in the window is stored as a candidate. When a valley is detected, the image with the highest similarity among the candidates that belong to the cluster is chosen as representative.

However, change-points usually correspond to places with very little information (e.g., turns in a corridor are a common change-point but they all tend to look alike). If topological segmentation is carried out as part of a larger topological mapping or SLAM system, it seems preferable to assign a representative that can be identified with the least possible uncertainty and that cannot be easily missed when revisiting that place even if the frame rate is not high.

Unfortunately, selecting the frame that is most similar to the rest of the members of the cluster involves many pairwise comparisons, especially if the cluster is large. Therefore, in order to keep computational burden under control, this calculation is approximated by taking advantage of the fact that some of these comparisons are already performed to build the affinity matrix (Figure 4.5). In each iteration, the sum of the similarities of every frame considered in the affinity matrix is obtained, which are in fact the elements of the main diagonal of the degree matrix. From these, the one with the highest value, along with its total similarity, is set aside. When a representative needs to be determined, it is just a matter of choosing the frame that has the highest similarity value among the stored ones. Another alternative is to choose the representative from the window that led to the highest algebraic connectivity. The main drawback of this solution is that if one or several captures considered do not match the rest of the images because of a temporary occlusion (e.g., due to a moving entity), the value of the algebraic connectivity will go down and a good representative might be discarded.

4.3. Results and discussion

The algorithm presented above, programmed in C++ using OpenCV [Bra00] for image processing and the Armadillo library [San10] for linear algebra computations, has been tested with the publicly accessible KTH-IDOL2 [Luo+07] and COLD [PC09] image databases that correspond to different office environments. For further details on the datasets, refer to Appendix A.

4.3.1. KTH-IDOL2 database

In the KTH-IDOL2 database, two different feature extraction and description methods have been used in order to validate the proposed method. On the one hand, a bag-of-words model constructed from SIFT features [Low04], like the one employed by Angeli *et al.* [Ang+08b]. A 100 word vocabulary was built using *k*-means clustering with the center initialization technique

suggested by Arthur and Vassilvitskii [AV07] from the first dataset of KTH-IDOL2 in cloudy illumination conditions, different from the datasets taken at night that were used for testing. These last datasets, which will be referred to as *Dumbo night 1* and *Dumbo night 2* hereafter, correspond to two different traversals of the same environment following approximately the same path. The former consists of 965 images whereas the latter has 952 captures.

On the other hand, VPACK with Star keypoints (see Chapter 3) was used. However, in this case the inverse entropy algorithm proposed to obtain the weighting factor to combine unigram, bigram, and trigram models cannot be applied because it is the pairwise similarity between two images what is being computed, not the probability of being at any of the previously visited locations. For the tests presented in this chapter, the arithmetic mean was used, although any alternative weighting criterion is possible.

These two feature extraction and description methods have been chosen to conduct the experiments because they are qualitatively different. The descriptor of a bag-of-words model is of constant length (the size of the vocabulary) and thus permits to easily apply statistical techniques if desired. This is one of the reasons why they are employed so extensively. By contrast, VPACK results in variable length descriptors, depending on how many structural vertical edges are identified in each capture, which makes it more difficult to accommodate into a statistical framework. All the results presented below were obtained with a sliding window n of size 25 captures, $\gamma = 0.5$, and $\delta = 0.25$. The maximum algebraic connectivity for this window size is $\lambda_{2_{\max}} = 1.04$. The sliding window size has been adjusted for an average robot speed of 0.3 or 0.35 m/s and a frame rate of 5 fps, which translates into an image being captured every 60 or 70 cm. The length span considered is therefore approximately between 1.5 and 1.75 m, which is reasonable for a normal house or office environment where the visual appearance tends to change rather quickly. Small variations of n for a given image acquisition rate do not have a great impact on the nodes detected. The other two parameters are even more flexible: γ has been chosen to prevent asigning nodes where the connectivity is too low because these places are not robust, whereas δ was adjusted to ensure that the noise that may be appear in λ_2 when traversing a repetitive area like a corridor (see Figure 4.13, for instance) are filtered out.

Figure 4.6 depicts the locations where the representative images of each topological place identified using the proposed algorithm were captured using the bag-of-words model in the two datasets considered. For completeness, the Fiedler values for *Dumbo night 2* with the clusters identified is also provided in Figure 4.7. Most of the detected topological places are consistent in the two traversals, especially when the robot moves forward for a relatively long time, and they usually correspond to corridors and room entrances and exits, although in large rooms additional nodes are sometimes found. Nonetheless, occasionally supplementary nodes are introduced (e.g., node 10 in Figure 4.6a and node 13 in Figure 4.6b). This may be due to two facts: either the variation in the path followed made a difference in terms of the images acquired or, most probably, the robot moved noticeably slower in that particular area, which resulted in additional similar images being captured. A solution to the latter problem would be to ignore images if the robot is not moving, or increase the sliding window size. As information on the robot's speed could be obtained based on the commands issued, encoders, or simple visual odometry. If this technique were to be employed for a topological SLAM system, any of these extra nodes that is spurious will disappear after several revisits and vice versa, if these nodes are really meaningful, they will end up being detected most of the times. Some of the node representative images obtained in both traversals are displayed in Figure 4.8. It seems

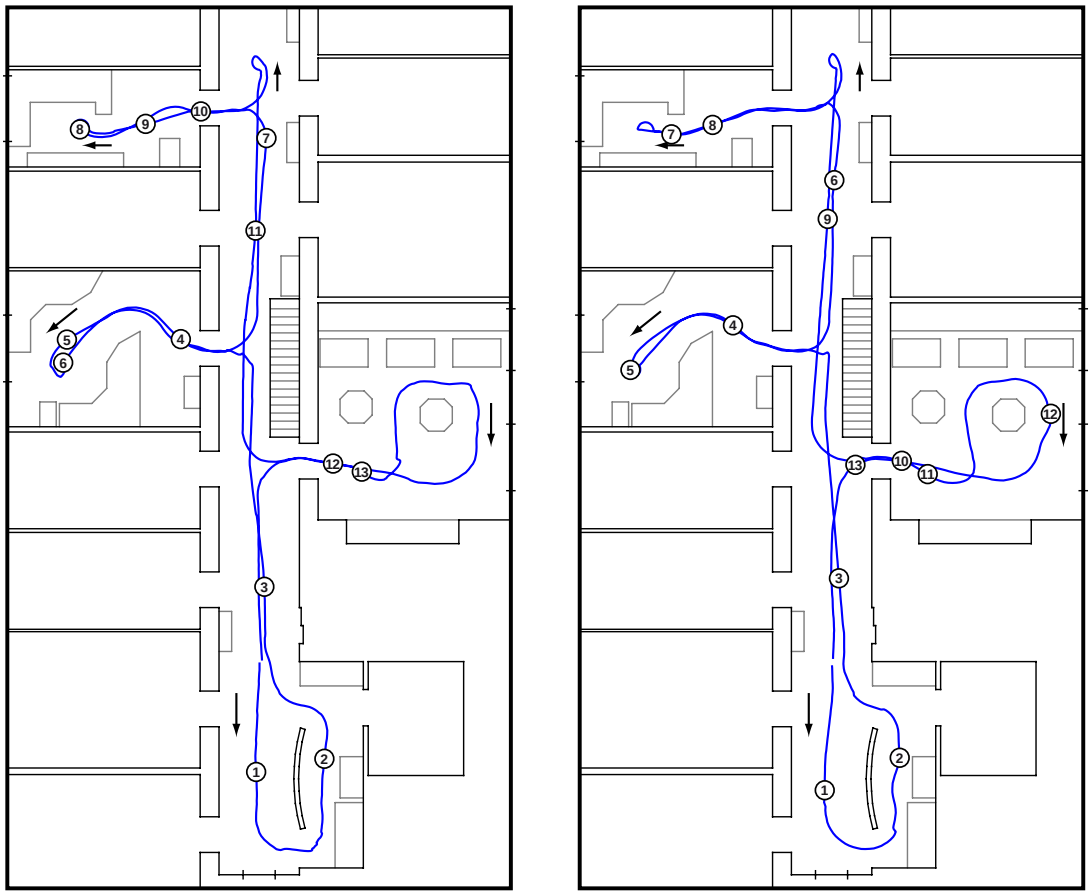


Figure 4.6. Location of the cluster representatives obtained using the bag-of-words model in *Dumbo night 1* (a) and *Dumbo night 2* (b). The numbers inside the nodes indicate the order in which they were detected. The robot's path is provided for illustrative purposes only; no odometric information was used.

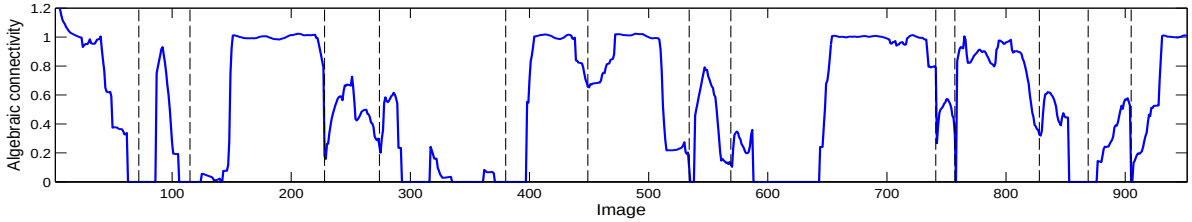


Figure 4.7. Fiedler values obtained from the *Dumbo night 2* dataset using the bag-of-words model. The dashed lines indicate the division between clusters.

Figure 4.8. Sample node representatives from the same topological locations obtained from *Dumbo night 1* (top row) and *Dumbo night 2* (bottom row) using the bag-of-words model.

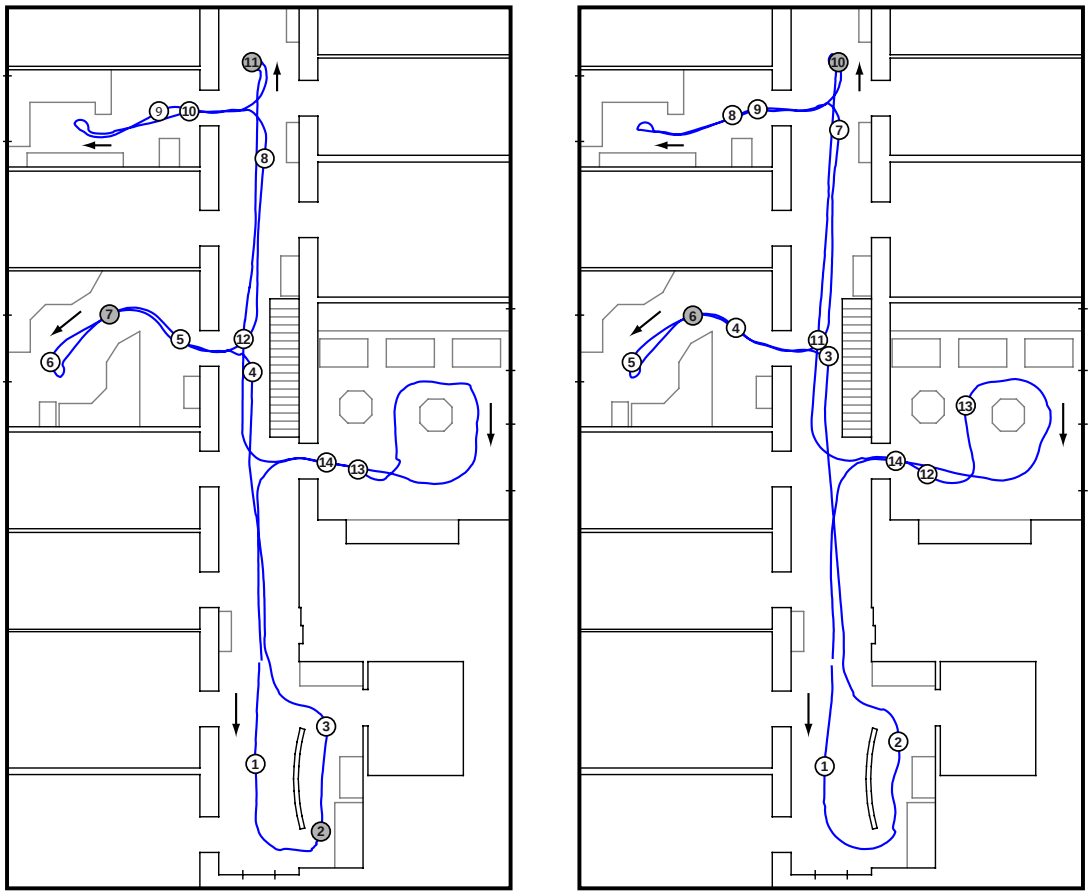


Figure 4.9. Location of the cluster representatives obtained using VPACK in *Dumbo night 1* (a) and *Dumbo night 2* (b). Numbers indicate the detection order. Gray nodes are removed because they have uninformative representatives. The robot's path is provided for illustrative purposes only.

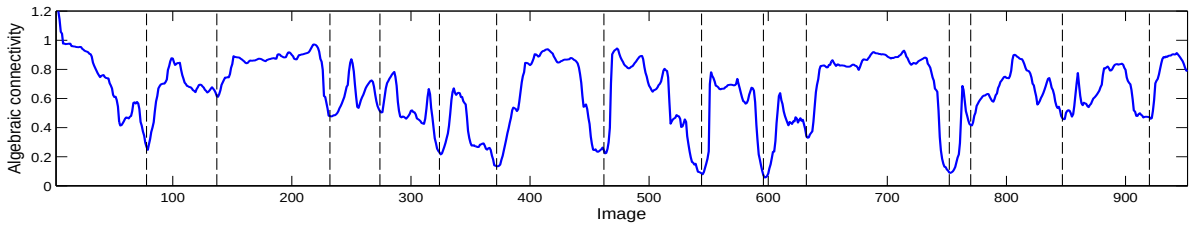


Figure 4.10. Clusters identified in the *Dumbo night 2* dataset using VPACK features. Divisions are marked with dashed lines.

Figure 4.11. Sample node representatives from corresponding topological places obtained from *Dumbo night 1* (top row) and *Dumbo night 2* (bottom row) using VPACK.

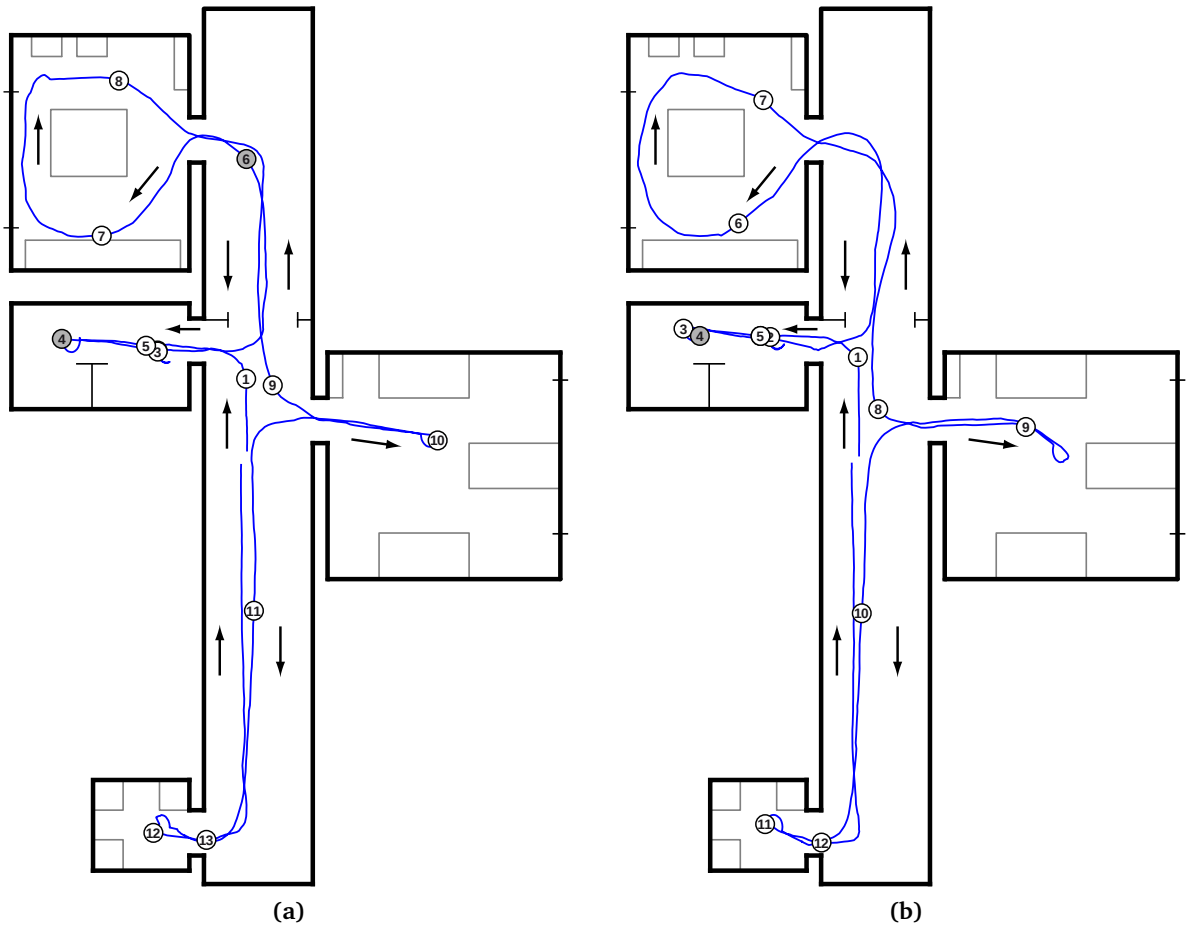


Figure 4.12. Cluster representatives obtained using VPACK in *Saarbrücken sunny 1* (a) and *Saarbrücken sunny 2* (b). Node numbers indicate the detection order. The approximate path followed by the robot is provided as reference.

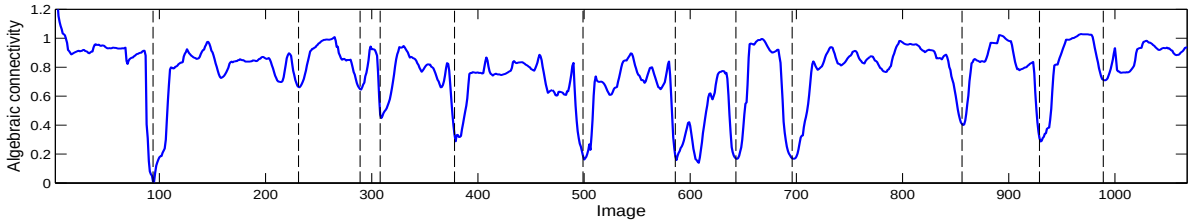


Figure 4.13. Clusters divisions found (depicted with dashed lines) in the *Saarbrücken sunny 2* dataset using VPACK features

Figure 4.14. Example of node representatives from corresponding topological places obtained from *Saarbrücken sunny 1* (top row) and *Saarbrücken sunny 2* (bottom row) using VPACK.

clear that the images chosen as representatives of corresponding topological places match to a high extent.

Regarding VPACK, the same places are also repeatedly identified in both traversals as can be seen in Figure 4.9. In fact, many of the resulting node representatives are similar to those obtained with the bag-of-words model (Figure 4.11). The variations are due to the different nature of the descriptors; the bag-of-words model tends to favor representative images with many keypoints, while VPACK tends to prefer captures with several structural vertical lines. As a consequence, some of the detected topological locations may correspond to fairly homogeneous regions that are not distinctive enough. In order to get rid of those uninformative nodes, the locations whose representative has a very limited amount of keypoints (e.g., below 15) may be discarded when using VPACK as image descriptor. The nodes that would disappear after applying this post-processing step are grayed out in Figures 4.9 and 4.12.

In addition, if one compares the evolution of the algebraic connectivity with both descriptors (Figures 4.7 and 4.10) it can be clearly seen that whereas the Fiedler value reaches zero multiple times with the bag-of-words model, this never happens with VPACK. The reason behind this behavior is that the wall and furniture colors are similar across the environment and, therefore, that part of the descriptor is never completely different. This is common in an office environment, but should not happen that frequently in a home environment.

4.3.2. COLD database

In order to evaluate the performance of the algorithm in conjunction with VPACK in a different environment, another experiment was carried out in the Saarbrücken dataset that is included in the COLD database. The extended version of Path B in sunny weather light conditions was chosen. The two datasets employed are referred to as *Saarbrücken sunny 1* and *Saarbrücken sunny 2*, and consist of 1104 and 1068 images, respectively, acquired using a camera installed at a height of 140 cm. Note that, contrary to the previous experiment, this camera setup allows to capture images that are more similar to what a human being would see.

In this case, the nodes representatives obtained in the two traversals using VPACK are almost identical (Figure 3.11) except for the fact that in *Saarbrücken sunny 1* there is an additional node (node 6). Once again, nodes correspond mainly to entrances, exits, and corridors. It is also important to note that in the first room the robot enters (the bathroom), four nodes are identified and, even some of them overlap. This can be easily explained by the fact that the robot moved back and forth and then stayed still twice. The limitation that was detected in the previous environment also occurs in this experiment: the robot's speed has an impact on the number of nodes detected.

4.3.3. Time profiling

As one of the objectives of the thesis is that the global solution developed permits real-time operation, the running time of the node extraction algorithm when used in conjunction with VPACK has been assessed on the same computer used in Section 3.4, which has a 2nd generation Intel® Core™ i5 CPU at 1.6 GHz and 4 GB of RAM.

The average computation time for *Dumbo night 1* was 198 ms, whereas for *Saarbrücken sunny 1* it took 140 ms. It is important emphasize that almost all the time is spent performing the pairwise comparisons. Once the adjacency matrix is updated, the rest of the algorithm is executed in only 1.4 ms. As the matching procedure proposed for VPACK is not symmetric,

50 comparisons are required in every iteration of the algorithm for a 25-image window like the one considered.

4.4. Conclusion

This chapter has introduced a method for online topological place extraction in visual sequences based on the algebraic connectivity of graphs that can be employed for any type of feature detector as long as non-negative pairwise similarities can be defined. Consequently, it is applicable to both constant and variable size feature descriptors. The results of the tests conducted seem to support this assertion. The topological locations obtained are persistent between traversals and could be used to build a complete topological SLAM system. The only step that remains unsolved is to determine, based on the past observations, whether a newly detected node has already been visited or is a truly new location. This problem is tackled in Chapter 5. In addition, execution times allow the robot to move at an acceptable pace. Still, if performance needs to be further improved, the focus should be put on the matching algorithm presented in Section 3.3.

Likewise other similar techniques in the literature, the main open issue of this method is that it assumes a fairly constant speed and, under some circumstances, can output spurious topological nodes when the robot stays still in the same place or moves slower. This is more likely to happen after a close turn in a cluttered room and much less probable during corridor navigation. As aforementioned, this problem can be overcome using odometric information to adjust the size of the sliding window based on the robot's speed. In principle, as there is no need for a precise estimate, direct encoder readings, a simple visual odometry algorithm, or even the navigation commands sent to the actuators should suffice.

5

Topological Simultaneous Localization and Mapping

The true logic of this world
is in the calculus of probabilities.
James Clerk Maxwell (1831–1879)

Once the robot has the capability of perceiving the environment and determining when it has moved to a distinct place, the final stage in topological SLAM is trying to establish relationships between the detected nodes in order to build a consistent map. As measurements are uncertain, the robot has to consider multiple hypotheses to prevent performing a wrong loop closure it cannot overcome. However, as enumerating all data association combinations is computationally intractable, a solution is to employ some sort of sampling method.

A particle filter is used in this chapter to build a topological map in conjunction with the previous developments of this thesis. Observation likelihoods are obtained by means of VPACK and the adjacency between the detected nodes is used for the transition model. However, as contrary to other approaches in the literature appearance probabilities do not rely on a bag-of-words model and no metric information is employed to make the algorithm more platform independent, some modifications have been introduced to be able to estimate the probabilities of unseen events. The experiments conducted in the same office environments than in the previous chapter show that, even in the absence of odometry, the correct topology can be retrieved with a reasonable number of particles that make the algorithm compatible with a real-time implementation.

5.1. Introduction

Every time the robot extracts a new topological node, no matter if it uses some kind of place segmentation algorithm, like the one presented in Chapter 4, or treats each sensory input as a node, it has to take two hypotheses into consideration: either it has arrived at an already visited location or it is in a previously unexplored place. Once this uncertainty is resolved, the

current location of the robot and the map, represented by the sequence of nodes the robot has traversed to reach its present position, become simultaneously unambiguous.

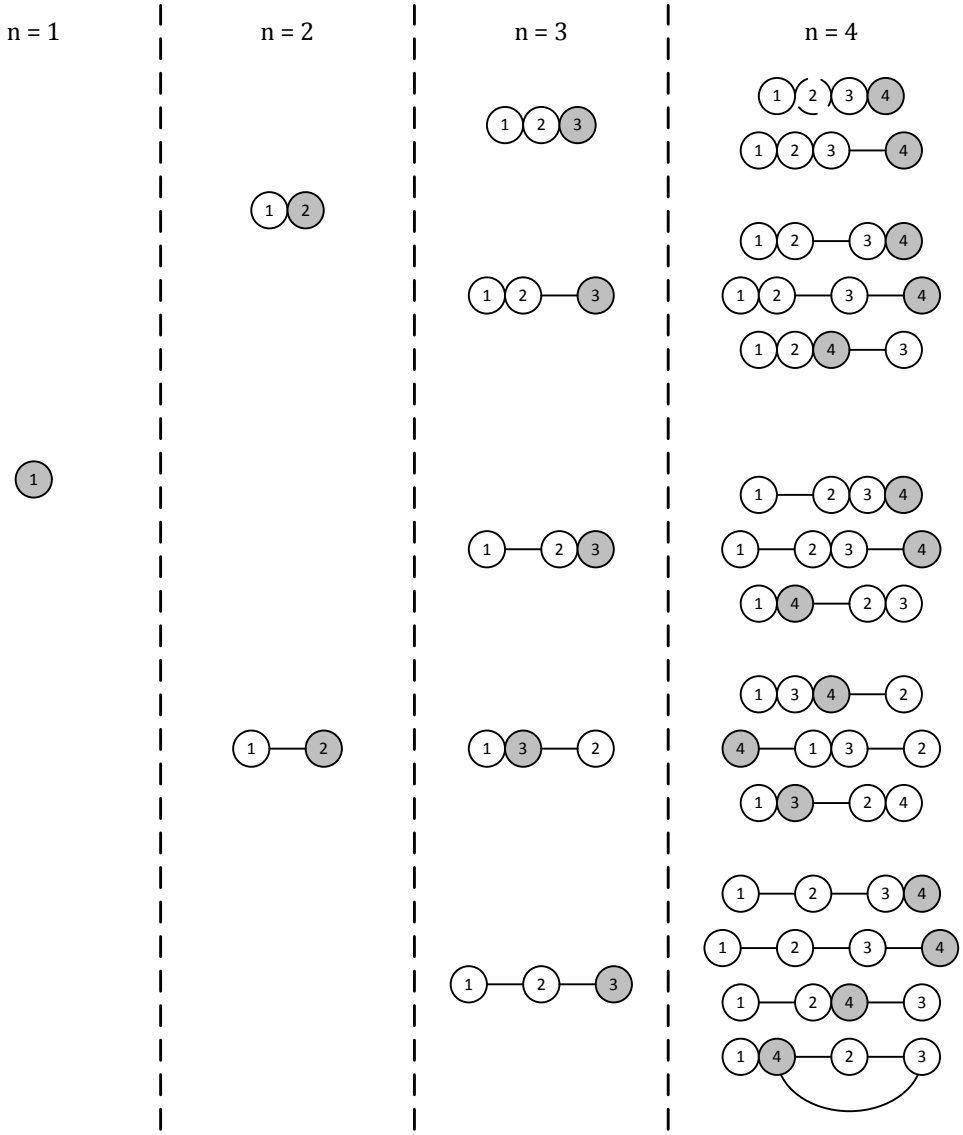


Figure 5.1. Possible topologies for up to four nodes. Contiguous nodes represent the same topological place, shadowed nodes show the current robot's location, and the numbers indicate the detection order.

In the best of cases, a decision can be locally made; however, this is not a common situation and the robot is forced to maintain and update multiple hypotheses over time before it can determine its position. Ideally, it would enumerate and keep track of all possible topologies until it can determine the correct one but, unfortunately, the number of hypotheses grows following the Bell number (5.1) and becomes rapidly intractable. The Bell number B_n represents the different ways a set of n elements can be split into nonempty subsets [Min07; Bel34]. Figure 5.1 shows the possible topologies for up to 4 nodes. When $n = 5$ there are already 52 combinations, and for only 10 nodes, 115975 topologies can be built.

$$B_n = \sum_{k=0}^{n-1} \binom{n-1}{k} B_k, \qquad B_0 = 1 \tag{5.1}$$

As a consequence, several approaches have been proposed in the literature in the context of topological SLAM to prevent having to deal with so many hypotheses. The simplest of them

all is to perform loop-closings as soon as possible. For instance, Romero and Cazorla [RC12] compare each new node with the rest of the representatives of the previously encoded places by means of a similarity threshold. The match that best satisfies the similarity constraint is taken for a loop closure. Otherwise, an additional node is included in the map. The main disadvantage of this solution is that it is extremely sensitive to the threshold value chosen and can easily output too cluttered or sparse topological maps. In addition, with these kinds of approaches a common criticism is that it is not possible to recover from incorrect loop-closings and that the map can easily become inconsistent in non-distinctive environments.

Tapus and Siegwart [TS05; Tap05] rely on a partially observable Markov decision process (POMDP) [Cas+96] to approximate the discrete probability distribution over all poses in the environment and find the control strategy that is likely to reduce the uncertainty to a greater extent. The POMDP is considered to be confident about its current state if the entropy of the probability distribution is sufficiently low, beneath an experimentally adjusted threshold. Only then the optimal action for that pose is executed. Otherwise, the robot tries to return to the corridor, because the algorithm was tested in an office environment, with the hope of gathering more information. Loop-closing relies also on the POMDP. As soon as the robot returns to a previously visited location, two peaks should appear in the pose distribution: one corresponding to the new node found by the robot, and another to a place already present in the map. A loop-closing is assumed if they behave likewise over time. The disadvantage of using POMDPs is that localization and mapping are dependent on navigation commands. It is certainly true that the robot can usually control its own movements, but we might not be willing to allow it to deviate from its path. Furthermore, this method cannot be employed as a stand-alone module in mobile entities where there is no access to the locomotion system.

Finally, the Bayesian formulation is probably the most widespread. Angeli *et al.* [Ang+08b] propose an appearance-based incremental topological SLAM implementation where nodes are identified using a similarity threshold, that is, an image is only considered if it is sufficiently different to the previous one. As a consequence, when the robot has to process a node, it can be fairly sure that it has moved. For this reason, the transition probability used to calculate the prior is modeled using a sum of Gaussians, to give more importance to the adjacent states and reduce the probability of being in the same node. The appearance likelihood is computed by means of an online bag-of-words model [FM03] and a voting scheme using the *tf-idf* coefficient [SZ03; Csu+04]. The likelihood of previously unvisited nodes is handled adding a virtual node characterized by the most frequent visual words. Once the normalized full posterior is obtained, if a small set of adjacent nodes have a probability higher than 0.8, the epipolar geometry of the keypoints is checked using RANSAC [Nis04]. The successful hypothesis is accepted as a loop closure. Otherwise, if all these conditions are not met, a new node is added to the map.

FAB-MAP (Fast appearance-based mapping) [CN08] follows a very similar approach. If the robot was at place i in time t, the transition function assigns equal probability to being at i or at its adjacent nodes in $t+1$. The observation likelihood is also computed using visual words. However, instead of computing an "average place" to model previously unseen places, the visual vocabulary is sampled to obtain several location models. A maximum a posteriori data association is performed after every time step. CAT-SLAM [Mad+11] incorporates odometry to FAB-MAP.

The main issue with all the previous solutions is that the robot cannot recover from incorrect loop closures. Rather than trying to estimate only the current node, Ranganathan and Dellaert [RD06a; RD11] suggest a Bayesian framework to obtain the posterior over all topologies given

the measurements. However, as it has been aforementioned, this problem rapidly becomes computationally intractable. For this reason, they propose using a particle filter to approximate the calculation and only keep track of the most probable hypotheses. As not only appearance but also laser scans and odometry are used, the filter is Rao-Blackwellized to separate discrete (appearance) and continuous (metric) state variables.

This chapter presents a Bayesian approach to topological SLAM that uses only appearance measurements and adjacency information with the aim of evaluating if VPACK and the nodes extracted in Chapter 4 are a viable alternative for topological map building and, at the same time, determining the need for metric data, which has to be estimated differently depending on the robot's locomotion system (e.g., wheeled, legged, flying) and is therefore platform dependent. In order to deal with false loop closures, a particle filter is employed to maintain several topologies and gather enough information before a loop closure decision is made. The formulation is similar to the one introduced by Ranganathan and Dellaert [RD11] but as there are no continuous variables involved, there is no need to apply a Rao-Blackwellized filter.

The remainder of the chapter is structured in the following manner. First, the algorithm used and the implementation details are set forth in Section 5.2. Afterwards, the results of the tests carried out in the same office environments used in Chapter 4 are discussed in Section 5.3 and, finally, the main conclusions are put forward in Section 5.4.

5.2. Topological SLAM algorithm

This section describes the procedure that is followed to update the candidate topological maps every time the node extraction algorithm proposed in Chapter 4 identifies a new representative. The revision depends on whether the robot believes it has arrived at a new place or it has returned to an already mapped location, according to the appearance of the representative image and adjacency data extracted from the maps built up to this time step.

5.2.1. Bayesian formulation

The aim of topological SLAM is to determine the sequence of nodes traversed, based on a series of measurements of different nature. Before proceeding, a quick remark on the notation is required. Throughout the mathematical derivation, X_k denotes the state of variable X at time k and X^k is equivalent to $X_{1:k}$, the set of all states up to time k. Therefore, if L stands for the robot's location, the probability that needs to be inferred is $p(L^k|z^k)$, the posterior on topologies given the measurements z which, in this case, correspond only to appearance information. Applying Bayes' theorem and dropping the normalization term leads to

$$p(L^k|z^k) \propto p(z_k|L^k, z^{k-1}) \cdot p(L^k|z^{k-1}) \tag{5.2}$$

where $p(z_k|L^k, z^{k-1})$ is the appearance measurement likelihood, and the prior on the topologies $p(L^k|z^{k-1})$ can be further factorized as

$$p(L^k|z^{k-1}) = p(L_k|L^{k-1}, z^{k-1}) \cdot p(L^{k-1}|z^{k-1}) \tag{5.3}$$

with $p(L_k|L^{k-1}, z^{k-1})$ representing the transition model and $p(L^{k-1}|z^{k-1})$ being the posterior from the previous step. Therefore, the final expanded expression results in

$$p(L^k|z^k) \propto p(z_k|L^k, z^{k-1}) \cdot p(L_k|L^{k-1}, z^{k-1}) \cdot p(L^{k-1}|z^{k-1}) \tag{5.4}$$

The following sections are devoted to explaining in detail how each of these terms are computed and how the algorithm has been implemented in practice using a particle filter.

5.2.2. Appearance measurement likelihood

The observation likelihood can be further simplified if the *Markov property*, which states that given knowledge of the current state, the future becomes independent of the past [Thr+05], is assumed true. This means that the probability of perceiving a particular set of landmarks depends only on where the robot currently is, not on where it has been before or, mathematically, that $z_k \perp\!\!\!\perp L^{k-1}|L_k$. Therefore, the simplified appearance likelihood ends up being $p(z_k|L_k)$.

VPACK with Star keypoints (Chapter 3) is used to estimate this term by comparing the fingerprint of the latest identified node with the representatives of each of the known locations giving the same importance to each of the *n*-gram models. If there is more than one representative per node, only the highest likelihood is considered. As usual in topological SLAM, the problem that arises is how to estimate the observation likelihood when the robot is in a previously unvisited location. As mentioned earlier, whenever a visual vocabulary is available, the typical solution is to employ the most common words [Ang+08b] or sample from the dictionary [CN08] and use these features as the representative for the unknown node. Unfortunately, this is not possible with VPACK and another heuristic is used instead. If the features and their associated matching procedure are distinctive enough, they should output high likelihoods when the two images are similar, and low probabilities otherwise. Hence, one of the simplest solutions is to set the measurement likelihood for a new location to one minus the maximum of the likelihoods of the set of mapped places $\mathcal{L}^k$.

$$p(z_k|L_k = \text{new}) = 1 - \max_{m \in \mathcal{L}^k} p(z_k|L_k = m) \tag{5.5}$$

However, it may happen that the maximum likelihood is not very high but that it is still significantly higher than that of the rest of alternatives. This would imply that the robot is somewhat more confident about being in that location than if there were two or three additional nodes with a likelihood close to the maximum. In order to take this effect into account, $p(z_k|L_k = \text{new})$ is weighted using the normalized entropy η that is a measure of the information of a probability distribution [Gol06]. All the measurement likelihoods of the known nodes are normalized and each resulting value $p(x_i)$ is used to compute η with the following equation

$$\eta = -\sum_{i=1}^{|\mathcal{L}^k|} \frac{p(x_i)\log(p(x_i))}{\log(|\mathcal{L}^k|)} \tag{5.6}$$

The normalized entropy has a value of one if all the normalized likelihoods are equal, and tends to zero as one option prevails over the others. Therefore, the likelihood of a new node will be assigned a lower value when there is a known node that stands out from the rest. The probability of observation z_k being originated from a new location is finally estimated as

$$p(z_k|L_k = \text{new}) = \eta \cdot \left(1 - \max_{m \in \mathcal{L}^k} p(z_k|L_k = m)\right) \tag{5.7}$$

One final remark is required before concluding this section. As at the beginning there is a single known node, its normalized probability is one and, due to the logarithm in the denominator, the computed normalized entropy would tend to infinity. In this case, it is

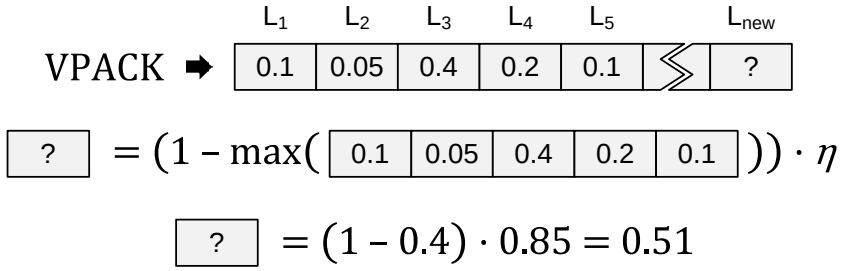


Figure 5.2. Sample computation of the appearance measurement likelihood for an unknown location.

assumed that $\eta = 1$ because every representative detected in the initial steps is more likely to correspond to a new node.

A simple application example is shown in Figure 5.2. Assume there are five known locations and that VPACK has already been used to obtain their appearance likelihoods. The measurement likelihood of a new location is computed as one minus the maximum, which amounts to 0.6 in this case, and weighted with the inverse entropy of the known locations that is computed by normalizing the values of the known likelihoods and applying equation (5.6).

5.2.3. Transition model

The motion model expression can be reduced to $p(L_k|L^{k-1})$ if it is assumed that the current location is independent of the past measurements given that the previous node sequence is known or, in other words, that the expected robot's current position depends on where it has been but not on what it has seen there. Two different situations need to be considered in this case. Either the robot was in a new location at time $k-1$, or it was in a previously mapped node. If it was in a new node, there is no adjacency information available and it is estimated using the following equation

$$p(L_k|L_{k-1} = \text{new}, L^{k-2}) = \begin{cases} \frac{1}{|\mathcal{L}^{k-1}|+\alpha} & \forall L_k \in \mathcal{L}^{k-1} \\ \frac{\alpha}{|\mathcal{L}^{k-1}|+\alpha} & \text{otherwise} \end{cases} \tag{5.8}$$

where α is a parameter that expresses the prior belief on the size of the environment (i.e., the expected number of nodes). It can be derived from the equation that the more nodes the map has, the less likely the robot has arrived at a new node. Other authors suggest employing a Dirichlet process prior instead [RD11], which is similar to the proposed formula but increases the probability of revisiting a location every time it is seen. This is also a reasonable assumption but has the inconvenient that after traversing the same path a few times, the probability of having moved to a new node will be really low. Hence, should the robot start to explore a different area at some point, the resulting map will be erroneous.

Conversely, if the robot moved from a known node, the most likely location would be the closest subsequent node. However, it may happen that the topological segmentation procedure missed it, so it could be in the following with less probability, and so on. Nevertheless, other alternatives cannot be disregarded. The robot could have also stayed in the same location or arrived at a completely new node. A gamma distribution with shape parameter $k = 10$ and scale parameter $\theta = 0.25$ is used to model the probability of not having moved, or being in one of the next nodes (Figure 5.3). The probability of being in a new node is given a fixed quantity (e.g., 0.05) and the gamma distribution is normalized so that the probability adds up to one. If no high-order neighbors exist, their probability is assigned to that of the new node.

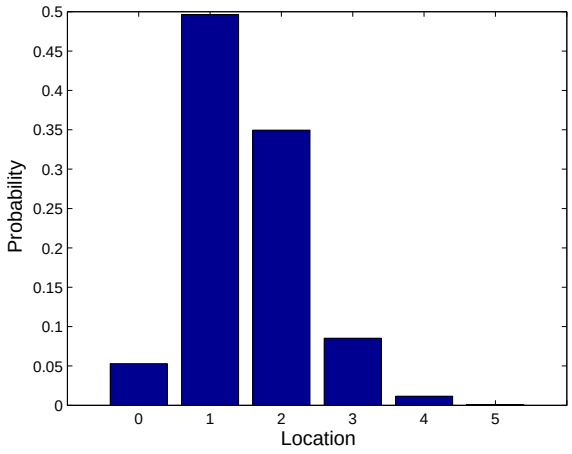


Figure 5.3. Probability of staying in the same node (0) or moving to the *n*th subsequent location.

Notice that the possibility of returning to a preceding node is not considered. This is an inherent limitation of frontal computer vision because if a node is entered from another direction the image captured is different.

5.2.4. Particle filter

The recursive nature of the Bayesian formulation makes it amenable to particle filtering, a non-parametric approach (i.e., data is not assumed to belong to any particular distribution) to approximate posteriors that is capable of dealing with multimodal beliefs [Thr+05]. The idea behind particle filters is that they represent a distribution with random samples, called particles, drawn from this same distribution. Consequently, it is an appropriate technique to solve the topological SLAM problem because it allows to keep track only of the most likely topologies and discard those that have negligible probabilities. The number of particles M gives control over the computational load, and unless the environment is prone to perceptual aliasing, the results do not vary much as long as this parameter is not too small (usually in the order of the hundreds) [Thr+05].

In this thesis, a standard particle filter (Algorithm 5.1) is employed to estimate the posterior on topologies $p(L^k|z^k)$ with a set of weighted particles

$$T_k = \left\{ t_k^{(1)}, t_k^{(2)}, \ldots, t_k^{(M)} \right\} = \left\{ L^{k,(i)}, \omega_k^{(i)} \right\}_{i=1}^{M} \tag{5.9}$$

where $L^{k,(i)}$ is the topology of the ith particle at time step k and $\omega_k^{(i)}$ is its weight. In short, for each particle, a new topology is first proposed based on the map built up to the previous time step and the robot's motion model. As a result, the robot's current position becomes known for this particle and an appearance likelihood can be computed. To prevent having to compute the similarities between the current representative and those of all the nodes in the map for every particle, which is a computationally demanding task, the comparisons are performed in advance, and each particle assigns the corresponding precomputed value to its importance weight. If the new sample location is a new node, only the entropy needs to be calculated to estimate the observation likelihood. Finally, a new posterior is obtained by drawing samples proportional to these likelihoods.

Bearing in mind that particle filters are an approximation, they are subject to the following errors that need to be taken into account. For more details refer to [Thr+05].

Algorithm 5.1. Particle filter algorithm

$T_k = \overline{T}_k = \varnothing$

for $i = 1$ to M **do**

1: Propose a new location for the robot $L_k^{(i)} \sim p(L_k|L^{k-1})$ and update the topology $L^{k,(i)} = \left\{L_k^{(i)}, L^{k-1,(i)}\right\}$.

2: Compute the importance weight $\omega_k^{(i)} = p(z_k|L_k^{(i)})$.

3: Add $t_k^{(i)}$ to $\overline{T}_k$.

end

for $i = 1$ to M **do**

4: Draw a particle $t_k^{(j)}$ from $\overline{T}_k$ with probability proportional to $\omega_k^{(j)}$.

5: Add $t_k^{(j)}$ to T_k.

end

- As a finite number of particles is used, a systematic bias is introduced. Imagine that there was a single particle ($M = 1$). In this case, the measurement likelihood would be irrelevant because the same particle would always be drawn in step 4 of the algorithm regardless of the information obtained from the sensor readings. Although this is an extreme situation, it may happen that all particles converge to the same, not necessarily correct, topology, thus reducing the effective number of particles that can lead to a scenario similar to that of having a single particle. Fortunately, this becomes more unlikely as M increases.
- The performance of the particle filter depends on how much information the measurements provide but, especially, on the transition model. If there is a great deviation from the real motion of the robot, the true topology will never be proposed and the algorithm will inevitably converge to an incorrect solution.
- A third problem is *particle deprivation*. Due to the random nature of the resampling process, the actual topology may not survive. In practice, this only tends to occur if M is small relative to the number of states with high measuremnt likelihood. That is why the features employed need to be distinctive enough.

5.2.5. Map update

Once the particle filter is computed, the information of the latest representative extracted must be incorporated to the map for further reference. In each of the surviving topologies, this fingerprint is added to the collection of representatives of the current node. For instance, if in a given topology the robot considers to have returned to the initial node, after the update step this node would have two representatives, the initial one and the last fingerprint captured. In order to avoid having to perform too many comparisons in the long term, the oldest representatives could be progressively deleted. This is a better solution than trying to merge representatives because with this approach, in the end, the representatives of all nodes tend to look alike.

5.3. Results and discussion

The method explained above was programmed in C++ and validated with the same image sequences employed in Chapter 4, that is, the KTH-IDOL2 and COLD databases (see Appendix A for more information). As each of the image sequences in these databases correspond to a single traversal of the environment with a common start and end point, two sequences are

stacked together in each of the cases in order to be able to test the loop closing capabilities of the algorithm.

As far as the implementation details are concerned, identical parameters were used in both cases: 500 particles were chosen as a compromise between taking a relatively large number of alternative topologies into account and computational effort, and a prior on the map size of $\alpha = 15$ was assumed.

5.3.1. KTH-IDOL2 database

The topological map obtained using *Dumbo night 1* and *Dumbo night 2* is shown in Figure 5.4. The blue and red lines indicate the approximate path followed by the robot in the first and second traversals respectively. It is provided for illustrative purposes only. Each numbered circle stands for the location where the candidate node representatives were found with the segmentation procedure presented in Chapter 4 and their detection order, whereas the triangles indicate the robot's orientation approximately. Therefore, representatives 1 through 11 correspond to the first traversal and 12 to 23, to the second. Obviously, the algorithm is not aware of this information. The circles that share color have been assigned to the same topological node. This means that representatives 2 and 13 form a unique node, that is connected to the one composed by representatives 3 and 14, that in turn heads to the node constituted by 4 and 15, and so forth.

In general, all representatives that are close to each other and were captured with similar orientations are correctly grouped together. The topology depicted in Figure 5.4 accounted for 0.50 of the probability up to, but not including, the detection of representative 22. As this representative was not identified during the first traversal, the robot is uncertain about whether it is a new node or not and, assuming it is a new node, what is the next node it is bound to encounter. Therefore, after this point the probability of the most likely topology plunges and when the last representative has been extracted and processed the probability is only 0.14. Another traversal of the environment would be required to make it rise again. Finally, note that that representatives 1 and 12 are quite distant and assigned to different nodes because in the second traversal, the robot explores a longer section of the corridor and finds a better image to describe it elsewhere.

5.3.2. COLD database

In the case of the COLD database, *Saarbrücken sunny 1* and *Saarbrücken sunny 2* were used for testing. As in the previous environment, the algorithm correctly finds the correspondences between locations that are physically near, and thus produce similar sensory information. The topology depicted in Figure 5.5 received a posterior probability of 0.96 and the rest of the non-negligible alternatives are minor variations of it.

Nevertheless, there are a few interesting things that deserve being highlighted. First, in spite of the robot moving rather erratically in the first room it entered, the algorithm successfully managed to assign all the representative images detected to the same topological node. Furthermore, it decided to group together representatives 5 and 6 (16 and 17 in the second traversal), which were detected consecutively and correspond to the same room, even though they are not in the same spatial coordinates, probably because they share some features in common like the predominant colors. This behavior would have been more unlikely if an odometry model had been used to propose the topologies. Despite this, both alternatives are

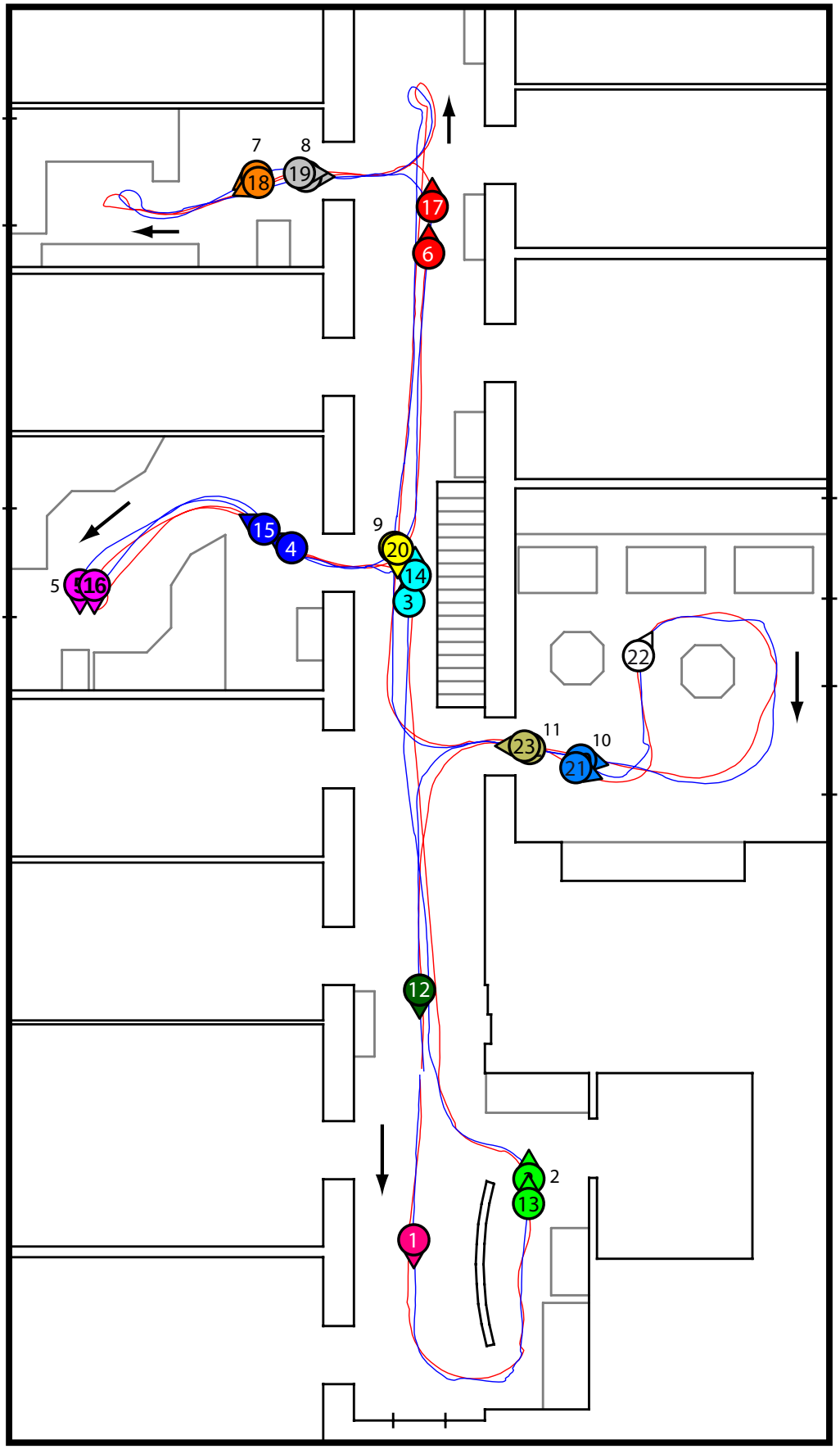


Figure 5.4. KTH-IDOL2 database. Topological SLAM results in *Dumbo night 1* and *Dumbo night 2*. Uncircled numbers indicate the detection order of the occluded nodes.

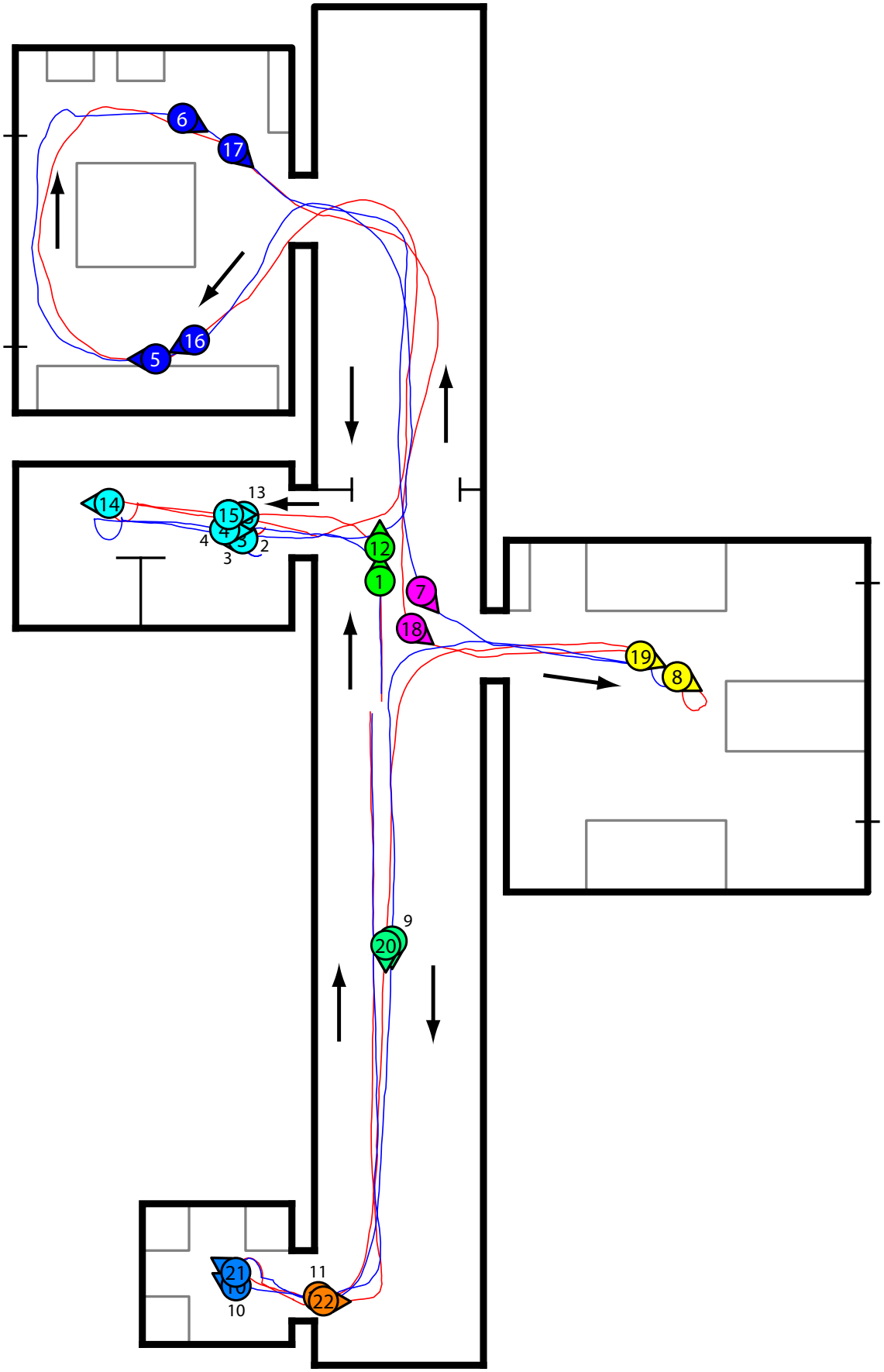


Figure 5.5. COLD database. Topological SLAM results in *Saarbrücken sunny 1* and *Saarbrücken sunny 2*. Uncircled numbers indicate the detection order of the occluded nodes.

correct and choosing one over the other depends more on the particular application the map is intended to be used for. Finally, the green (1 and 12) and magenta (7 and 18) nodes are different although they are almost in the same coordinates, because the images captured are dissimilar due to the different orientation of the camera.

5.3.3. Time requirement analysis

In order to assess if the method could be applied in real-time, some time measurements have been carried out on the same computer used throughout the dissertation that is equipped with a 2nd generation Intel® Core™ i5 CPU at 1.6 GHz and 4 GB of RAM. The average computation time of the particle filter, which includes the time spent performing the appearance likelihood comparisons, was 108 ms on the KTH-IDOL2 datasets, whereas it took 158 ms for the sequences of the COLD database.

Finally, now that all the modules of the thesis have been explained, the total execution time of the solution proposed can be analyzed. Figures 5.6 and 5.7 show the time employed processing each of the input images. This time always includes the computation of the node extraction procedure (Chapter 4) and, only when a new representative is found, of the particle filter. On average, the total time required for the experiments conducted on the KTH-IDOL2 database was 223 ms, with a maximum of 670 ms. In the case of the COLD database, the mean was 230 ms and the highest value 930 ms.

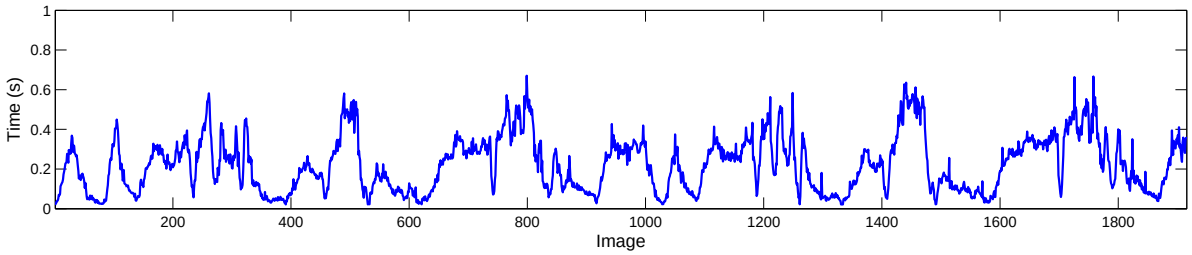


Figure 5.6. Total computation time required for every input image of the KTH-IDOL2 database.

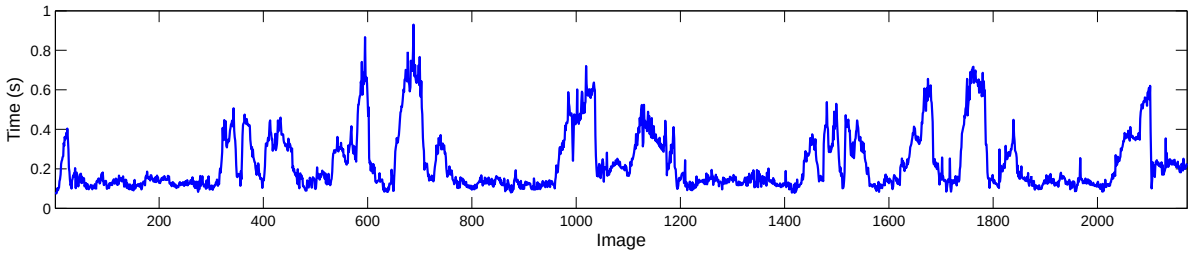


Figure 5.7. Total execution time of each image in the Saarbrücken datasets.

5.4. Conclusion

The utilization of a particle filter to build a topological map of the environment without metric information has been studied in this chapter. Conversely to other similar appearance-only approaches where a Bayesian framework is also used, the system has been applied to a set of nodes that are physically and visually separated rather than treating each input image as a different node. As a consequence, the system is computationally lighter and allows to maintain several alternatives over time and make more informed decisions, instead of having to perform a single data association every time step, which apart from being more demanding can lead

to incorrect loop closures. The trade-off is that the position of the robot is known with less precision. This might be an issue for some applications, but is generally enough indoors where the interest mostly lies in detecting in which room the robot is.

With respect to other approaches that employ particle filters, the absence of metric information allows the algorithm to consider as a single node places that are relatively apart but make sense to be regarded as a unique location (e.g., two nodes in the same room) and that would otherwise be treated as different nodes due to the metric constraints. Thus, this results in more sparse maps with richer nodes. Nevertheless, it might be interesting to analyze in more detail the impact of incorporating weak odometric information, perhaps following the approach by Bazeille and Filliat [BF11], to improve the topologies proposed by the transition model and reduce the uncertainty introduced by those locations that are not consistently detected by the node candidate extraction algorithm.

The outcome of the experiments carried out in this chapter encompass the developments of the whole thesis, as these have been used to implement the different terms of the Bayes filter. VPACK (Chapter 3) is employed to estimate the appearance likelihoods, whereas the different candidate nodes are obtained with the topological segmentation procedure based on the algebraic connectivity of graphs presented in Chapter 4. The resulting topological maps match the actual structure of the environment and it has been proved that the robot is capable of determining the most probable map and localizing itself within it after only two traversals.

In addition, as far as time is concerned, in the worst of cases, which occurs when the robot arrives at a new location and, apart from the node extraction procedure, it has to run the particle filter to update the candidate topological maps, all the required operations are performed in under one second, which makes it clearly amenable to real-time implementation in embedded systems with limited computational resources. Notwithstanding, the computation of the particle filter can be split over several time steps because, at least for SLAM purposes, the result is not required until the robot leaves its current location.

6

Conclusions, Contributions and Future Work

Science never solves a problem
without creating ten more.
George Bernard Shaw (1856–1950)

This last chapter, summarizes the developments of this thesis. The main conclusions that can be drawn from the experiments carried out are set forth and the most original contributions are highlighted. Finally, the open issues that have not been tackled in the thesis, as well as possible future research lines, are discussed.

6.1. Summary and conclusions

For a mobile robot, holding a reliable map it can use to navigate the environment is essential to enable it to perform other higher level tasks. Having a human being to produce this map in a format understandable by the robot is not practical, and therefore it is a much better idea to provide the robot with the capability of building the map on its own. This map can be either metric or topological. Despite metric maps being much more accurate, they are also more computationally demanding and do not scale well with the size of the environment. By contrast, topological maps provide a more compact representation that is useful for service robots, which rarely need to know their pose with precision of millimeters and degrees. To some extent, the topological approach recalls human map building techniques. By means of vision, we acquire relevant information from the different locations we visit and establish relationships of proximity and order between these places. Therefore, it seems reasonable to think that a robot equipped with a forward-facing camera should be capable of operating in a similar manner.

This thesis has proposed a comprehensive solution to the topological SLAM problem that covers all the modules involved in the process rather than concentrating on a single step. It is incremental, computationally lightweight, and fairly easy to integrate in any robotic platform, because it is independent of the robot's locomotion system (e.g., wheeled, legged, flying), due

to the absence of metric information, and because the only sensor used is a monocular camera. Notwithstanding, although the different modules have been conceived to operate jointly, they can also work as separate units. This allows to replace any of them for other alternatives that may suit a particular application better (e.g., use a different feature detection method) or, conversely, apply one of the algorithms presented in other contexts.

Although there are many different visual features in the literature for topological feature extraction, as has been shown in Chapter 2, in the end there exist only two prevailing research lines: keypoints and a collection of complementary features. A combination of both approaches, coined VPACK, is proposed in Chapter 3 with the intention of making the most of the individual characteristics of each type of feature and improve distinctiveness. Vertical edges, naturally divide the image into meaningful regions, color histograms provide holistic information of the scene, and keypoints concentrate on the details. This combination of features very much resembles how human beings process visual information. We acquire an approximate idea of how objects are arranged in space, collect what describes the scene as a whole (e.g., the predominant color), and concentrate on the distinctive details that make the place unique. Along with the VPACK descriptor, a tailored matching procedure designed to supply the features with geometric meaning through simple ordering is introduced. This matching framework allows to seamlessly include additional types of features in the fingerprint. It has been experimentally verified that VPACK provides high scene recognition accuracy even in challenging office environments where the wall colors and the furniture of all rooms look alike and color histograms cannot exhibit their full potential. Furthermore, it produces fairly good results in environments with moderate semi-permanent dynamics (i.e., changes that persist for a prolonged period of time) like those caused by human activity over a four month timespan.

In order to reduce the computational load of solving a loop-closure problem every time an image is captured, as often occurs with appearance-based SLAM, an online node extraction procedure that can accommodate any kind of feature has been presented in Chapter 4 to segment space into persistent and perceptually salient topological places. Each detected node is represented with the image that best captures the essence of the location. This is also similar to how a person's mind works. When we think about a place, we only recall one or two mental images. According to the results of the tests carried out, nodes are consistently detected in the same location in successive traversals of the environment. The topological places found are similar even if completely different descriptors and matching algorithms are employed, which suggests that the proposed algorithm succeeds in finding relevant places in the environment like, for instance, doorways.

Finally, all the developments of the thesis are put together in Chapter 5, where a particle filter is used to track the most probable topologies of the environment using only appearance information and the adjacency between the different nodes extracted. According to the experiments, which were conducted in the same publicly available datasets of two different office environments used throughout the dissertation, after two traversals the robot is capable of closing the loop and determining its position correctly within the map it has created from scratch with a fairly high probability.

With regard to computational burden, the most demanding moment occurs when a new candidate node is identified because it is only then that the particle filter needs to be computed. Using a reasonable number of particles, computation times never exceed one second in a moderately powerful laptop. Still, if the robot cannot wait slightly longer as a human being

would do when trying to determine where it is, the computation of the particle filter could be distributed along several time steps, because its output is not required for mapping purposes until the following node is identified. Consequently, the topological SLAM solution presented in this dissertation is compatible with a real-time implementation.

Throughout the thesis, an attempt to reduce the number of critical parameters has been made. In Chapter 3, several values like the minimum space between vertical lines, the amount of keypoints, or the number of bins used for the chromatic and achromatic histograms have been preset. However, there are no significant variations in the results as long as reasonable values, similar to the ones proposed, are used. In node extraction (Chapter 4) there are three main parameters: the sliding window size and two values required to identify the transition between places. Only the first has a great impact on the outcome because if it is too large, very few nodes are found and, conversely, if it is too small, almost all images would be treated as a new node, thus losing map sparsity. Lastly, in Chapter 5, two parameters are employed: one that provides an insight of the number of places in the environment and the number of particles in the filter. The former is just a rough guess that may only affect the final map produced should there be a large deviation with respect to the actual environment. As for the particles, any number large enough to prevent the problems associated with the algorithm produces no noticeable differences.

6.2. Original contributions

This thesis was conceived with the objective of pushing forward research on topological simultaneous localization and mapping for service robots using a single affordable and easy to install sensor like a forward-facing camera. The main original contributions of this dissertation are the following:

- A meticulous revision of the state-of-the-art, carried out with the intention of bringing together, in a structured manner, the most relevant developments made in topological SLAM, that has resulted in a published journal article [Boa+14b]. As nothing similar could be found in the literature, effort was put into producing a comprehensive survey that could serve as a starting point for new researchers into the field and, at the same time, provide current researchers with a broader overview of all the different approaches that exist up-to-date.

- A new fingerprint for topological place identification, named VPACK, based exclusively on visual features and specifically targeted at monocular cameras, as opposed to omnidirectional cameras that are currently *de facto* standard in visual topological SLAM. As monocular cameras have a narrower field of view, more informative information needs to be extracted. For this reason, a fingerprint based exclusively on weak features (e.g., edges, color) is not generally sufficient, hence highly distinctive keypoints like SIFT [Low04] and SURF [Bay+08] are widely used instead. However, the latter are usually too computationally demanding for robots running on low specification hardware. VPACK was designed with the aim of bringing the best of both worlds together by combining a few robust keypoints, which do not necessarily need to be SIFT or SURF, with weak features, like vertical edges and color histograms, that can help disambiguate between challenging locations.

- Of the three kinds of features that constitute a VPACK fingerprint, special attention has been put on color histograms because color is often overlooked due to its sensitivity to

lighting variations. With a view to reducing the impact of illumination changes, a method to distinguish between chromatic (where color information is relevant) and achromatic pixels (where only grayscale values should be employed) has been introduced.

- A matching procedure for VPACK inspired by the natural language processing field and the concept of *n*-grams that allows to compare two images based not only on the similarity between features, but also on the order in which they appear. This ordering is possible thanks to the structural vertical edges extracted in VPACK, which naturally split the image into subregions based on the content of the scene instead of on an arbitrary division as occurs with spatial pyramids. Furthermore, in the context of localization, a method based on inverse entropy that gives more weight to the *n*-gram model that is more confident about its prediction is also proposed.
- An online and fairly easy-to-implement node extraction method that relies on the concept of algebraic connectivity of graphs and that can be generally employed with no matter what feature descriptor and matching procedure as long as a non-negative pairwise similarity measure can be computed. It is an alternative to arbitrarily defining thresholds on similarity measures that is backed up by graph theory, which has been extensively used in batch clustering applications. When applied to an image sequence, the algorithm identifies a topological location whenever several consecutive captures are similar enough among themselves. Once the robot leaves a place, a single image representative is selected for the node left behind.
- Finally, it has been demonstrated that a standard particle filter can be used without metric data for topological SLAM applications. New strategies to estimate the probabilities of unseen events in the absence of a database to sample from have been suggested, along with a slightly more sophisticated method to take more advantage of the adjacency information.

This research has given rise to the following journal articles:

- J. Boal, Á. Sánchez-Miralles, and Á. Arranz, "Topological simultaneous localization and mapping: A survey," *Robotica*, vol. 32, no. 5, pp. 803–821, Aug. 2014. DOI: 10.1017/S0263574713001070
- J. Boal, Á. Sánchez-Miralles, and M. Alvar, "Matching monocular lightweight features using *n*-gram techniques for topological location identification," *Robotica*, vol. FirstView, pp. 1–15, May 2014. DOI: 10.1017/S0263574714001076

Additionally, at the time of writing, another paper is under review:

- J. Boal and Á. Sánchez-Miralles, "Online topological segmentation of visual sequences using the algebraic connectivity of graphs," submitted to *Robotica*.

6.3. Future work

In the hope that this dissertation inspires readers to continue research in pursuit of a mapping system that is generally applicable to low-cost platforms, some ideas to further extend the solutions presented, and that may open new research areas, are set forth below:

- Throughout this thesis, a monocular camera has been used as the primary sensory source because it is inexpensive and can be easily installed in any robot. However, there are

currently multiple affordable RGB-D sensors available that provide fairly good depth estimates, in addition to monocular images of sufficient resolution. For these reasons, this kind of sensors are being progressively adopted in robotics. The inclusion of an RGB-D camera would enable to readily incorporate other weak and fast-to-compute features to VPACK like, for instance, depth histograms.

- With respect to matching, some preliminary tests have been conducted on VPACK by replacing the current individual keypoint matching procedure with histograms of visual words in each of the subregions. The results so far suggest that there is a significant reduction in computation time with similar, or even slightly better, performance if the dictionary size is appropriately chosen. However, in this dissertation it has been argued against offline vocabulary building, because it is environment dependent. Therefore, it might be worth applying an incremental clustering algorithm to construct the vocabulary online. A promising alternative is Adaptive Incremental Neural Gas (AING) [Bou+13] that overcomes some of the issues with the incremental nearest neighbor classifier proposed by [Fil07]. For instance, several cluster representatives are updated with every new data point that arrives, which makes it more immune to noise and to the feature extraction and processing order. In addition, it incorporates a merging mechanism to prevent the number of clusters from growing indefinitely.
- As far as node extraction is concerned, the main issue of the algorithm presented is that the number of nodes detected is sensitive to the robot's motion commands. For instance, if the robot stands still, all the input images will be almost identical and the algebraic connectivity will inevitably rise. Rather than just ignoring these spurious nodes based on odometry measurements, a better solution would be to derive a method to adjust the sliding window size according to the robot's speed, the camera's frame frequency, and the rate of change observed in the environment.
- It might be interesting to test the developed algorithms in conjunction with a metric SLAM implementation and evaluate the advantages and disadvantages of a hybrid approach. In particular, enhancing topological nodes with local metric map patches would enable the robot to propose better topologies and perform more complex goal-directed tasks.

A

Datasets

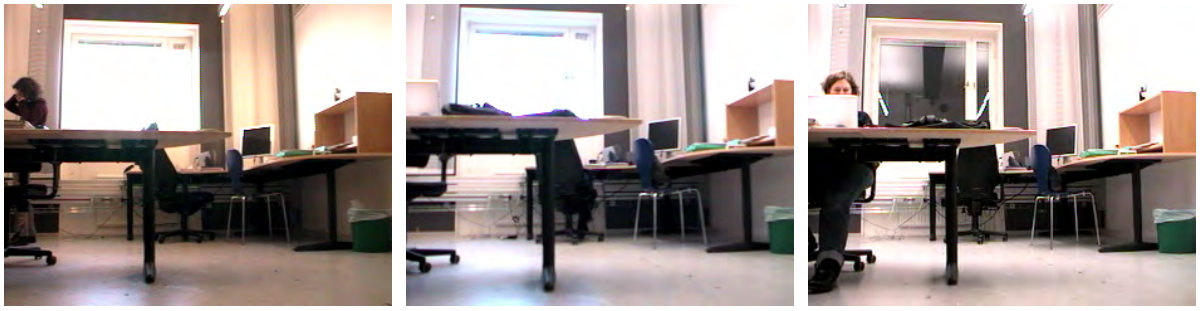
As a scientist, you're not supposed to make decisions without the data.
Francis Collins (1950–)

In this appendix, the specific characteristics of the image databases employed throughout the thesis to validate the algorithms proposed are set forth. These include information on the image sizes, the frame rate at which they were acquired, the height of the camera, the robot navigation speed, and peculiarities of the environment, among others.

A.1. KTH-IDOL2 database

The second version of the KTH Image Database for rObot Localization (KTH-IDOL2) [Luo+06; Luo+07] comprises 24 datasets captured with two different robots, a PeopleBot™ Minnie and a PowerBot™ Dumbo, that were manually commanded to navigate through an office environment at an average speed of approximately 0.3 m/s. Both robots acquired 309x240 pixel images at 5 fps with automatic exposure turned off. This atypical resolution is due to the fact that all the images were cropped to remove an 11 pixel wide black line introduced by the camera. The main difference between the two robotic platforms lies in the perspective camera setup; whereas Minnie's was installed at a height of 98 cm, Dumbo's was only at 36 cm. In addition, in the latter configuration the camera was tilted 13º upwards from the horizontal to reduce the amount of floor captured, which is not informative.

This database is aimed at capturing different variations that may appear in a typical office environment, like changes introduced by different lighting conditions (Figure A.1), and short and long term modifications caused by everyday human activity. The latter includes people appearing in different rooms, moving furniture and other objects, or even changing the decoration of a room. Hence, the environment, which consists of a corridor, a printer area, a kitchen, and two different offices (Figure A.3), was traversed 12 times with each of the robots to capture these effects. Three illumination conditions are considered: sunny weather, cloudy weather, and artificial lighting at night. For each of these conditions, two consecutive traversals were recorded and, then, four or five months later, another two were acquired.



(a) **(b)** **(c)**

Figure A.1. KTH-IDOL2 database. Example of images acquired in sunny (a), cloudy (b), and night (c) illumination conditions.

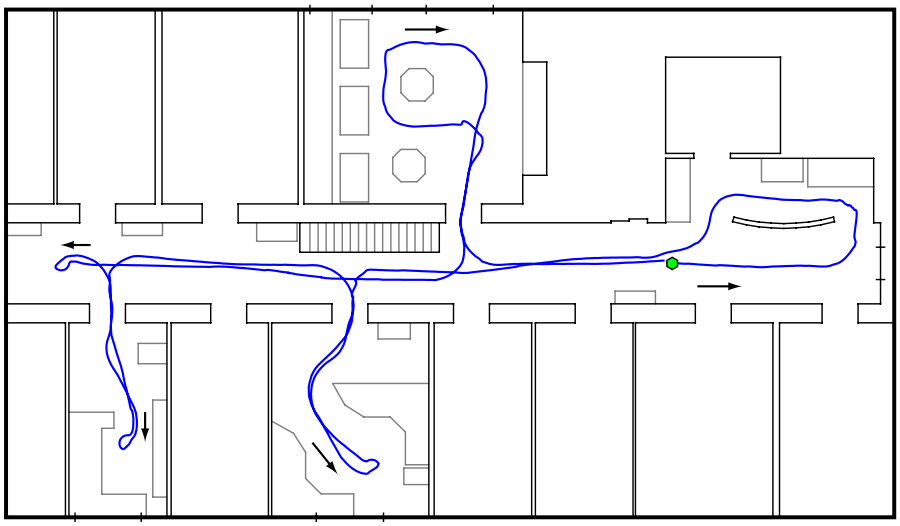


Figure A.2. KTH-IDOL2. Map of the environment and sample path. The green hexagon indicates the starting location.

(a) **(b)** **(c)**

(d) **(e)**

Figure A.3. KTH-IDOL2. Images of the five rooms captured with PowerBot Dumbo in order of acquisition: printer area (a), corridor (b), two-people office (c), one-person office (d), and kitchen (e).

A.2. COLD database

The COsy Localization Database (COLD) [Ull+07; PC09] consists of three independent datasets recorded in three laboratory/office environments from three different European cities: Saarbrücken and Freiburg in Germany, and Ljubljana in Slovenia. Like in the KTH-IDOL2 database, each environment was traversed multiple times to capture diverse illumination conditions (i.e., sunny, cloudy, and night), and the variations caused by human activity, in this case, during a two or three day span. Therefore, only minor changes, such as chair and object displacements, are perceptible.

As solely the Saarbrücken dataset is used in this dissertation, the rest of the section will focus on this particular location. In Saarbrücken, image acquisition was carried out using a remotely controlled ActivMedia PeopleBot™ equipped with a perspective camera mounted 140 cm above the floor. The robot was driven through the environment at roughly 0.3 m/s to capture 640x480 images at a frame rate of 5 fps with automatic exposure.

The Saarbrücken dataset is divided in two parts, A and B, each of which consists of a standard and an extended, longer, path. It contains the typical rooms that are expected to find in an office environment, such as, a corridor, a printer area, offices for one and several people, a bathroom, or a kitchen (Figure A.7).

A.3. Home environment

Finally, a home environment dataset was recorded as no image collections of non-office indoor enviroments captured using a forward-facing camera that fulfill the camera's setup requirements described in Section 3.2.1 (i.e., focal plane parallel to the walls and no roll) could be found. Images of a resolution of 640x480 pixels were captured using the left camera of a Point Grey Bumblebee®2 stereo vision camera [Wwwa] installed at a height of around 30 cm on an RC crawler that was modified for this thesis to integrate a notebook and the camera (Figure A.4). The robot was remotely driven at an average speed of 0.1 m/s and the frame rate was 1 fps. In total, six different locations were explored: the kitchen, the entrance hall, the living room, the terrace, the grass area, and the pergola (Figure A.8).

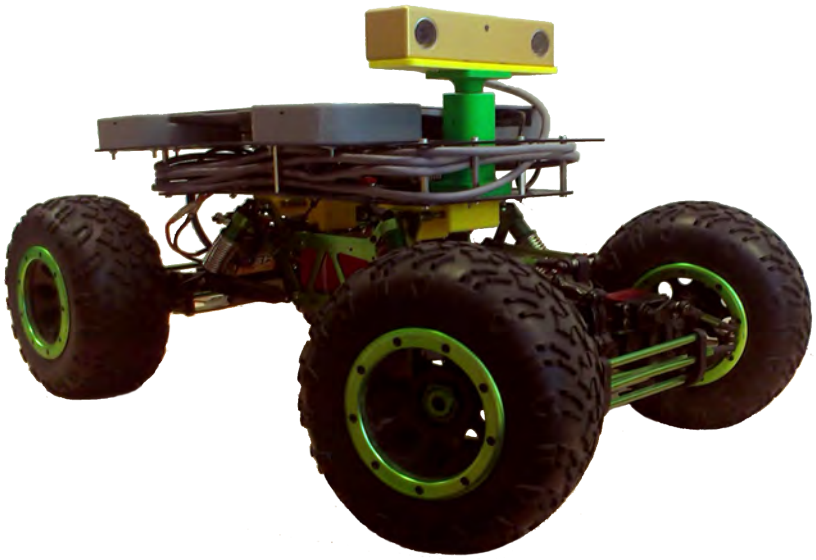
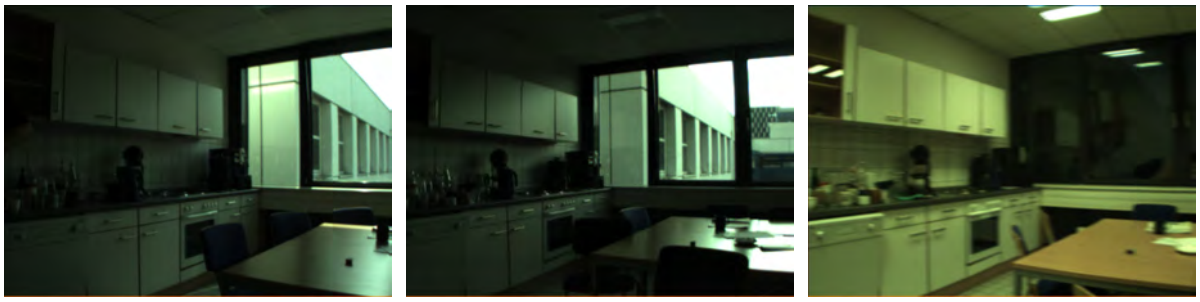


Figure A.4. Robotic platform employed to record the home environment dataset.



(a) **(b)** **(c)**

Figure A.5. COLD Saarbrücken. Example of images captured in sunny (a), cloudy (b), and night (c) illumination conditions.

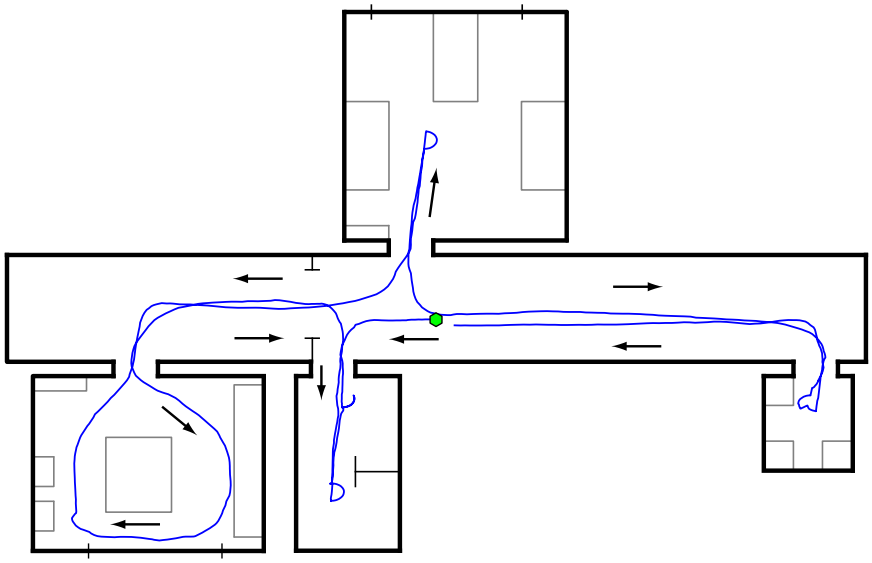
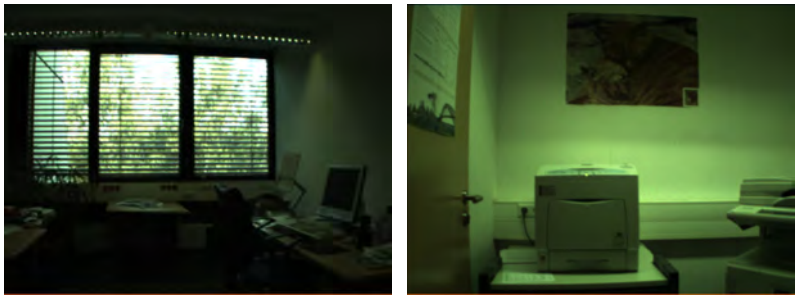


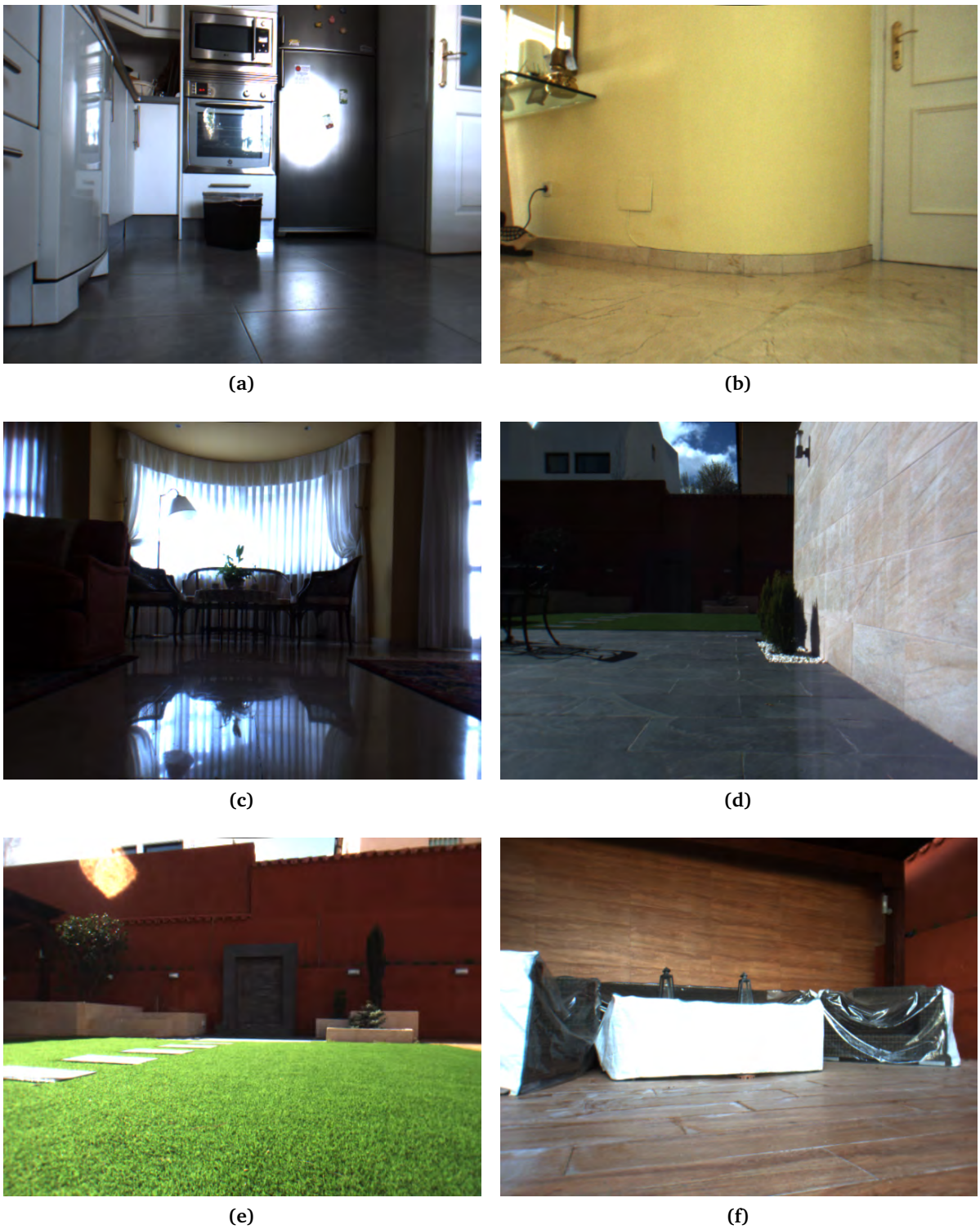
Figure A.6. COLD Saarbrücken. Map of part B with the extended path superimposed. The green hexagon indicates the starting point.

(a) **(b)** **(c)**



(d) **(e)**

Figure A.7. COLD Saarbrücken. Sample images of the five rooms visited in order of acquisition: corridor (a), bathroom (b), one-person office (c), kitchen (d), and printer area (e).



(a) **(b)**

(c) **(d)**

(e) **(f)**

Figure A.8. Home environment. Sample images of the six locations considered in order of acquisition: kitchen (a), entrance hall (b), living room (c), terrace (d), grass area (e), and pergola (f).

The robotic platform is equipped with a lightweight 11-inch MacBook Air® with 4 GB of RAM and a 2nd generation Intel® Core™ i5 processor at 1.6 GHz, the aforementioned Bumblebee2 camera, which is mounted on a stand that allows it to turn, and a electronic board based on two dsPICs, designed and programmed in parallel to the development of this thesis, that generates the control commands for the actuators. The camera is connected to the laptop using a FireWire to Thunderbolt adapter, and the control board, via USB. The platforms where the different components are installed and the camera stand were custom-designed and manufactured using 3D printing technology and laser cutting.

Bibliography

[Agr+08] M. Agrawal, K. Konolige, and M. R. Blas, "CenSurE: center surround extremas for realtime feature detection and matching", in *European Conf. Computer Vision*, ser. Lecture Notes in Computer Science, D. Forsyth, P. Torr, and A. Zisserman, Eds., vol. 5305, Springer, 2008, pp. 102–115.

[Agu+09] W. Aguilar, Y. Frauel, F. Escolano, M. E. Martínez-Pérez, A. Espinosa-Romero, and M. Á. Lozano, "A robust graph transformation matching for non-rigid registration", *Image and Vision Computing*, vol. 27, pp. 897–910, 2009.

[Ala+12] A. Alahi, R. Ortiz, and P. Vandergheynst, "FREAK: Fast retina keypoint", in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, Providence, RI, USA, Jun. 16–21, 2012, pp. 510–517.

[AM07] R. P. Adams and D. J. C. MacKay, "Bayesian online changepoint detection", University of Cambridge, Cambridge, UK, Tech. Rep. arXiv:0710.3742v1 [stat.ML]. Oct. 19, 2007.

[And+05] H. Andreasson, A. Treptow, and T. Duckett, "Localization for mobile robots using panoramic vision, local features and particle filter", in *Proc. IEEE Int. Conf. Robotics and Automation*, Barcelona, Spain, Apr. 18–22, 2005, pp. 3348–3353.

[Ang+04] D. Anguelov, D. Koller, E. Parker, and S. Thrun, "Detecting and modeling doors with mobile robots", in *Proc. IEEE Int. Conf. Robotics and Automation*, New Orleans, LA, USA, Apr. 26–May 1, 2004, pp. 3777–3784.

[Ang+08a] A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer, "A fast and incremental method for loop-closure detection using bags of visual words", *IEEE Trans. Robotics*, vol. 24, no. 5, pp. 1027–1037, Oct. 2008.

[Ang+08b] A. Angeli, S. Doncieux, J.-A. Meyer, and D. Filliat, "Incremental vision-based topological SLAM", in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Nice, France, Sep. 22–26, 2008, pp. 1031–1036.

[Ang+08c] A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer, "Real-time visual loop-closure detection", in *Proc. IEEE Int. Conf. Robotics and Automation*, Pasadena, CA, USA, May 19–23, 2008, pp. 1842–1847.

[Asi42] I. Asimov, "Runaround", in *Astounding Science Fiction*, Street & Smith, Mar. 1942.

[AV07] D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding", in *Proc. ACM-SIAM Symp. Discrete Algorithms*, New Orleans, LA, USA: Society for Industrial and Applied Mathematics, Jan. 7–9, 2007, pp. 1027–1035.

[Bay+08] H. Bay, A. Ess, T. Tuytelaars, and L. van Gool, "SURF: speeded up robust features", *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, Jun. 2008.

[BDW06] T. Bailey and H. F. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): Part II", *IEEE Robotics and Automation Magazine*, vol. 13, no. 3, pp. 108–117, Sep. 2006.

[Bee+05] P. Beeson, N. K. Jong, and B. Kuipers, "Towards autonomous topological place detection using the extended Voronoi graph", in *Proc. IEEE Int. Conf. Robotics and Automation*, Barcelona, Spain, Apr. 18–22, 2005, pp. 4373–4379.

[Bel34] E. T. Bell, "Exponential numbers", *American Mathematical Monthly*, vol. 41, pp. 411–419, 1934.

[BF11] S. Bazeille and D. Filliat, "Incremental topo-metric SLAM using vision and robot odometry", in *Proc. IEEE Int. Conf. Robotics and Automation*, Shanghai, China, May 9–13, 2011, pp. 4067–4073.

[Bil12] E. Billauer. (2012). Peakdet: Peak detection using MATLAB, [Online]. Available: http://billauer.co.il/peakdet.html (visited on 04/27/2014).

[Bla+08] J. L. Blanco, J. A. Fernández-Madrigal, and J. González, "Toward a unified Bayesian approach to hybrid metric-topological SLAM", *IEEE Trans. Robotics*, vol. 24, no. 2, pp. 259–270, Apr. 2008.

[BN03] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation", *Neural Computation*, vol. 15, pp. 1373–1396, 2003.

[Boa+14a] J. Boal, Á. Sánchez-Miralles, and M. Alvar, "Matching monocular lightweight features using n-gram techniques for topological location identification", *Robotica*, vol. FirstView, pp. 1–15, May 2014.

[Boa+14b] J. Boal, Á. Sánchez-Miralles, and Á. Arranz, "Topological simultaneous localization and mapping: A survey", *Robotica*, vol. 32, no. 5, pp. 803–821, Aug. 2014.

[Bol+09] G. Bologna, B. Deville, and T. Pun, "Blind navigation along a sinuous path by means of the See ColOr interface", in *Proc. Int. Work–Conference on the Interplay Between Natural and Artificial Computation. Part II: Bioinspired Applications in Artificial and Natural Computation*, J. Mira, J. R. Álvarez, F. de la Paz, and J. M. Ferrández, Eds., Santiago de Compostela, Spain: Springer-Verlag, Jun. 2009, pp. 235–243.

[Boo+07] O. Booij, B. Terwijn, Z. Zivkovic, and B. Kröse, "Navigation using an appearance based topological map", in *Proc. IEEE Int. Conf. Robotics and Automation*, Rome, Italy, Apr. 10–14, 2007, pp. 3927–3932.

[Bou+13] M.-R. Bouguelia, Y. Belaïd, and A. Belaïd, "An adaptive incremental clustering method based on the growing neural gas algorithm", in *Proc. Int. Conf. Pattern Recognition Applications and Methods*, Barcelona, Spain, Feb. 15–18, 2013, pp. 42–49.

[Bra00] G. Bradski. (2000). The OpenCV library, [Online]. Available: http://www.opencv.org (visited on 05/24/2014).

[Bre+12] T. Breuer, G. R. Giorgana Macedo, R. Hartanto, N. Hochgeschwender, D. Holz, F. Hegger, Z. Jin, C. Müller, J. Paulus, M. Reckhaus, J. A. Álvarez Ruiz, P. G. Plöger, and G. K. Kraetzschmar, "Johnny: An autonomous service robot for domestic environments", English, *Journal of Intelligent & Robotic Systems*, vol. 66, no. 1–2, pp. 245–272, 2012.

[Bro85] R. A. Brooks, "Visual map making for a mobile robot", in *Proc. IEEE Int. Conf. Robotics and Automation*, St. Louis, MO, USA, Mar. 25–28, 1985, pp. 824–829.

[Bro90] ——, "Elephants don't play chess", *Robotics and Autonomous Systems*, vol. 6, pp. 3–15, 1990.

[BSM] J. Boal and Á. Sánchez-Miralles, "Online topological segmentation of visual sequences using the algebraic connectivity of graphs", Under review in Robotica.

[Bur+99] W. Burgard, B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun, "Experiences with an interactive museum tour-guide robot", *Artificial Intelligence*, vol. 114, no. 1–2, pp. 3–55, Oct. 1999.

[Bül02] T. Bülow, "Multiscale image processing on the sphere", English, in *Pattern Recognition*, ser. Lecture Notes in Computer Science, L. van Gool, Ed., vol. 2449, Springer Berlin Heidelberg, 2002, pp. 609–617.

[Cad+12] C. Cadena, D. Gálvez-López, J. D. Tardós, and J. Neira, "Robust place recognition with stereo sequences", *IEEE Trans. Robotics*, vol. 28, no. 4, pp. 871–885, Aug. 2012.

[Cal+10] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary robust independent elementary features", in *European Conference on Computer Vision*, ser. Lecture Notes in Computer Science, K. Daniilidis, P. Maragos, and N. Paragios, Eds., vol. 6314, Springer Berlin Heidelberg, 2010, pp. 778–792.

[Cas+96] A. R. Cassandra, L. P. Kaelbling, and J. A. Kurien, "Acting under uncertainty: discrete Bayesian models for mobile-robot navigation", in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, vol. 2, Osaka, Japan, Nov. 4–8, 1996, pp. 963–972.

[Cha+10] C.-K. Chang, C. Siagian, and L. Itti, "Mobile robot vision navigation & localization using Gist and saliency", in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Taipei, Taiwan, Oct. 18–22, 2010, pp. 4147–4154.

[Cha+13] A. Chapoulie, P. Rives, and D. Filliat, "Appearance-based segmentation of indoors/outdoors sequences of spherical views", in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Tokyo, Japan, Nov. 3–8, 2013, pp. 1946–1951.

[Cha07] S.-H. Cha, "Comprehensive survey on distance/similarity measures between probability density functions", *International Journal of Mathematical Models and Methods in Applied Sciences*, vol. 1, no. 4, pp. 300–307, 2007.

[Cha08] S. Chawla, "Sparsest cut", in *Encyclopedia of Algorithms*, M.-Y. Kao, Ed., Springer, 2008, pp. 868–870.

[Chu97] F. K. Chung, *Spectral Graph Theory*, ser. CBMS Regional Conference Series in Mathematics. American Mathematical Society, 1997, vol. 92.

[CL68] C. K. Chow and C. N. Liu, "Approximating discrete probability distributions with dependence trees", *IEEE Trans. Information Theory*, vol. 14, no. 3, pp. 462–467, May 1968.

[CL85] R. Chatila and J.-P. Laumond, "Position referencing and consistent world modeling for mobile robots", in *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 2, St. Louis, MO, USA, Mar. 25–28, 1985, pp. 138–145.

[CN01] H. Choset and K. Nagatani, "Topological simultaneous localization and mapping (SLAM): toward exact localization without explicit localization", *IEEE Trans. Robotics and Automation*, vol. 17, no. 2, pp. 125–137, Apr. 2001.

[CN07] M. Cummins and P. Newman, "Probabilistic appearance based navigation and loop closing", in *Proc. IEEE. Int. Conf. Robotics and Automation*, Rome, Italy, Apr. 10–14, 2007, pp. 2042–2048.

[CN08] ——, "FAB-MAP: Probabilistic localization and mapping in the space of appearance", *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 647–665, Jun. 2008.

[CN10a] ——, "Accelerating FAB-MAP with concentration inequalities", *IEEE Trans. Robotics*, vol. 26, no. 6, pp. 1042–1050, Dec. 2010.

[CN10b] ——, "FAB-MAP: Appearance-based place recognition and mapping using a learned visual vocabulary model", in *Proc. 27th Int. Conf. Machine Learning*, Haifa, Israel, Jun. 21–24, 2010, pp. 3–10.

[CN11] ——, "Appearance-only SLAM at large scale with FAB-MAP 2.0", *The International Journal of Robotics Research*, vol. 30, no. 9, pp. 1100–1123, Aug. 2011.

[Com+03] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–577, May 2003.

[Csu+04] G. Csurka, C. Dance, L. Fan, J. Williamowski, and C. Bray, "Visual categorization with bags of keypoints", in *Proc. European Conf. Computer Vision–Workshop Statistical Learning in Computer Vision*, Prague, Czech Republic, May 11–14, 2004, pp. 59–74.

[Cum09] M. Cummins, "Probabilistic localization and mapping in appearance space", PhD thesis, University of Oxford, Oct. 2009.

[dAb07] N. M. M. de Abreu, "Old and new results on algebraic connectivity of graphs", *Linear Algebra and its Applications*, vol. 423, no. 1, pp. 53–73, May 2007.

[Dem67] A. P. Dempster, "Upper and lower probabilities induced by a multivalued mapping", *The Annals of Mathematical Statistics*, vol. 38, no. 2, pp. 325–339, Apr. 1967.

[DM01] Y. Deng and B. S. Manjunath, "Unsupervised segmentation of color-texture regions in images and video", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 8, pp. 800–810, Aug. 2001.

[Doh+09] N. L. Doh, K. Lee, W. K. Chung, and H. Cho, "Simultaneous localisation and mapping algorithm for topological maps with dynamics", *IET Control Theory and Applications*, vol. 3, no. 9, pp. 1249–1260, 2009.

[Dou+00a] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering", *Statistics and Computing*, vol. 10, pp. 197–208, 2000.

[Dou+00b] A. Doucet, N. de Freitas, K. Murphy, and S. Russell, "Rao-Blackwellised particle filtering for dynamic Bayesian networks", in *Proc. 16th Conf. Uncertainty in Artificial Intelligence*, Stanford, CA, USA: Morgan Kaufmann, Jun. 30–Jul. 3, 2000, pp. 176–183.

[Dou+01] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*, ser. Information Science and Statistics. Springer, 2001.

[Duc+00] T. Duckett, S. Marsland, and J. Shapiro, "Learning globally consistent maps by relaxation", in *Proc. IEEE Int. Conf. Robotics and Automation*, San Francisco, CA, USA, Apr. 2000, pp. 3841–3846.

[Dud+93] G. Dudek, P. Freedman, and S. Hadjres, "Using local information in a non-local way for mapping graph-like worlds", in *Proc. Int. Joint Conf. Artificial Intelligence*, Chambéry, France: Morgan Kaufmann, Aug. 28–Sep. 3, 1993, pp. 1639–1647.

[DWB06] H. F. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part I", *IEEE Robotics and Automation Magazine*, vol. 13, no. 2, pp. 99–110, Jun. 2006.

[EMC09] M. Ebrahimi and W. W. Mayol-Cuevas, "SUSurE: speeded up surround extrema feature detector and descriptor for realtime applications", in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition Workshops*, Miami, FL, USA, Jun. 20–25, 2009, pp. 9–14.

[FB81] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography", *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

[Fer+13] F. Ferreira, G. Veruggio, M. Caccia, and G. Bruzzone, "Binary visual features for ROV motion estimation", in *Proc. MTS/IEEE OCEANS*, Bergen, Norway, Jun. 10–14, 2013, pp. 1–7.

[Fie73] M. Fiedler, "Algebraic connectivity of graphs", *Czechoslovak Mathematical Journal*, vol. 23, no. 2, pp. 298–305, 1973.

[Fil07] D. Filliat, "A visual bag of words method for interactive qualitative localization and mapping", in *Proc. IEEE Int. Conf. Robotics and Automation*, Rome, Italy, Apr. 10–14, 2007.

[FM+13] E. Fernández-Moral, W. Mayol-Cuevas, V. Arévalo, and J. González-Jiménez, "Fast place recognition with plane-based maps", in *Proc. IEEE Int. Conf. Robotics and Automation*, Karlsruhe, Germany, May 6–10, 2013, pp. 2719–2724.

[FM03] D. Filliat and J.-A. Meyer, "Map-based navigation in mobile robots: i. A review of localization strategies", *Cognitive Systems Research*, vol. 4, no. 4, pp. 243–282, Dec. 2003.

[FP03] D. A. Forsyth and J. Ponce, *Computer vision: A modern approach*. Prentice Hall, 2003.

[Fra+07] F. Fraundorfer, C. Engels, and D. Nistér, "Topological mapping, localization and navigation using image collections", in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, San Diego, CA, USA, Oct. 29–Nov. 2, 2007, pp. 3872–3877.

[FS97] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and application to boosting", *Journal of Computer and System Sciences*, vol. 55, pp. 119–139, 1997.

[Gag10] N. D. Gagunashvili, "Chi-square tests for comparing weighted histograms", *Nuclear Instruments and Methods in Physics Research A*, vol. 614, pp. 287–296, 2010.

[GG08] T. Goedemé and L. van Gool, "Robust vision-only mobile robot navigation with topological maps", in *Mobile robots motion planning, new challenges*, X.-J. Jing, Ed., Austria: InTech, Jun. 2008, ch. 4, pp. 63–88.

[GK99] J.-S. Gutmann and K. Konolige, "Incremental mapping of large cyclic environments", in *Proc. IEEE Int. Symp. Computational Intelligence in Robotics and Automation*, Monterey, CA, USA, Nov. 8–9, 1999, pp. 318–325.

[Goe+04] T. Goedemé, T. Tuytelaars, and L. van Gool, "Fast wide baseline matching for visual navigation", in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, vol. 1, Washington, DC, USA, Jun. 27–Jul. 2, 2004, pp. 24–29.

[Goe+05] T. Goedemé, T. Tuytelaars, L. van Gool, G. Vanacker, and M. Nuttin, "Feature based omnidirectional sparse visual path following", in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Edmonton, AB, Canada, Aug. 2–6, 2005, pp. 1806–1811.

[Goe+07] T. Goedemé, M. Nuttin, T. Tuytelaars, and L. van Gool, "Omnidirectional vision based topological navigation", *International Journal of Computer Vision*, vol. 74, no. 3, pp. 219–236, 2007.

[Gol06] A. Golan, *Information and Entropy Econometrics — A Review and Synthesis*, ser. Foundations and Trends in Econometrics 1–2. Now Publishers Inc., 2006, vol. 2, pp. 1–145.

[GR+12] J. García-Rodríguez, A. Angelopoulou, J. M. García-Chamizo, A. Psarrou, S. Orts Escolano, and V. Morell Giménez, "Autonomous growing neural gas for applications with time constraint: Optimal parameter estimation", *Neural Networks*, vol. 32, pp. 196–208, 2012.

[Gro+03] H.-M. Gross, A. Koenig, C. Schroeter, and H.-J. Boehme, "Omnivision-based probabilistic self-localization for a mobile shopping assistant continued", in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, vol. 2, Las Vegas, NV, USA, Oct. 27–31, 2003, pp. 1505–1511.

[Gro+08] H. M. Gross, H.-J. Boehme, C. Schroeter, S. Mueller, A. Koenig, C. Martin, M. Merten, and A. Bley, "ShopBot: Progress in developing an interactive mobile shopping assistant for everyday use", in *Proc. IEEE Int. Conf. Systems, Man and Cybernetics*, Singapore, Oct. 12–15, 2008, pp. 3471–3478.

[GW08] R. C. González and R. E. Woods, *Digital Image Processing*, 3rd. Pearson Education, 2008.

[Haf00] H. H. Hafner, "Learning places in newly explored environments", in *Proc. Int. Conf. Simulation of Adaptive Behavior*, Meyer, Berthoz, Floreano, Roitblat, and Wilson, Eds., Honolulu, HI, USA: International Society for Adaptive Behavior, 2000, pp. 111–120.

[Ham50] R. W. Hamming, "Error detecting and error correcting codes", *Bell System Technical Journal*, vol. 29, no. 2, pp. 147–160, 1950.

[HB13] D. Holz and S. Behnke, "Fast range image segmentation and smoothing using approximate surface reconstruction and region growing", in *Intelligent Autonomous Systems 12*, ser. Advances in Intelligent Systems and Computing, S. Lee, H. Cho, K.-J. Yoon, and J. Lee, Eds., vol. 194, Springer Berlin Heidelberg, 2013, pp. 61–73.

[HJ08] N. Ho and R. Jarvis, "Vision based global localisation using a 3D environmental model created by a laser range scanner", in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Nice, France, Sep. 22–26, 2008, pp. 2964–2969.

[HO00] A. Hyvärinen and E. Oja, "Independent component analysis: Algorithms and applications", *Neural Networks*, vol. 13, no. 4–5, pp. 411–430, 2000.

[Hor08] M. Horvath. (2008). ShapeGrid macro — Isometricland, [Online]. Available: http://isometricland.net/povray/povray.php (visited on 05/24/2014).

[IB05] L. Itti and P. Baldi, "A principled approach to detecting surprising events in video", in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, vol. 1, San Diego, CA, USA, Jun. 20–26, 2005, pp. 631–637.

[Itt+98] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, Nov. 1998.

[Jac+95] C. E. Jacobs, A. Finkelstein, and D. H. Salesin, "Fast multiresolution image querying", in *Proc. Annu. Conf. Computer Graphics and Interactive Techniques*, Los Angeles, CA, USA, Aug. 6–11, 1995, pp. 277–286.

[JK12] C. Johnson and B. Kuipers, "Efficient search for correct and useful topological maps", in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Vilamoura, Algarve, Portugal, Oct. 7–12, 2012, pp. 5277–5282.

[JM09] D. Jurafsky and J. H. Martin, *Speech and language processing*. Prentice Hall, 2009.

[Jol05] I. T. Jolliffe, *Principal Component Analysis*, 2nd, ser. Springer Series in Statistics. Springer, 2005.

[JY13a] E. Johns and G.-Z. Yang, "Dynamic scene models for incremental, long-term, appearance-based localisation", in *Proc. IEEE Int. Conf. Robotics and Automation*, Karlsruhe, Germany, May 6–10, 2013, pp. 2731–2736.

[JY13b] ——, "Feature co-occurrence maps: Appearance-based localisation throughout the day", in *Proc. IEEE Int. Conf. Robotics and Automation*, Karlsruhe, Germany, May 6–10, 2013, pp. 3212–3218.

[Kae+98] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains", *Artificial Intelligence*, vol. 101, no. 1–2, pp. 99–134, May 1998.

[Kai67] T. Kailath, "The divergence and Bhattacharyya distance measures in signal selection", *IEEE Trans. Communication Technology*, vol. 15, no. 1, pp. 52–60, Feb. 1967.

[KB02] B. Kuipers and P. Beeson, "Bootstrap learning for place recognition", in *Proc. AAAI 18th Nat. Conf. Artificial Intelligence*, Edmonton, AB, Canada, Jul. 28–Aug. 1, 2002, pp. 174–180.

[KB91] B. Kuipers and Y.-T. Byun, "A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations", *Robotics and Autonomous Systems*, vol. 8, no. 1, pp. 47–63, 1991.

[KL51] S. Kullback and R. A. Leibler, "On information and sufficiency", *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[KL88] B. Kuipers and T. Levitt, "Navigation and mapping in large-scale space", *AI Magazine*, vol. 9, no. 2, pp. 25–43, 1988.

[Koe+06] S. Koenig, A. Mudgal, and C. Tovey, "A near-tight approximation lower bound and algorithm for the kidnapped robot problem", in *Proc. Symp. Discrete Algorithms*, Miami, FL, USA, Jan. 22–26, 2006, pp. 133–142.

[Koe+08] A. Koenig, J. Kessler, and H.-M. Gross, "A graph matching technique for an appearance-based, visual SLAM-approach using Rao-Blackwellized particle filters", in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Nice, France, Sep. 22–26, 2008, pp. 1576–1581.

[Kon04] K. Konolige, "Large-scale map-making", in *Proc. AAAI 19th Nat. Conf. Artificial Intelligence*, San Jose, CA, USA, Jul. 25–29, 2004, pp. 457–463.

[KS96] S. Koenig and R. G. Simmons, "Unsupervised learning of probabilistic models for robot navigation", in *Proc. IEEE Int. Conf. Robotics and Automation*, Minneapolis, MN, USA, Apr. 1996, pp. 2301–2308.

[Kui+04] B. Kuipers, J. Modayil, P. Beeson, M. MacMahon, and F. Savelli, "Local metrical and global topological maps in the hybrid spatial semantic hierarchy", in *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 5, New Orleans, LA, USA, Apr. 26–May 1, 2004, pp. 4845–4851.

[Kui00] B. Kuipers, "The spatial semantic hierarchy", *Artificial Intelligence*, vol. 119, pp. 191–233, 2000.

[KW94] D. Kortenkamp and T. Weymouth, "Topological mapping for mobile robots using a combination of sonar and vision sensing", in *Proc. AAAI 12th Nat. Conf. Artificial Intelligence*, Seattle, WA, USA, Jul. 31–Aug. 4, 1994, pp. 979–984.

[Lam+01] P. Lamon, I. Nourbakhsh, B. Jensen, and R. Siegwart, "Deriving and matching image fingerprint sequences for mobile robot localization", in *Proc. IEEE Int. Conf. Robotics and Automation*, Seoul, Korea, May 21–26, 2001, pp. 1609–1614.

[Laz+06] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories", in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, New York, NY, USA, Jun. 17–22, 2006, pp. 2169–2178.

[LDW91a] J. J. Leonard and H. F. Durrant-Whyte, "Mobile robot localization by tracking geometric beacons", *IEEE Trans. Robotics and Automation*, vol. 7, no. 3, pp. 376–382, Jun. 1991.

[LDW91b] ——, "Simultaneous map building and localization for an autonomous mobile robot", in *Proc. IEEE/RSJ Int. Intelligent Robots and Systems Workshop*, Osaka, Japan, Nov. 3–5, 1991, pp. 1442–1447.

[LeC86] L. LeCam, *Asymptotic Methods in Statistical Decision Theory*. Springer-Verlag, 1986.

[Leu+11] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints", in *Proc. IEEE Int. Conf. Computer Vision*, Barcelona, Spain, Nov. 6–13, 2011, pp. 2548–2555.

[Lie+03] R. Lienhart, A. Kuranov, and V. Pisarevsky, "Empirical analysis of detection cascades of boosted classifiers for rapid object detection", in *Pattern Recognition*, ser. Lecture Notes in Computer Science, B. Michaelis and G. Krell, Eds., vol. 2781, Springer Berlin Heidelberg, 2003, pp. 297–304.

[Lis+03] B. Lisien, D. Morales, D. Silver, G. Kantor, I. Rekleitis, and H. Choset, "Hierarchical simultaneous localization and mapping", in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Las Vegas, NV, USA, Oct. 27–Nov. 1, 2003, pp. 448–453.

[Liu+09] M. Liu, D. Scaramuzza, C. Pradalier, R. Siegwart, and Q. Chen, "Scene recognition with omnidirectional vision for topological map using lightweight adaptive descriptors", in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, St. Louis, MO, USA, Oct. 11–15, 2009, pp. 116–121.

[Liu+11] M. Liu, F. Colas, and R. Siegwart, "Regional topological segmentation based on mutual information graphs", in *Proc. IEEE Int. Conf. Robotics and Automation*, Shanghai, China, May 9–13, 2011, pp. 3269–3274.

[LJ10] W. L. D. Lui and R. Jarvis, "A pure vision-based approach to topological SLAM", in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Taipei, Taiwan, Oct. 18–22, 2010, pp. 3784–3791.

[LK81] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision", in *Proc. 7th Int. Joint Conf. Artificial Intelligence*, Vancouver, BC, Canada: William Kaufmann, Aug. 24–28, 1981, pp. 674–679.

[LM13] M. Labbé and F. Michaud, "Appearance-based loop closure detection for online large-scale and long-term operation", *IEEE Trans. Robotics*, vol. 29, no. 3, pp. 734–745, Jun. 2013.

[LM97] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping", *Autonomous Robots*, vol. 4, pp. 333–349, 1997.

[LO12] Y. Li and E. B. Olson, "IPJC: The incremental posterior joint compatibility test for fast feature cloud matching", in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Vilamoura, Algarve, Portugal, Oct. 7–12, 2012, pp. 3467–3474.

[Low04] D. G. Lowe, "Distinctive image features from scale-invariant keypoints", *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[Low99] ——, "Object recognition from local scale-invariant features", in *Proc. IEEE Int. Conf. Computer Vision*, vol. 2, Kerkyra, Greece, Sep. 20–27, 1999, pp. 1150–1157.

[LS12] M. Liu and R. Siegwart, "DP-FACT: Towards topological mapping and scene recognition with color for omnidirectional camera", in *Proc. IEEE Int. Conf. Robotics and Automation*, Saint Paul, MN, USA, May 14–18, 2012, pp. 3503–3508.

[LS14] ——, "Topological mapping and scene recognition with lightweight color descriptors for an ominidirectional camera", *IEEE Trans. Robotics*, vol. 30, no. 2, pp. 310–324, Apr. 2014.

[Luo+06] J. Luo, A. Pronobis, B. Caputo, and P. Jensfelt, "The KTH-IDOL2 database", KTH Royal Institute of Technology, CVAP/CAS, Stockholm, Sweden, Tech. Rep. CVAP304, Oct. 2006.

[Luo+07] ——, "Incremental learning for place recognition in dynamic environments", in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, San Diego, CA, USA, Oct. 29–Nov. 2, 2007, pp. 721–728.

[LZ13] Y. Liu and H. Zhang, "Towards improving the efficiency of sequence-based SLAM", in *Proc. IEEE Int. Conf. Mechatronics and Automation*, Takamatsu, Japan, Aug. 4–7, 2013, pp. 1261–1266.

[MA09] R. Maini and H. Aggarwal, "Study and comparison of various image edge detection techniques", *International Journal of Image Processing*, vol. 3, no. 1, pp. 1–11, Jan. 2009.

[Mac67] J. MacQueen, "Some methods for classification and analysis of multivariate observations", in *Proc. Berkeley Symp. Mathematical Statistics and Probability*, vol. 1, Berkeley, CA, USA: University of California Press, 1967, pp. 281–297.

[Mad+11] W. Maddern, M. Milford, and G. Wyeth, "Continuous appearance-based trajectory SLAM", in *Proc. IEEE Int. Conf. Robotics and Automation*, Shanghai, China, May 9–13, 2011, pp. 3595–3600.

[Mad+12a] ——, "Capping computation time and storage requirements for appearance-based localization with CAT-SLAM", in *Proc. IEEE Int. Conf. Robotics and Automation*, Saint Paul, MN, USA, May 14–18, 2012, pp. 822–827.

[Mad+12b] ——, "Towards persistent indoor appearance-based localization, mapping and navigation using CAT-SLAM", in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Vilamoura, Algarve, Portugal, Oct. 7–12, 2012, pp. 4224–4230.

[Mat+02] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions", in *Proc. British Machine Vision Conf.*, Cardiff, Wales, UK, Sep. 2–5, 2002, pp. 384–393.

[MD+07] M. Magimai-Doss, D. Hakkani-Tür, Ö. Çetin, E. Shriberg, J. Fung, and N. Mirghafori, "Entropy based classifier combination for sentence segmentation", in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, vol. 4, Honolulu, HI, USA, Apr. 15–20, 2007, IV–189–IV–192.

[Mik+06] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. van Gool, "A comparison of affine region detectors", *International Journal of Computer Vision*, vol. 65, no. 1, pp. 43–72, Nov. 2006.

[Min07] S. Minetola, "Principii di analisi combinatoria", *Giornale di Matematiche*, vol. 45, pp. 333–366, 1907.

[MM+05] Ó. Martínez-Mozos, C. Stachniss, and W. Burgard, "Supervised learning of places from range data using AdaBoost", in *Proc. IEEE Int. Conf. Robotics and Automation*, Barcelona, Spain, Apr. 18–22, 2005, pp. 1730–1735.

[Mod+04] J. Modayil, P. Beeson, and B. Kuipers, "Using the topological skeleton for scalable global metrical map-building", in *Proc. IEEE/RSJ Int. Conf. lntelligent Robots and Systems*, Sendai, Japan, Sep. 28–Oct. 2, 2004, pp. 1530–1536.

[Mon+02] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem", in *Proc. AAAI 18th Nat. Conf. Artificial Intelligence*, Edmonton, AB, Canada, Jul. 28–Aug. 1, 2002, pp. 593–598.

[Ng+02] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm", *Advances in Neural Information Processing Systems*, vol. 14, pp. 849–856, 2002.

[NG06] A. Neubeck and L. van Gool, "Efficient non-maximum suppression", in *Proc. 18th Int. Conf. Pattern Recognition*, vol. 3, Hong Kong, China, Aug. 20–24, 2006, pp. 850–855.

[Ngu+05] V. Nguyen, A. Martinelli, N. Tomatis, and R. Siegwart, "A comparison of line extraction algorithms using 2D laser rangefinder for indoor mobile robotics", in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Edmonton, AB, Canada, Aug. 2–6, 2005, pp. 1929–1934.

[NH08] A. Nüchter and J. Hertzberg, "Towards semantic maps for mobile robots", *Robotics and Autonomous Systems*, vol. 56, pp. 915–926, 2008.

[Nie+04] J. I. Nieto, J. E. Guivant, and E. M. Nebot, "The hybrid metric maps (HYMMs): a novel map representation for DenseSLAM", in *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 1, New Orleans, LA, USA, Apr. 26–May 1, 2004, pp. 391–396.

[Nis04] D. Nistér, "An efficient solution to the five-point relative pose problem", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756–770, Jun. 2004.

[NP33] J. Neyman and E. S. Pearson, "On the problem of the most efficient tests of statistical hypotheses", *Phil. Trans. of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 231, no. 694–706, pp. 289–337, Jan. 1933.

[NS06] D. Nistér and H. Stewénius, "Scalable recognition with a vocabulary tree", in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, vol. 2, New York, NY, USA, Jun. 17–22, 2006, pp. 2161–2168.

[NT01] J. Neira and J. D. Tardós, "Data association in stochastic mapping using the joint compatibility test", *IEEE Trans. Robotics and Automation*, vol. 17, no. 6, pp. 890–897, Dec. 2001.

[NW70] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins", *Journal of Molecular Biology*, vol. 48, pp. 443–453, 1970.

[NZ06] M.-E. Nilsback and A. Zisserman, "A visual vocabulary for flower classification", in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, vol. 2, New York, NY, USA, Jun. 17–22, 2006, pp. 1447–1454.

[ON98] C. Owen and U. Nehmzow, "Landmark-based navigation for a mobile robot", in *Proc. Int. Conf. Simulation of Adaptive Behaviour*, MIT Press, 1998, pp. 240–245.

[PC09] A. Pronobis and B. Caputo, "COLD: The cosy localization database", *The International Journal of Robotics Research*, vol. 28, pp. 588–594, May 2009.

[Pfi+03] S. T. Pfister, S. I. Roumeliotis, and J. W. Burdick, "Weighted line fitting algorithms for mobile robot map building and efficient data representation", in *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 1, Taipei, Taiwan, Sep. 14–19, 2003, pp. 1304–1311.

[Phi+07] J. Philbin, R. Arandjelović, and A. Zisserman. (Nov. 2007). Oxford buildings dataset, [Online]. Available: http://www.robots.ox.ac.uk/~vgg/data/oxbuildings (visited on 05/24/2014).

[Pir+03] P. Pirjanian, N. Karlsson, L. Goncalves, and E. di Bernardo, "Low-cost visual localization and mapping for consumer robotics", *Industrial Robot: An International Journal*, vol. 30, no. 2, pp. 139–144, Apr. 2003.

[PN10] R. Paul and P. Newman, "FAB-MAP 3D: Topological mapping with spatial and visual appearance", in *Proc. IEEE Int. Conf. Robotics and Automation*, Anchorage, AK, USA, May 3–8, 2010, pp. 2649–2656.

[Ram+05] F. T. Ramos, B. Upcroft, S. Kumar, and H. F. Durrant-Whyte, "A Bayesian approach for place recognition", in *Proc. Int. Joint Conf. Artificial Intelligence– Workshop Reasoning with Uncertainty in Robotics*, Edinburgh, Scotland, UK, Jul. 30, 2005.

[Ran+06] A. Ranganathan, E. Menegatti, and F. Dellaert, "Bayesian inference in the space of topological maps", *IEEE Trans. Robotics*, vol. 22, no. 1, pp. 92–107, Feb. 2006.

[Ran08] A. Ranganathan, "Probabilistic topological maps", PhD thesis, Georgia Institute of Technology, Apr. 2008.

[Ran10] ——, "PLISS: Detecting and labeling places using online change-point detection", in *Proc. Robotics: Science and Systems*, Zaragoza, Spain, Jun. 2010.

[Ran12] ——, "PLISS: Labeling places using online changepoint detection", *Autonomous Robots*, vol. 32, no. 4, pp. 351–368, 2012.

[RC10] A. Romero and M. Cazorla, "Topological SLAM using omnidirectional images: Merging feature detectors and graph-matching", in *Proc. Advanced Concepts for Intelligent Vision Systems*, Sydney, Australia, Dec. 13–16, 2010, pp. 464–475.

[RC12] ——, "Topological visual mapping in robotics", *Cognitive Processing*, vol. 13, no. 1 Supplement, pp. 305–308, Aug. 2012.

[RD04] A. Ranganathan and F. Dellaert, "Inference in the space of topological maps: an MCMC-based approach", in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Sendai, Japan, Sep. 28–Oct. 2, 2004, pp. 1518–1523.

[RD06a] ——, "A Rao-Blackwellized particle filter for topological mapping", in *Proc. IEEE Int. Conf. Robotics and Automation*, Orlando, FL, USA, May 15–19, 2006, pp. 810–817.

[RD06b] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection", in *Proc. European Conf. Computer Vision*, Graz, Austria, May 7–13, 2006, pp. 430–443.

[RD08] A. Ranganathan and F. Dellaert, "Automatic landmark detection for topological mapping using Bayesian surprise", Georgia Institute of Technology, Tech. Rep. GT-IC-08-04, 2008.

[RD09] ——, "Bayesian surprise and landmark detection", in *Proc. IEEE Int. Conf. Robotics and Automation*, Kobe, Japan, May 12–17, 2009, pp. 2017–2023.

[RD11] ——, "Online probabilistic topological mapping", *The International Journal of Robotics Research*, vol. 30, no. 6, pp. 755–771, May 2011.

[RK04] E. Remolina and B. Kuipers, "Towards a general theory of topological maps", *Artificial Intelligence*, vol. 152, pp. 47–104, 2004.

[RN10] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd. Pearson Education, 2010.

[Ros+10] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 105–119, Jan. 2010.

[Rub+11] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: an efficient alternative to SIFT or SURF", in *Proc. IEEE Int. Conf. Computer Vision*, Barcelona, Spain, Nov. 6–13, 2011, pp. 2564–2571.

[Sab+10] D. Sabatta, D. Scaramuzza, and R. Siegwart, "Improved appearance-based matching in similar and dynamic environments using a vocabulary tree", in *Proc. IEEE Int. Conf. Robotics and Automation*, Anchorage, AK, USA, May 3–7, 2010, pp. 1008–1013.

[Sab08] D. G. Sabatta, "Vision-based topological map building and localisation using persistent features", in *Robotics and Mechatronics Symp.*, Bloemfontein, South Africa, Nov. 11, 2008, pp. 1–6.

[San10] C. Sanderson, "Armadillo: An open source C++ linear algebra library for fast prototyping and computationally intensive experiments", NICTA, Tech. Rep., 2010. [Online]. Available: http://arma.sourceforge.net (visited on 05/24/2014).

[Se+05] S. Se, D. G. Lowe, and J. J. Little, "Vision-based global localization and mapping for mobile robots", *IEEE Trans. Robotics*, vol. 21, no. 3, pp. 364–375, Jun. 2005.

[SI05] C. Siagian and L. Itti, "Gist: A mobile robotics application of context-based vision in outdoor environment", in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition–Workshop Attention and Performance in Computer Vision*, San Diego, CA, USA, Jun. 20–26, 2005, pp. 1–7.

[SI07] ——, "Rapid biologically-inspired scene classification using features shared with visual attention", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 300–312, Feb. 2007.

[SK04] F. Savelli and B. Kuipers, "Loop-closing and planarity in topological map-building", in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Sendai, Japan, Sep. 28–Oct. 2, 2004, pp. 1511–1517.

[SK07] B. J. Stankiewicz and A. A. Kalia, "Acquisition of structural versus object landmark knowledge", *Journal of Experimental Psychology: Human Perception and Performance*, vol. 33, no. 2, pp. 378–390, 2007.

[ST06] D. A. Spielman and S.-H. Teng, "Spectral partitioning works: Planar graphs and finite element meshes", *Linear Algebra and its Applications*, vol. 421, pp. 284–305, 2006.

[ST94] J. Shi and C. Tomasi, "Good features to track", in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Seattle, WA, USA, Jun. 1994, pp. 593–600.

[Sta+05] C. Stachniss, O. Martínez-Mozos, A. Rottmann, and W. Burgard, "Semantic labeling of places", in *Proc. Int. Sym. Robotics Research*, San Francisco, CA, USA, Oct. 12–15, 2005.

[SZ03] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos", in *Proc. IEEE Int. Conf. Computer Vision*, vol. 2, Nice, France, Oct. 14–17, 2003, pp. 1470–1477.

[Tao+11] T. Tao, S. Tully, G. Kantor, and H. Choset, "Incremental construction of the saturated-GVG for multi-hypothesis topological SLAM", in *Proc. IEEE Int. Conf. Robotics and Automation*, Shanghai, China, May 9–13, 2011, pp. 3072–3077.

[Tap05] A. Tapus, "Topological SLAM – Simultaneous localization and mapping with fingerprints of places", PhD thesis, École Polytechnique Fédérale de Lausanne, Switzerland, 2005.

[TC92] D.-C. Tseng and C.-H. Chang, "Color segmentation using perceptual attributes", in *Proc. 11th IAPR Int. Conf. Pattern Recognition*, vol. 3, The Hage, Netherlands, Aug. 30–Sep. 3, 1992, pp. 228–231.

[Thr+05] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. The MIT Press, Aug. 2005.

[Thr02] S. Thrun, "Robotic mapping: A survey", in *Exploring Artificial Intelligence in the New Millenium*, G. Lakemeyer and B. Nebel, Eds., Morgan Kaufmann, 2002, pp. 1–35.

[TK91] C. Tomasi and T. Kanade, "Detection and tracking of point features", Carnegie Mellon University, Tech. Rep. CMU-CS-91-132, Apr. 1991.

[Tom+02] N. Tomatis, I. Nourbakhsh, and R. Siegwart, "Hybrid simultaneous localization and map building: closing the loop with multi-hypotheses tracking", in *Proc. IEEE Int. Conf. Robotics and Automation*, Washington, DC, USA, May 11–15, 2002, pp. 2749–2754.

[TS05] A. Tapus and R. Siegwart, "Incremental robot mapping with fingerprints of places", in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Edmonton, AB, Canada, Aug. 2–6, 2005, pp. 2429–2434.

[Tse+06] G. Tsechpenakis, D. N. Metaxas, C. Neidle, and O. Hadjiliadis, "Robust online change-point detection in video sequences", in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition Workshops*, New York, NY, USA, Jun. 17–22, 2006.

[Tul+09] S. Tully, G. Kantor, H. Choset, and F. Werner, "A multi-hypothesis topological SLAM approach for loop closing on edge-ordered graphs", in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, St. Louis, MO, USA, Oct. 11–15, 2009, pp. 4943–4948.

[Ull+07] M. M. Ullah, A. Pronobis, B. Caputo, J. Luo, and P. Jensfelt, "The COLD database", KTH Royal Institute of Technology, CVAP/CAS, Stockholm, Sweden, Tech. Rep. TRITA-CSC-CV 2007:1, Oct. 2007.

[Ulr+97] I. Ulrich, F. Mondada, and J.-D. Nicoud, "Autonomous vacuum cleaner", *Robotics and Autonomous Systems*, vol. 19, no. 3, pp. 233–245, 1997.

[UN00] I. Ulrich and I. Nourbakhsh, "Appearance-based place recognition for topological localization", in *Proc. IEEE Int. Conf. Robotics and Automation*, San Francisco, CA, USA, Apr. 24–28, 2000, pp. 1023–1029.

[Vas+07] S. Vasudevan, S. Gächter, V. Nguyen, and R. Siegwart, "Cognitive maps for mobile robots – an object based approach", *Robotics and Autonomous Systems*, vol. 55, pp. 359–371, 2007.

[VJ01] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features", in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, vol. 1, Kauai, HI, USA, Dec. 8–14, 2001, pp. 511–518.

[vLu07] U. von Luxburg, "A tutorial on spectral clustering", *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[Wer+08a] F. Werner, C. Gretton, F. Maire, and J. Sitte, "Induction of topological environment maps from sequences of visited places", in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Nice, France, Sep. 22–26, 2008, pp. 2890–2895.

[Wer+08b] F. Werner, J. Sitte, and F. Maire, "Visual topological mapping and localisation using colour histograms", in *Proc. Int. Conf. Control, Automation, Robotics and Vision*, Hanoi, Vietnam, Dec. 17–20, 2008, pp. 341–346.

[Wer+09a] F. Werner, F. Maire, and J. Sitte, "Topological SLAM using fast vision techniques", in *Proc. Advances in Robotics: FIRA RoboWorld Congress*, Incheon, Korea, Aug. 16–20, 2009, pp. 187–198.

[Wer+09b] F. Werner, F. Maire, J. Sitte, H. Choset, S. Tully, and G. Kantor, "Topological SLAM using neighbourhood information of places", in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, St. Louis, MO, USA, Oct. 11–15, 2009.

[Wer+12] F. Werner, J. Sitte, and F. Maire, "Topological map induction using neighbourhood information of places", *Autonomous Robots*, vol. 32, no. 4, pp. 405–418, May 2012.

[Wwwa] Bumblebee2 stereo camera, Point Grey, [Online]. Available: http://www.ptgrey.com/products/bumblebee2/bumblebee2_stereo_camera.asp (visited on 05/24/2014).

[Wwwb] Street View, Google Inc., [Online]. Available: http://www.google.com/intl/en/maps/about/behind-the-scenes/streetview/ (visited on 05/24/2014).

[Yam+98] B. Yamauchi, A. Schultz, and W. Adams, "Mobile robot exploration and map-building with continuous localization", in *Proc. IEEE Int. Conf. Robotics and Automation*, Leuven, Belgium, May 16–20, 1998, pp. 3715–3720.

[Ziv+05] Z. Zivkovic, B. Bakker, and B. Kröse, "Hierarchical map building using visual landmarks and geometric constraints", in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Aug. 2–6, 2005, pp. 2480–2485.

[ZT98] D. Ziou and S. Tabbone, "Edge detection techniques: An overview", *International Journal of Pattern Recognition and Image Analysis*, vol. 8, no. 4, pp. 537–559, 1998.